

TESI DI LAUREA MAGISTRALE

**A volume integral formulation on polyhedral meshes
for eddy current problems in fusion devices**

Formulazione integrale su mesh poliedriche per problemi di correnti indotte
in macchine a confinamento magnetico

Studente:
Mauro Passarotto
Matricola 1084172

Docente:
Prof. Paolo Bettini
Correlatore:
Prof. Ruben Specogna

A mio nonno

Abstract

Integral formulations can be more convenient than 3D finite-element-method (FEM) codes for the numerical solution of quasi-magnetostatic (eddy currents) problems in large and complex domains, consisting of many interconnected parts or components (e.g. magnetic confinement fusion devices), since they do not require the discretization of non-conducting subdomains. As a matter of fact, a good accuracy is often achieved with a relatively coarse discretization, thus reducing the need of allocated memory and computing time.

This thesis work aims at developing a new volume integral code, in the MATLAB framework, able to solve magneto quasi-static problems in frequency domain. The main innovation of this work is the formulation based on a domain discretization with generic polyhedral elements whose unique specification is to have plane faces. Moreover, a fast and accurate algorithm (closed-form formulas) has been implemented for the computation of the magnetic field and vector potential components produced by a general polyhedron. This tool is used to compute both the known term of the problem (integral sources) and the effects of the current induced in the conducting regions (numerical domain), in the post processing phase.

As far as the validation of the code is concerned, first the code has been benchmarked against some reference configurations (with either analytical or numerical solution), then the proposed approach is applied to a challenging eddy current problem on a real fusion device (RFX-mod).

Keywords: volume integral formulation, eddy current, polyhedral mesh, fusion devices.

Sommario

I reattori da fusione sono macchine complesse e dalla geometria intricata. Nel momento in cui ci si trovi a dovere realizzare delle simulazioni numeriche il poter disporre di uno strumento efficiente è un aspetto fondamentale. In questo senso, differentemente dai più diffusi software a elementi finiti (FEM), l' utilizzo di un codice a formulazione integrale rappresenta un grosso vantaggio dal momento che non si ha più la necessità di dovere *meshare* l' aria intorno alle strutture conduttrici che si stanno analizzando.

In questa tesi si è perciò sviluppato un nuovo codice a formulazione integrale in MATLAB per la risoluzione di un problema a correnti indotte in strutture conduttrici. La principale novità è rappresentata dall' utilizzo di poliedri, la cui unica specifica è avere tutte facce piane, per la discretizzazione del dominio. Oltre a ciò sarà sviluppato un codice per il post processing della soluzione, basato non su integrazioni numeriche, ma bensì su formule esatte (di cui proveremo la precisione). Infine il codice così formato sarà testato prima su problemi schematici (una lastra e una sfera metalliche) per valutarne la precisione e l' affidabilità e solo successivamente sarà impiegato su un dispositivo a confinamento magnetico come RFX-mod per dimostrarne l' applicabilità nella soluzione di un problema reale.

Oltre a questi contenuti, durante lo svolgimento di questo lavoro si è anche sfruttata la duttilità della formulazione utilizzata per il "*data post processing*" per ottenere uno strumento numerico che realizzi la discretizzazione di sorgenti di campo magnetico come bobine induttive di forme diverse e che ne calcoli i campi da queste generati. In particolare si è lavorato su bobine circolari, rettangolari ed elicoidali per ottenere, da quast' ultime, una bobina di magnetizzazione tipo "Stellarator", ampiamente diffusa negli omonimi reattori da fusione.

In futuro si conta di continuare lo sviluppo di questo codice approntando sistemi di discretizzazione del dominio più flessibili e integrando la sua formulazione con strumenti di sparsificazione matriciale che ne aumentino drasticamente la velocità ampliando di conseguenza il numero delle possibili applicazioni. Inoltre si auspica l' introduzione di tecniche di adattività per il raffinamento mirato della mesh in modo da aumentare la precisione numerica dei risultati.

Parole chiave: formulazione integrale, correnti indotte, mesh a poliedri, reattori da fusione.

Ringraziamenti

I primi più sinceri ringraziamenti vanno indistintamente al professore Paolo Bettini e al professore Ruben Specogna: per l'attenzione e l'impegno dedicatomi e per avermi trasmesso la grande passione per il loro lavoro, passione sempre dimostratami in questi mesi. Un grazie ancora per la loro sincera disponibilità e per i preziosi consigli senza i quali non avrei portato a termine questo percorso umano e professionale. Oltre a loro un ruolo fondamentale nel trasmettermi nuove preziose conoscenze e nel migliorare il mio modo di lavorare lo ha avuto l'ingegnere Nicolò Marconato che sinceramente ringrazio per la pazienza e l'inesauribile cortesia.

Un ringraziamento alla mia famiglia tutta, per il costante supporto in tutti questi anni e per la cieca fiducia da sempre riposta in me. Spero nel corso della vita di poterli ripagare per i loro sforzi e di serbare sempre in me i loro consigli e i loro valori. A tal proposito un grazie particolare va sicuramente ai miei nonni, perchè con la loro saggia umiltà mi hanno sempre trasmesso un'incredibile e serena determinazione.

Un grazie va poi a Maria, per l'infinita pazienza e la comprensione dettate da un Sentimento vero che in questi mesi e spero anche in futuro non verrà mai meno; le semplici parole non basteranno mai a descrivere la mia gratitudine.

Un ringraziamento particolare lo dedico inoltre a Riccardo e Matteo: per il supporto professionale e tecnico che tante volte mi è stato d'aiuto per rendere questo lavoro migliore, risparmiandomi frustrazioni e perdite di tempo.

Infine, a tutti gli amici che in questi anni mi hanno accompagnato contribuendo a rendermi chi sono oggi, un ringraziamento di cuore, con la speranza che l'amicizia non svanisca mai nonostante i percorsi di ciascuno possano dividerci. Un grazie in particolare a Carlo, con il reciproco augurio che, nonostante gli studi ci abbiano allontanati, possa sempre conservarsi quella stima reciproca che persiste da anni.

Contents

1	Introduction	1
2	Integral formulation with DGA	3
2.1	Maxwell's equations	4
2.2	Maxwell's equations discretization	5
2.3	Magnetic vector potential introduction	7
2.4	Boundary conditions and reduced arrays	8
2.5	Constitutive relations	9
2.6	Final equations system	10
3	Basis functions in a DGA geometry	11
3.1	The grids	11
3.1.1	Primal grid G	11
3.1.2	Dual grid \tilde{G}	11
3.2	Geometrical definitions and properties	12
3.2.1	Partition of hexahedron	14
3.3	Basis vector functions	16
3.4	Constitutive matrices R and M	17
3.4.1	R matrix	17
3.4.2	M matrix	18
4	Planning a volume integral code	21
4.1	Starting data	21
4.2	Domain meshing with polyhedra	21
4.3	R and M matrices building	22
4.4	Equations system assembly	23
4.5	Code schematic	24
5	Data post processing	25
5.1	Why an exact formulation	25
5.2	Magnetic vector potential calculation	26
5.2.1	W_f calculation	27
5.3	Magnetic induction calculation	29
5.4	Discontinuities and singularities	29
5.5	Code structure	30
5.6	Results comparisons	31

6	Starting a simulation	35
6.1	Ω_s discretization	35
6.2	Source calculation	35
6.3	Discretized magnetisation coils	36
6.3.1	Circular coils	36
6.3.2	Square coils	39
6.3.3	Helical coils: stellarators	41
6.3.4	Filamentary coils	45
7	Numerical simulations	47
7.1	Matrices assembly testing	47
7.2	Code benchmark: slab in a vertical field	49
7.3	Code benchmark: sphere in a vertical field	61
7.3.1	Analytical solution of a sphere in uniform \mathbf{B} field	61
7.3.2	ANSYS solution	66
7.3.3	Volume integral code solution	67
7.4	Simulations on a real fusion device: RFX-mod	79
7.4.1	Simulation A: external \mathbf{B} field vertically applied	81
7.4.2	Simulation B: load assembly and saddle coils	82
8	Conclusions and future developments	87
A	Volume integral code: Matlab scripts	89

List of Figures

1.1	RFX-mod view	2
3.1	Hexahedral volume \mathbf{v}	12
3.2	Dual elements of \mathbf{v}	13
3.3	Hexahedron partition $\tau_j^f = \tau_j^{\tilde{f}}$	15
4.1	Code schematic	24
5.1	\mathbf{B}_x field comparison	33
5.2	Magnetic vector potential calculation comparison	34
6.1	Coil faces and barycenters	37
6.2	Circular discretized coil	37
6.3	Circular coil \mathbf{B} field	38
6.4	Solenoidal coil \mathbf{B}_{int} field	40
6.5	Square coil construction	41
6.6	Square coil: $\mathbf{B}(\mathbf{r})$ field calculation	42
6.7	Square coil with not-splitting edges and with edges rotation ($\phi = 45^\circ$)	42
6.8	Fusion reactors: Tokamak VS Stellarator	43
6.9	Helix construction	44
6.10	Helical coil faces construction and matching	44
6.11	From hexahedron to polyhedron (face splitting)	45
6.12	$\mathbf{B}(\mathbf{r})$ field calculation: filamentary and solid coils comparison	46
7.1	Examples of discretization grids: hexahedral (on the left) and polyhedral grid (on the right)	48
7.2	Magneto static field test results	48
7.3	Induced eddy currents in hexahedral meshed volume ($l = 0.04 [m]$, $h = 0.004 [m]$)	50
7.4	Induced eddy currents in polyhedral meshed volume ($l = 0.04 [m]$, $h = 0.004 [m]$)	51
7.5	Discretization grid of the slab	53
7.6	Volume integral solution: current density \mathbf{J}	54
7.7	Reaction \mathbf{B} field above slab due to imaginary current density	55
7.8	Slab section displaying for current flow calculation	56
7.9	Air volume mesh (not visible slab inside) and conducting slab mesh	56
7.10	ANSYS current density \mathbf{J} (<i>top view</i>)	57
7.11	ANSYS imaginary reaction field B_{im} (<i>frontal and 3D views</i>)	58
7.12	\mathbf{B} field evaluation path	59

7.13	Imaginary \mathbf{B}_y field comparison between ANSYS and the volume integral code	61
7.14	Imaginary \mathbf{B}_z field comparison between ANSYS and the volume integral code	62
7.15	Imaginary $ \mathbf{B} $ field comparison between ANSYS and the volume integral code	62
7.16	Spherical polar coordinate reference	63
7.17	Overall view of ANSYS domain discretization: sphere sector and vacuum	66
7.18	Volumes distinction: sphere eighth (on the left) and vacuum (on the right)	67
7.19	J_{im} in the sphere (top view)	68
7.20	B_{im} in the sphere	68
7.21	Comparison between analytical computation of current density \mathbf{J} and ANSYS solution	69
7.22	Coarse sphere grid: on the left an overall view, on the right a top view	70
7.23	Dense sphere grid: on the left an overall view, on the right a top view	71
7.24	Polyhedral volumes details inside coarser spherical domain	71
7.25	Current density \mathbf{J} in the sphere with coarse grid: on the left J_{real} on the right $J_{imaginary}$	72
7.26	Current density \mathbf{J} in the sphere with dense grid: on the left J_{real} on the right $J_{imaginary}$	72
7.27	Induction imaginary field B_{im} on a 2D grid of point	73
7.28	B_{im} field lines map	74
7.29	Current density error ε_{real}^J for real component on coarse grid	75
7.30	Current density error ε_{imag}^J for imaginary component on coarse grid	76
7.31	Current density error ε_{real}^J for real component on dense grid	76
7.32	Current density error ε_{imag}^J for imaginary component on dense grid	76
7.33	Analytical J field trend on a line	77
7.34	B_{im} field on a line: values comparison and related error	78
7.35	B_{im} error maps on a plane	78
7.36	RFX-mod load assembly	79
7.37	RFX-mod toroidal supporting structure (TSS) schematic	80
7.38	RFX-mod TSS mesh	81
7.39	RFX-mod toroidal supporting structure: mesh detail nearby the cuts	81
7.40	RFX-mod simulation results: current density \mathbf{J} in TSS with vertically applied field	82
7.41	Real J : results detail	83
7.42	RFX-mod saddle coils: on the top the 192 coils, on the bottom a detail	84
7.43	Saddle coils simulation: imaginary current density J_{im} results	85
7.44	Saddle coils simulation: progressive enlargement in correspondence of fed coil	86

List of Tables

5.1	B_x values	33
6.1	B_{int} values of a solenoidal coil	39
6.2	B values comparison for solid and filamentary coils modelling	46
7.1	Current flow i values comparison	60
7.2	Current flow i errors	60
7.3	Test problem on a sphere specifications	65
7.4	TSS geometry specifications	80

Chapter 1

Introduction

Nowadays it is undeniable that researches in nuclear fusion to build better and better reactors have produced wide improvements. Knowledges in this field are such that we are absolutely not too far from obtaining an actual machine able to generate a relatively green energy with the aim of producing electric power [11]. Nevertheless, despite theoretical feasibility was demonstrated, whenever the construction of a real device becomes day by day more concrete (and I am referring to ITER [25]) new problems appears. We can thus say that “*complexity*” is the main feature of these reactors: physical theory complexity, dimensions complexity, operations complexity and, finally, structure and geometry complexity. As a consequence it is clear that to face with more and more complex problem we have to provide more and more efficient tools.

If we consider these issues from engineering point of view, and in particular we limit to electrical engineering, we can state that electro-magnetic behaviour of these machines is the main interesting aspect to investigate and that finding an accurate and efficient way to simulate it, is a fundamental requirement to produce a development in this field. An advancement in this sector will allow to better manage all this kind of devices and improve their technical applicability since an efficient simulator is essential in order to have a proficient real time control on machine.

Among various electro-magnetic phenomena, magneto quasi static problems, in particular eddy currents problems, are the most significative in relation with these mentioned devices. Consequently, we will try to build a new code aiming at developing fields simulations with complex geometries to study an eddy current problem in the frequency domain.

In the last two decades researches and improvements in this field were numerous. In particular, **Finite Elements Method** approach, developed initially for mechanical problems and successively adapted for computational electrical simulations, has represented a robust solution and nowadays it is still the main adopted in many cases. Nevertheless, recent enhancements in computational tools and researches in mathematics issues (e.g. matrices representation in calculators), have allowed the development of a wide range of **Integral Formulations** as a valid alternative, especially for electro-magnetic simulations.

In fact the main problem with a FEM code is the necessity of considering all the domain around the magnetic device: in other words, you have to mesh air. This fact represents a limitation since in most cases we are not interested in field outside objects but we are only interested in the inner one (e.g. if we are testing a conducting device). To avoid this fact the use of an integral formulation is possible: mesh is limited only to conducting and magnetic regions and Maxwell’s equations are discretized from their

integral form. This solution from one hand allows to have a smaller domain but to the other hand it leads to have a full system of equations which represents an obstacle in term of computational efficiency.

Despite this issue the choice of an integral formulation represents an unquestionable advantage if we want to study complex and intricate machine such as fusion reactor devices because it allow to accomplish good results even with a relatively coarse discretization grid.

For instance we can consider a magnetic confinement machine such as RFX-mod of which we report in figure 1.1 a toroidal section. From the picture we can immediately understand this “complexity” of which we were talking about before: the machine is indeed composed by several different layers of distinct material each of them with a specific function and each of them not trivially shaped.

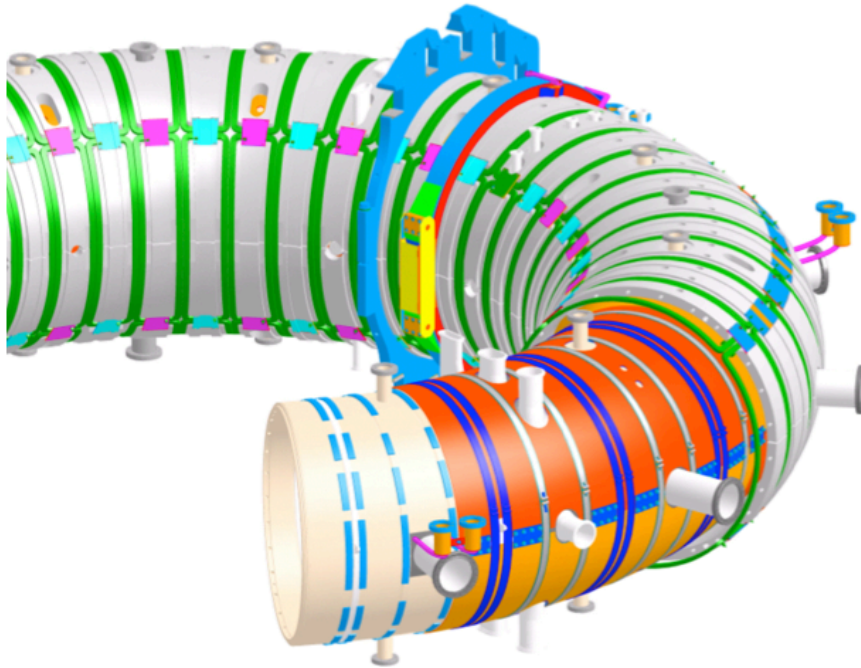


Figure 1.1: RFX-mod view

If we want to perform some computations on this geometry, in order to study and improve MHD stability [8] or to study some particular behaviours of the machine to modify it [9] we will necessary do some simplifications since a correct and complete discretization of this geometry with a FEM software is too difficult or requires too much time. Actually, as every computational engineer confirms, most of problems whenever you have to create a mesh on a such intricate domain, are not due to geometrical complexity itself but are a consequence of trying to correctly mesh the borders between actual object and air (or vacuum) surrounding it, in particular if surface is not plane. Trying to avoid this unpleasant drawback is surely a good reason to explore new solutions.

Chapter 2

Integral formulation with DGA

The discretization of a physical domain is the first basic step on which is generally built a FEM simulator. With respect to an electro-magnetic problem, the discretization of partial differential Maxwell's equations presents some critical aspects which would not be involved if we were considering a mechanical application. For this reason, the recent development of the so called **Discrete Geometric Approach** has represented an important step: by this way, exploiting geometrical properties of the domain grid, is possible to write Maxwell's equation in term of fluxes and circulations on the chosen grid. This method brings to a sparse system of equations similar to the one obtainable with a pure finite element formulation such as the Galérkin approach.

As said the choice of an integral formulation represents an advantage in many configurations (for instance, if we are trying to solve a field in an intricate region), for this reason the code that will be developed is precisely based on an **integral magneto static formulation with a discrete geometry domain**. The aim is exploiting the advantages of the DGA in term of equations accuracy and use continuity law, Ampere's law and Faraday's law in their integral formulation to obtain a compactly meshed domain.

The goal of this chapter is thus to derive a unique set of equations and then solve them with a calculator [1]. The equations set is built from:

- electro-magnetic relations i.e. Maxwell's relations in integral form (continuity, Ampere's and Faraday's laws);
- two constitutive relations written in discrete configuration (i.e. as matrices) which will be the discrete approximation of Ohm's constitutive relation (electro-static behaviour of matter) and of the link between current density and the magnetic vector potential (magnetic behaviour of matter).

For our computational aim it is essential to underline, as previously said, the substantial difference between the two groups of equation once they will be discretized: the latter will exactly satisfied the original laws thanks to the use of DGA, the former will be only an approximation¹. The consequent error is known as *constitutive error*.

In order to reach our goal a discrete geometry subdivision of the domain Ω is needed. In this first step the domain meshing is given and its construction and its properties will be detailed later. We mark in Ω a conducting region Ω_c with its resistivity in

¹Regarding a FEM code the behaviour is opposite: the constitutive relations are exactly while the Maxwell's laws are approximated.

function of position vector $\eta(\mathbf{r})$ and a source region Ω_s . The remaining region is the insulating region $\Omega_a = \Omega - \Omega_c \cup \Omega_s$. The magnetic matter is considered linear and its permeability μ is uniform in Ω .

The conducting region Ω_c is subdivided with a pair of oriented grids G and \tilde{G} . The **primal grid** G is constituted by polyhedral elements characterized by v volumes, f faces, e edges and n nodes. Each of them has an orientation. The **dual grid** \tilde{G} is built from the primal one, as its barycentric subdivision, and has \tilde{v} volumes quantitatively equal to n nodes, \tilde{f} faces quantitatively equal to e edges, \tilde{e} edges quantitatively equal to f faces and, finally, \tilde{n} nodes quantitatively equal to v volumes. For these elements orientation is chosen by the orientation of primal grid G .

2.1 Maxwell's equations

Before starting the derivation of the discretized integral formulation for an eddy currents problem the equations which rule the electromagnetic phenomena are here recalled: Maxwell's equations. These relations are reported in their differential formulation and only successively they will be used in integral form.

The well known equations set is:

1. **Gauss's law** for the **electric flux density** \mathbf{D}

$$\nabla \cdot \mathbf{D} = \rho \quad (2.1)$$

with ρ as volume charge density.

2. **Gauss's law** for the **magnetic flux density** \mathbf{B}

$$\nabla \cdot \mathbf{B} = \mathbf{0} \quad (2.2)$$

3. **Faraday-Neumann' law**

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.3)$$

where \mathbf{E} is the electric field.

4. **Ampere-Maxwell' law**

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (2.4)$$

with \mathbf{H} as magnetic field and \mathbf{J} current density.

From these four relations a fifth law is derivable, the **continuity law**:

$$\begin{aligned} \nabla \cdot \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \right) &= \nabla \cdot (\nabla \times \mathbf{H}) = \mathbf{0} \\ \implies \nabla \cdot \mathbf{J} &= -\nabla \cdot \frac{\partial \mathbf{D}}{\partial t} = -\frac{\partial (\nabla \cdot \mathbf{D})}{\partial t} = -\frac{\partial \rho}{\partial t} \end{aligned}$$

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} \quad (2.5)$$

For our studied case, an eddy current problem which belongs to the magnetoquasi-static problems category, with respect to the above relations there is only a further consideration to do: in (2.1), $\frac{\partial \mathbf{D}}{\partial t}$ term is negligible; since we are not working with high frequencies (*MHz* or *GHz*) Ampere-Maxwell' equation can be approximated as

$$\nabla \times \mathbf{H} \approx \mathbf{J} \quad (2.6)$$

2.2 Maxwell's equations discretization

As previously said we are now interested in discretizing the Maxwell's equations for a domain subdivided as a discrete geometry.

Continuity law

The discretized continuity law results to be:

$$\mathbf{D}\mathbf{i}(t) = \mathbf{0} \quad (2.7)$$

Terms involved are:

- **volume-face incidence matrix \mathbf{D}** , written on the primal grid G , whose dimensions are $v \times f$. This matrix operates in the discrete domain such as the divergence operator “ ∇ ” does in continuous domain;
- **current array $\mathbf{i}(t)$** whose elements $i_i(t)$ are the f -currents sampled through the faces of G . As a consequence this vector is a $f \times 1$ vector. This array has ideally the function of representing the real current density vector $\mathbf{j}(\mathbf{r}, t)$. More precisely it represents the fluxes of \mathbf{j} through the f faces of the primal grid. Symbolically this is marked as $\rho_f \mathbf{j}(\mathbf{r}, t)$ where the operator ρ_f yields the fluxes of \mathbf{j} through the f -surface of G . Approximation between $\mathbf{i}(t)$ and $\rho_f \mathbf{j}(\mathbf{r}, t)$ is as more precise as smaller is the grain h chosen for the mesh grid G .

By this way the differential relation presented in (2.5), with $\rho = 0$, is integrated and discretized as:

$$\nabla \cdot \mathbf{J} = 0 \implies \int_{\Omega} (\nabla \cdot \mathbf{J}) d\Omega = \int_{\partial\Omega} (\mathbf{J} \cdot \mathbf{n}) dS$$

Ampere's law

As done above for the continuity law here is reported the **discretization of right hand side** of Ampere's law:

$$\mathbf{i}(t) = \mathbf{C}\mathbf{T}(t) \quad (2.8)$$

This relation is referred to (2.6) considering that:

$$\nabla \times \mathbf{H} = \mathbf{J} \implies \int_{\partial\Omega} (\nabla \times \mathbf{H}) dS = \int_{\partial\Omega} \mathbf{J} dS$$

and thanks to Kelvin-Stokes' theorem we have

$$\oint \mathbf{H} \cdot \mathbf{t} dl = \int_{\partial\Omega} \mathbf{J} dS \quad (2.9)$$

Now, being us interested only on right equation member, it becomes:

$$\int_{\partial\Omega} \mathbf{J} dS = \oint \mathbf{T}(\mathbf{r}, t) \cdot \mathbf{t} dl \quad (2.10)$$

Comparing (2.8) with (2.10) it is clear that:

- **face-edge incidence matrix \mathbf{C}** is used as the discrete counterpart of circulation. This matrix has dimensions $f \times e$ and, as it was for \mathbf{D} , it is referred to the primal grid G ;
- the introduced array $\mathbf{T}(t)$ stays for the **integral of electric potential** along primal grid edges. Consequently the right hand side of (2.8) approximates the circulation $\rho_e \mathbf{T}(\mathbf{r}, t)$ along the edges of primal grid, of **actual electric potential integral $\mathbf{T}(\mathbf{r}, t)$** . $\mathbf{T}(t)$ has dimensions $e \times 1$. Similarly to the previous operator, ρ_e yields an array of circulations along the primal grid edges.

Faraday's law

If we integrate both sides of (2.1), applying Kelvin-Stokes' theorem as done previously, we obtain:

$$\begin{aligned} \int_{\partial\Omega} (\nabla \times \mathbf{E}) dS &= - \int_{\partial\Omega} \left(\frac{\partial \mathbf{B}}{\partial t} \right) dS \\ \implies \oint \mathbf{E} \cdot \mathbf{t} dl &= - \frac{\partial}{\partial t} \int_{\partial\Omega} (\mathbf{B} \cdot \mathbf{n}) dS \end{aligned}$$

Since the left side is known and defined as the voltage \mathbf{v} , defined the flux φ as

$$\varphi = \int_{\partial\Omega} \mathbf{B} \cdot \mathbf{n} dS$$

the previous equations yield the integral formulation of Faraday's law (for a fixed line):

$$\mathbf{v} = - \frac{d\varphi}{dt} \quad (2.11)$$

Now, if we apply this relation to our discrete domain Ω , the following equation is obtained:

$$\mathbf{C}_r^T \tilde{\mathbf{v}}_r(t) = - \frac{d\tilde{\varphi}_r(t)}{dt} \quad (2.12)$$

The “ r ” subscript is referred to the reduced arrows of the involved entities. This expedient, as later will be explained, is due to boundary conditions applied to the conductive surfaces of Ω_c .

Focusing on the discretized Faraday's relation it is possible to identify:

- the already known face-edge incidence matrix \mathbf{C} which is here transposed and reduced to the only faces not belonging to the boundary regions;
- **voltage array** $\tilde{\mathbf{v}}(t)$, which has been reduced in the same way of its incidence matrix. This vector, whose dimensions are $\tilde{e} \times 1$, is referred to the dual grid \tilde{G} and it approximates the $\rho_{\tilde{e}}\mathbf{e}(\mathbf{r}, t)$ array of the circulations of the electric field $\mathbf{e}(\mathbf{r}, t)$ along the edges of the dual grid. The operator $\rho_{\tilde{e}}$ is similar to the ones of the previous sections with the peculiarity that this one acts on dual grid edges \tilde{e} .
- **flux array** $\tilde{\varphi}(t)$ (from which the reduced $\tilde{\varphi}_r(t)$ is obtainable) approximates the array $\rho_{\tilde{f}}\mathbf{b}(\mathbf{r}, t)$ of the fluxes of the actual magnetic induction $\mathbf{b}(\mathbf{r}, t)$ through the faces of the dual grid \tilde{G} . Consequently this vector has dimensions $\tilde{f} \times 1$. As above, $\rho_{\tilde{f}}$ is an operator referred to dual grid faces \tilde{f} .

2.3 Magnetic vector potential introduction

Before introducing the last two missing relations, in order to obtain the final equations system a magnetic vector potential \mathbf{A} is needed.

This potential, which depends on time and space, is defined as:

$$\nabla \times \mathbf{A} = \mathbf{B} \quad (2.13)$$

Since, by definition, for a given field \mathbf{F} is true that $\nabla \cdot \nabla \times \mathbf{F} = \mathbf{0}$, by this way (2.1) is satisfied and \mathbf{A} is defined except for an arbitrary field ψ .

The differential relations reported above can be integrally recast for our studied case. In particular we are interested in separating the magnetic vector potential $\mathbf{A}(\mathbf{r}, t)$ due to unknown current density $\mathbf{j}(\mathbf{r}, t)$ in the conducting region Ω_c from the magnetic potential $\mathbf{A}_s(\mathbf{r}, t)$ due to known current density $\mathbf{j}_s(\mathbf{r}, t)$ in the source region Ω_s .

Symbolically:

$$\mathbf{A}(\mathbf{r}, t) = \frac{\mu}{4\pi} \int_{\Omega_c} \frac{\mathbf{j}(\mathbf{p}, t)}{|\mathbf{r} - \mathbf{p}|} d\Omega_p \quad (2.14)$$

$$\mathbf{A}_s(\mathbf{r}, t) = \frac{\mu}{4\pi} \int_{\Omega_s} \frac{\mathbf{j}_s(\mathbf{p}, t)}{|\mathbf{r} - \mathbf{p}|} d\Omega_p \quad (2.15)$$

In previous expressions $d\Omega_p$ is the volume differential around calculating point \mathbf{p} .

Discretization

Last two relations have to be discretized in order to obtain a coherent set of equations. For this reason an array $\tilde{\mathbf{a}}(t)$ is introduced: this array approximates the array $\rho_{\tilde{e}}\mathbf{A}(\mathbf{r}, t)$ of circulation of the magnetic vector potential $\mathbf{A}(\mathbf{r}, t)$ due to unknown current density along the edges of dual grid \tilde{G} . Moreover an array $\rho_{\tilde{e}}\mathbf{A}_s(\mathbf{r}, t)$ is needed, with the aim of taking into account the contribution of the circulation of known magnetic vector potential $\mathbf{A}_s(\mathbf{r}, t)$ along \tilde{e} . Differently from the other, this array will not be approximated since it is well known, being due to the known current density in source region. In conclusion this last one will be considered as a known term. Both of them has dimensions equal to $\tilde{e} \times 1$.

Thanks to the above consideration and since

$$\varphi = \int_{\partial\Omega} \mathbf{B} \cdot \mathbf{n} dS = \int_{\partial\Omega} (\nabla \times \mathbf{A}) \cdot d\mathbf{S} = \oint \mathbf{A} \cdot \mathbf{t} dl \quad (2.16)$$

it is possible to rewrite the flux array used in (2.12) as follows:

$$\tilde{\varphi}(t) = \mathbf{C}^T (\tilde{\mathbf{a}}(t) + \rho_{\tilde{e}} \mathbf{A}_s(\mathbf{r}, t)) \quad (2.17)$$

2.4 Boundary conditions and reduced arrays

All the previous relations (except for the (2.12)) were indiscriminately written for each element of the domain. This is uncorrected because that relations are not constantly true in whole Ω , therefore corrections are necessary. These corrections are a consequence of boundary conditions.

We have to consider that, on the surface $\partial\Omega_c$ of the conducting region

$$\mathbf{j}(\mathbf{r}, t) \cdot \mathbf{n} = \mathbf{0}$$

Therefore:

1. there are some null elements in the current array $\mathbf{i}(t)$ since some faces of G in Ω_c are not crossed by current. These faces, which will be marked as f_b , will be eliminated from the complete vector yielding a reduced vector $\mathbf{i}_r(t)$ whose dimensions are $(f - f_b) \times 1$;
2. also the approximate array of electric vector potential $\mathbf{T}(t)$ has to be reduced in order to take into consideration the influence of the previous null current faces into the approximate circulation of $\mathbf{T}(\mathbf{r}, t)$. This implies that the f_b boundary faces can be associated to l_b boundary edges, belonging to $\partial\Omega_c$, whose circulation should be null too or should anyway have an established value. Therefore, in such a way, it is possible to define a reduced array $\mathbf{T}_r(t)$ of dimension $(l - l_b) \times 1$;
3. with the same procedure used for the previous array, we can establish a **one-to-one correspondence** between geometrical elements belonging to primal and dual grid and we can “reflect” the boundary conditions, reported so far, from G to \tilde{G} . So, considering the flux vector $\tilde{\varphi}(t)$, it is possible to define its reduced one, marked as $\tilde{\varphi}_r(t)$, which will not contain that fluxes linked with the \tilde{f}_b faces of \tilde{G} numerically equal and linked to the l_b edges of $\partial\Omega_c$ on G identified in the previous section. Consequently this new vector has dimension $(l - l_b) \times 1$. Similarly, a $\tilde{\mathbf{v}}_r(t)$ array of dimension $(\tilde{f} - \tilde{f}_b) \times 1$ will exist considering and eliminating that voltages correlated with the \tilde{l}_b edges of \tilde{G} linked to the f_b faces on G . Finally, a reduced magnetic vector potential array $\tilde{\mathbf{a}}_r(t)$ of dimension $(\tilde{f} - \tilde{f}_b) \times 1$ is defined.

After last considerations hereafter the complete set of reduced equations is reported:

$$\mathbf{D}_r \mathbf{i}_r(t) = \mathbf{0} \quad (2.18)$$

$$\mathbf{i}_r(t) = \mathbf{C}_r \mathbf{T}_r(t) \quad (2.19)$$

$$\mathbf{C}_r^T \tilde{\mathbf{v}}_r(t) = -\frac{d\tilde{\varphi}_r(t)}{dt} \quad (2.20)$$

$$\tilde{\varphi}_r(t) = \mathbf{C}_r^T \left(\tilde{\mathbf{a}}_r(t) + \rho_{\tilde{e}_r} \mathbf{A}_s(\mathbf{r}, t) \right) \quad (2.21)$$

2.5 Constitutive relations

As initially said, besides the discretized Maxwell's equations a link between current and voltage and between current and magnetic vector potential is needed. Discretization process of these laws is the main issue of this work and, for this reason, it will be deeply debated in a dedicated chapter.

Ohm's constitutive relation

The first equation concerns matter behavior under an electric field. This law, microscopically, can be written as:

$$\mathbf{e}(\mathbf{r}, t) = \boldsymbol{\eta}(\mathbf{r}) \mathbf{j}(\mathbf{r}, t)$$

where $\boldsymbol{\eta}(\mathbf{r})$ is the medium resistivity tensor.

For now we limit to report the final result, which leads to

$$\tilde{\mathbf{v}}(t) = \mathbf{R} \mathbf{i}(t) \quad (2.22)$$

whose reduced form, considering the boundary conditions, is

$$\tilde{\mathbf{v}}_r(t) = \mathbf{R}_r \mathbf{i}_r(t) \quad (2.23)$$

Therefore, the introduced \mathbf{R} matrix of dimension $f \times f$ is responsible of the link between \mathbf{j} and \mathbf{e} ; in other words \mathbf{R} is the discrete counterpart of $\boldsymbol{\eta}(\mathbf{r})$.

Magnetic vector potential linking

The following relation is the discretization of (2.14) and it can be written as

$$\tilde{\mathbf{a}}(t) = \mathbf{M} \mathbf{i}(t) \quad (2.24)$$

where, similarly to what has been done above, M is a $f \times f$ matrix responsible of the link between \mathbf{A} and \mathbf{j} .

Even for this equation it is possible a reduced formulation, due to boundary conditions, such as

$$\tilde{\mathbf{a}}_r(t) = \mathbf{M}_r \mathbf{i}_r(t) \quad (2.25)$$

2.6 Final equations system

The substitution in ((2.20)) of ((2.23)) for $\tilde{\mathbf{v}}(t)$, ((2.21)) for $\tilde{\boldsymbol{\varphi}}_r(t)$, ((2.25)) for $\tilde{\mathbf{a}}_r(t)$ and ((2.19)) for $\mathbf{i}_r(t)$, yields to the final linear differential system:

$$\mathbf{C}_r^T \mathbf{M}_r \mathbf{C}_r \frac{d}{dt} \mathbf{T}_r(t) + \mathbf{C}_r^T \mathbf{R}_r \mathbf{C}_r \mathbf{T}_r(t) = -\mathbf{C}_r^T \frac{d}{dt} \rho_{\tilde{e}_r} \mathbf{A}_s(\mathbf{r}, t) \quad (2.26)$$

This system of equations can be directly solved in frequency domain if we are investigating an isofrequential phenomenon with sinusoidal sources; otherwise if we are treating a general situation, for instance a transient phenomenon, it can be solved in time domain using a theta-method.

Chapter 3

Basis functions in a DGA geometry

As previously announced, the discretization of continuity law and of magnetic vector potential in function of current is not a trivial process.

First of all it is necessary to start from geometrical elements of the domain Ω whose properties will lead to define a set of basis functions $\mathbf{w}_i(\mathbf{p})$ from which \mathbf{R} and \mathbf{M} matrices are obtained. These functions firstly have to ensure the consistency and the stability of the final equations system. This request is reflected on constitutive matrices and thus on the choice of basis functions: **stability** requirement is satisfied whenever \mathbf{R} and \mathbf{M} are symmetric and positive defined, while **consistency** is reached if, for element-wise uniform fields, matrices exactly map circulation onto fluxes or viceversa.

Presently used basis functions are supposed to be combined with an energetic approach and they yield to a method valid for general polyhedral grids with the only specification that every face of the grid must be planar [2].

3.1 The grids

As already partially known, the geometric subdivision is based on two interlocked grids, one as barycentric subdivision of the other. In order to well define and well understand the construction of the various geometric elements used to developed the integral formulation of the problem, now we will refer only upon a single hexhedral volume \mathbf{v} of volume v as reported in figure 3.1.

3.1.1 Primal grid G

With respect to this volume \mathbf{v} other geometrical entities, belonging to the primal grid, are identified: f faces, l edges and n nodes. Each of them can be represented through an appropriate array with a given orientation. Consequently it is useful to assemble these vectors into geometrical tensors, one for each geometric category; we will obtain a face tensor \mathbf{f} , an edge tensor \mathbf{e} and a node tensor \mathbf{n} . Although the edge-vector e_j definition is banal (a vector linking one node to another), the same cannot be said for what regard the face-vector definition: it is intended as that normal vector f_j , whose module is equal to the face area, applied at the centre of gravity of the face.

3.1.2 Dual grid \tilde{G}

Once primal grid geometric elements are defined, it is possible to derive their **dual elements**. Differently from what previously seen but always using a *tilde* above when

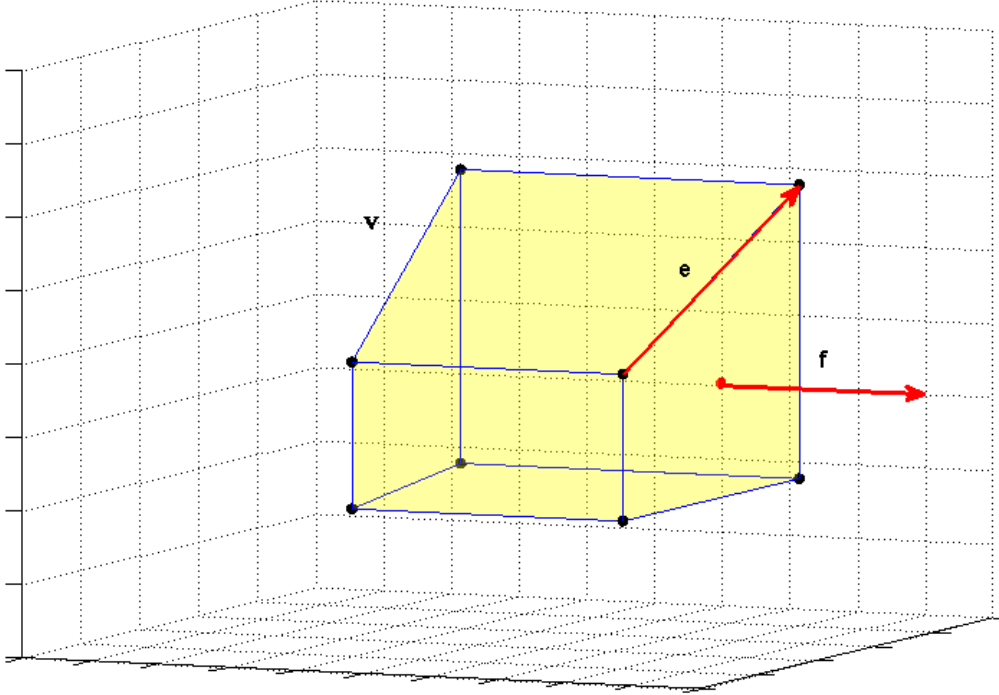


Figure 3.1: Hexahedral volume \mathbf{v}

referring to a dual element, we will now denote the dual of a primal volume v_j as the dual node \tilde{v}_j , the dual of a primal face f_j as the dual edge \tilde{f}_j , the dual of a primal edge e_j as the dual face \tilde{e}_j and finally, the dual of a primal node n_j as the dual volume \tilde{n}_j ¹. As it has happened for the primal grid elements, also these geometric elements will be assembled into tensors, yielding $\tilde{\mathbf{v}}$, $\tilde{\mathbf{f}}$, $\tilde{\mathbf{e}}$ and $\tilde{\mathbf{n}}$.

The construction of this second grid, as shown in figure 3.2, is based on barycentric points of the primal one. The basic element is volume barycenter, precisely an arbitrary inner point of the volume \mathbf{v} . This point, which has been defined as the dual node \tilde{v}_j , will exactly be a node of $\tilde{\mathcal{G}}$. Starting from this point, the other elements will come out: connecting the volume barycenter to a face barycenter a dual edge vector \tilde{f}_j is obtained; by analogy, a proper set of dual edges will mark a dual face \tilde{e}_j in a one-to-one correspondence with a primal edge e_j , as its barycentric subdivision (in fact, a dual face, besides being delimited by some dual edges, is always crossed by a primal edge in its barycenter).

3.2 Geometrical definitions and properties

The following properties are fundamental in order to successively define the basis functions. They originate from basic algebraic operations between tensors. The geometric elements involved, in particular dual faces \tilde{e}_j and dual edges \tilde{f}_j , are supposed to be

¹Due to theory complexity the choice of a different notation between chapter 2 and chapter 3 is necessary in order to assure from one hand an immediate comprehension of concepts in the previous chapter, to the other hand an easy equation representation in the current chapter.

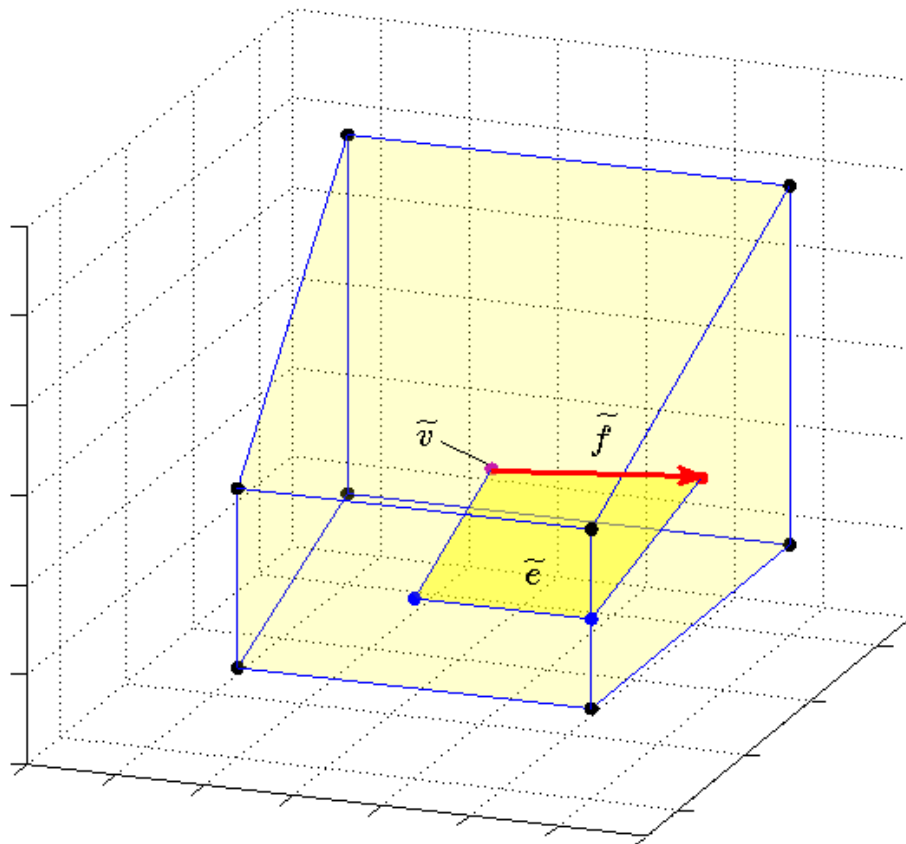


Figure 3.2: Dual elements of \mathbf{v}

outer oriented thus each element pair (f_j, \tilde{f}_j) and (e_j, \tilde{e}_j) has a congruent orientation.

In the following relations we will refer to a general geometric element “ r_j ” which can equally be either a face or an edge thus its dual one will result to be labelled as “ \tilde{r}_j ”.

Definition We will define the **double tensor** T_i as

$$T_i = \tilde{r}_i \otimes r_i \quad \forall i = 1, \dots, R \quad (3.1)$$

with R equal to either number of faces or number of edges.

In (3.1) the Kronecker product \otimes was used: by this way an arbitrary component (h, k) of T_i will result to be as $(T_i)_{hk} = (\tilde{r}_i)_h (r_i)_k$ where $(r_i)_k$ is the h -th component of r_i with $h, k = 1, \dots, 3$.

Definition Directly from the previous definition, **trace of \mathbf{T}** follows as

$$t_i = tr(T_i) = \tilde{r}_i \cdot r_i \quad \forall i = 1, \dots, R \quad (3.2)$$

where “ \cdot ” is the inner product between \tilde{r}_i and r_i .

Property The product $T_i x$ between T_i and a generic vector x is a vector and it is true that

$$T_i x = (r_i \cdot x) \tilde{r}_i \quad (3.3)$$

Property It has been proved that the following identity is valid:

$$\mathbf{T} = \sum_{i=1}^R T_i = \sum_{i=1}^R \tilde{r}_i \otimes r_i = |v| \mathbf{I} \quad (3.4)$$

where \mathbf{I} is the identity matrix and $|v|$ the volume value.

As a consequence, \mathbf{T} tensor is symmetric and it is valid that $tr(\mathbf{T}) = 3|v|$.

3.2.1 Partition of hexahedron

Before proceeding with basis vector functions dissertation, we now pause on hexahedron \mathbf{v} partition.

In fact, for each volume of the domain Ω , it is possible to identify some subregions according to the geometrical elements previously described such as faces and dual edges or edges and dual faces. Each of these elements pairs is responsible of a different subdivision. In particular we will now focus on partition due to *face-dual edge* correspondence which is the one chosen for our integral code construction. Nevertheless the same results are obtainable also with *edge-dual face* partition of which will be reported only a brief description.

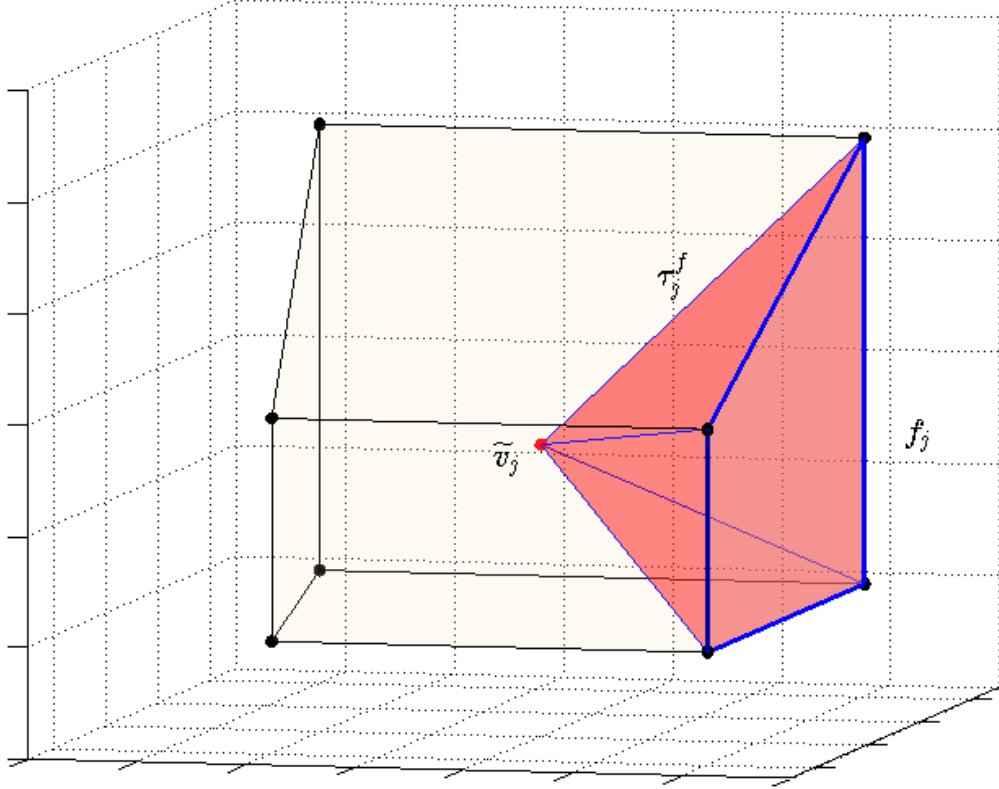


Figure 3.3: Hexahedron partition $\tau_j^f = \tau_j^{\tilde{f}}$

In figure 3.3 is shown a partition $\tau_j^f = \tau_j^{\tilde{f}}$. This subregion, which is practically a tetrahedron, can be delimited and identified either by the j -th face f_j or equivalently by the j -th dual edge \tilde{f}_j thus the double notation employed.

It is clear that the result is the pyramid τ_j^f which has as base f_j and as apex the \tilde{v}_j dual node. Similarly the same partition is obtained considering the pyramid $\tau_j^{\tilde{f}}$ identified by the same dual node and by dual edge \tilde{f}_j which univocally determines the pyramid base thanks to the one-to-one correspondence between primal and dual elements.

With a specular reasoning it is determined the subregion $\tau_i^e = \tau_i^{\tilde{e}}$ as that “double tetrahedron” whose tetrahedra have as vertices the dual node \tilde{v}_j , the pair of node of e_j , the faces barycenters which have e_j in common and finally the edge e_j barycenter.

Property In a subregion τ_i^r we have that

$$t_i = tr(T_i) = 3|\tau_i^r| \quad (3.5)$$

with $|\tau_i^r|$ subregion volume value.

3.3 Basis vector functions

For a given vector field $x(p)$ in v , such as an electric field \mathbf{E} or a current density field \mathbf{J} , the following integral quantity is defined:

$$X_i^r = \int_{r_i} x(p) \cdot dr \quad (3.6)$$

where X_i^r is also called “*Degree of Freedom*”.

This generic integral can be interpreted either as a flux if r_i is a face of the volume or as a circulation if r_i is an edge (faces and edges belonging indifferently to primal or dual grid). By this way we can obtain the electro-motive force along a primal edge e_i integrating the electric field \mathbf{E} or we can obtain the current through a face integrating \mathbf{J} .

Under the conditions that $x(p)$ is uniform in v , assuring that x is constant in each point of the volume, from (3.4), right multiplying it by x , considering that $X_i^r = x \cdot r_i$ and using the previous relations, we can write that

$$x = \frac{1}{|v|} \sum_{i=1}^R X_i^r \tilde{r}_i \quad (3.7)$$

with $i = 1, \dots, R$.

Moreover, the identity

$$\mathbf{I} = \frac{T_j}{t_j} + \left(\mathbf{I} - \frac{T_j}{t_j} \right) \quad (3.8)$$

by right multiplying by x both its members, thanks to relation (3.3), yields

$$x = \frac{X_j^r}{t_j} \tilde{r}_j + \left(\mathbf{I} - \frac{T_j}{t_j} \right) x \quad (3.9)$$

Now, expressing x through (3.7) in the right member of (3.9), the result is:

$$x = \sum_{i=1}^R w_i^r(p) X_i^r \quad (3.10)$$

where $w_i^r(p)$ are the searched **basis vector functions** whose expressions are:

$$w_i^r(p) = \frac{\tilde{r}_j}{t_j} \delta_{ij} + \left(\mathbf{I} - \frac{T_j}{t_j} \right) \frac{\tilde{r}_i}{|v|} \quad (3.11)$$

for each $p \in \tau_j^r$, with $j = 1, \dots, R$. Operator δ_{ij} is the so called “*Delta of Kronecker*” which is equal to 1 if $i = j$ otherwise its value is 0.

Here defined basis vector functions are piece-wise uniform in v and uniform in each τ_j^r for each $j = 1, \dots, R$. For coherent construction of constitutive matrices \mathbf{R} and \mathbf{M} the following three properties are necessary ².

²Proofs of these properties are reported in [2].

Property 1. The above defined basis functions are linearly independent and are such that

$$\int_{r_j} w_i^r(p) \cdot dr = \delta_{ij} \quad (3.12)$$

for $i, j = 1, \dots, R$.

Property 2. According to (3.10), the basis vector functions allow to **exactly** represent a uniform vector field from its *Degrees of Freedom*.

Property 3. Consistency conditions is

$$\int_v w_i^r(p) dv = \tilde{r}_i \quad (3.13)$$

with $i = 1, \dots, R$.

3.4 Constitutive matrices R and M

Once geometry has been described, basis vector functions and their properties has been defined, we now have all the necessary instruments to find a coherent expression for both \mathbf{R} and \mathbf{M} matrix.

3.4.1 Ohm's constitutive relation: R matrix

We are now interested in finding a double tensor, i.e. a matrix, which will be called \mathbf{R} , with the aim of mapping a vector field x into another one referred as y considering a single polyhedron v . In symbol:

$$y = rx \quad (3.14)$$

where the tensor r , representing the material properties, is symmetric, positive defined and homogeneous in v .

If y is the voltage and x the current, the previous relation is the discrete counterpart of Ohm's law we were looking for in 2.5 and thus (3.14) will lead to (2.22).

If we consider an arbitrary pair of geometric elements on primal and dual grids r_i and \tilde{r}_i with $i = 1, \dots, R$ we can consequently derive the corresponding Degrees of Freedom respectively of x and y as done in the previous section which will be

$$X_i^r = \int_{r_i} x \cdot dr$$

and

$$Y_i^{\tilde{r}} = \int_{\tilde{r}_i} y \cdot dr$$

and referring to their arrays of dimensions $R \times 1$ denoted as \mathbf{X}^r and $\mathbf{Y}^{\tilde{r}}$ we can recast (3.14), in order to discretize it, such as:

$$\mathbf{Y}^r = \mathbf{R}^{rr}(m)\mathbf{X}^r \quad (3.15)$$

where m is the material parameter, in our case it will be the resistivity η .

Just found $\mathbf{R}^{rr}(\eta)$ matrix, whose dimensions are $R \times R$, is the constitutive matrix which maps \mathbf{X}^r onto \mathbf{Y}^r . It is *not exactly known* thus it is an approximation of Ohm's law which leads to the so called "**constitutive error**" affecting the overall solution of the system. This matrix respects the expected requirements since it is symmetric and positive defined and moreover, it guarantees the consistency requirement i.e. it exactly map the pair of given fields at least when they are uniform in v .

The R matrix expression is obtained through the forecast energetic approach which practically allows to write the matrix only thanks to basis vector functions and their previously explained properties. As a consequence, for a volume v , we have

$$R_{ij}^{rr}(\eta) = \int_v w_i^r(p) \cdot \eta w_j^r(p) dv \quad i, j = 1, \dots, R \quad (3.16)$$

Because of basis vector functions properties, integration of (3.16) can be exactly done without resorting to approximated numerical formulations, thus, for each $p_k \in \tau_k^r$ with $k = 1, \dots, R$ we can re-write the previous relation as

$$R_{ij}^{rr}(\eta) = \sum_{k=1}^R w_i^r(p_k) \cdot \eta w_j^r(p_k) \frac{t_k}{3} \quad (3.17)$$

The reported expression shows that, for a complete domain Ω with more than a single polyhedral volume v_j the overall matrix does not depend on relations between two different volumes but it is defined for each volume independently from the others. For this reason the final matrix $\mathbf{R}^{rr}(\eta)$ will be a sparse matrix with lots of null entries.

3.4.2 Constitutive relation between current and magnetic vector potential: M matrix

We are looking for a matrix M which can map the current density $\mathbf{j}(\mathbf{r}, t)$ onto the magnetic vector potential $\mathbf{A}(\mathbf{r}, t)$.

Firstly we need to express circulation of the magnetic vector potential along the edges of dual grid \tilde{G} , previously denoted as $\rho_e \tilde{\mathbf{A}}(\mathbf{r}, t)$, exploiting here introduced basis functions. In fact, if $\mathbf{A}(\mathbf{r}, t)$ is uniform in a volume v_k thanks to (3.13), it is true that:

$$\int_{\tilde{f}_i} \mathbf{A}(\mathbf{r}, t) dl_r = \int_{v_k} \mathbf{A}(\mathbf{r}, t) \cdot w_i^k(\mathbf{r}) dv_r \quad i = 1, \dots, f^k \quad (3.18)$$

being dl_r and dv_r respectively an infinitesimal length or volume around \mathbf{r} , and being f^k the number of faces of v_k .

Since the left hand side of the expression indeed is the circulation of $\mathbf{A}(\mathbf{r}, t)$, we can conclude, according to the definition of $\tilde{a}_i(t)$ given in section 2.3, that

$$\tilde{a}_i^k(t) = \int_{v_k} \mathbf{A}(\mathbf{r}, t) \cdot \mathbf{w}_i^k(\mathbf{r}) dv_r \quad i = 1, \dots, f^k \quad (3.19)$$

and doing that we are approximating $\mathbf{A}(\mathbf{r}, t)$ as locally uniform in the considered volume.

After having expressed the discretized circulation of magnetic vector potential $\tilde{a}_i^k(t)$ using the basis vector functions, we now have to exploit relation between the current density $\mathbf{j}(\mathbf{r}, t)$ and the integral magnetic vector potential expression reported in (2.14).

In order to express this relation we now introduce the quantity $\boldsymbol{\pi}_f^k(\mathbf{r})\mathbf{i}^k(t)$ as current density field, where $\boldsymbol{\pi}_f^k(\mathbf{r})$ yields a vector field from the vector of fluxes through the faces of v_k , previously denoted as $\mathbf{i}^k(t)$. In such a way the just defined vector field approximates the current density vector $\mathbf{j}(\mathbf{r}, t)$.

Beside above definition, we will need following theorem [1].

Theorem *The functions $\mathbf{w}_i^k(\mathbf{r})$ with $i = 1, \dots, f^k$ form a basis and for the current density field*

$$\boldsymbol{\pi}_f^k(\mathbf{r})\mathbf{i}^k(t) = \sum_{i=1}^{f^k} i_i^k(t)\mathbf{w}_i^k(\mathbf{r}) \quad (3.20)$$

the degrees of freedom $i_i^k(t)$ with $i = 1, \dots, f^k$ are fluxes of $\boldsymbol{\pi}_f^k(\mathbf{r})\mathbf{i}^k(t)$ across the faces of v_k .

Whenever $\mathbf{j}(\mathbf{r}, t)$ can be considered as locally uniform in each v_k , thanks to (3.20), substituting it in (2.14) we obtain:

$$\mathbf{A}(\mathbf{r}, t) = \frac{\mu}{4\pi} \sum_{k=1}^v \sum_{j=1}^{f^k} i_j^k(t) \int_{v_k} \frac{\mathbf{w}_j^k(\mathbf{p})}{|\mathbf{r} - \mathbf{p}|} dv_p \quad (3.21)$$

where v is obviously the total number of volumes.

If we substitute (3.21) into (3.19) we will have the final discretized researched expression of \mathbf{A} in function of $\mathbf{i}(t)$:

$$\tilde{a}_i^h(t) = \frac{\mu}{4\pi} \sum_{k=1}^v \sum_{j=1}^{f^k} i_j^k(t) \int_{v_h} \int_{v_k} \frac{\mathbf{w}_i^h(\mathbf{r}) \cdot \mathbf{w}_j^k(\mathbf{p})}{|\mathbf{r} - \mathbf{p}|} dv_r dv_p \quad (3.22)$$

which can be vectorially recast as

$$\tilde{\mathbf{a}}^h(t) = \sum_{k=1}^v \mathbf{M}^{hk} \mathbf{i}^k(t) \quad (3.23)$$

and consequently the entries of local \mathbf{M}^{hk} matrix, between two given volume v_h and v_k of dimension $f^k \times f^k$ will be:

$$M_{ij}^{hk} = \frac{\mu}{4\pi} \int_{v_h} \int_{v_k} \frac{\mathbf{w}_i^h(\mathbf{r}) \cdot \mathbf{w}_j^k(\mathbf{p})}{|\mathbf{r} - \mathbf{p}|} dv_r dv_p \quad (3.24)$$

Differently from \mathbf{R} constitutive matrix, this one is not sparse anymore. In fact from (3.24) it is clear that it will exist a local \mathbf{M}^{hk} for each arbitrary pair of mesh volumes and so this matrix will have to be calculated for all volumes with respect to a fixed one, and so on. This procedure will indeed yield a full matrix.

Regards properties of overall matrix \mathbf{M} it will result to be symmetric and positive definite although being affected by the so called *constitutive error*. Nevertheless, whenever $\mathbf{A}(\mathbf{r}, t)$ and $\mathbf{j}(\mathbf{r}, t)$ are uniform in each considered v_j , thus (3.22) will exactly map the current density $\mathbf{j}(\mathbf{r}, t)$ onto the circulation of $\mathbf{A}(\mathbf{r}, t)$. By this way consistency requirement will be satisfied.

Chapter 4

Planning a volume integral code

Accomplishing a numerical code in MATLAB to solve the eddy current problem equation presented in (2.26) is final goal of this work. If we analyse that relation we can identify the different terms which compose it: these terms will be obtained through code processes, starting from a defined geometry and using informations and properties described in previous chapters to obtain a simple system of equations which can easily be solved with a calculator.

4.1 Starting data

Initial data set is composed by an array containing vertices coordinates, denoted as “ $\mathbf{P0}$ ”, and three incidence matrices introduced and defined in chapter 2 called \mathbf{D} , \mathbf{C} and \mathbf{G} . Dimensions of the matrices are already known while $\mathbf{P0}$ array dimensions will be equal to $(n \times 3)$, where obviously n is the total number of vertices. These elements are the only needed elements to make the code start working. They are produced during domain discretization procedures which require specific codes and proper studies about topology. Since our work only concerns problem solution given an already existing mesh, we will now limit only on reporting a brief description of the procedures to obtain a polyhedral mesh [14].

4.2 Domain meshing with polyhedra

The procedure starts with the definition of a basic square geometry which will be meshed by hexahedral elements forming a first coarse grid. This initial grid will be as a sort of reference to shape the actual geometry of the desired domain. Once this grid is set, a subgridding of previous elements takes place, aiming at refining the previously created hexahedra. Also for this second step discretization is obtained using hexahedral elements; subgridding can be indiscriminately performed onto each element or it can be exclusively dedicated to particular groups of meshing volumes. After having set this more refined grid, geometrical shape equation and material properties are considered in order to obtain the final discretizing grid.

During this last phase earlier defined, grid points are shifted and volumes are re-assembled to satisfy geometrical or physical requirements of the domain. In particular, an implicit distance function is set to decide whether a point is internal or external with respect to geometry border. For instance, if we consider a sphere defined by its equation $(x^2 + y^2 + z^2 - r_0^2 = 0)$ it will be easy to determine the distance of a grid point

from its surface and, looking at the distance-function sign, it will be easy establishing the point position and deciding if this point is inside or outside the sphere. Once we know distance and position, the point laying near to geometry boundary within a certain distance is shifted and projected onto geometry surface. With this method the point will now correctly stay on sphere border but, unfortunately, the faces linked to the shifted point are plane no more. Consequently, to obtain a polyhedron with all plane faces this deformed surfaces are separated into two triangular faces. Thus, by this way, from an hexahedral element, a polyhedral one is obtained;

Another possible region, where a point can be shifted exploiting implicit distance-functions, is in proximity of material change borders. In fact if we set a function describing surface where two different materials are touching we can separate different elements which belong to the one or to the other material (e.g. metal and vacuum). Consequently other polyhedral volumes are generated.

Once vertices \mathbf{P}_0 have been established with this procedure, incidence matrices are successively assembled in order to provide correct information about mesh topology.

From incidence matrices to geometry arrays

Thanks to earlier described initial data, we can correctly write the desired equations system exploiting geometrical relations expressed by incidence matrices. In fact, by questioning these matrices we can respectively discover the geometrical relations between volume and faces using \mathbf{D} , faces and edges using \mathbf{C} and, finally, between edges and nodes using \mathbf{G} . More precisely each matrix conveys in every row which elements of the columns belong to a chosen row, e.g. if we are looking at a particular volume we can question \mathbf{D} and discover which faces are the borders of that volume and then, thanks to \mathbf{C} , we can identify the edges that delimit each face, and so on until knowing all the nodes of the volume.

By this way, “`geoarr_Enh`” has been developed, a geometrical MATLAB function which, starting from above elements, produces a set of fundamental geometrical arrays such as edges array, faces array and volumes array, respectively denoted as $\mathbf{L1}$, $\mathbf{L2}$ and $\mathbf{L3}$. Moreover, during this process, we can create variables containing barycenters of the different elements of the primal grid such as edges, faces and volumes barycenters, respectively called $\mathbf{P1}$, $\mathbf{P2}$ and $\mathbf{P3}$.

4.3 \mathbf{R} and \mathbf{M} matrices building

Once geometry has been reconstructed using incidence matrices, the other missing elements of the final equations system are constitutive matrices \mathbf{R} and \mathbf{M} and the right hand side of (2.26) i.e. the magnetic vector potential $\mathbf{A}_s(\mathbf{r}, t)$ caused by an external source in the domain Ω_s . This last term will be further discussed in this work, so hereafter we will only focus on constitutive matrices construction.

\mathbf{R} matrix

Expression that lead to \mathbf{R} matrix has been reported in (3.17). This equation clearly shows that construction of this matrix is based on piece-wise uniform vector functions $w^r(\mathbf{p})$ definition, which are based on geometrical elements found thanks to `geoarr_Enh` function as earlier explained. Consequently, since geometry has already been reported

into vectorial quantities, we have only to correctly compute the requested $w^r(\mathbf{p})$ and then implement the discretized equation for each volume of the primal grid. Finally we will sum and store partial results of each local matrix to achieve the overall matrix. All this described instructions are contained in `Rmatrix_Enh`, while calculation of a local \mathbf{R} matrix will be performed thanks to `Rloc_mtrx_En` function.

M matrix

Similarly to the previous matrix, \mathbf{M} matrix calculation is based on computing a basis vector function $w^r(\mathbf{p})$ as shown in equation (3.24) which reports the expression of a local matrix for two fixed volumes v_h and v_k . Differences respect to the prior procedure are essentially two:

1. we will have a double loop over volumes since we will have a different local M_{ij}^{hk} for every possible pair of volumes;
2. the expression reported is not directly implementable: we will need a function able to compute double volume integral of $\frac{1}{|\mathbf{r}-\mathbf{p}|}$ since this factor is not uniform in the volumes but it depends on calculation points \mathbf{r} and \mathbf{p} . This aspect will be treated with a numerical integration tool which will perform an integration of the fourth order whenever the volume h and k coincide, otherwise, if volumes are different, it will perform an easier integration approximated at the first order.

As a consequence, we will need a main script, denoted as `Mmatrix_Enh` which computes the overall \mathbf{M} matrix. This code will be constituted by a function that calculates a local M_{ij}^{hk} , called “`Mloc_mtrx_Enh`” and by the necessary integration tools of the 1st and 4th order (respectively “`int1ordbis`” and “`int4ordbis`”).

4.4 Equations system assembly

After having found constitutive matrices we have all the necessary terms to assemble the overall system that can be determined. To solve this system we can successively proceed working in frequency domain defining a frequency f and thus an angular frequency ω or in time domain (e.g. if we want to study a transient case) using a theta method. The code `EddySolv_Enh` can work only in frequency domain but a future improvement to make a transient solution possible is easily feasible.

Before starting a simulation we have to set the conditions at domain boundary as explained in chapter 2 and then we have to properly stimulate the grid with magnetic vector potential generated by sources in Ω_c as will later be discussed in chapter 6. The equations can now be solved. Result is an array \mathbf{T} expressing integral of electric vector potential along every edge of the primal grid G . Finally, using (2.8), we can find currents array \mathbf{i} for each grid face.

Since final expected solution is represented by the magnetic fields in a desired space point the obtained currents vector must be efficiently post-processed to acquire this goal.

4.5 Code schematic

All previous descriptions leads to the general code schematic shown in figure 4. The mentioned MATLAB codes are partially reported in appendix A.

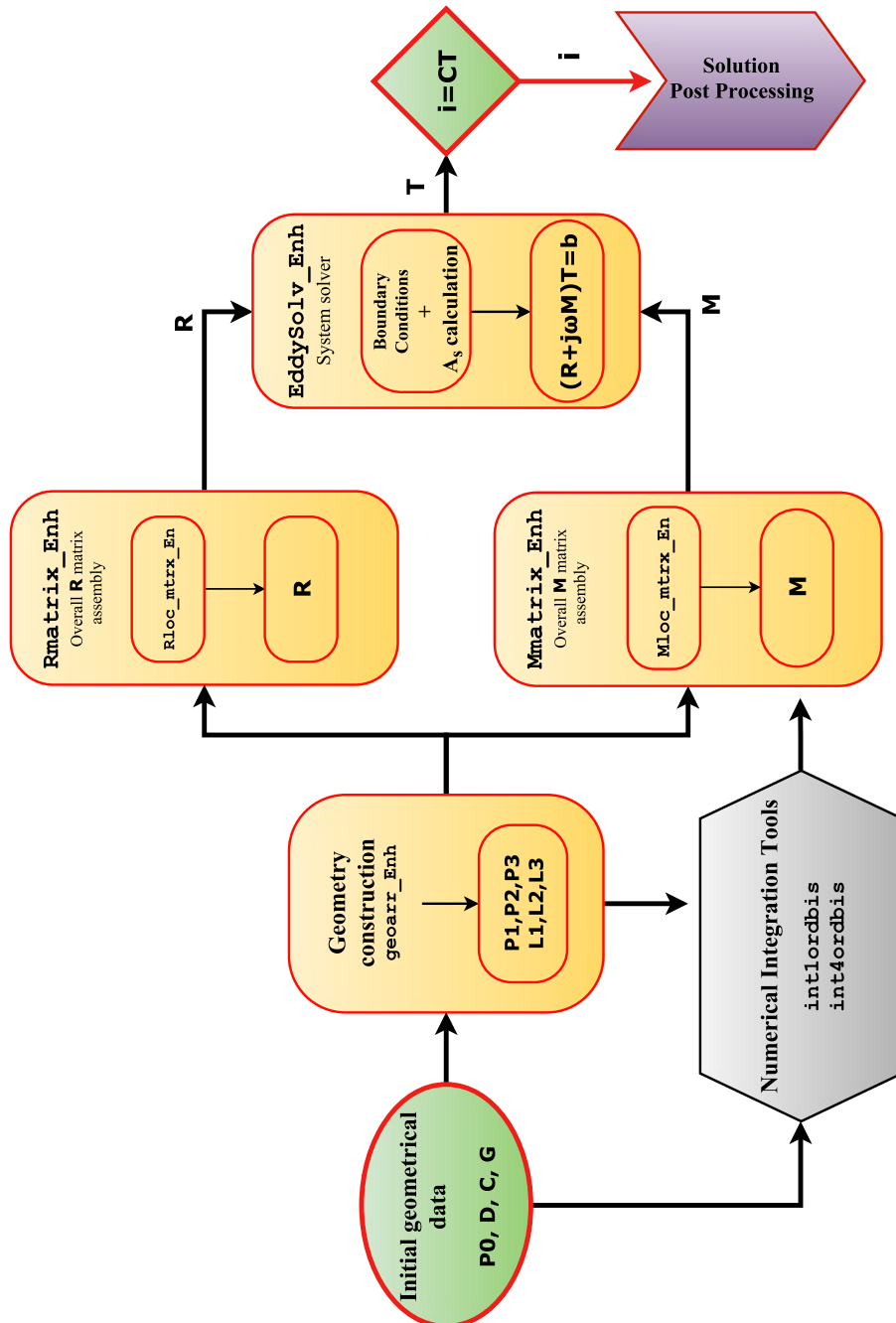


Figure 4.1: Code schematic

Chapter 5

Data post processing

In previous chapters how to solve a problem with a volume-integral approach has been shown. It is thus clear that obtained results, for an eddy-currents problem, will be in term of current density: we will have one value of \mathbf{J} for each element of the mesh, supposing this value constant in the volume. As a consequence, an efficient and robust tool is needed in order to evaluate the magnetic field induction \mathbf{B} and the magnetic vector potential \mathbf{A} generated by a given current density. In other words we need a valid data post processing method which provides us magnetic field values in each desired point.

In scientific literature there are several proposed alternative ways to achieve the prospected results, either exploiting numerical integration of magnetic induction and magnetic vector potential formulas, or using specific analytical formulations which are more precise but on average less general. Our proposal is based on an exact formulation, valid for a generic polyhedron with arbitrary faces and edges as proposed by M. Fabbri. This choice will guarantee both high accuracy and a satisfying level of generality.

5.1 Why an exact formulation

As just said, the main advantage in using an analytical formulation is due to the more accuracy of this method with respect to a numerical integration which, in smallest length needs an high quadrature order to avoid an unacceptable error. Moreover, as high is quadrature order, there will always be an unavoidable error in results. As a consequence, many simulators have introduced the use of analytical formulas to delete this error.

These formulations, even though are generally more efficient in saving calculation time, have a basic limit: each of them is built on a fixed and particular geometry. This fact has two consequences: it often leads to define a local coordinate system for each mesh element and, moreover, it leads to have a wide number of different routines to perform the calculus in the right way depending on different possible shapes of the various mesh elements. This last aspect is deeply problematic if we are working with polyhedra since the given expression are usually based on bars, bricks, tetrahedra or prisms with particular sections. In conclusion, both these features will cause an increasing level of complexity in the final numerical code, and consequently they will introduce new possible errors sources.

For these reasons introduction and development of a new general exact formulation, based on generic polyhedra is really useful to have a post processing tool which is not

error affected, general, and easy to implement [3].

5.2 Magnetic vector potential calculation

We will now focus on single polyhedron \mathbf{v} , and we will focus on the magnetic vector potential produced by a uniform current density \mathbf{J} in a given domain point \mathbf{r} . As already reported in previous chapters we have:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\mathbf{v}} \frac{\mathbf{J}}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}' \quad (5.1)$$

where \mathbf{r}' is an arbitrary point of \mathbf{v} and thus $d^3\mathbf{r}'$ is an infinitesimal volume around it.

Since \mathbf{J} is uniform in the considered volume it does not depend on \mathbf{r}' so, exploiting the vector identity

$$\frac{1}{|\mathbf{r} - \mathbf{r}'|} = \nabla' \cdot \left(\frac{\mathbf{r}' - \mathbf{r}}{2|\mathbf{r}' - \mathbf{r}|} \right) \quad (5.2)$$

we obtain

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 \mathbf{J}}{4\pi} \int_{\mathbf{v}} \frac{1}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}' = \frac{\mu_0 \mathbf{J}}{4\pi} \int_{\mathbf{v}} \nabla' \cdot \left(\frac{\mathbf{r}' - \mathbf{r}}{2|\mathbf{r}' - \mathbf{r}|} \right) d^3\mathbf{r}' \quad (5.3)$$

where the “prime” symbol on ∇ signifies that divergence is calculated in \mathbf{r}' .

Thus, using Gauss’ theorem, we arrive at

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0 \mathbf{J}}{8\pi} \oint_{\partial \mathbf{v}} \frac{(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}}{|\mathbf{r}' - \mathbf{r}|} d^2\mathbf{r}' \quad (5.4)$$

Obviously, $d^2\mathbf{r}'$ is an infinitesimal area around \mathbf{r}' whose outgoing normal unit vector is \mathbf{n} . Besides, $\partial \mathbf{v}$ is polyhedron surface.

Since every polyhedron face is plane, we can define an outgoing normal unit vector for each face, calling it \mathbf{n}_f , and we can replace \mathbf{n} with its corresponding \mathbf{n}_f vector. As a consequence, integrating over each plane face S_f of the considered volume, the quantity $(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}_f$ can be assumed constant over the face S_f and equal to $(\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f$, where \mathbf{r}_f is an arbitrary face point. In conclusion, being the last quantity constant and not depending by \mathbf{r}' , equation (5.4) becomes:

$$\begin{aligned} \mathbf{A}(\mathbf{r}) &= \frac{\mu_0 \mathbf{J}}{8\pi} \sum_{S_f \in \partial \mathbf{v}} \int_{S_f} \frac{(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}_f}{|\mathbf{r}' - \mathbf{r}|} d^2\mathbf{r}' \\ \mathbf{A}(\mathbf{r}) &= \frac{\mu_0 \mathbf{J}}{8\pi} \sum_{S_f \in \partial \mathbf{v}} (\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f W_f(\mathbf{r}) \end{aligned} \quad (5.5)$$

where we define the function $W_f(\mathbf{r})$ as

$$W_f(\mathbf{r}) = \int_{S_f} \frac{1}{|\mathbf{r}' - \mathbf{r}|} d^2\mathbf{r}' \quad (5.6)$$

Relation (5.5) provides the analytical expression of magnetic vector potential produced by a uniform current density \mathbf{J} .

5.2.1 W_f calculation

In order to correctly compute $\mathbf{A}(\mathbf{r})$ a discretized (but not approximate) expression of (5.6) is needed.

First of all the following vector relation is given:

$$\frac{1}{|\mathbf{r}' - \mathbf{r}|} = \mathbf{n}_f \cdot \nabla' \times \left(\mathbf{n}_f \times \frac{\mathbf{r}' - \mathbf{r}}{|\mathbf{r}' - \mathbf{r}|} \right) - \frac{[(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}_f]^2}{|\mathbf{r}' - \mathbf{r}|^3} \quad (5.7)$$

Substituting it in (5.6) and using Stoke's theorem it leads to:

$$W_f(\mathbf{r}) = \oint_{\partial S_f} \mathbf{n}_f \times \frac{\mathbf{r}' - \mathbf{r}}{|\mathbf{r}' - \mathbf{r}|} d^1\mathbf{r}' - [(\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f] \int_{S_f} \frac{(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}_f}{|\mathbf{r}' - \mathbf{r}|^3} d^2\mathbf{r}' \quad (5.8)$$

where, ∂S_f is the boundary of each S_f i.e. the l_e edges of the S_f face, $d^1\mathbf{r}'$ is an infinitesimal length around \mathbf{r}' and where, as previously done in (5.5), factor $[(\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f]$ has been taken outside integral considering a constant \mathbf{n}_f for each face S_f and choosing a new arbitrary point $\mathbf{r}_f \in S_f$.

With respect to (5.5) here the choice of \mathbf{n}_f can be different from the prior array since $W_f(\mathbf{r})$ is a scalar quantity and orientation of its arrays must be innerly coherent but may be independent from previously used conventions because it does not affect the final result. In order to have a coherent inner orientation in (5.8) this must be in such a way that the direction of each face normal is related with a freely chosen edges orientation path for each face S_f . This choice can easily be done using the incidence matrices defined in chapter 2.

We can now proceed considering that integral factor

$$\int_{S_f} \frac{(\mathbf{r}' - \mathbf{r}) \cdot \mathbf{n}_f}{|\mathbf{r}' - \mathbf{r}|^3} d^2\mathbf{r}'$$

is the solid angle seen from \mathbf{r} and subtended by an arbitrary face S_f . This angle will be hereafter defined as $\Omega_f(\mathbf{r})$. If now we denote with \mathbf{u}_e the edge tangent unit vector oriented as described in above paragraph, we can recast last equation as

$$W_f(\mathbf{r}) = \sum_{l_e \in \partial S_f} \int_{l_e} \mathbf{n}_f \times \frac{\mathbf{r}' - \mathbf{r}}{|\mathbf{r}' - \mathbf{r}|} \cdot \mathbf{u}_e d^1\mathbf{r}' - [(\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f] \Omega_f(\mathbf{r}) \quad (5.9)$$

where, another time, we can simplify first integral term taking outside a constant factor $\mathbf{n}_f \times (\mathbf{r}_e - \mathbf{r}) \cdot \mathbf{u}_e$ since we are integrating over straight lines and so this factor is constant. Point \mathbf{r}_e is an arbitrary point of the edge l_e .

Once a new function $w_e(\mathbf{r})$ has been defined as follows

$$w_e(\mathbf{r}) = \int_{l_e} \frac{d^1 \mathbf{r}'}{|\mathbf{r}' - \mathbf{r}|} \quad (5.10)$$

we finally have:

$$W_f(\mathbf{r}) = \left[\sum_{l_e \in \partial S_f} \mathbf{n}_f \times (\mathbf{r}_e - \mathbf{r}) \cdot \mathbf{u}_e w_e(\mathbf{r}) \right] - [(\mathbf{r}_f - \mathbf{r}) \cdot \mathbf{n}_f] \Omega_f(\mathbf{r}) \quad (5.11)$$

which is the researched discretized expression of W_f .

How to compute $\Omega_f(\mathbf{r})$ and $w_e(\mathbf{r})$

Last given equation still contains two undetermined expressions. Their computation is easy and immediate but it is hereafter reported for the sake of completeness.

Solid angle Ω_f

The solid angle seen from calculation point \mathbf{r} subtended by a face S_f can be found using additivity property of solid angles, splitting the face S_f into triangles, (e.g. one triangle for each face edge l_e) and computing each “sub-angle” $\Omega_T(\mathbf{r})$ subtended by the triangular face considered. By this way we can use the well known formula

$$\Omega_T(\mathbf{r}) = 2 \arctan \left[\frac{(\mathbf{r}_1 - \mathbf{r}) \cdot (\mathbf{r}_2 - \mathbf{r}) \times (\mathbf{r}_3 - \mathbf{r})}{D} \right] \quad (5.12)$$

with

$$D = |\mathbf{r}_1 - \mathbf{r}| |\mathbf{r}_2 - \mathbf{r}| |\mathbf{r}_3 - \mathbf{r}| + |\mathbf{r}_3 - \mathbf{r}| (\mathbf{r}_1 - \mathbf{r}) \cdot (\mathbf{r}_2 - \mathbf{r}) + |\mathbf{r}_2 - \mathbf{r}| (\mathbf{r}_1 - \mathbf{r}) \cdot (\mathbf{r}_3 - \mathbf{r}) + |\mathbf{r}_1 - \mathbf{r}| (\mathbf{r}_2 - \mathbf{r}) \cdot (\mathbf{r}_3 - \mathbf{r}) \quad (5.13)$$

where $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are the arrays of the considered triangle vertices. Choice of vertices order is not trivial, since it must be such that the triangle normal, defined as

$$\mathbf{n}_T = \frac{(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)}{|(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)|} \quad (5.14)$$

is equal to chosen \mathbf{n}_f orientation of (5.11).

Calculation of w_e function

For each edge l_e of the sum in (5.11), delimiting the face S_f of the volume \mathbf{v} , we can define a function $w_e(\mathbf{r})$ as shown in (5.10). This function can be easily computed as

$$w_e(\mathbf{r}) = \ln \left(\frac{|\mathbf{r}_2 - \mathbf{r}| + |\mathbf{r}_1 - \mathbf{r}| + |\mathbf{r}_2 - \mathbf{r}_1|}{|\mathbf{r}_2 - \mathbf{r}| + |\mathbf{r}_1 - \mathbf{r}| - |\mathbf{r}_2 - \mathbf{r}_1|} \right) \quad (5.15)$$

where \mathbf{r}_1 and \mathbf{r}_2 are endpoints arrows of the edge l_e . Order is arbitrary.

5.3 Magnetic induction calculation

As initially announced we will now find a discretized analytical expression of magnetic field induction $\mathbf{B}(\mathbf{r})$ in an arbitrary space point \mathbf{r} . As done for magnetic vector potential, the considered induction source is a current density \mathbf{J} uniform in the volume v .

The well known integral relation between $\mathbf{B}(\mathbf{r})$ and \mathbf{J} is

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_v \frac{\mathbf{J} \times (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} d^3\mathbf{r}' \quad (5.16)$$

in which symbols conventions are the same of that used in (5.1).

Once again, since \mathbf{J} is uniform it does not depend on \mathbf{r}' so it can be taken outside the integral symbol. In addition, we can resort to the following new vector identity:

$$\frac{(\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} = \nabla' \frac{1}{|\mathbf{r} - \mathbf{r}'|} \quad (5.17)$$

Similarly to what has been done in section (5.2), but now exploiting Green's theorem, we obtain:

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0 \mathbf{J}}{4\pi} \times \int_v \nabla' \frac{1}{|\mathbf{r} - \mathbf{r}'|} d^3\mathbf{r}' = \frac{\mu_0 \mathbf{J}}{4\pi} \times \oint_{\partial v} \frac{\mathbf{n}}{|\mathbf{r} - \mathbf{r}'|} d^2\mathbf{r}' \quad (5.18)$$

with the already known meaning of ∂v and $d^2\mathbf{r}'$.

Now we can recall already used normal face unit vector \mathbf{n}_f firstly employed in (5.5) to take outside from the integral a constant vector \mathbf{n}_f for each considered face S_f . Consequently, from (5.18) we achieve

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_{S_f \in \partial v} \mathbf{J} \times \mathbf{n}_f \int_{S_f} \frac{1}{|\mathbf{r} - \mathbf{r}'|} d^2\mathbf{r}' \quad (5.19)$$

and recalling $W_f(\mathbf{r})$ definition in (5.6) we finally obtain

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_{S_f \in \partial v} \mathbf{J} \times \mathbf{n}_f W_f(\mathbf{r}) \quad (5.20)$$

which exactly is the discretized analytical expression of $\mathbf{B}(\mathbf{r})$ for a given uniform current density \mathbf{J} .

5.4 Discontinuities and singularities

The last necessary aspect to be treated to complete our dissertation about analytical formulations, and especially that one above proposed and used in our code, regards how to avoid problems of discontinuities and singularities in discretized reported expressions.

First of all we have to **distinguish discontinuities from singularities** problem: the first one is due to use of inverse functions such as inverse goniometric functions which can lead to different values calculation for a given situation and so it can be classified as a *programming risk*; the second one is indeed due to mathematics issues such as zeroing of denominators or indefiniteness of ratios for particular values.

As a consequence:

- we have a **discontinuity** in (5.12) expression. In fact, it is useful to notice that a solid angle spans from 0 to 4π . Unfortunately inverse tangent function “arctan” by definition returns a number from $-\pi/2$ to $+\pi/2$ so that the result of (5.12) will be between $-\pi$ and $+\pi$. This problem will be solved thanks to the use of “atan2” MATLAB function which allows to have a result that automatically spans from -2π to 2π in a such way that if the position of calculation point \mathbf{r} moves from one side to the other (with respect to the plane of triangle T), sign of $\Omega_T(\mathbf{r})$ will change too;
- we have some singularities problems in more than one relation; in particular we can identify a logarithmic singularity in (5.10) whenever calculation point \mathbf{r} belongs to the edge l_e . Moreover there is a singularity in equation (5.12) too: when \mathbf{r} coincided with one of the triangle vertices, ratio would not exist.

To avoid these singularities issues we can modify the Euclidean norm such as:

$$|\mathbf{r}' - \mathbf{r}|_\varepsilon = \sqrt{|\mathbf{r}' - \mathbf{r}| + \varepsilon^2} \quad (5.21)$$

This choice introduce an error in computation process but, being ε essentially a length, whenever it is small with respect to the volume \mathbf{v} dimensions, this error will be negligible since it will affect only a small cylindrical volume whose radius is equal to ε around volume edges.

5.5 Code structure

In prior sections we have described different involved equations if we want to reconstruct magnetic vector potential and magnetic induction field caused by a uniform current density inside a single volume \mathbf{v} . Hereafter we are expected to explain how that equations have been implemented using our data structure.

We start reminding that we have just solved an eddy current problem as explained in chapter 2, using (2.26), so we have found $\mathbf{T}(t)$ array. Through (2.8) we could obtain an array, whose dimensions will be $f \times 1$, of the fluxes of \mathbf{j} through volumes faces i.e. the current crossing each face of primal grid. Exploiting these informations, thanks to (3.7), we could reconstruct the field using the flux of each face: in conclusion we have found a uniform field \mathbf{J} approximating $\mathbf{j}(\mathbf{r}, t)$ in each volume. We are exactly in previously described conditions.

Now we focus our attention on a specific given point \mathbf{r} of the space where we are interested to know $\mathbf{A}(\mathbf{r})$ and $\mathbf{B}(\mathbf{r})$. We will produce a MATLAB function to calculate the contribution of a single volume \mathbf{v} on that specific point. If we want to compute total contribution of each current in each volume we will simply loop over all volumes of primal grid G to obtain the total field.

In order to implement (5.5) the first step is considering our data format. We have incidence matrices \mathbf{D} , \mathbf{C} , \mathbf{G} to know which faces, edges and nodes belong to a considered volume. Consequently, we can chose a face of \mathbf{v} (denoted as S_f) using \mathbf{D} and focus on it since in considered equation we need the sum of each face contribution. Following procedure would thus be replied for each face of \mathbf{v} .

$W_f(\mathbf{r})$ implementation

Contribution of each face on (5.5) depends on $W_f(\mathbf{r})$. So we will use (5.11) relation which contains a sum on the edges of S_f . Questioning \mathbf{C} in the correct row we can find which edges belong to S_f face. Through geometric construction we identify the endpoints of this edges (we will exploit knowledge of edges midpoints called “ $\mathbf{P1}$ ” and edges vectors denoted in the code as “ $\mathbf{L1}$ ”) also used in $w_e(\mathbf{r})$ calculation ¹ (see (5.15) formula). Now there is nothing to do except calculating coherent \mathbf{n}_f and \mathbf{u}_e vectors: the first one is obtained considering faces arrays $\mathbf{L2}$, the second vector using again incidence matrix \mathbf{C} indications and edges arrays $\mathbf{L1}$. Choice of coherent \mathbf{n}_f and \mathbf{u}_e is surely the most critical step and this can be assured only correctly questioning \mathbf{C} .

Resulting function to be operated will need even another last routine to compute $\Omega_f(\mathbf{r})$.

$\Omega_f(\mathbf{r})$ function

Similarly to previous procedure, $\Omega_f(\mathbf{r})$ calculation, as said in prior corresponding paragraph, is articulated into a sum of sub-angles generated splitting the face into triangles. The choice we will make for this formula implementation, is considering a triangle for each edge delimiting the face. So for example, if in S_f we have four edges we consider four triangles and so on. This choice is particularly simple and “natural” since it well fits our data structure and because we can re-use previous logic to question each edge of the face S_f . After having identified each triangle (considering as third vertex the faces barycenters stored in $\mathbf{P2}$) we have only to use (5.12) formula and then sum partial contributions.

Final expression

Once we have computed $W_f(\mathbf{r})$ we can finally calculate (5.5): in addition to W_f we only need \mathbf{r}_f and \mathbf{n}_f . The former is chosen to be the face S_f barycenter (simply found using global face index and $\mathbf{P2}$) the latter has to be such that the volume v would be outer oriented. To reach this requirement we question \mathbf{D} and then we chose the corresponding face vector in $\mathbf{L2}$, **multiplying the array for a coherent unit value** (± 1) as determined in \mathbf{D} matrix.

5.6 Results comparisons

Fabbri’s Formulation VS Analytical Solution

An important possible comparison is between our analytical discretized formulation and an exact analytical integration for a fixed given geometry. In particular we propose a comparison with an exact formula calculated for a brick-shaped element, considering for simplicity only the x component of the induction field $\mathbf{B}(\mathbf{r})$ computed in a space point \mathbf{r} . The brick element is one meter high and wide and 100 meters long (with the same coordinates convention of the axes: $1 \times 100 \times 1$) centred in $(0, 0, 0)$ and it is crossed by a uniform unitary current along y direction $J_y = 1 [A/m^2]$. The point \mathbf{r} moves on a

¹Precisely the last previous procedure i.e. w_e and endpoints calculation would be done in $\Omega_f(\mathbf{r})$ calculation function to improve efficiency and avoid arrays questioning repetitions.

line through the element and so its x coordinate varies between $r_x = -2$ and $r_x = +2$ while the other coordinates are fixed and equal to $r_z = 0.25$ and $r_y = 0$.

Analytical formula

The exact formula used to benchmark our code is the result of the integration of Biot-Savart law as reported in (5.16). If we denote with $X_1, X_2, Y_1, Y_2, Z_1, Z_2$ coordinates extremes of the brick geometry (i.e. \mathbf{r}' coordinates extremes), and with (x, y, z) \mathbf{r} point position, we can separately integrate the considered equation along each direction after calculating $\mathbf{J} \times (\mathbf{r} - \mathbf{r}')$ for a constant current along y direction J_y . We obtain:

$$\mathbf{B}(\mathbf{r}) = \begin{cases} -\frac{\mu_0 |J_y|}{4\pi} \int_V \frac{(Z - z)}{|\mathbf{r}' - \mathbf{r}|^3} dX dY dZ \\ 0 \\ -\frac{\mu_0 |J_y|}{4\pi} \int_V \frac{(X - x)}{|\mathbf{r}' - \mathbf{r}|^3} dX dY dZ \end{cases} \quad (5.22)$$

where (X, Y, Z) is a generic point in the brick volume.

Now, focusing our attention on x component and integrating its volume-dependent factor along dX, dY, dZ , we can define, for arbitrary points $\mathbf{r} = (x, y, z)$ and $\mathbf{r}' = (X, Y, Z)$, a function:

$$\begin{aligned} I_B^x(X, Y, Z) = & -(Y - y) \log(|\mathbf{r}' - \mathbf{r}| + (X - x)) + \\ & - (X - x) \log(2(|\mathbf{r}' - \mathbf{r}| + (Y - y))) + \\ & + \arctan\left(\frac{(X - x)(Y - y)}{(Z - z)|\mathbf{r}' - \mathbf{r}|\right) + \\ & - (Z - z) \arctan\left(\frac{Y - y}{Z - z}\right) + Y \end{aligned} \quad (5.23)$$

such as

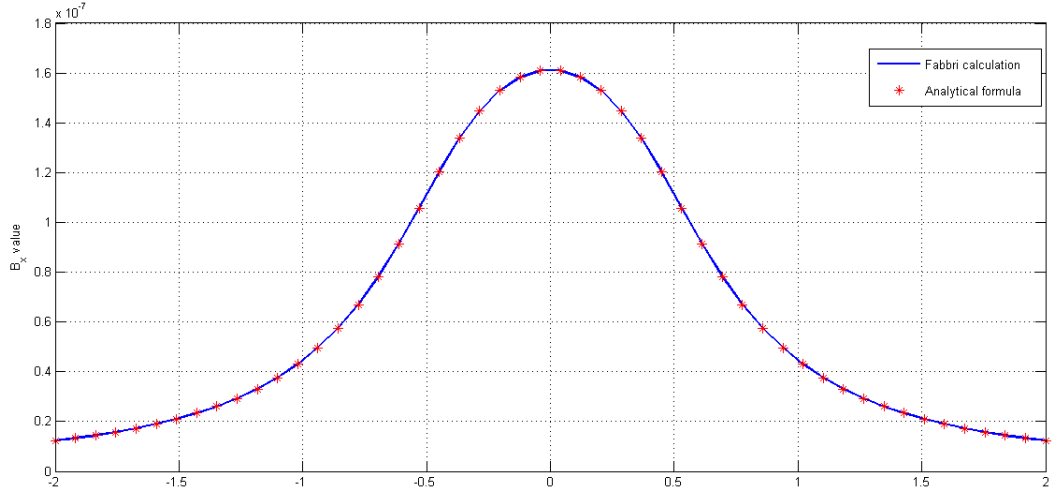
$$I_B^x(X, Y, Z) = \int_V \frac{(Z - z)}{|\mathbf{r}' - \mathbf{r}|^3} dX dY dZ \quad (5.24)$$

Consequently, the resulting expression is:

$$\begin{aligned} \mathbf{B}_x(\mathbf{r}) = & -\frac{\mu_0 |J_y|}{4\pi} \{ [(I_B^x(X_2, Y_2, Z_2) - I_B^x(X_1, Y_2, Z_2)) + \\ & - (I_B^x(X_2, Y_1, Z_2) - I_B^x(X_1, Y_1, Z_2))] + \\ & - [(I_B^x(X_2, Y_2, Z_1) - I_B^x(X_1, Y_2, Z_1)) + \\ & - (I_B^x(X_2, Y_1, Z_1) - I_B^x(X_1, Y_1, Z_1))] \} \end{aligned} \quad (5.25)$$

Comparison between (5.25) and the x component of (5.20) is reported in figure 5.1. From the graph is thus clear that the two expressions lead to the same precise results.

Finally we report in table 5.1 the numerical results for five calculation points to prove method precision.

Figure 5.1: B_x field comparison

Fabbri B_x [T]	Analytical B_x [T]
$0.012239788877157 \cdot 10^{-6}$	$0.012239788877179 \cdot 10^{-6}$
$0.015776776206583 \cdot 10^{-6}$	$0.015776776206627 \cdot 10^{-6}$
$0.033089455139092 \cdot 10^{-6}$	$0.033089455139142 \cdot 10^{-6}$
$0.091304199096174 \cdot 10^{-6}$	$0.091304199096172 \cdot 10^{-6}$
$0.160898768631363 \cdot 10^{-6}$	$0.160898768631182 \cdot 10^{-6}$

Table 5.1: B_x values

Fabbri's Formulation VS Numerical Integration

For the sake of completeness we also report a graph with the comparison between $\mathbf{A}(\mathbf{r})$ calculation performed with the proposed method (denoted as “*Fabbri A(r)*”) and one carried out with a standard numerical integration using Abramowitz formulas [16] (denoted as “*Abramowitz A(r)*”). The graphs are shown in figure 5.2: the first graph separately reports the numeric results with the two method, the second graph shows what has been denoted as *error* i.e. the absolute value of the difference between the two resulting integrals.

The comparison has been done considering a unitary uniform current density \mathbf{J} along z-axis inside a cube volume whose barycenter is on $(0.375, 0.375, 0.375)$, moving the x-coordinate of calculation point \mathbf{r} between $x = 0$ and $x = 0.8$. The cube edge measures $l = 0.25$. We can notice that Fabbri's result is more continuous and it is not influenced by \mathbf{r} position while the result obtained with numerical integration is affected by sudden variations due to singularities in its formulation (in fact used Abramowitz formulas are calculated at volume barycenter and at faces barycenters). Moreover, as we expected, when \mathbf{r} moves far away from the volume these two methods lead to the same results. This behaviour confirms what has been described at the beginning of this chapter.

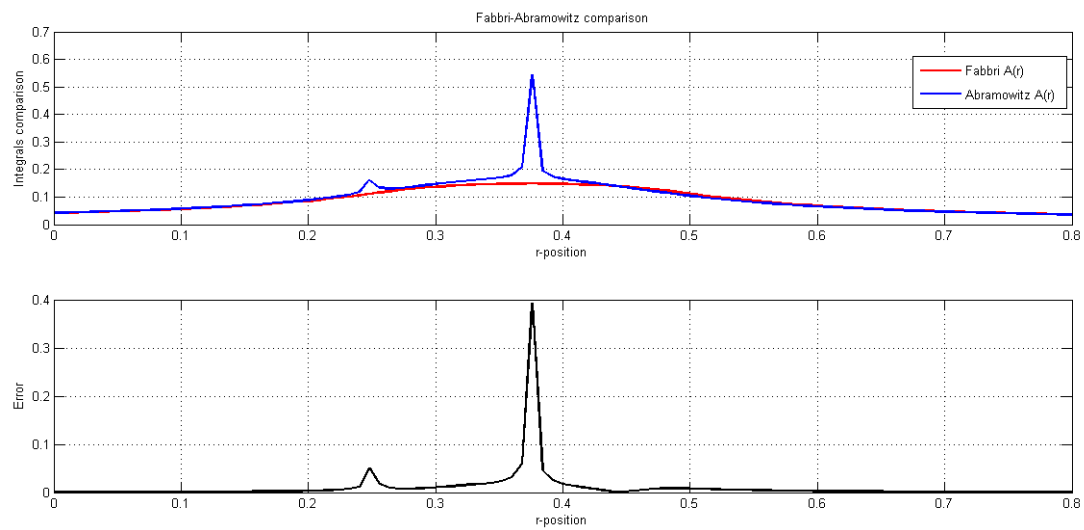


Figure 5.2: Magnetic vector potential calculation comparison

Chapter 6

Starting a simulation: source computing

In chapter 4 the code structure has been delineated in each necessary part to implement the final equations system, as reported in (2.26). Nevertheless a fundamental term has been momentarily neglected: calculation of $\mathbf{A}_s(\mathbf{r}, t)$. This term is indeed the magnetic vector potential caused by an external magnetic source on our analysed material (e.g. a simple source such as a coil or more generally a magnetic circuit). As a consequence the aim of this chapter is describe a method to compute the effect of this term on conductive domain Ω_c . Once this effect is known the simulation code can start.

6.1 Ω_s discretization

We consider a magnetic source represented by a coil. If we describe turns of this coil with a parameter curve we can obtain a mathematical representation of the barycenter of each turn. The final step is create a simple tool able to split the overall coil into small hexahedral slices, each of them determined thanks to its vertices. By this way we can discretized the original coil with a satisfactory level of approximation (the smaller the slices the higher the fidelity of the model) obtaining a series of hexahedra that describes the original domain Ω_s .

Once the domain is discretized we have to compute the magnetic vector potential produced by each volume when it is crossed by a uniform current. This configuration is the same that has been treated in the previous chapter, as a consequence that formulation can be re-adapted for this new situation.

6.2 Source calculation

The aim of this new code is clear: cycling over each volume of Ω_s and then sum the partial effect of a volume to obtain the total effect of every coil element in a single fixed point of Ω_c . Code procedures and implicated formulas to reach this goal are very similar to that illustrated in post processing chapter and more precisely in section 5.5. The tool differences are simply due to a dissimilar configuration of geometric starting data: for the post processing functions the involved beginning variables came from the mesh structure while for this new functions the same necessary data are locally produced and formulated in an independent template.

Calculation points

The last remaining aspect regards the choice of calculation points. If we look at (2.26) we can notice that, at the right hand side, the implicated quantity precisely is the circulation of $\mathbf{A}_s(\mathbf{r}, t)$ along the edges of the **dual** grid \tilde{G} . Consequently our calculation point will lay on this grid.

For the sake of simplicity the magnetic vector potential will be computed on each dual edge midpoint, and then it will be integrated along the dual edge. As a consequence we will obtain an integral value for each dual edge: this value will be that one used to complete the equations system. By this way we will accomplish a general method to “stimulate” the mesh grid and to study on the same conductive domain a wide range of different problems, changing the magnetic source.

6.3 Discretized magnetisation coils

We will now describe with an higher details level the discretization process for some specific magnetization coils often involved in fusion reactor. In particular we will focus our attention on **circular coils**, **square coils** and finally, on **helical coils**. The first two coils are more precisely used in tokamak-shaped reactors (e.g. ITER, JET or RFX-mod experiments) while the last ones are specifically used in Stellarators.

6.3.1 Circular coils

The simplest magnetizing device is a circular coil. This device is composed by a series of conductors all crossed by the same given current I_w , each of them winded around a circular path. As a consequence, the coil section will be crossed by an overall current I_s equal to I_w times the number of turns n_t of these conductors around the circular structure: $I_s = n_t I_w$.

Our numerical tool will be able to compute the magnetic effect of this coil in a space point \mathbf{r} i.e. it will be able to compute the magnetic vector potential \mathbf{A} , the magnetic induction \mathbf{B} and so also the magnetic field \mathbf{H} (since $\mathbf{H} = \mathbf{B}/\mu$) given the point, the current I_w of a coil turn and the number of turns n_t of the desired coil.

Firstly we set the barycentric path position choosing the circumference radius r_0 and the circumference centre c_0 . For simplicity we will assume the circular area described by the coil as parallel to the plane determined by x and y axes. Using this few input data we can proficiently describe the coil path and choose a sufficiently wide number of points on it. These points will be the barycenters of each face of the various hexahedra composing the coil.

Once the faces barycenters are determined we can set a proper orthonormal frame to pose the four face vertices. This orthonormal vectors base will be such that each face will radially lay perpendicular to the circumference plane (as in figure 6.1). By this way in each face barycenter the face normal will coincide with the circumference tangent vector.

After having found the vertices of each face we can determine an hexahedron correctly matching the vertices of two consequent faces. Matching all the faces vertices the accomplished final result shown in figure 6.2 is a circular discretized coil.

The so far described process has led to the determination of coil geometry. The next step is obtain the magnetic field produced by this coil exploiting processes and

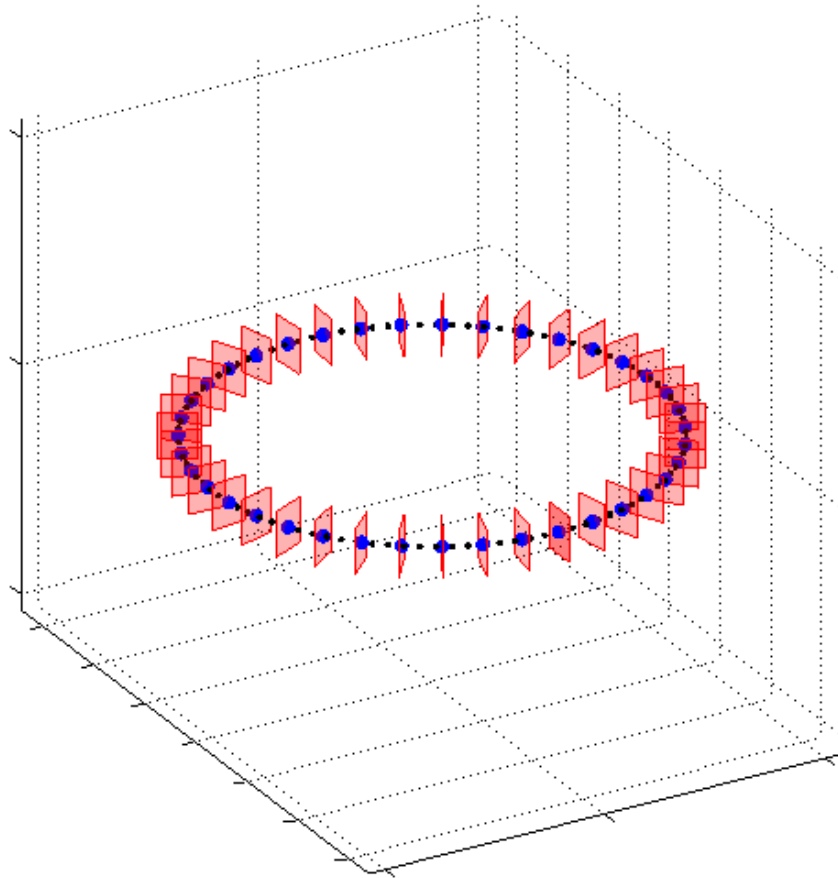


Figure 6.1: Coil faces and barycenters

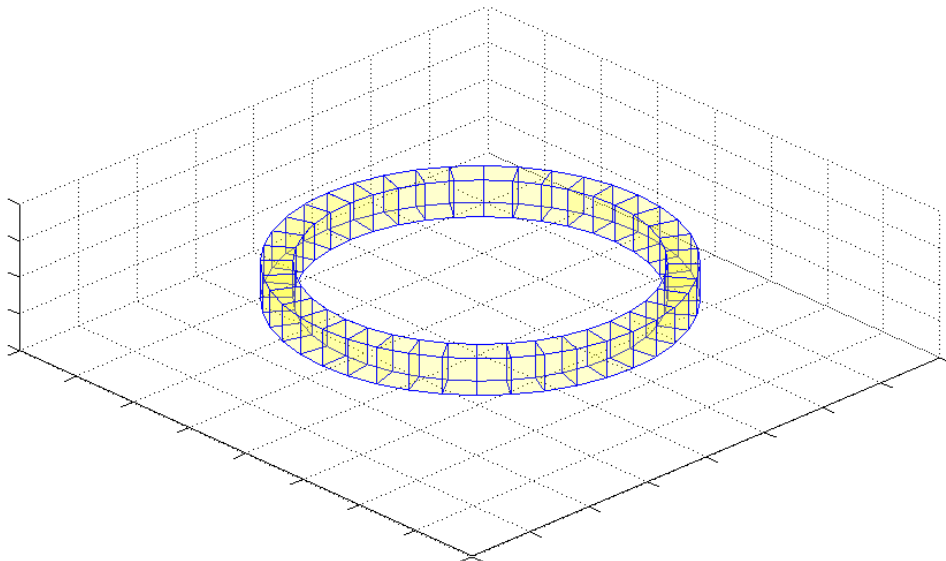


Figure 6.2: Circular discretized coil

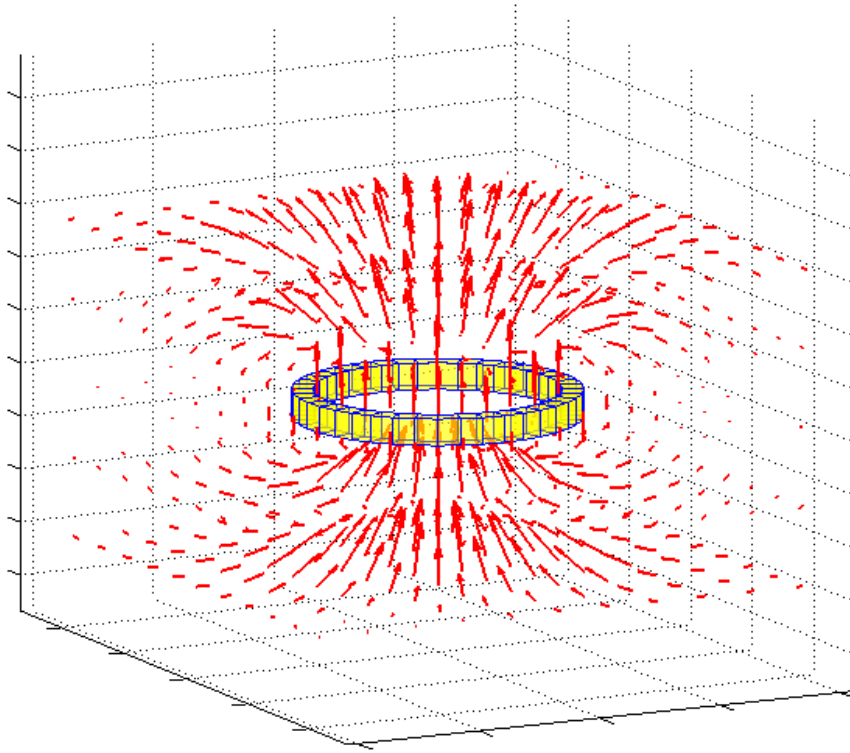


Figure 6.3: Circular coil \mathbf{B} field

equations reported in the previous chapter. Consequently, since volumes are now given and known, we have only to compute the current density vector \mathbf{J} in each volume, and then, assuming it constant in each specific element, we can apply equations (5.5) and (5.20) to find $\mathbf{A}(\mathbf{r})$ and $\mathbf{B}(\mathbf{r})$ as desired. The current density vector can be estimate starting from I_s using the already known following equation

$$\mathbf{J} = \frac{1}{|v|} \sum_{i=1}^R (I_s)_i \tilde{\mathbf{r}}_i \quad (6.1)$$

obtained recasting equation (3.7) for $x = \mathbf{J}$ and $X_i = (I_s)_i$. We underline that since current is supposed to be perfectly parallel to the coil path, the current across the lateral faces of the volume is zero. As a consequence (6.1) has to be computed for $R = 2$ i.e. only respect to ingoing and outgoing faces.

The last step is creating a MATLAB function which can compute the fields we are interested in, produced by an hexahedron given the previously calculated current density \mathbf{J} and the eight vertices that define the faces. This function will be the core of a code able to loop over each volume and able to compute the contribution of all these elements in a single point. If we create a grid of point in the space and we compute the induction field $\mathbf{B}(\mathbf{r})$ in every point, we obtain the overall result which is visible in figure 6.3.

	B_{int} [mT] value
Ideal solenoid formula	0.3141592653e-02
Discretized coil 10 elements	0.3137982738e-02
Discretized coil 30 elements	0.3137703038e-02
Discretized coil 100 elements	0.3137675124e-02

Table 6.1: B_{int} values of a solenoidal coil

Solenoidal coils A paradigmatic study case with respect to circular coils is surely represented by a solenoidal coil. As it is well known, the internal axial induction field B_{int} of a solenoid immersed in a vacuum space, formed by N wires, crossed by a current I and whose length is equal to l , is easily evaluable thanks to

$$B_{int} = \mu_0 \frac{N}{l} I \quad (6.2)$$

If we use our code to create a circular coil whose faces height is equal to l , we can make a comparison between the exact analytical ideal solution of (6.2) and the value of \mathbf{B} computed with our code in the central point of the coil. If l is much greater than faces width the two value will coincide proving the method accuracy.

In table 6.1 we report the results for a solenoidal coil of radius $R_0 = 0.5$ [m] and length $l = 20$ [m] (the face width for the discretized geometry is equal to $a = 0.02$ [m] such as $l/a = 1000$). The solenoid is crossed by a unitary current $I = 1$ [A] and is composed by $N = 50$ turns. Three different tests with increasing level of discretization have been done with our code. The testing coils are:

- a solenoidal coil composed by 10 hexahedral volumes;
- a solenoidal coil composed by 30 hexahedral volumes;
- a solenoidal coil composed by 100 hexahedral volumes (reported in figure 6.4).

These different elements densities have the scope of investigating possible influence of a coarse grid on the solution.

Numerical results evidently show that, in this case, discretization level does not significantly affect the outcomes. This behaviour is due to the fact that, has already said in the dedicated chapter, our computational method is not based on a numerical integration but on an analytical one. Consequently the sole error we introduce comes from the assumption of a uniform current density in each volume. So, if we perform a good level of discretization we can assume that the discretized current density value will be close enough to the real “physical” value. In this case the “*good level of discretization*” is verified to be relatively low so, even with a coarse grid of ten elements the error is small. In conclusion, this comparison shows a difference between the ideal value of (6.2) and B_{int} value with 100 elements lower than 0.12% giving a further proof of the method accuracy.

6.3.2 Square coils

Other coils widely employed in fusion devices are square coils. These coils are particularly useful since their shape is simply feasible, and they represent a valid general shape

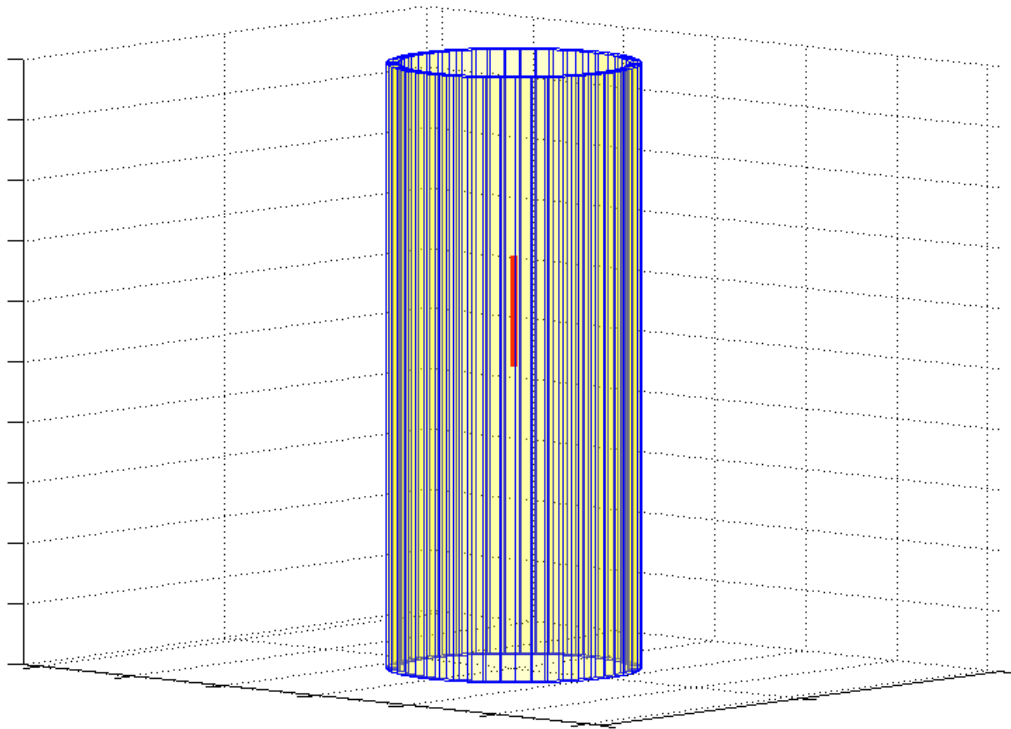


Figure 6.4: Solenoidal coil B_{int} field

that can be adapted to produce other different shaped coils (e.g. saddle coils which are such as a square coil adapted for a cylindrical geometry). For this reason we produced a code that is able to compute the fields generated by a square coil. For obvious reasons this new tool will be very similar to the one used for circular coils. Consequently we will now limit only on reporting final results.

As earlier done for circular coils, the code is able to generate a series of faces along a predefined path. In order to define the coil barycentric path the code needs coil borders measures (precisely it needs the semi-axes measures of the square shape) and the position of coil plane with respect to the axis origin (i.e. the plane height respect to z-axis). Once the barycentric path is defined, given faces dimensions, the code produce the faces which correctly matched give shape to the hexahedra composing the coil (as reported in figure 6.5).

Once the geometry has been built, the code can compute fields by cycling over volumes and over calculation points with the same processes already explained. The overall result is shown in figure 6.6.

Further considerations In order to provide a complete description about code possibilities in figure 6.7 we report other two different versions of a square coil. Both of them have edges composed by a single hexahedral element. The important aspect is that this new peculiarity does not affect the precision of the process because the mathematical formulation to compute fields is exact: whether we split the edges into more than a single volume or we consider a whole element, in each volume the current density \mathbf{J} will always be the same since in each edge of the square coil \mathbf{J} must be uniform.

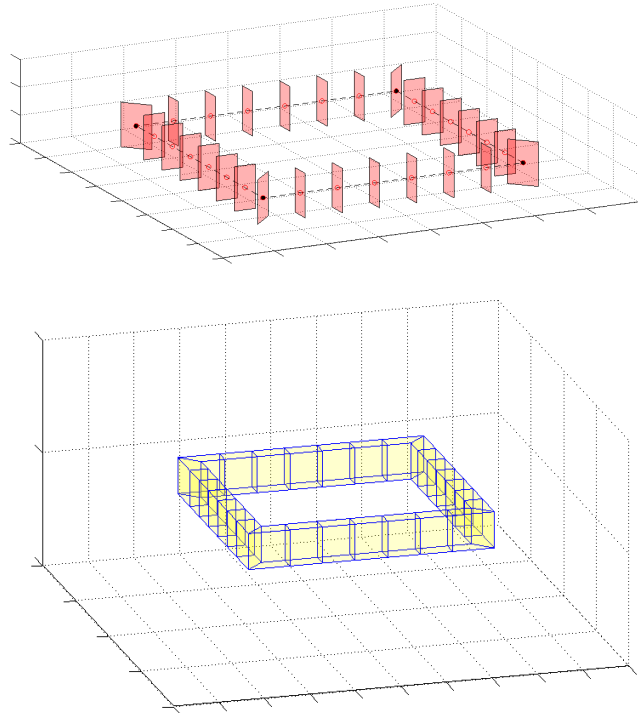


Figure 6.5: Square coil construction

This behaviour, which is different respect to a circular coil, allows to discretize a square plane coil using only four element, proficiently thus reducing computation time.

Finally, we have generated a square coil with its edges rotated with respect to the cartesian axes. This last coil has been produced with a trick: the generation code in fact was not the one specifically written for square coil but we used the circular coils modelling code, setting only four subdivision points! By this way it is thus possible to create a square coil generally rotated of an arbitrary angle ϕ .

6.3.3 Helical coils: stellarators

After having described coils with simple shapes, we now face with a more challenging geometry: helix.

This type of coils is widely used in particular fusion reactors called “*Stellarators*” whose design is specifically studied to ensure an higher plasma MHD stability because they avoid that bothersome plasma drift typical of tokamaks. Even if nowadays the shapes of current reactors is more complex than a simple helix (as shown in figure 6.8) the first prototype of these machines was actually surrounded by simple helically shaped coils. Actual coils are only their natural evolution. For these reasons the possibility of creating an helical coil to use it as magnetic field source for our volume integral code become an interesting challenge.

Procedures to obtain the desired coil are similar to those used for square and circular inductors. The main difference is the barycentric path delimitation. An helical path, indeed, is obtained from the motion of two distinct circumferences: a principal one and a smaller one (moving at the same time respect to the first). These two circumferences

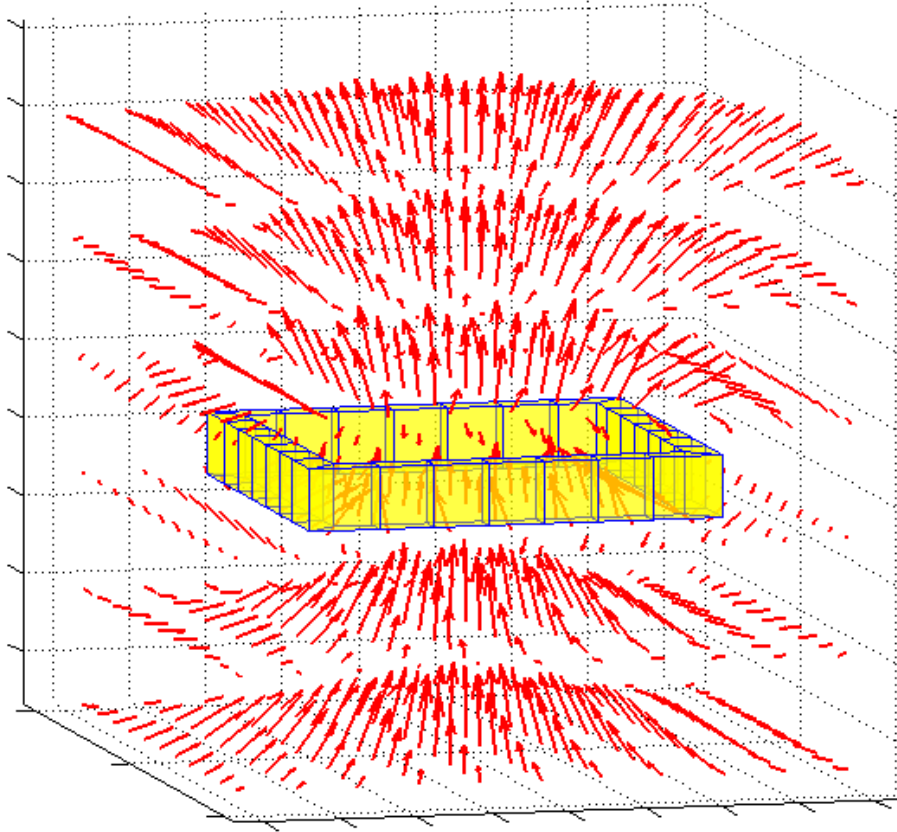


Figure 6.6: Square coil: $B(\mathbf{r})$ field calculation

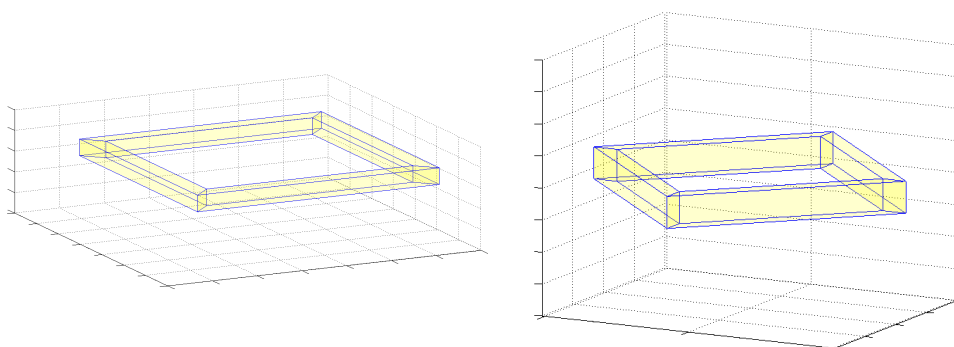


Figure 6.7: Square coil with not-splitting edges and with edges rotation ($\phi = 45^\circ$)

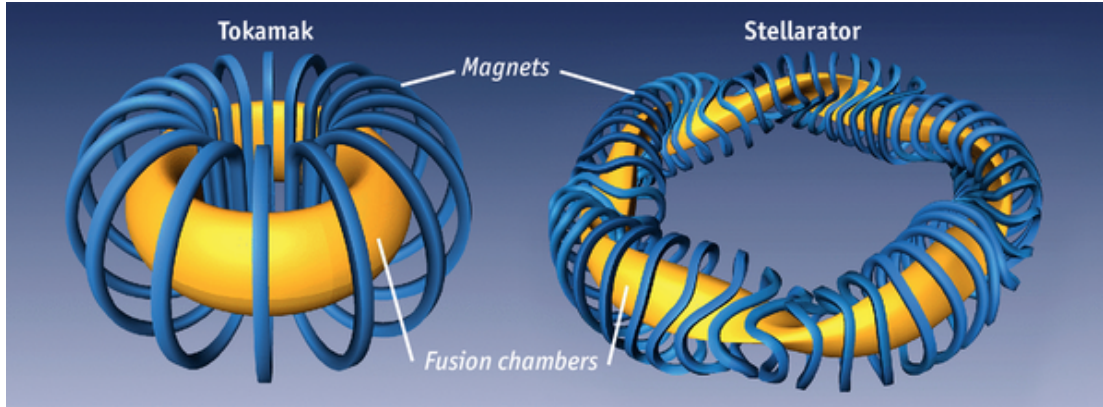


Figure 6.8: Fusion reactors: Tokamak VS Stellarator

have distinct angular speeds ω_1 and ω_2 with $\omega_2/\omega_1 = k, k \in \mathbb{N}$. Once the circumferences radius, R_1 and R_2 , and principal circumference height z_0 have been set (assuming the helix laying on a plane parallel to x and y axes) we can describe the movement of a point along the principal circumference as

$$\mathbf{r}_1 \begin{cases} x = R_1 \cos(\omega_1 t) \\ y = R_1 \sin(\omega_1 t) \\ z = z_0 \end{cases} \quad (6.3)$$

where (x, y, z) are the component of a generic array \mathbf{r}_1 describing the point position.

Starting from this point we can determinate the position \mathbf{r}_2 of a point moving along the secondary circumference, laying on a plane perpendicular to the first one:

$$\mathbf{r}_2 \begin{cases} x = R_2 \cos(\omega_2 t) \cos(\omega_1 t) \\ y = R_2 \cos(\omega_2 t) \sin(\omega_1 t) \\ z = R_2 \sin(\omega_2 t) \end{cases} \quad (6.4)$$

By this way for each time t , we obtain the position of a point along an helical path simply from $\mathbf{r} = \mathbf{r}_1 + \mathbf{r}_2$:

$$\mathbf{r} = (\mathbf{r}_1 + \mathbf{r}_2) = \begin{cases} (R_1 + R_2 \cos(\omega_2 t)) \cos(\omega_1 t) \\ (R_1 + R_2 \cos(\omega_2 t)) \sin(\omega_1 t) \\ z_0 + R_2 \sin(\omega_2 t) \end{cases} \quad (6.5)$$

Resulting curve is shown in figure 6.9.

Whenever helical path is determined we can proceed as done with circular and square coils delimiting hexahedra faces points which are established considering a local reference frame laying parallel to secondary circumference plane. This frame is built appropriately using again \mathbf{r}_1 and \mathbf{r}_2 . These last construction steps are reported in figure 6.10: each face is delimited by its four vertices disposed around a barycentric point laying on the helical path; successively these points are matched to obtain a 3D helical coil.

If we carefully look at the final result displayed in the figure we can see an anomaly respects to the other discretized coils: there is not any hexahedral volume. In fact

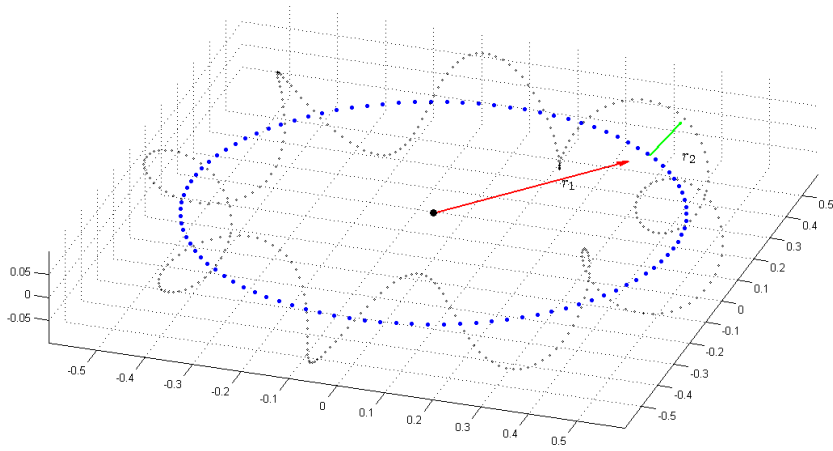


Figure 6.9: Helix construction

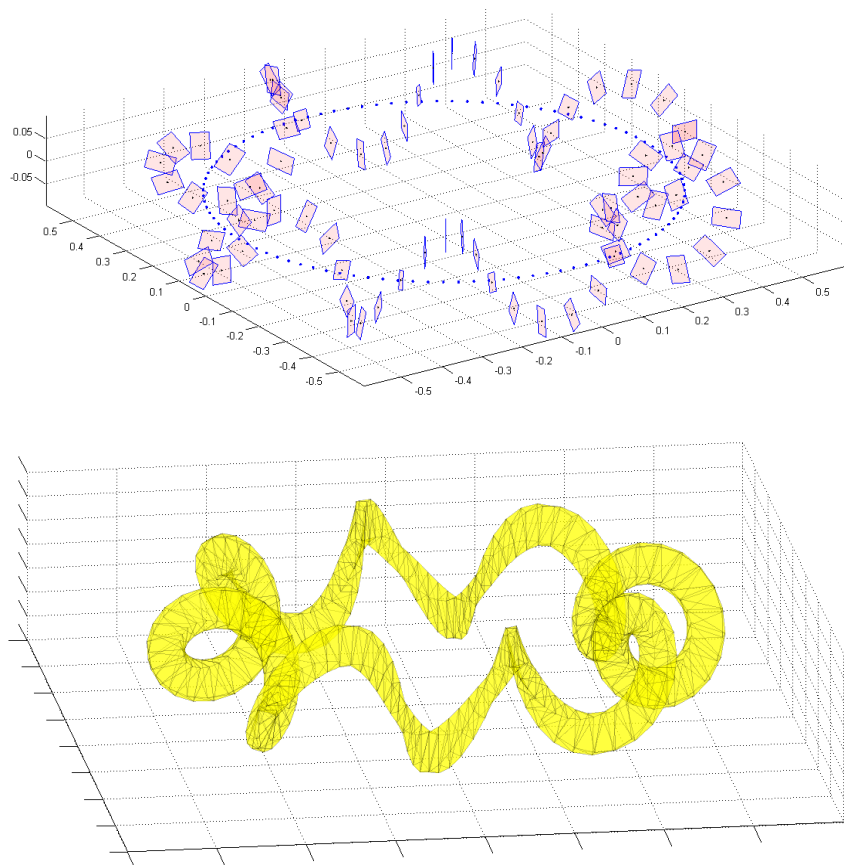


Figure 6.10: Helical coil faces construction and matching

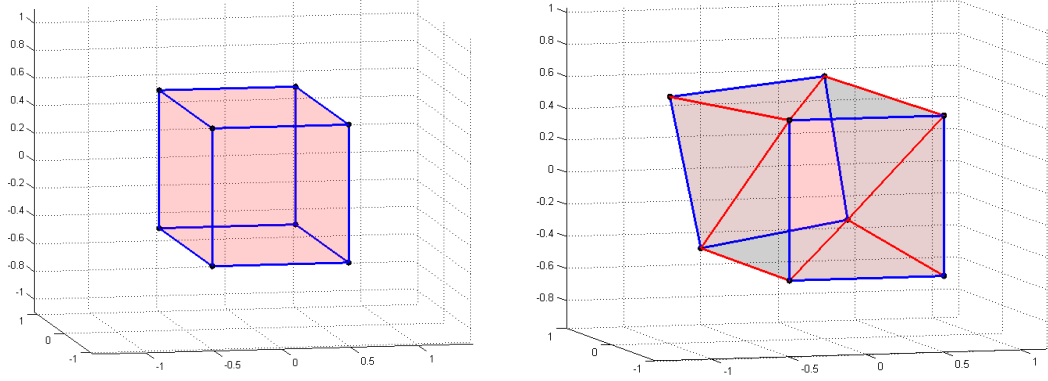


Figure 6.11: From hexahedron to polyhedron (face splitting)

to accomplish this result and use it to compute fields produced by this helical coil we had to introduce a novelty in our source calculation code. If we consider again figure 6.10, in particular the representation on the top where is reported each single face, we can realize that it is not possible matching two consequent faces and obtain a polyhedron with all plane faces, since the two given faces are not aligned but are rotate one another. Consequently lateral faces will result as rotating concave surfaces. Since for our formulation, as explained in chapter 3, all plane faces are requested, we are obliged, in order to continue applying the already used equations, to split these lateral not plane faces into triangle (because, as it is well known, through three non collinear points always pass only one plane) to assuring we are respecting the theory conditions. By this way we will have hexahedra no more, but we will work with generic polyhedra denoted by eight vertices and ten faces. In figure 6.11 on the left an hexahedron with all plane faces is shown, on the right a polyhedron with ingoing and outgoing rotated faces and eight triangular lateral faces.

Since our source calculation method is valid for a generic polyhedron, it will be sufficient re-adapting the old MATLAB functions to the new geometrical situation, simply adding some faces to the code iteration and distinguishing “frontal” quadrilateral faces from lateral triangular faces. Once we have a template-code for a single polyhedron, as done for the other coils, we can iterate on all helix volumes and compute the overall contribution on a grid of points.

6.3.4 Filamentary coils

A last useful tool to compute coils magnetic effects on a space point is represented by a code which is based on a filamentary approximation of a coil. In fact whenever a coil section is small compared to its length it will be appropriate considering it as a tight line crossed by a uniform current I .

In literature, on filamentary coils, there are lots of different proposed and validated formulations all referred to reviewed versions of Biot-Savart law. In this case we will refer to an expression for the Biot-Savart fields on a segment [10]. This method will lead to a MATLAB function, called `ABstick`, which will be involved on computing this field, in particular, magnetic induction field $\mathbf{B}(\mathbf{r})$. This method is thus based on filamentary coil subdivision into segments, of which we will compute the field generated on a space

point.

Filamentary approximation of a circular coil We now propose a simple comparison between a computation on a circular coil performed both with 3D discretization techniques and with filamentary approximation to verify the Biot-Savart law approximation precision. The calculation of \mathbf{B} field using this new function is also reported in figure 6.12 comparing it with its corresponding computation with 3D solid modelling tool.

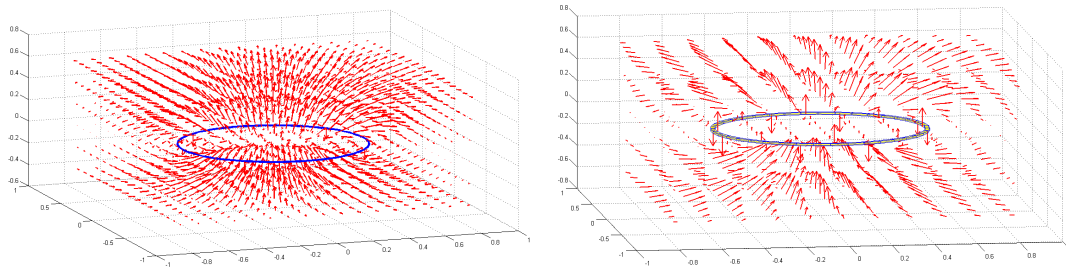


Figure 6.12: $\mathbf{B}(\mathbf{r})$ field calculation: filamentary and solid coils comparison

In table 6.2 is reported a comparison between modules of induction field computed in the same grid of points performed with the two proposed method, with the corresponding error. This error results always to be much lower than 0.5%.

$ \mathbf{B} $ 3D model [mT]	$ \mathbf{B} $ filamentary model [mT]	error%
0.0623	0.0620	0.3849
0.1208	0.1204	0.3094
0.0623	0.0620	0.3849
0.1208	0.1204	0.3094
0.2221	0.2219	0.0884
0.1208	0.1204	0.3094
0.0623	0.0620	0.3849
0.1208	0.1204	0.3094
0.0623	0.0620	0.3849

Table 6.2: B values comparison for solid and filamentary coils modelling

Chapter 7

Numerical simulations

The last remaining term of the overall equations system, the calculation of magnetic vector potential \mathbf{A}_s produced by a conductive domain Ω_s crossed by a determined current is now clearly well known. We are thus ready to use this new volume integral code in numeric simulations in order to verify its reliability and accuracy.

This new chapter will report all the obtained results, using the code to face with more and more complex settings. In particular we will start with easy, basic problems, necessary to verify the correctness of constitutive matrices \mathbf{R} and \mathbf{M} assembling. This initial step will consist on standard situations where a trivial conductive domain will be specifically stimulated in order to obtain a predetermined desired result. After these tests we will resort to a commercial reliable software such as ANSYS [24] to implement parallel simulation with the aim of benchmarking our code with complex geometries. With this two instruments (our volume integral code from one side and ANSYS to the other side) we will initially study eddy currents in a slab and a sphere in uniform vertical induction field \mathbf{B}_0 . To improve the test consistency, for the case of a sphere, we will also compare the results with an exact analytical solution. Finally we will engage in a fusion situation, testing the MATLAB code with RFX-mod experiment.

7.1 Matrices assembly testing

The first fundamental test has been verifying correctness of matrices assembly i.e. testing the properly implementation of (3.17) and (3.24).

Magneto static test problem

We obviously started with Ohm's constitutive matrix \mathbf{R} , assembled through the function `Rloc_mtrx_En` contained in `Rmatrix_Enh` script, as illustrated in chapter 4.

Aiming to avoid errors superposition this matrix has been singly tested in a magneto static problem (after having verified its positive definiteness). Firstly a simple testing domain discretized with hexahedra has been used, successively we have improved the code complexity trying to solve the same magneto static problem but with a polyhedral grid. An example of these two grids is reported in figure 7.1.

The aim of the test is stimulating the grid to obtain a vertical symmetric static \mathbf{B} field along z direction equal to 1 Tesla ($B = 1 [T]$). Whenever there were assembly errors in \mathbf{R} matrix the solution would not be as expected. This test has been passed without complications, yielding the results in figure 7.2 (due to numerical banality of solution, the values are not here reported).

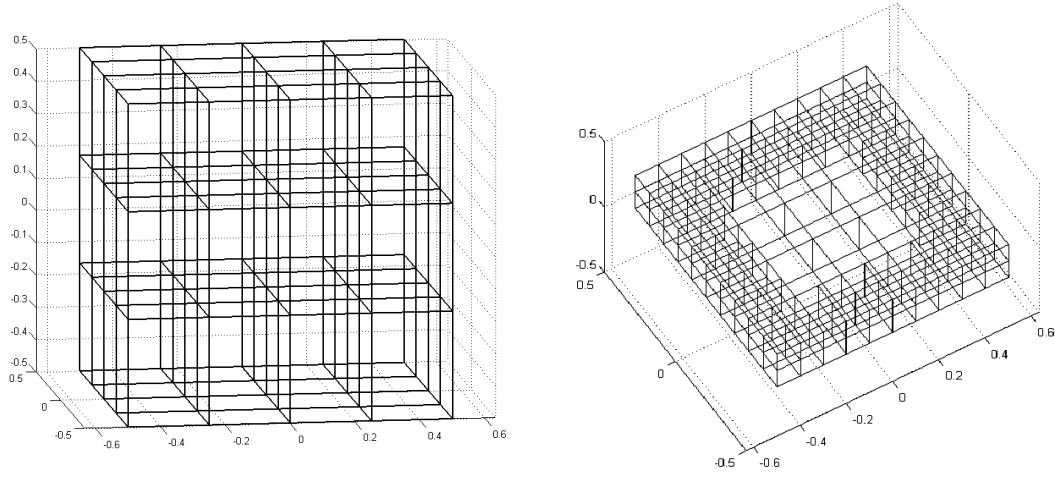


Figure 7.1: Examples of discretization grids: hexahedral (on the left) and polyhedral grid (on the right)

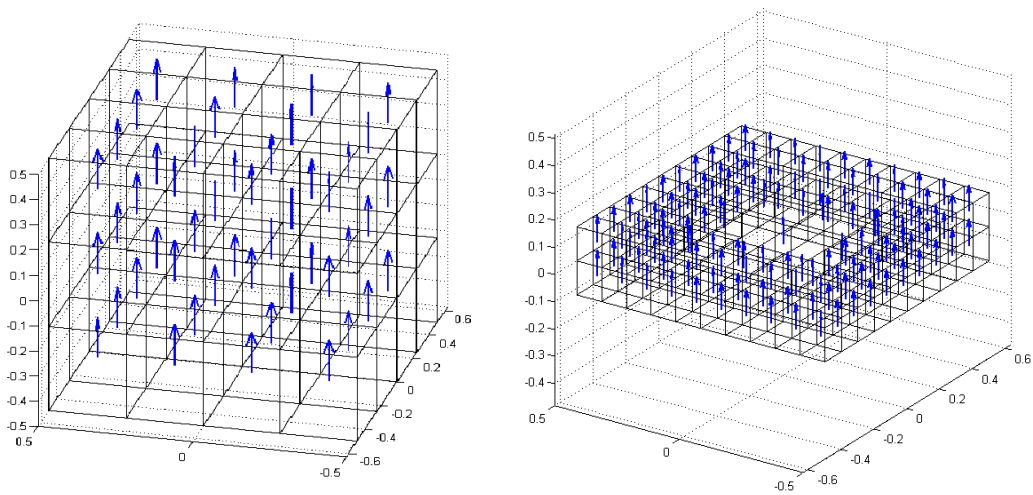


Figure 7.2: Magneto static field test results

M matrix testing: first eddy current problem solution

After the validation of Ohm's constitutive relation we proceeded with a fundamental step: solution of the first eddy current problem with the new volume integral code. Whenever the code is able to correctly face with this problem we can conclude that even the constitutive relation between current and magnetic vector potential matrix, \mathbf{M} , has been accurately assembled. Even this time, the test has been done both with hexahedral and polyhedral grid.

The slab was immersed in a vertical external field, directed along positive z axis, whose amplitude was equal to $\hat{B}_0 = 1 [T]$. The analysis was executed with a frequency $f = 50 [Hz]$, on a volume made of copper, whose resistivity ρ was chosen to be equal to $1.68 \cdot 10^{-8} [\Omega \cdot m]$ with magnetic permeability $\mu = \mu_0 = 4\pi \cdot 10^{-7} [H/m]$ (since for copper $\mu_r \approx 1$). In this initial phase, since the aim was to validate the matrix assembly (and not to test numerical values yet), we did not care about domain dimensions, taking into account only the material depth in such a way that this was not higher than skin depth in that conditions i.e.

$$\delta = \sqrt{\frac{2}{\mu\sigma\omega}} \quad (7.1)$$

where, the conductivity σ is equal to $1/\rho$ (as it is well known that $\rho = 1/\sigma$) and $\omega = 2\pi f [rad/s]$ is the angular speed. It follows that, in our conditions, $\delta \approx 9.225 [mm]$. We underline that, differently from previously presented domains used for static case testing, hereafter new domains with different discretization grids are involved in order to assure the above described requests on current penetration depth (δ) related with each discretizing sub-volume dimensions.

Since we want to solve an eddy current problem and thus we are now running a simulation in magneto quasi static conditions, the overall equation (2.26), will not be solved in time domain, but, thanks to Steinmetz transforms, it will be solved in frequency domain as (reordering left hand side terms):

$$\left(\mathbf{C}_r^T \mathbf{R}_r \mathbf{C}_r + j\omega \mathbf{C}_r^T \mathbf{M}_r \mathbf{C}_r \right) \mathbf{T}_r = -j\omega \mathbf{C}_r^T \rho_{\tilde{e}_r} \mathbf{A}_s(\mathbf{r}) \quad (7.2)$$

where j is the imaginary unit.

As a consequence we will present a **real** and an **imaginary solution**: the former is slab reaction at the present time (or $t = 0 [s]$) and it is the ohmic contribution on overall slab current, while the latter is slab magnetic reaction due to Faraday-Lenz' law i.e. it is the inductive contribution on overall slab current.

In conclusion we obtain results reported in figures 7.3 and 7.4 where a typical eddy current disposition in the tested slabs is clearly visible. This suggests the correctness of \mathbf{M} matrix implementation. Further and definitive proofs will be presented in next sections.

7.2 Code benchmark: slab in a vertical field

Once matrices proper assembly has been verified, we can go on with the first actually significant simulation that, differently from the previous ones, has been done with the aim of testing numerical results in order to provide a first proof of code precision.

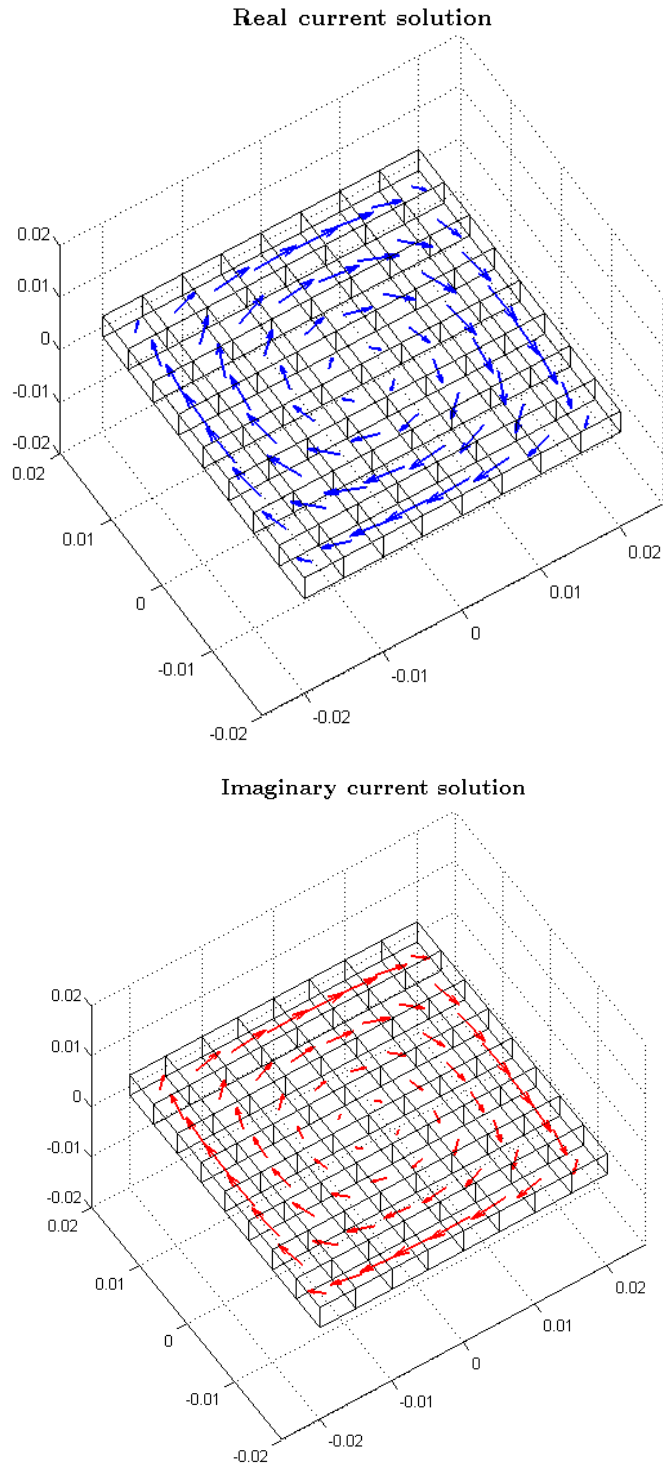


Figure 7.3: Induced eddy currents in hexahedral meshed volume ($l = 0.04 [m]$, $h = 0.004 [m]$)

As already outlined, during this phase our benchmark code will be represented by ANSYS software. Consequently, we aim at testing our volume integral code in a parallel trial between this two software that will lead to a comparison of numerical results.

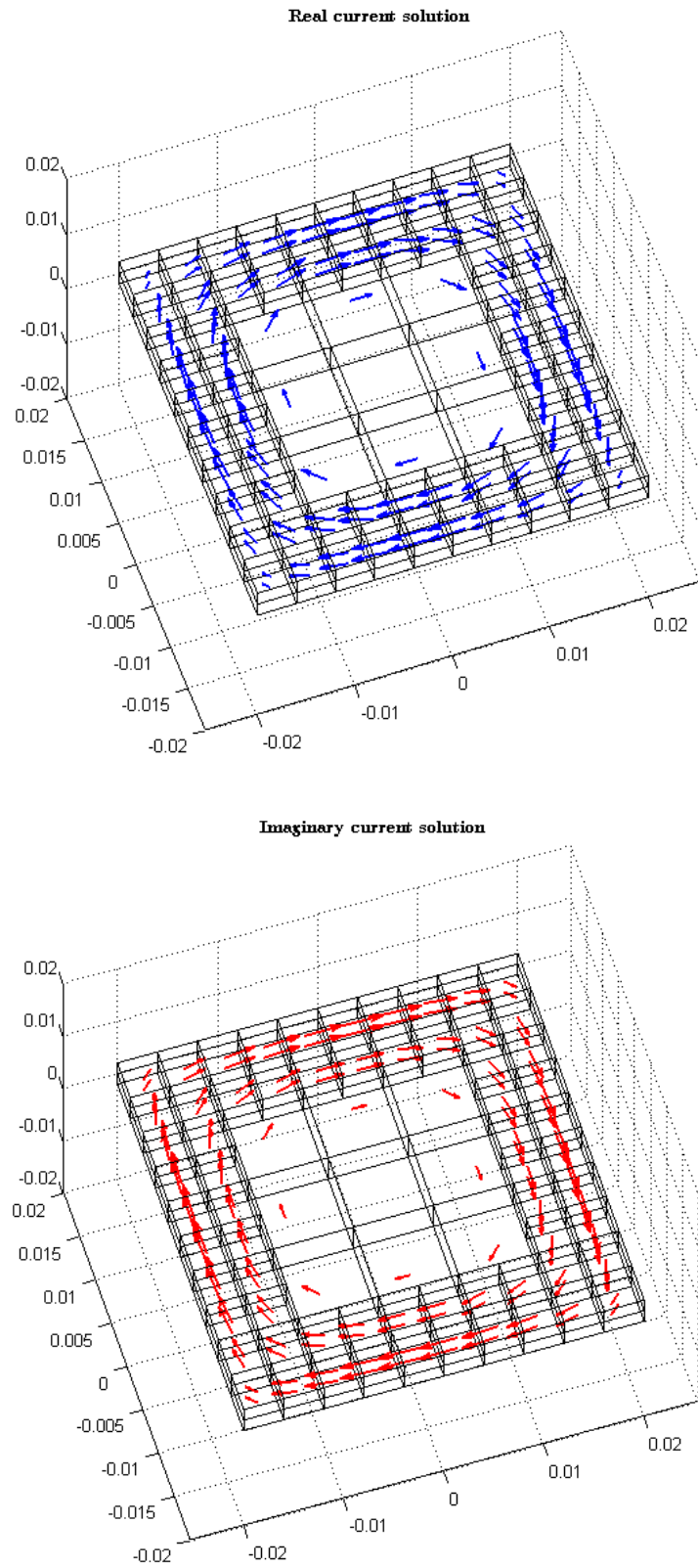


Figure 7.4: Induced eddy currents in polyhedral meshed volume ($l = 0.04 [m]$, $h = 0.004 [m]$)

This procedure implies a substantial difference with respect to above simulations, and it request a novelties in our code: use of earlier described data post processing tools. In fact during this benchmarking test we will compare not only the obtained current density arrays but we will also compute induction field $\mathbf{B}(\mathbf{r})$ in different space points. It is thus clear that by this way post processing codes will be tested too.

Problem data Our test problem will be conducted on a metallic slab, whose dimensions are $l = 40 [mm]$ and $h = 4 [mm]$ (such as the total volume will be $l \times l \times h$). Differently from previous sections we will set $\rho = 1 \cdot 10^{-7} [\Omega \cdot m]$ and $\mu = \mu_0 = 4\pi \cdot 10^{-7}$ (to avoid excessive skin depth issues). The slab will be stimulated with an external alternating field \mathbf{B}_0 whose pick value is $\hat{B}_0 = 1 [T]$ varying with sinusoidal law at frequency $f = 50 [Hz]$. For the sake of completeness we also report the resulting skin depth equal to $\delta = 22.5 [mm]$.

$\mathbf{A}_s(\mathbf{r})$ calculation Considering the simplicity of externally imposed field, this last one will not be calculated with sources tools codes described in chapter 6 but, needing a vertical field along z direction, this known term will be computed as:

$$\mathbf{A}_s(\mathbf{r}) = \begin{cases} -\frac{1}{2}\hat{B}_0 r_y \\ +\frac{1}{2}\hat{B}_0 r_x \\ 0 \end{cases} \quad (7.3)$$

where $\mathbf{r} = (r_x, r_y, r_z)$ are calculation point components. As already explained in section 6.2 this calculation point will be decided to be in correspondence of each dual edge \tilde{f}_j barycenter. Then this calculated value is integrated along each dual edge taking into account a properly coherent overall orientation. This just presented expression is derived considering that:

$$\mathbf{B} = \nabla \times \mathbf{A} = \begin{vmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ A_x & A_y & A_z \end{vmatrix} \quad (7.4)$$

where $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ are the axes versors. Since we want $\mathbf{B} = (0, 0, B_z)$ (with $B_z = \hat{B}_0$) it follows that (7.3) verifies:

$$\mathbf{B} = \begin{cases} 0 \\ 0 \\ \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right) \hat{\mathbf{z}} \end{cases} \quad (7.5)$$

Volume integral code solution

After having given problem specifications we can continue and solve the problem. The discretization grid, shown in figure 7.5, is composed by a single layer of 8×8 elements each of them of dimensions $5 \times 5 \times 4 [mm]$.

A graphic representation of solution, in term of current density \mathbf{J} , can be found in figure 7.6.

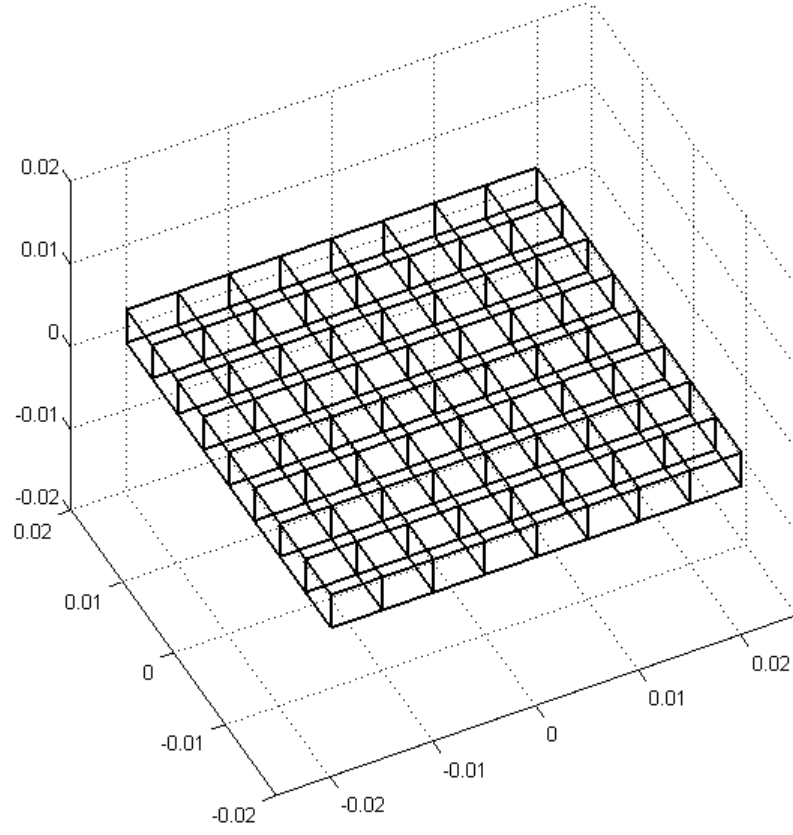


Figure 7.5: Discretization grid of the slab

As announced, beyond current density, also the induction field \mathbf{B} generated by the imaginary current has been computed. This reaction field has been calculated with post processing codes on a grid of points located above the conducting domain at a height equal to $z_0 = 0.01 [m]$ (the slab is centrally positioned respect to Cartesian axes). Results are as in figure 7.7.

Current flux $i(t)$ on a section

The last considered parameter for our comparison is current flux flowing through a slab section. The established section is displayed in figure 7.8. This section will be the same used for calculating the current flow in ANSYS simulation too.

To compute this current in our volume integral code we will recur to

$$i(t) = \mathbf{CT}(t) \quad (7.6)$$

already described and reported as (2.8). This formula computes the flow in each face of the domain; then, only the desired faces will be selected and results will be summed to obtain the current on that section.

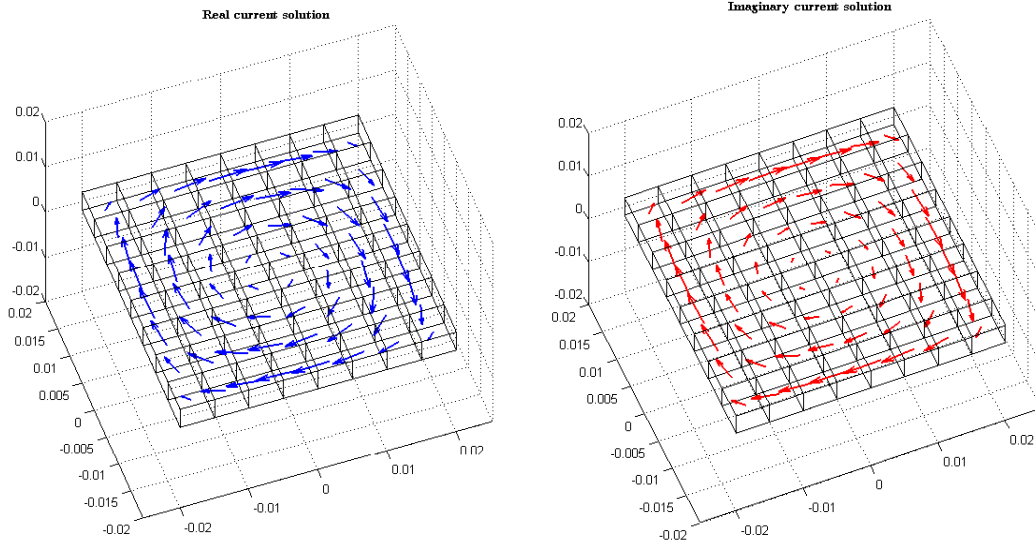


Figure 7.6: Volume integral solution: current density \mathbf{J}

The so calculated current on the section results to be:

$$I_{sec} \begin{cases} I_{real} = 82.67 [A] \\ I_{imaginary} = 1495.5 [A] \end{cases} \quad (7.7)$$

These values will be compared to the ones coming from ANSYS simulation.

Ansyes simulation

As previously said, the same problem on the slab has been resolved also using a commercial software such as ANSYS. Dimensions and magnetic stimuli are obviously equal to that used for earlier volume integral simulation.

Since this software, differently from our code, is a FEM simulator we will need to mesh a sufficient amount of vacuum around the slab under investigation. Consequently we will build a cube around the slab, and we will create a **mapped mesh**¹. In figure 7.9 both vacuum and slab meshes are reported.

Before solving the problem, boundary conditions must be applied to the meshed volume. These conditions will be such that the resulting field will be vertically directed along z axis and equal to $\hat{B}_0 = 1 [T]$. In order to obtain this field we will recur again to (7.4) and (7.5). Imposing

$$\begin{cases} \frac{\partial A_y}{\partial x} = \hat{B}_0 \\ \frac{\partial A_x}{\partial y} = 0 \\ A_z = 0 \end{cases} \quad (7.8)$$

the desired vertical field is obtained.

¹This mesh type is fundamentally constituted by multiple layer of hexahedra.

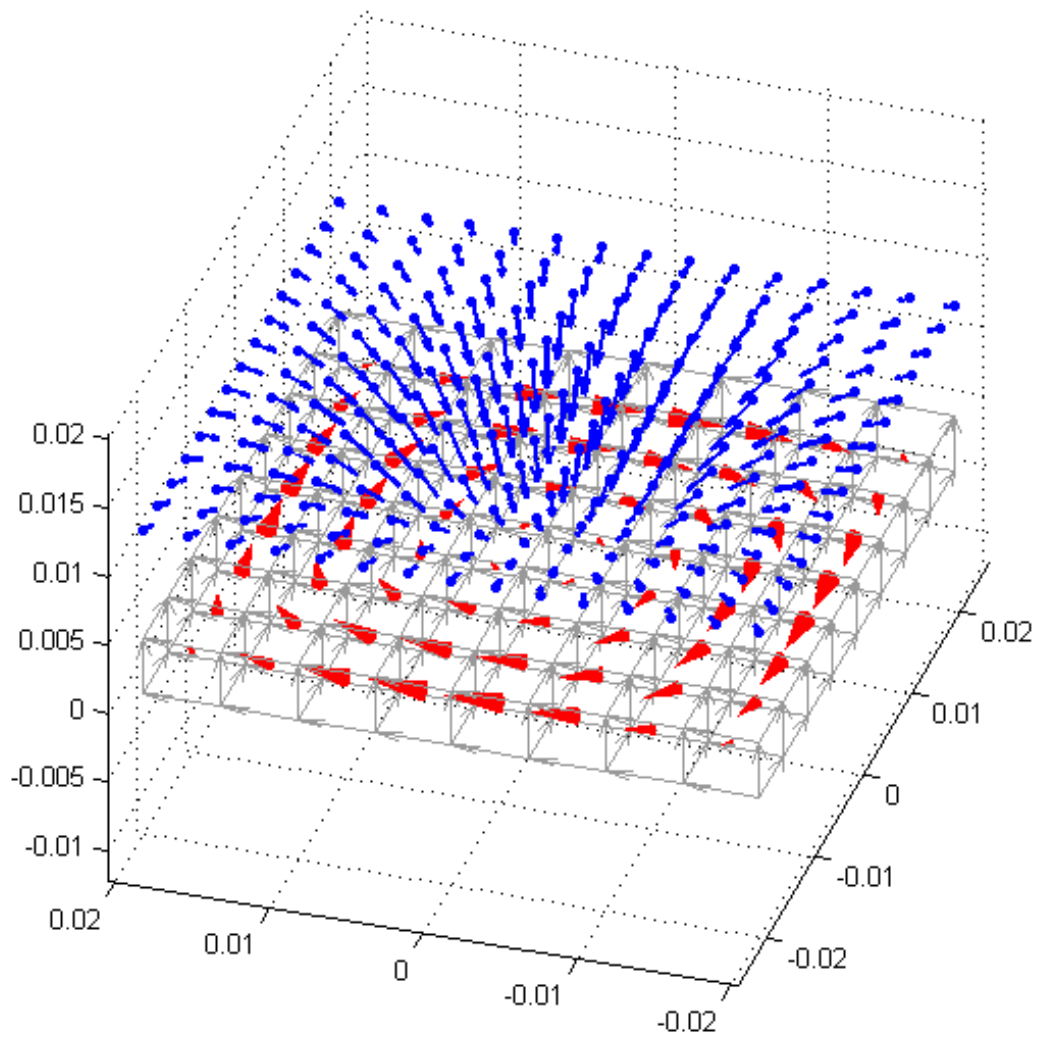


Figure 7.7: Reaction B field above slab due to imaginary current density

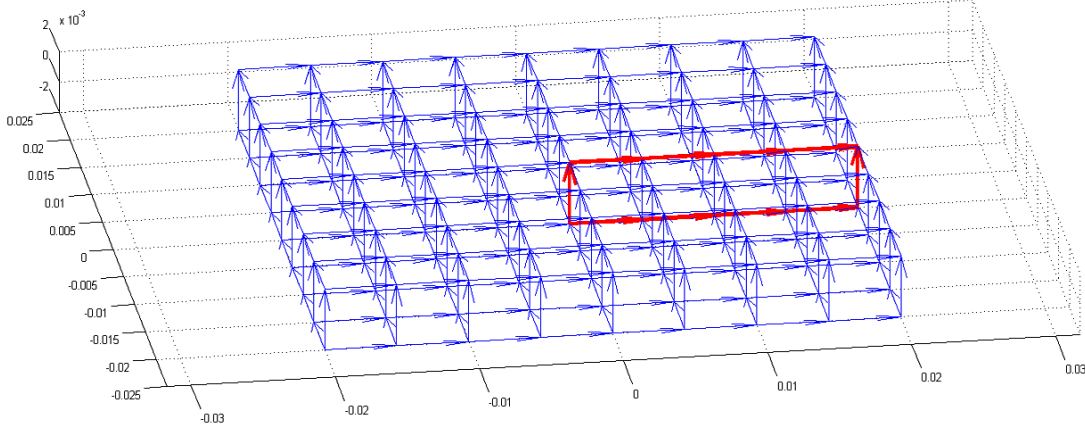


Figure 7.8: Slab section displaying for current flow calculation

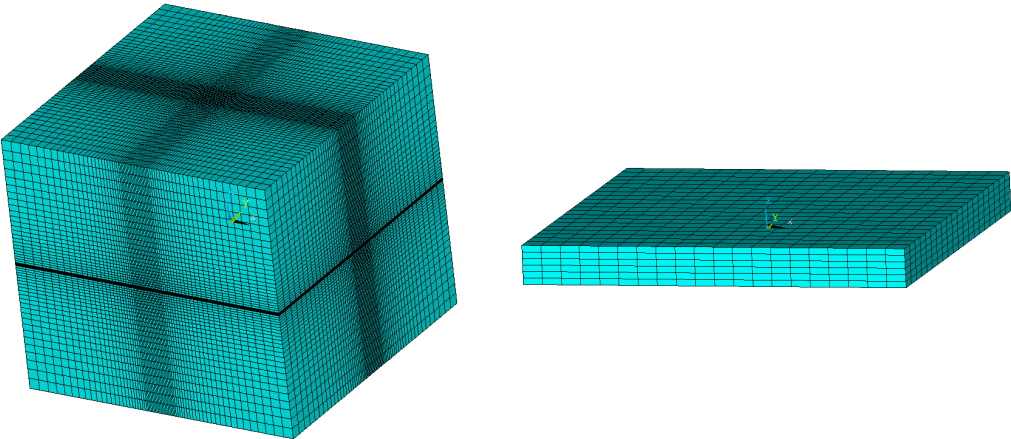


Figure 7.9: Air volume mesh (not visible slab inside) and conducting slab mesh

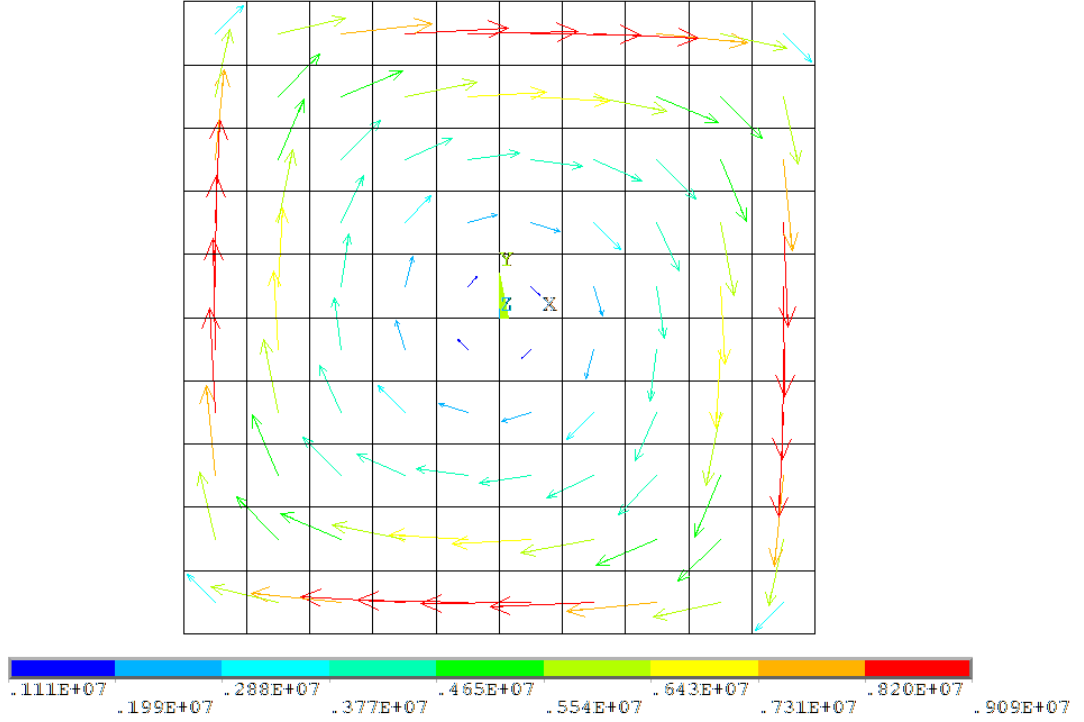


Figure 7.10: ANSYS current density J (top view)

Solution post processing

Once the solution is achieved a post processing is necessary to compare results with the volume integral solution.

In particular we will focus on:

1. current flow i on a slab section (previously identified) computed using integral form of Ampere's Law

$$\oint_{\partial S} \mathbf{H} \cdot \mathbf{t} = I_{sec} \quad (7.9)$$

performing an integration of magnetic field \mathbf{H} with ANSYS, along a rectangular closed path ∂S around the assigned slab section S ;

2. induction field \mathbf{B} computed on a line of points above the slab. This line will be coincident with the central line of points of the grid previously shown for volume integral solution post processing.

The problem solution leads to results reported in figure 7.10 and 7.11 where the total current density \mathbf{J} and the imaginary reaction field \mathbf{B}_{im} generated by the current in the slab are respectively shown. Numerical values will be reported in next section.

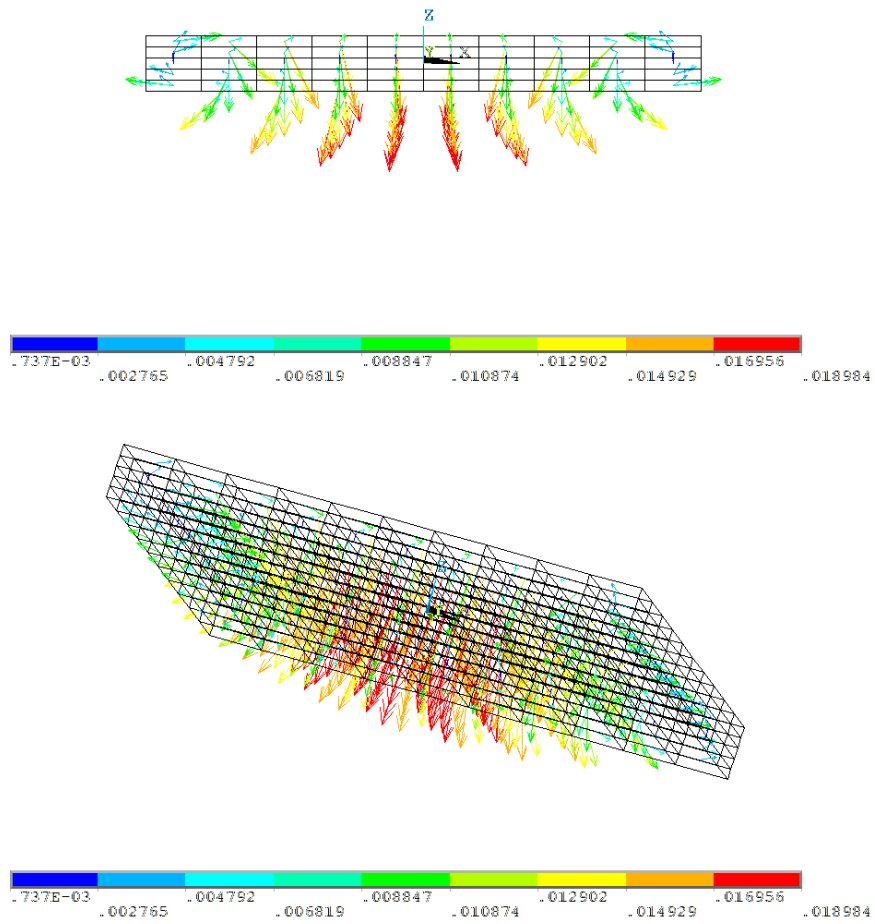


Figure 7.11: ANSYS imaginary reaction field B_{im} (frontal and 3D views)

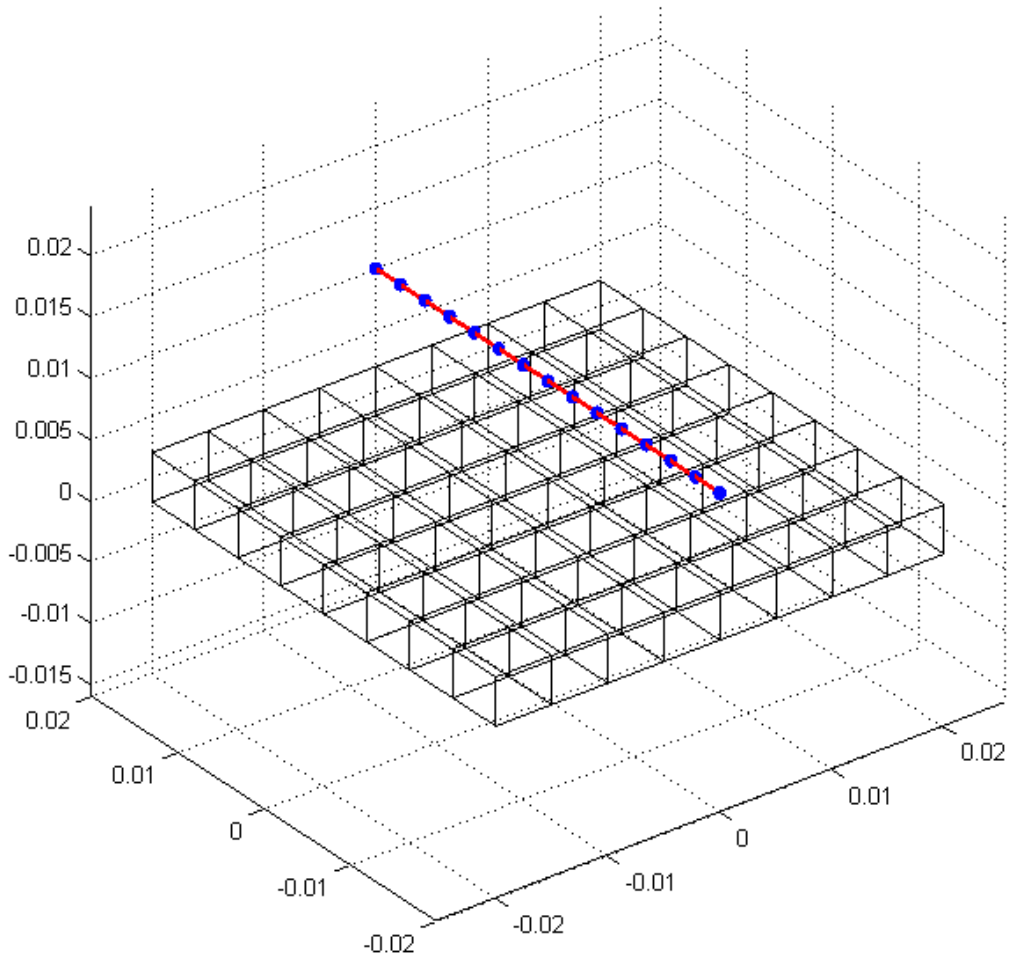


Figure 7.12: B field evaluation path

Results comparison

As already explained results comparison will be done with respect to two physical quantities:

- the current flow i on a slab section (already displayed in 7.8);
- the imaginary induction field B_{im} computed on points disposed along a line. This line path is exhibited in figure 7.12; from this scheme we can see that path is extended above the slab at $z_0 = 0.01 [m]$, $x_0 = 0 [m]$ and with y varying from $y_0^A = -0.02 [m]$ to $y_0^B = +0.02 [m]$.

Current flow i

Related results are scheduled in table 7.1. Current values computed with ANSYS versus the same values calculated with our volume integral code are reported. We can notice that for what regards the volume integral results are reported two similar values denoted as “method 1” and “method 2”. These results are different because they are computed

in different ways: the first one is evaluated thanks to (7.6) while the second method considers the resulting current computed using the current density \mathbf{J} in each volume next to the considered section multiplied for the corresponding face area.

	ANSYS	V. I. method 1	V. I. method 2
$I_{re} [A]$	80.93	82.67	79.71
$I_{im} [A]$	1448.22	1495.50	1455.50
$ \mathbf{I} [A]$	1450.48	1497.78	1457.68

Table 7.1: Current flow i values comparison

Finally in table 7.2 are reported the normalized percentage errors computed for each pair of value of the previous table. It will be reported $err_{re}^{\%}$ as the error related to real current, $err_{im}^{\%}$ as the error related to imaginary current and finally $err_{mod}^{\%}$ as the error related to current module. All these parameters will be computed for ANSYS versus volume integral “Method1” and for ANSYS versus volume integral “Method2” too.

	ANSYS VS m.1	ANSYS VS m.2
$err_{re}^{\%}$	2.10	1.53
$err_{im}^{\%}$	3.16	0.50
$err_{mod}^{\%}$	3.16	0.49

Table 7.2: Current flow i errors

All the errors are small considering that the discretization grid of volume integral simulation is rather coarse. Moreover we can see that with “Method 2” these errors are actually low: this discrepancy is due to post processing method used with ANSYS to evaluate the current in the section. Since the integration with Ampere’s law is not a really precise tool, using a similarly approximate procedure (multiplying J for the corresponding section) also for the volume integral we can obtain two nearer results. This explains numerical outputs.

Imaginary B field results

In figures 7.13, 7.14 and 7.15 are respectively reported: imaginary B_y field component along the path, imaginary B_z component and finally, imaginary B module (respect to field directions, not to real-imaginary components). These charts are immediately clear, each of them reports field values end error in the same corresponding calculation points. We can observe that the error is extremely low at the centre of the slab ($\approx 1\%$) while is higher at its borders ($\approx 5.5\%$). This is probably due to relation between discretizing grid and current distribution: the current tends to be higher at the slab border so to obtain a more accurate result a subgridding to have a finer mesh is needed.

Conclusion

With this first comparison we can state that the proposed new code provides correct and precise results. Moreover the most interesting aspect regards involved “numbers”.

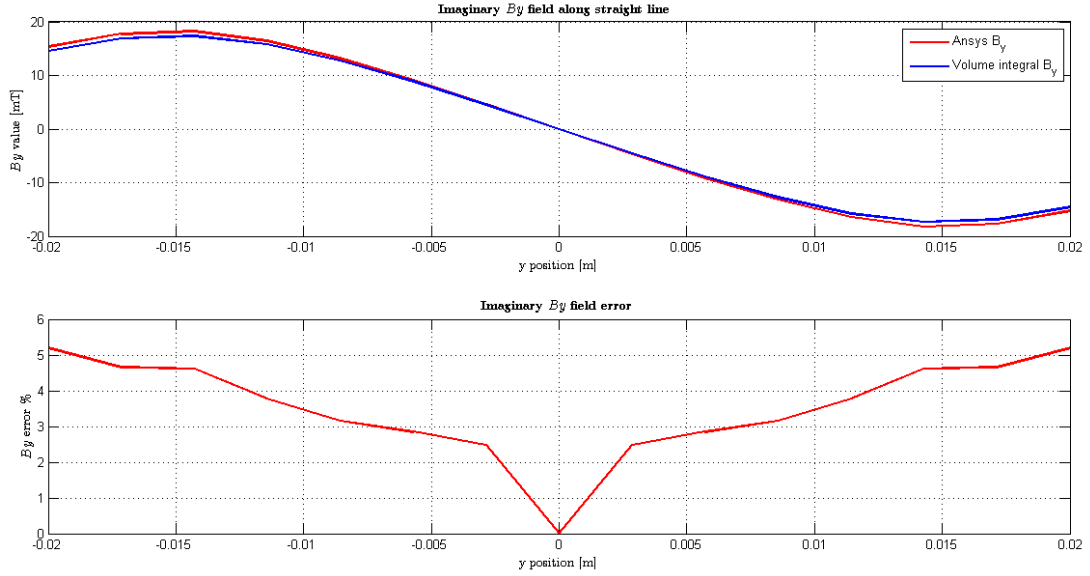


Figure 7.13: Imaginary B_y field comparison between ANSYS and the volume integral code

In fact, despite results are substantially the same for the two simulation, in the case of ANSYS simulation to accomplish these results a discretization grid composed by 249696 elements (of which 2400 belonging to the slab) was used while for volume integral simulation we were able to obtain good numerical results involving only 64 elements and 369 unknowns: an actual computational saving!

7.3 Code benchmark: sphere in a vertical field

After comparisons on a slab we now propose a more substantial experience; in fact we will benchmark our code against an analytical solution. The implied geometry is represented by a sphere immersed in a uniform vertical field directed along z axis. This choice requires two novelties respect to previous simulation:

- since we want to face with a spherical geometry, this needs a more complex and dense discretizing mesh;
- the analytical solution is an accurate paradigm, so the comparison results will be actually definite.

Before proceeding with the simulation we have to introduce analytic solution equations.

7.3.1 Analytical solution of a sphere in uniform B field

We consider a sphere immersed in an alternated vertical field $\mathbf{B} = (0, 0, \hat{B}_0)$ varying with sinusoidal law at frequency f and angular speed $\omega = 2\pi f$. Following equations we will be referred to a spherical polar coordinates system shown in figure 7.16. This reference will simplify the notation.

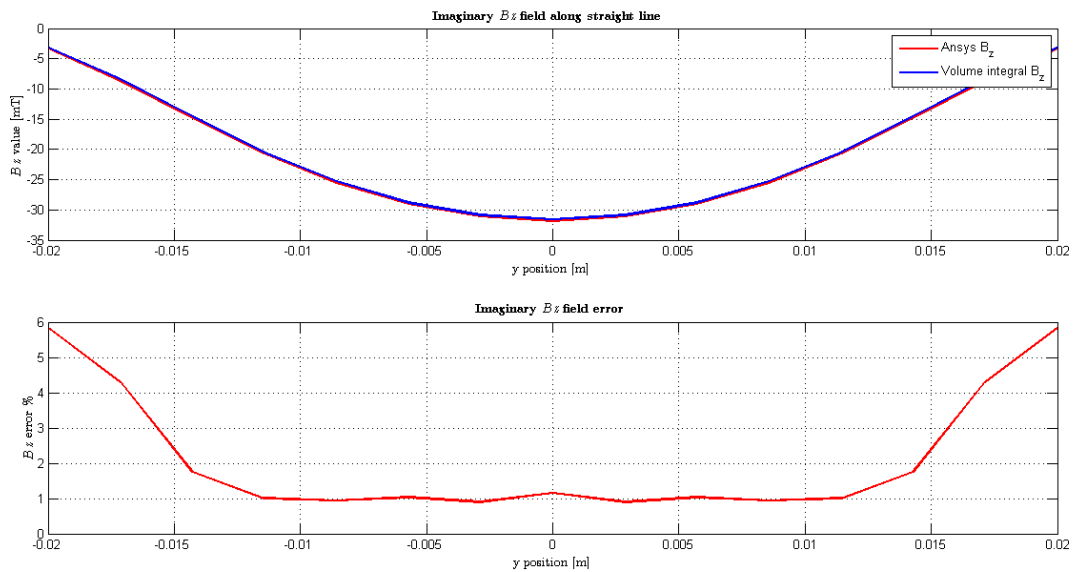


Figure 7.14: Imaginary B_z field comparison between ANSYS and the volume integral code

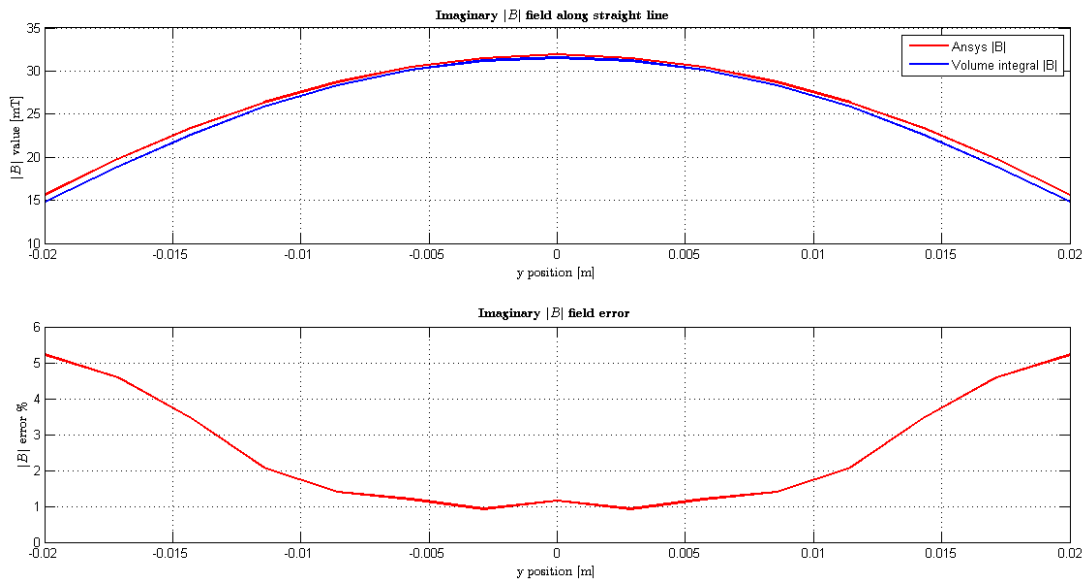


Figure 7.15: Imaginary $|B|$ field comparison between ANSYS and the volume integral code

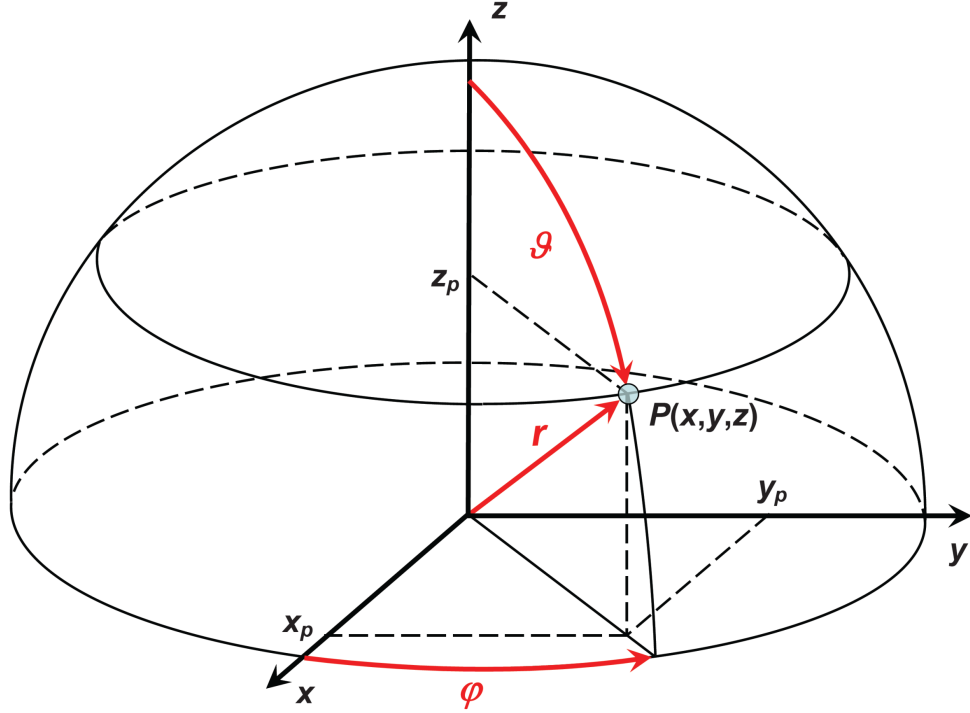


Figure 7.16: Spherical polar coordinate reference

Considering an induction field positively directed along z axis (i.e. from the bottom to the top of reference system), of magnitude $\hat{B}_0 [T]$, surrounding a sphere whose radius is equal to $R [m]$, with magnetic permeability $\mu = \mu_r \mu_0$ where μ_r depends on sphere material and $\mu_0 = 4\pi \cdot 10^{-7} [H/m]$, and with conductivity $\sigma [\Omega^{-1}m^{-1}]$ we can obtain [21] expressions of magnetic vector potential $\mathbf{A}(r, \theta)$ and of magnetic induction field $\mathbf{B}(r, \theta)$ in spherical polar coordinates as:

$$\mathbf{A}(r, \theta) = \begin{cases} a j_1(kr) \hat{B}_0 \sin(\theta) \hat{\varphi} & \text{if } r < R \\ \left(\frac{r}{2} + \frac{b}{r^2}\right) \hat{B}_0 \sin(\theta) \hat{\varphi} & \text{if } r > R \end{cases} \quad (7.10)$$

and

$$\mathbf{B}(r, \theta) = \begin{cases} \frac{2a}{r} j_1(kr) \hat{B}_0 \cos(\theta) \hat{r} - \frac{a}{2r} \gamma(r) \hat{B}_0 \sin(\theta) \hat{\theta} & \text{if } r < R \\ \left(1 + \frac{2b}{r^3}\right) \hat{B}_0 \cos(\theta) \hat{r} - \left(1 - \frac{b}{r^3}\right) \hat{B}_0 \sin(\theta) \hat{\theta} & \text{if } r > R \end{cases} \quad (7.11)$$

where

$$\gamma(r) = [j_1(kr) + kr(j_0(kr) - j_2(kr))] \quad (7.12)$$

and with

$$\begin{aligned}
k &= (-1 + j) \sqrt{\frac{\omega\mu\sigma}{2}} = \frac{-1 + j}{\delta} \\
a &= \frac{3R}{2D} \\
b &= \frac{R^3}{D} \left[j_1(kR) \left(1 + \frac{1}{2\mu_r} \right) - \frac{kR}{2\mu_r} j_0(kR) \right] \\
D &= j_1(kR) \left(1 - \frac{1}{\mu_r} \right) + \frac{kR}{\mu_r} j_0(kR)
\end{aligned} \tag{7.13}$$

The **functions** j_0, j_1 **and** j_2 are the Bessel's function defined as

$$\begin{aligned}
j_0(z) &= \frac{1}{z} \sin z \\
j_1(z) &= \frac{1}{z} \left(\frac{1}{z} \sin z - \cos z \right) \\
j_2(z) &= \frac{1}{z} \left[\left(\frac{3}{z^2} - 1 \right) \sin z - \frac{3}{z} \cos z \right]
\end{aligned} \tag{7.14}$$

Once we have the expression of magnetic induction field we can also calculate equation of current density $\mathbf{J}(r, \theta)$ from

$$\mathbf{J} = \nabla \times \mathbf{H} = \nabla \times \frac{\mathbf{B}}{\mu} \tag{7.15}$$

which, if $\mu = \text{const}$, leads to:

$$\mu \mathbf{J} = \nabla \times \mathbf{B} \tag{7.16}$$

in other words we can compute the current density calculating the curl of \mathbf{B} (in spherical coordinates).

The performed calculation² starts considering that [7]:

$$\nabla \times \mathbf{B} = \frac{1}{r^2 \sin \theta} \begin{vmatrix} \hat{r} & r\hat{\theta} & r \sin \theta \hat{\varphi} \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \theta} & \frac{\partial}{\partial \varphi} \\ B_r & rB_\theta & r \sin \theta B_\varphi \end{vmatrix} \tag{7.17}$$

which, since $B_\varphi = 0$ and since terms contained in (7.11) are independent of φ leads to

$$\mu \mathbf{J} = \nabla \times \mathbf{B} = \frac{1}{r} \left[\frac{\partial(rB_\theta)}{\partial r} - \frac{\partial B_r}{\partial \theta} \right] \hat{\varphi} \tag{7.18}$$

We thus proceed applying (7.11) to (7.18); we can distinguish two cases:

- if $r > R$ i.e. **point is external** respects to the sphere, current density results to be

$$\mathbf{J}(r, \theta) = 0 \tag{7.19}$$

and this is correct since there can not be current out of conducting domain;

²This calculus was not be developed in [21] but it has been independently carried out for this work.

- if $r < R$ i.e. the point is inside the sphere, the current density is not null and it can be calculated thanks to (7.18).

As a consequence we will now focus only on $r < R$ situation.

With respect to (7.18) we obtain:

$$\begin{aligned} \frac{\partial(rB_\theta)}{\partial r} = & m(\theta) \left[\frac{\partial(j_1(kr))}{\partial r} + k(j_0(kr) - j_2(kr)) \right] + \\ & + m(\theta) \left[kr \left(\frac{\partial(j_0(kr))}{\partial r} - \frac{\partial(j_2(kr))}{\partial r} \right) \right] \end{aligned} \quad (7.20)$$

and

$$\frac{\partial B_r}{\partial \theta} = -\frac{2a\hat{B}_0}{r} j_1(kr) \sin \theta \quad (7.21)$$

with

$$m(\theta) = -\frac{a\hat{B}_0 \sin \theta}{2} \quad (7.22)$$

and where

$$\frac{\partial(j_0(kr))}{\partial r} = \frac{r \cos(kr) - (1/k) \sin(kr)}{r^2} \quad (7.23)$$

$$\frac{\partial(j_1(kr))}{\partial r} = \frac{1}{kr^2} [\cos(kr) - j_0(kr)] + \frac{1}{kr} \left[\frac{\partial(j_0(kr))}{\partial r} \right] + \frac{1}{r} \sin(kr) \quad (7.24)$$

$$\frac{\partial(j_2(kr))}{\partial r} = -\frac{6j_0(kr)}{k^2 r^3} + \left(\frac{3}{(kr)^2} - 1 \right) \frac{\partial(j_0(kr))}{\partial r} + \frac{kr \sin(kr) + 2 \cos(kr)}{r^3} \quad (7.25)$$

These reported equation has been implemented in a MATLAB function which is thus able to analytically compute the fields generated by the sphere and in the sphere. These fields values will be used as a standard reference an they will be compare to solution provided by volume integral code simulation.

Problem specifications

As far as problem specifications are concerned, we will perform this test on a conductive sphere whose radius is equal to $r_0 = 0.05 [m]$ and centered on axis origin ($c_0 = (0, 0, 0)$). The other sphere parameters are decided to be as in table 7.3.

$r_0 [m]$	$\rho [\Omega \cdot m]$	$\mu_0 [H/m]$	μ
0.05	$1 \cdot 10^{-7}$	$4\pi \cdot 10^{-7}$	1

Table 7.3: Test problem on a sphere specifications

The sphere is immersed in a uniform vertical field $\mathbf{B} = (0, 0, \hat{B}_0)$ with $\hat{B}_0 = 1 [mT]$.

7.3.2 Ansys solution

The kernel of problem solution with ANSYS software is represented by domain meshing. First of all problem dimensions have been reduced exploiting geometry symmetries: in place of the whole sphere only one eighth has been studied and thus meshed, particularly that spherical sector laying on the first octant (octant with $x, y, z > 0$). Around the sphere sector a vacuum cube containing the sphere inside has been constructed. Finally, both these volumes has been meshed with a mapped mesh constituted by hexahedra. Overall, 84800 elements has been used, of which 32000 for the sphere mesh; the unknowns are 89965.

Domain meshing result is shown in figures 7.17 and 7.18.

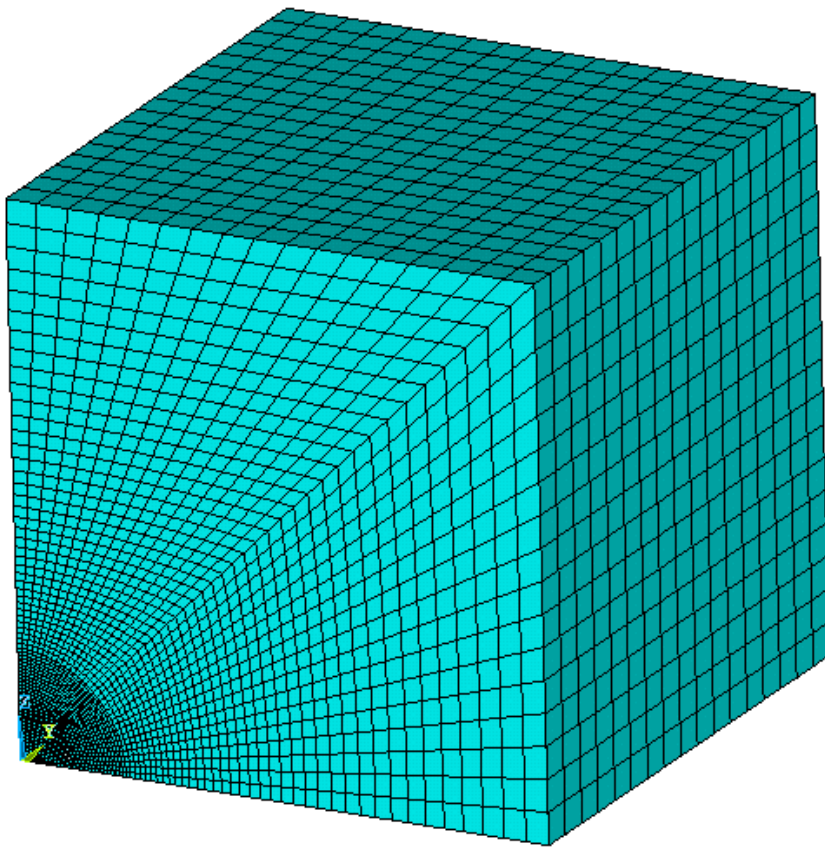


Figure 7.17: Overall view of ANSYS domain discretization: sphere sector and vacuum

Boundary conditions With respect to the slab problem, boundary conditions are very similar except for a further request. In fact we already need a vertical induction field as done in previous simulation (whose mathematical conditions are reported in (7.8)), but now also solution symmetry on the sphere planes with $x = 0$ and $y = 0$ has to be guaranteed. In particular, current density \mathbf{J} must be perpendicular to these two planes. This condition is reflected on imposing a null electric potential difference in each mesh node belonging to the mentioned surfaces: by this way \mathbf{J} is prevented to have a not null component parallel to symmetry planes.

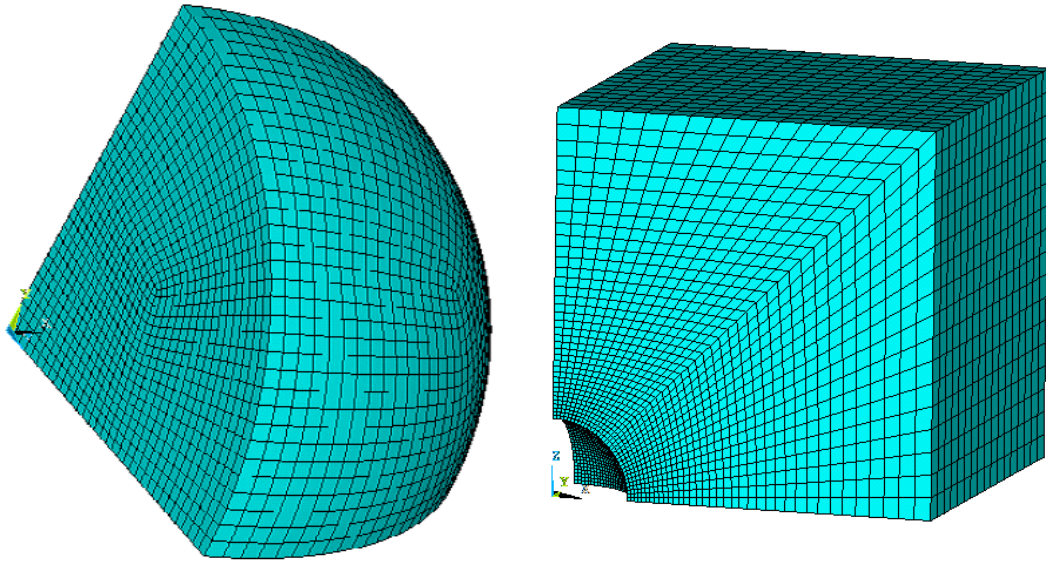


Figure 7.18: Volumes distinction: sphere eighth (on the left) and vacuum (on the right)

Ansys solutions Once boundary conditions have been setted and geometry has been meshed we can proceed and solve the problem finding the magnetic induction field \mathbf{B} and the current density \mathbf{J} in each point of the discretized domain. In figure 7.19 and 7.20 the imaginary current density in the sphere and the consequential induction field B_{im} caused by the current flowing in the conducting domain are respectively shown. These two quantities will be used as terms of comparison for next investigations.

Analytical formulas implementation testing

Before studying the same problem with the new volume integral code we want to benchmark, we have used just presented ANSYS solution to test the MATLAB functions which have been implemented to analytically compute current density field in the sphere. In fact, since related formulas have been independently computed for this work, before comparing them with our code solution their correctness has to be verified.

In figure 7.21 is thus reported a graph showing real (top figure) and imaginary (bottom figure) current density computed both with ANSYS and analytical formulas along a path coinciding with Cartesian axis x . As it is clearly visible values perfectly coincide, giving proof of the correct implementation and computation of the formulas (and mutually validating ANSYS simulation too).

7.3.3 Volume integral code solution

After solution has been found with ANSYS and with an analytic calculation, we can focus the attention on the volume integral code solution.

To face the problem we need the discretization of a sphere. This discretization has been obtained with an apposite MATLAB code already existing (and thus not developed in this work). Precisely, different polyhedral grids with increasing elements density levels have been produced, in order to study the behaviour of the solution with coarser

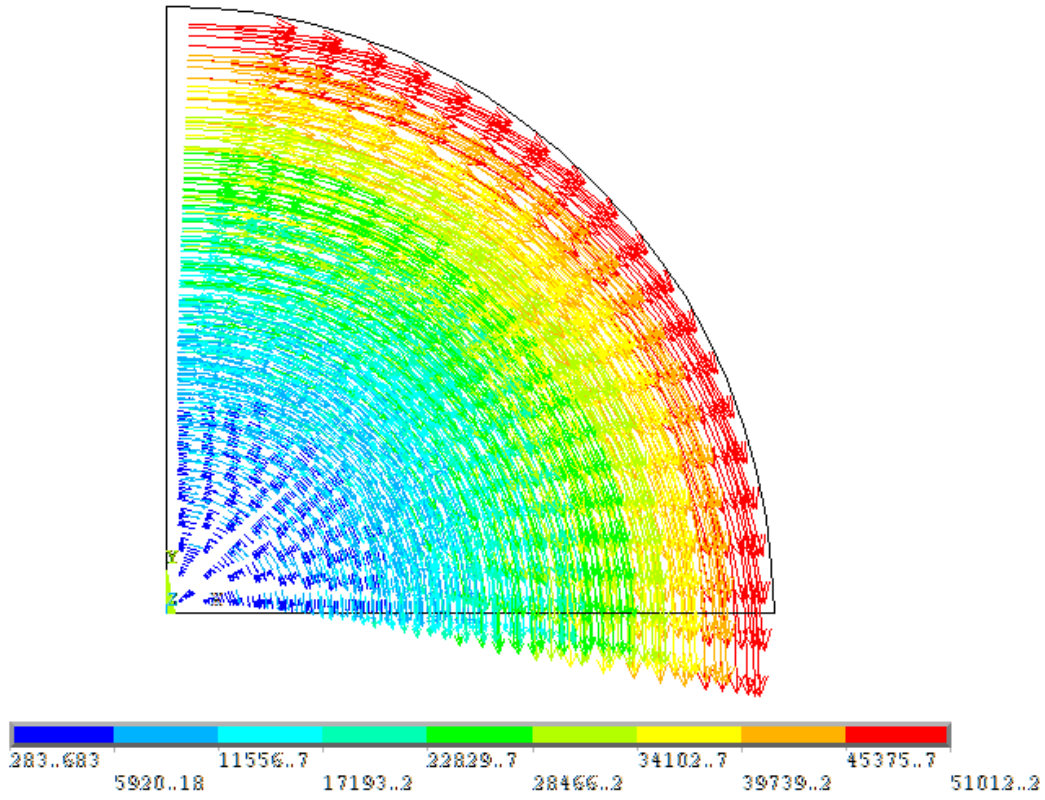


Figure 7.19: J_{im} in the sphere (top view)

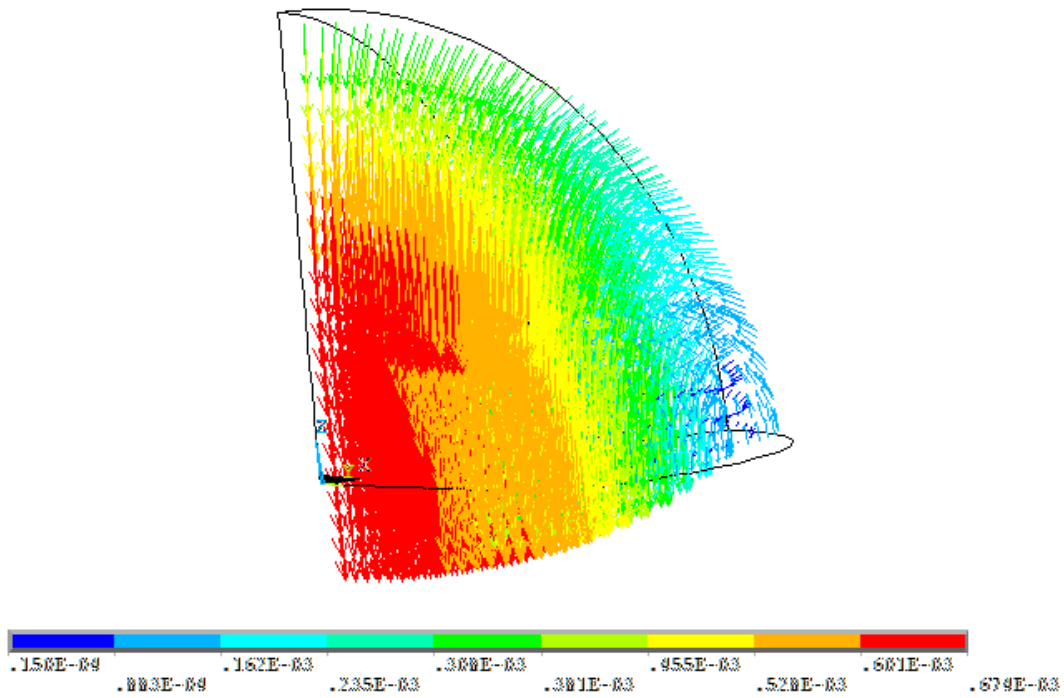


Figure 7.20: B_{im} in the sphere

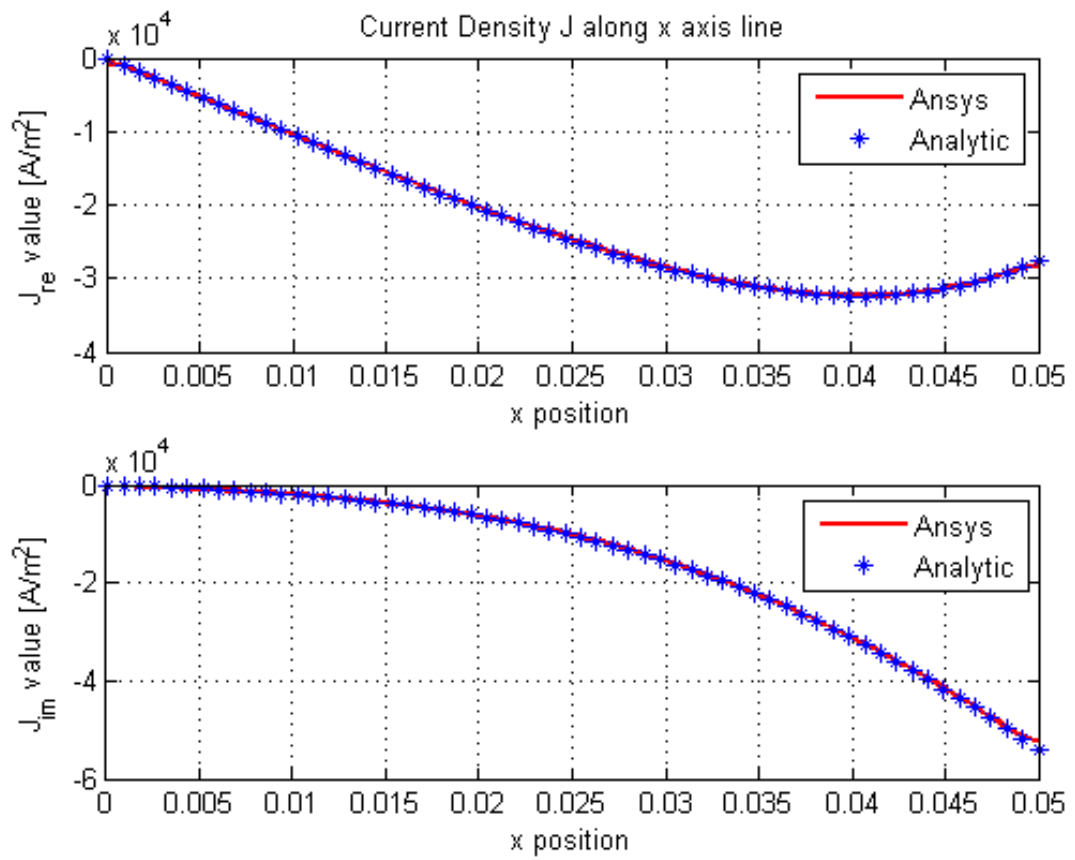


Figure 7.21: Comparison between analytical computation of current density J and ANSYS solution

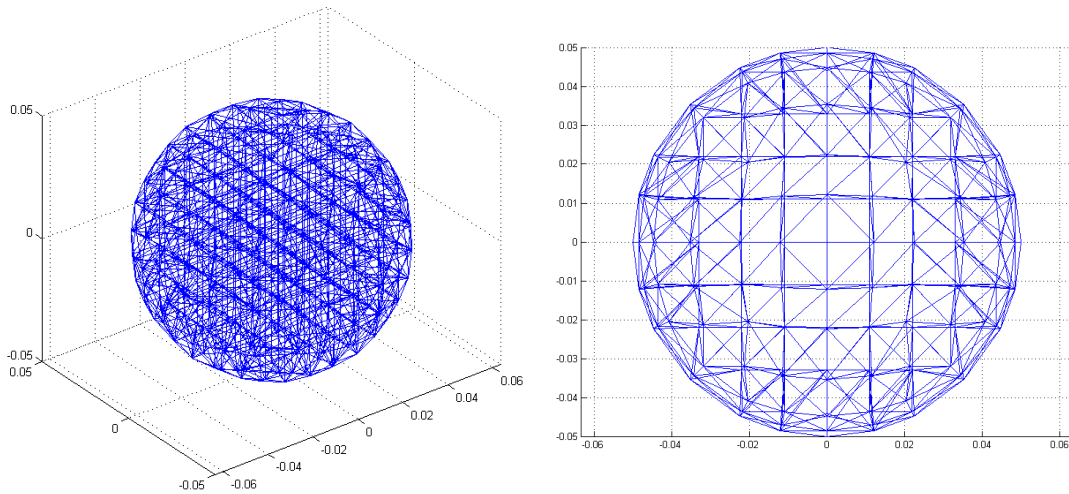


Figure 7.22: Coarse sphere grid: on the left an overall view, on the right a top view

and denser grids. In figure 7.22 and 7.23 coarse and dense mesh are respectively shown. The former is constituted by 456 volumes and 491 nodes while the latter has 1840 volumes and 1911 nodes; the unknowns are respectively 2010 and 7326. Moreover, in figure 7.24 some discretizing volumes (referred to coarser grid) are shown in detail: a tetrahedral volume, an hexahedral and a polyhedral one (which is composed by eleven faces).

Due to wide elements number of the grid a new issue has to be solved. The code that has been developed since now with MATLAB, takes about one hour to run a grid of one hundred of volumes. It follows that, since time consumption is not linear with volumes number but it is quadratic, it is not possible performing simulations with hundreds of elements. For this reason a FORTRAN version of the main functions (i.e. that functions which compute \mathbf{R} and \mathbf{M} matrices) has been realized and put into a *MEX file* so that this file can be run with MATLAB as a *Matlab EXecutable* function. This procedure has reduced computation time up to 5000 times so, by this way, grids of one hundred elements takes only few seconds to run with the help of the MEX file.

Results We hereafter report the volume integral simulation results in term of current density \mathbf{J} (both real and imaginary) and reaction \mathbf{B}_{im} field on a grid of point. In particular, figure 7.25 reports results related to current density \mathbf{J} computed on the coarsely discretized domain, while figure 7.26 reports the same parameter calculated on dense spherical grid.

In addition, in figures 7.27 and 7.28 induction imaginary field B_{im} on a grid of point above the sphere ($z_0 = 0.07 [m]$) and the same field on a 3D grid are respectively shown. Precisely, the second image is obtained interpolating the discrete values to obtain a sort of field lines map. These two last figures are obtained with post processing tools described in chapter 5.

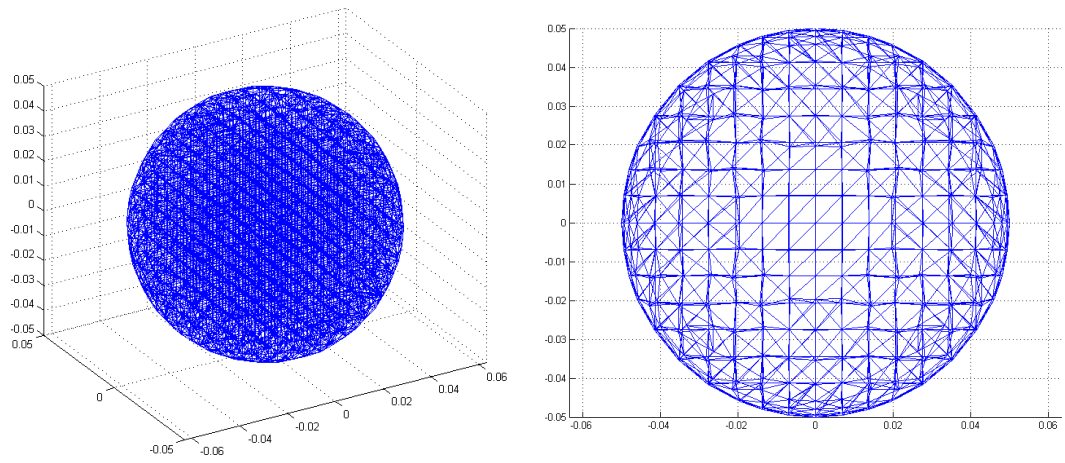


Figure 7.23: Dense sphere grid: on the left an overall view, on the right a top view

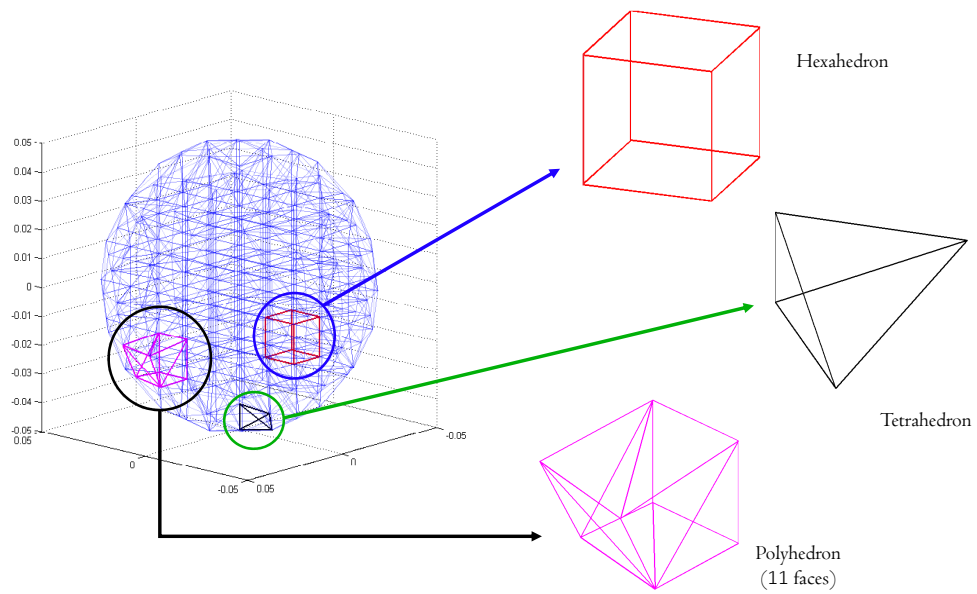


Figure 7.24: Polyhedral volumes details inside coarser spherical domain

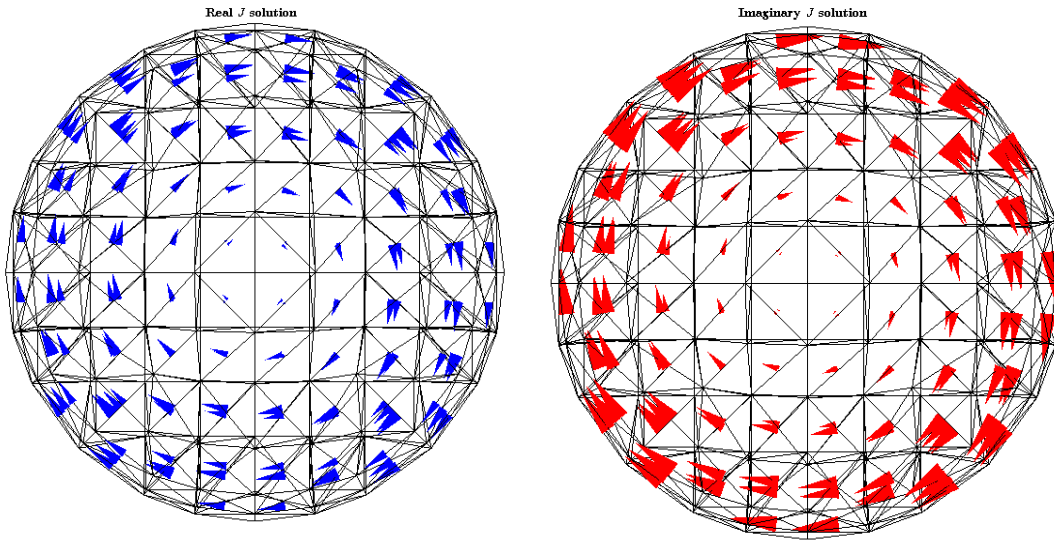


Figure 7.25: Current density J in the sphere with coarse grid: on the left J_{real} on the right $J_{imaginary}$

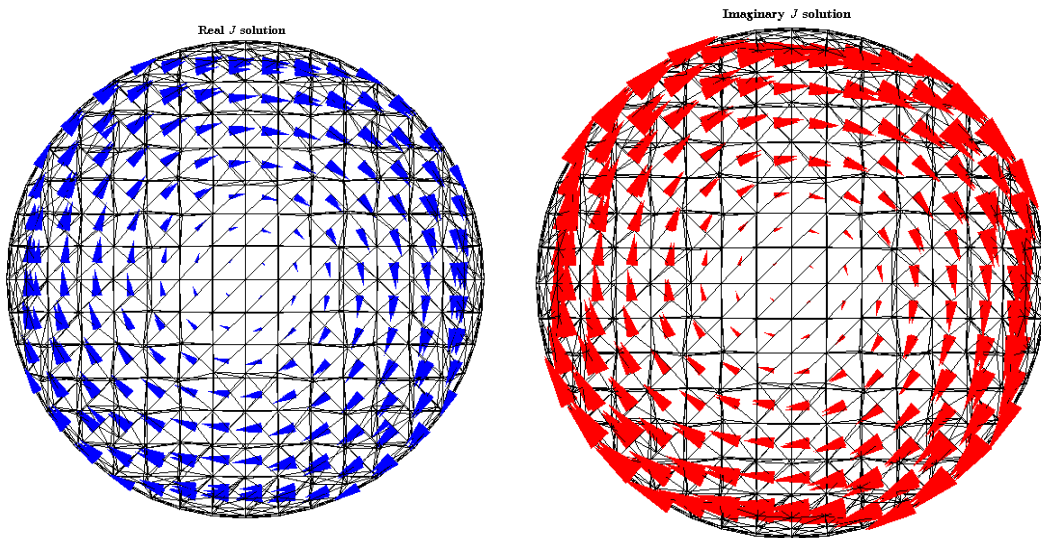


Figure 7.26: Current density J in the sphere with dense grid: on the left J_{real} on the right $J_{imaginary}$

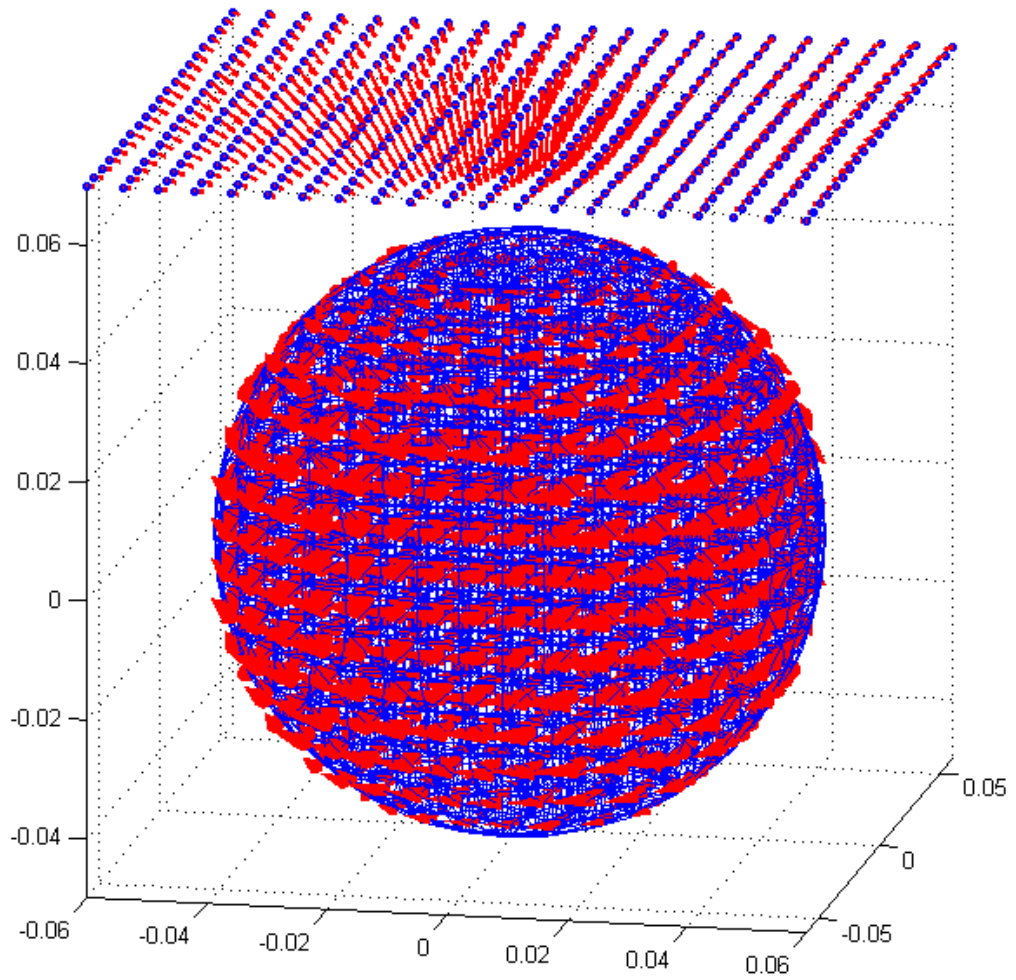


Figure 7.27: Induction imaginary field B_{im} on a 2D grid of point

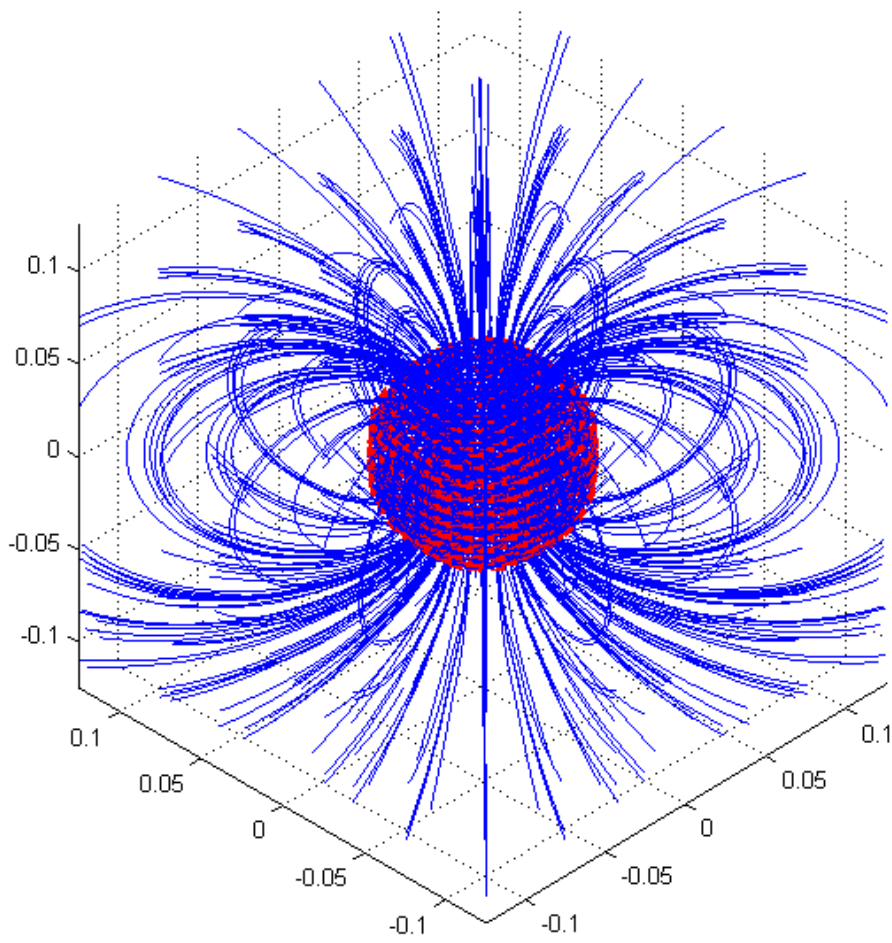


Figure 7.28: B_{im} field lines map

Results comparison: volume integral code VS analytical solution

Since an analytical solution of this problem is available, we will compare the obtained results above displayed with this exact comparison term.

Current density \mathbf{J} error

First of all we consider the solution in term of current density. Values of \mathbf{J} are computed by the volume integral code in each volume barycenter. Consequently we will compare the resulting current obtained with the volume integral code with the exact analytical solution provided by (7.18). Due to problem solution symmetry, clearly visible in figure 7.26, the comparison will be done on a diametrical plane parallel to x, y axis. To chose the plain height z_0 all the volumes barycenters of the mesh with $z \approx 0$ have been selected. Precisely for coarser grid this value is $z_0 = 0.0056 [m]$ while for the dense one is $z_0 = 0.0034 [m]$. Once the mesh volumes have been chosen, we have computed the current in each volume barycenter point with the analytical formula and then we have compared and plot the results (interpolating the obtained values)

The volume integral code error respect to the current density \mathbf{J} , which has been denoted as ϵ^J , has been computed as follows:

$$\begin{cases} \epsilon_{real}^J = |J_{re}^{v.i.} - J_{re}^*| \\ \epsilon_{imag}^J = |J_{imag}^{v.i.} - J_{imag}^*| \end{cases} \quad (7.26)$$

where the first expression is referred to real current component, while the second is referred to imaginary current component. Moreover, $J^{v.i.}$ is the **module** of current density (after either real or imaginary component has been chosen) computed with our volume integral code and J^* is the value of \mathbf{J} computed with exact analytical expression (and thus considered as reference).

In figure 7.29 and 7.30 a representation of the error respectively for real and imaginary component of \mathbf{J} computed on coarse grid for each point of xy plane (as previously explained) is reported. On the left a 3D representation on the right the consequent error map. Similarly in figures 7.31 and 7.32 the same parameters are shown but now related to dense grid.

Since the error reported is not a relative value but it is absolute, in figure 7.33 exact current density trend is reported to give a comparing parameter for absolute error plots.

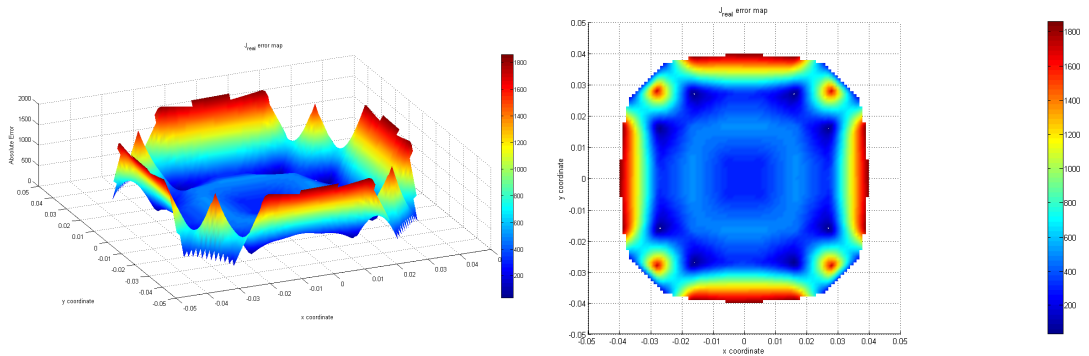


Figure 7.29: Current density error ϵ_{real}^J for real component on coarse grid

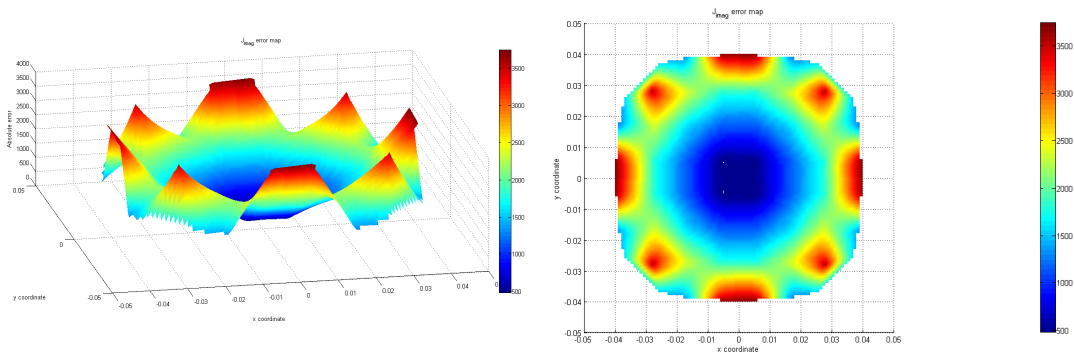


Figure 7.30: Current density error ε_{imag}^J for imaginary component on coarse grid

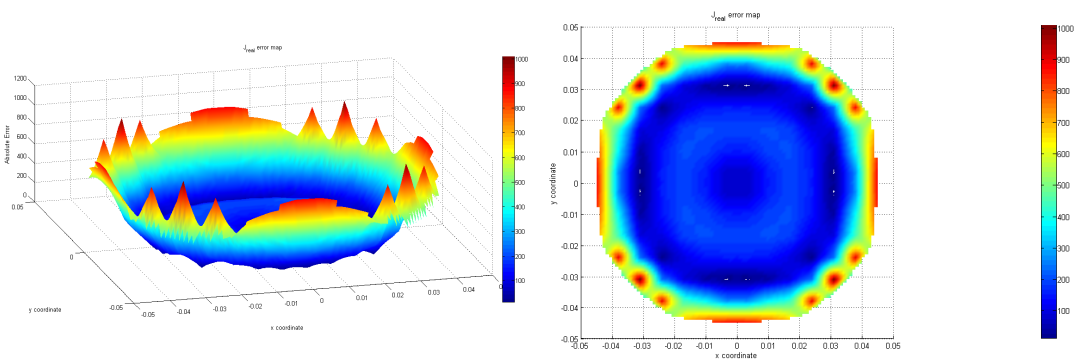


Figure 7.31: Current density error ε_{real}^J for real component on dense grid

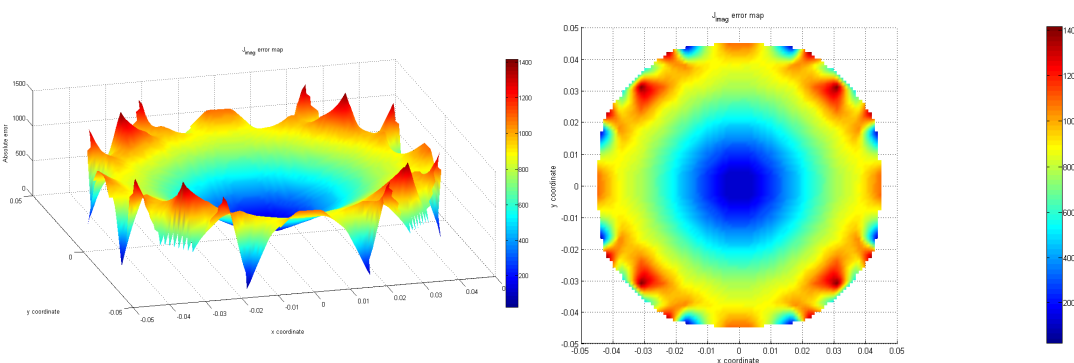


Figure 7.32: Current density error ε_{imag}^J for imaginary component on dense grid

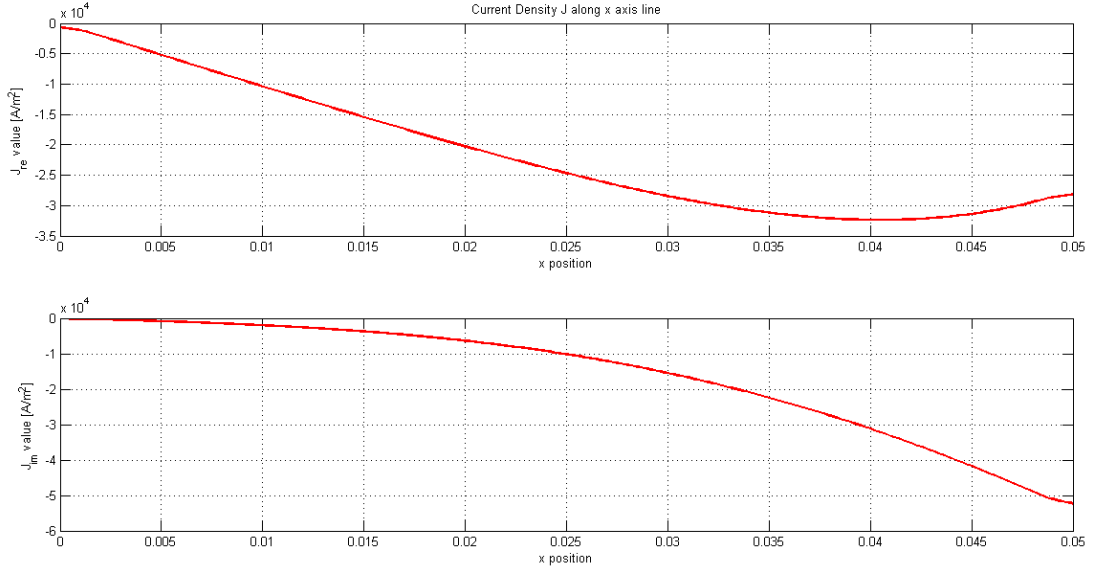


Figure 7.33: Analytical J field trend on a line

Comparing graphs referred to coarse grid and graphs related to dense discretizing grid, an error reduction due to a more accurate domain discretization is clearly visible. In fact, considering imaginary components the error is actually halved from more than $3500 [A/m^2]$ with coarse grid up to less than $1400 [A/m^2]$ with dense grid. This tendency is the same for real component too, but with a smaller discrepancy between the different grids solutions.

An explication of this described behaviour can be found in domain meshing issues. In fact, the relatively high error of imaginary current at the sphere boundary is surely caused by a scarce discretization of that sphere zone. Skin depth effect is responsible of a different distribution of current in section, consequently to well understand the phenomena is necessary having a grid as dense as possible in that zone. If the discretization is not fine enough an error occurs. This trend is a direct consequence of problem constitutive error, in fact whenever we discretize a volume sector with a single element we implicitly decide that in that element the field will be uniform. If actual field is not uniform enough, as in the case of sphere boundary where the current slope increase near to the borders, the solution will result to be error affected. Nevertheless, for the simulation performed on dense grid, if we parametrize the error at the border with respect to its absolute value we obtain an actually acceptable error of about $3 \div 4\%$.

Imaginary induction field B_{im} error

As already done testing the code with slab, in addition to current density comparison we also report the error related to the induction field generated by the current flowing in the sphere discretized with the dense grid. In particular we will report two test case:

1. we will compute the imaginary component of reaction B field on a line laying along x axis with $x \in [0; 0.2] [m]$ comparing results with exact analytical expression of (7.11);

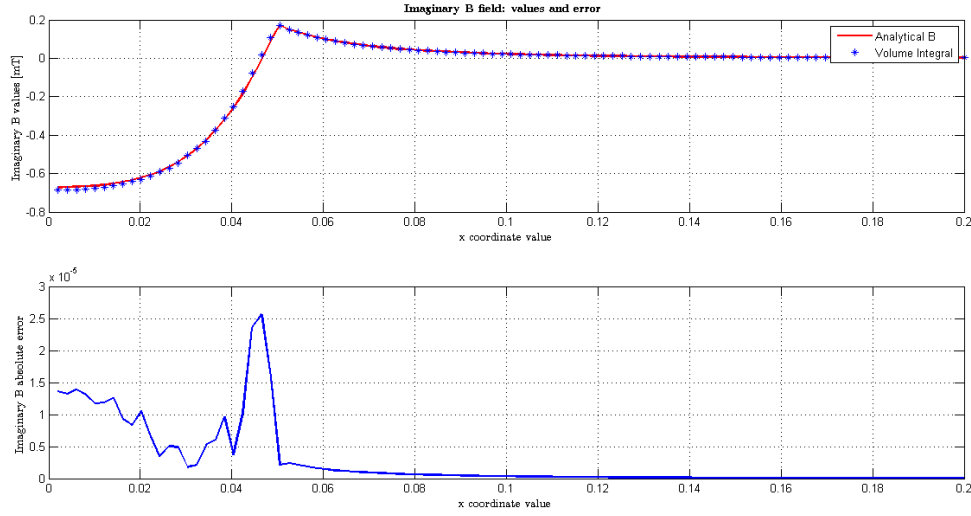


Figure 7.34: B_{im} field on a line: values comparison and related error

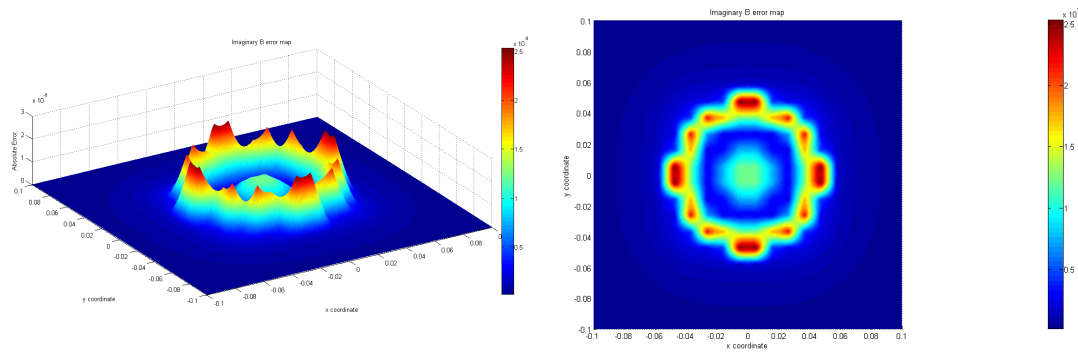


Figure 7.35: B_{im} error maps on a plane

- we will perform the same comparison but on a grid of point laying on a plane parallel to xy plane with $z_0 = 0 [m]$, and, as done for current density, we will then interpolate the errors to obtain an error map on that plane.

Related results are shown respectively in figures 7.34 and 7.35. Error has been computed as explained in equation (7.26) when in place of J , B field has been considered.

Comments With respect to the error referred to current density, B_{im} error is lower. In fact, if we consider figure 7.34 and we compute percentage error from its absolute values, on average the error will result to be between 1% and 3% except for a pick in correspondence of sphere border ($x \approx 0.05 [m]$). This behaviour is confirmed by the map in figure 7.35 where the error is approximatively null in each plane points except that near to circumference of radius $r_0 = 0.05 [m]$. Error decrease, respect to J comparison, is a consequence of the fact that on computing B_{im} we consider the effect of all the mesh volumes in a single space point; by this way each single volume error on J is neutralized by another opposite error in a different grid volume, thus the obtained results are better because they are not affected by local irregularities of the

problem solution. Another time, B_{im} error increase at sphere border is caused again by a too coarse discretization in that points where the field varies rapidly and it reach a pick before decreasing outside the sphere (see B_{im} tendency in figure 7.34 near to $x = 0.05 [m]$).

Consequently, thanks to all analysis performed, we can conclude that the volume integral code solution is as more precise as denser is the discretizing grid. More precisely, the grid has to be more and more fine in that domain points where field changes more rapidly from one point to another. As previously said, this feature agrees with the constitutive error introduced on writing problem equations which is caused by considering the field uniform in each volume of the discretized domain Ω_c .

7.4 Simulations on a real fusion device: Rfx-mod

Lastly performed simulations regard a real device actually existing in Padova at *Consiglio Nazionale delle Ricerche* currently used to study plasma stability and behaviour: RFX-mod. This device, already mentioned in the introduction, is constituted by a toroidal vacuum vessel surrounded by an highly conductive shell made of copper. Both these structures are enclosed into a stainless steel toroidal supporting structure. All these parts compose the so called “load assembly”. Overall, the load assembly presents several ancillary frameworks, each of them with a specific feature, which are integrated into this described device and which consequently contribute to make this machine a complex device with an intricate geometry, as shown in figure 7.36. It is thus clear the reason for which we will use this device as a reference to prove the efficiency of our implemented method when facing with complex problems.

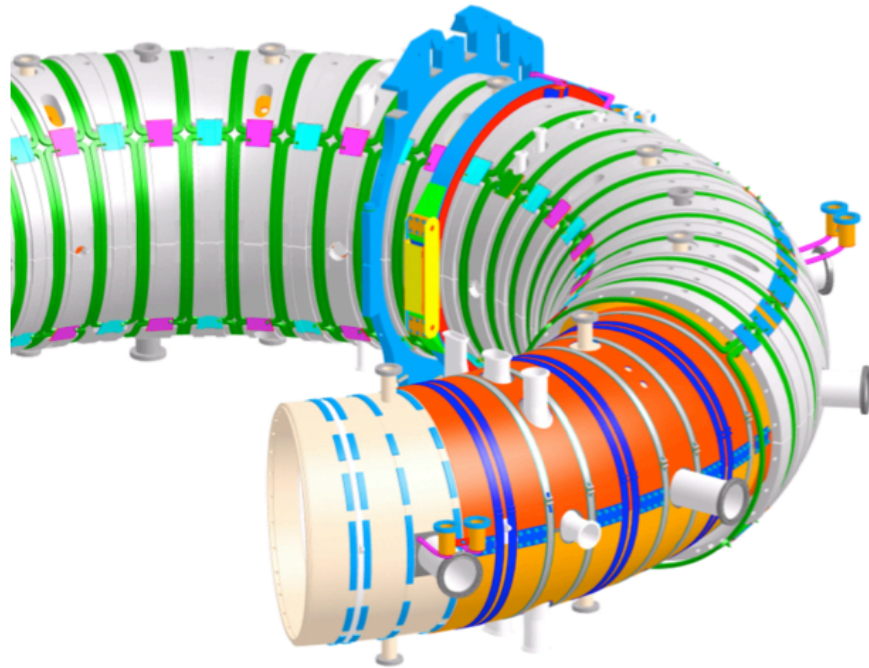


Figure 7.36: RFX-mod load assembly

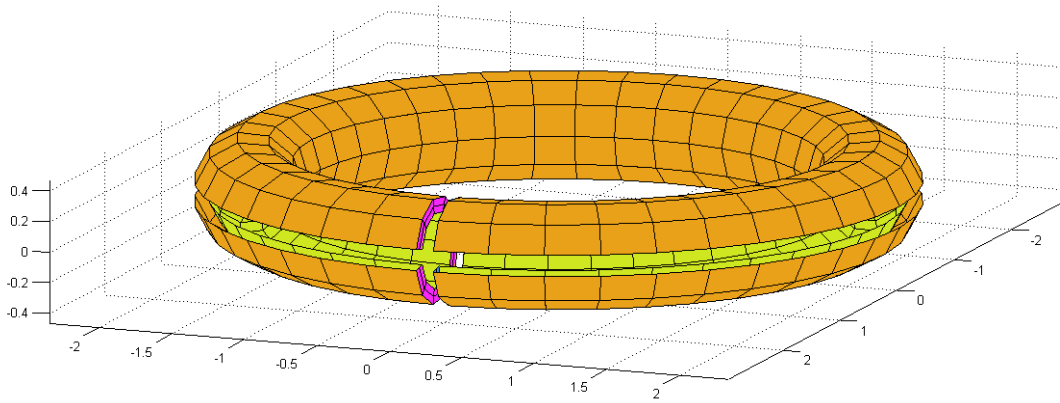


Figure 7.37: RFX-mod toroidal supporting structure (TSS) schematic

Problem setting: geometry

For these simulations we will focus only on a simplified version of the toroidal supporting structure (TSS), in order to study eddy currents generated by an externally applied magnetic induction field \mathbf{B} . This structure, as displayed in figure 7.37, presents two cuts: a toroidal cuts and a poloidal one. These two cuts, named “*gaps*” have the aim of allowing the penetration of the external fields (increasing the overall resistivity around the plasma inside the vessel) applied to control plasma Magneto Hydro Dynamic (**MHD**) stability. Moreover, the gaps reduce the eddy currents induced, onto TSS, by plasma fields variations.

Geometry specifications are reported in table 7.4 where R_0 denotes the major toroidal radius while a denotes the minor radius.

R_0 [m]	a [m]	Shell width [m]	Gaps width [m]
2	0.25	0.1	0.1

Table 7.4: TSS geometry specifications

Domain meshing

Before starting with simulations, a good domain discretization is needed to ensure fair results. The mesh, reported in figure 7.38, has been realized with a triple layer of hexahedral elements, with a subgridding nearby the gaps, where current density is forecast to concentrate. A detail of this subgridding is shown in figure 7.39.

This mesh is constituted by 14400 volumes delimited by 19844 nodes. The problem unknowns are 53923. Consequently, respects to previous test problems, this one is a really wide problem for which computational efficiency is fundamental. For this reason the MEX files, implemented for sphere test, have been further improved and problem parallelization has been applied to reduce computing time.

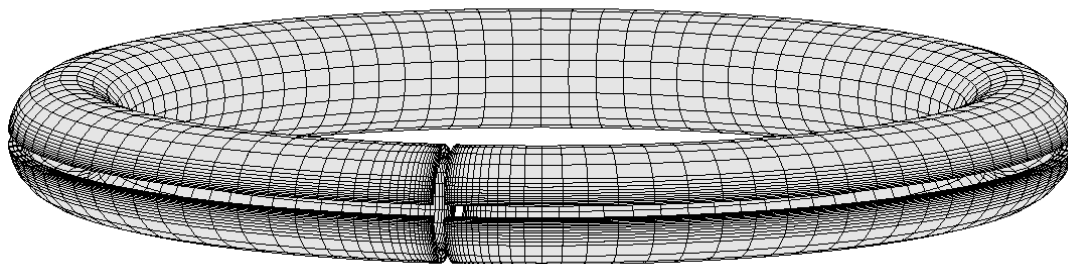


Figure 7.38: RFX-mod TSS mesh

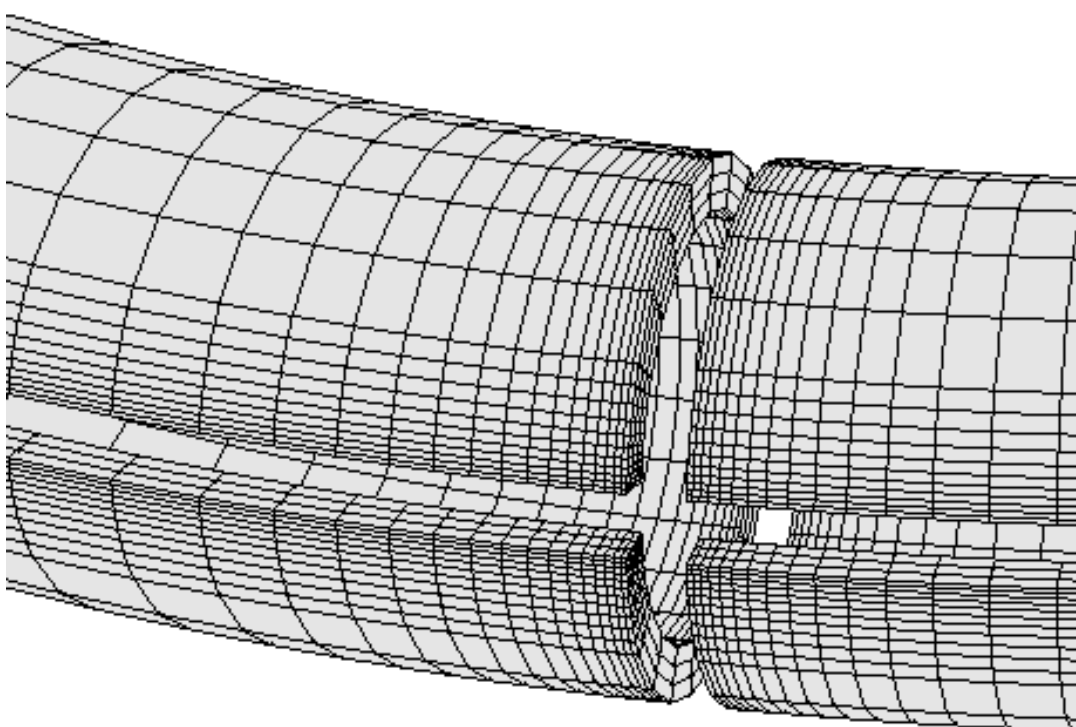


Figure 7.39: RFX-mod toroidal supporting structure: mesh detail nearby the cuts

7.4.1 Simulation A: external \mathbf{B} field vertically applied

This simulation was realized imposing an external induction field \mathbf{B} positively directed along z axis, whose module was chosen to be equal to $\hat{B}_0 = 1 [T]$. Material parameters values are $\rho = 1 \cdot 10^{-7} [\Omega \cdot m]$ and $\mu = \mu_0 = 4\pi \cdot 10^{-7} [H/m]$.

Results

Following results show the current density \mathbf{J} flowing in the TSS, current with its real and imaginary components distinct. To obtain these outcomes, the simulation takes 96 minutes to compute \mathbf{R} and \mathbf{M} matrices, and 11 minutes to assemble and solve the

equations. Results are reported in figure 7.40.

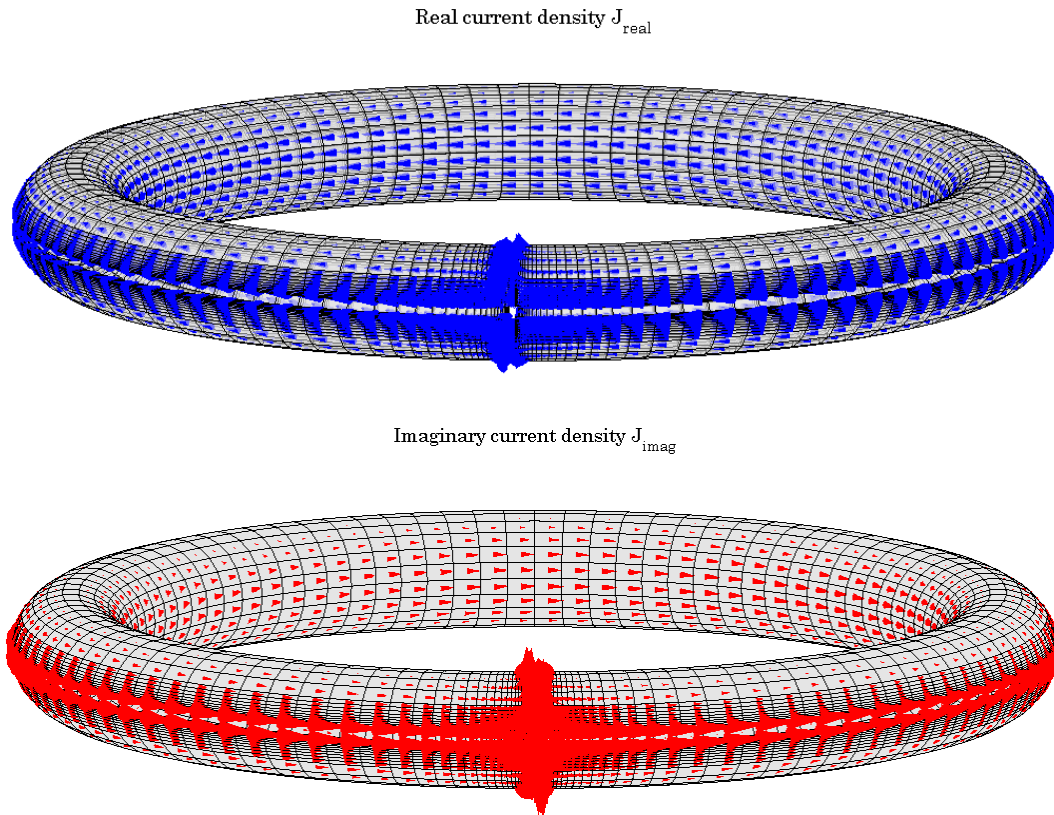


Figure 7.40: RFX-mod simulation results: current density \mathbf{J} in TSS with vertically applied field

From these figures we can see the behaviour of the structure under an induction field: as announced current density is higher nearby the gaps whose design has to be carefully carried out. Moreover, in figure 7.41, a detail of real current density, in correspondence of the cuts, is displayed: skin depth effect across TSS width is clearly visible.

7.4.2 Simulation B: load assembly and saddle coils

For this last problem, several local sources of magnetic induction field have been introduced: saddle coils. These 192 coils, reported in figure 7.42, aim at locally controlling plasma MHD instabilities thanks to a complex power supply system that can separately feed with current each coil in function of several fields measures, properly analysed and processed, coming from probes around plasma vessel. As shown in the figure, coils are organized into 48 coils groups surrounding the toroidal structure, each of them composed by 4 independent coils disposed along poloidal direction.

This configuration has been reported to give proof of a possible combination between a conducting domain Ω_c (the toroidal supporting structure) surrounded by local known sources of induction magnetic fields crossed by predetermined currents. Consequently, in this case, the known magnetic vector potential $\mathbf{A}_s(\mathbf{r})$ will be computed as described

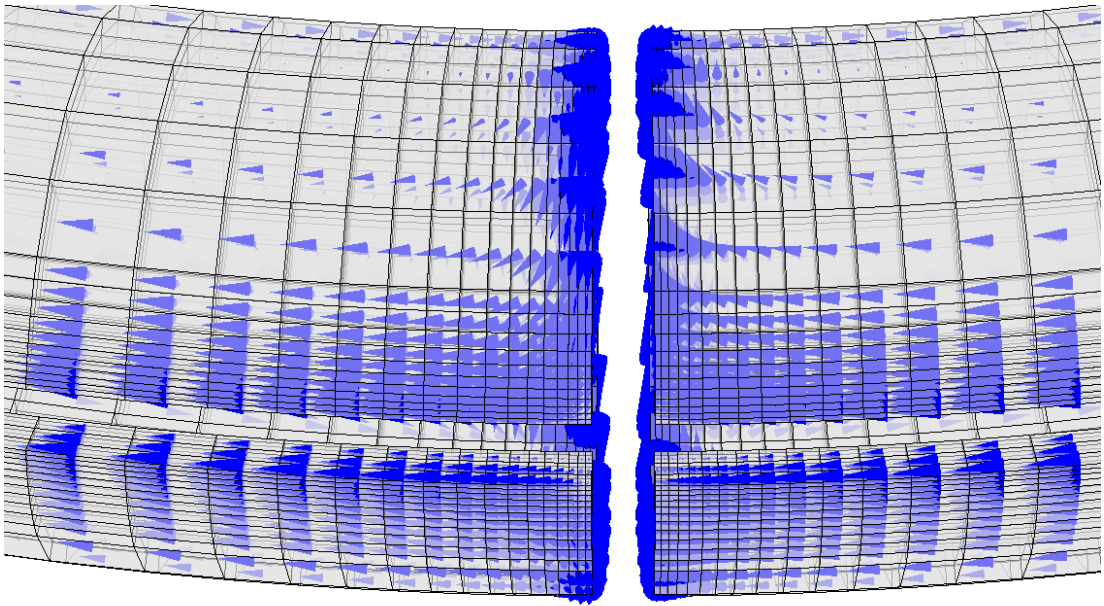


Figure 7.41: Real J : results detail

in chapter 4, to successively run the simulation in order to compute the reaction of TSS in terms of current density, under the saddle coils solicitations.

Results

In this last simulation to underline the effect of this particular situation, only one saddle coil has been fed with current. Consequently, eddy currents will be concentrated only nearby the powered coil. In figure 7.43 this powered coil (yellow coloured) and the related induced imaginary current density J_{im} in the TSS are visible. Moreover, in figure 7.44, two progressive enlargements nearby TSS cuts are shown. Current density is restricted to the powered coil as expected.

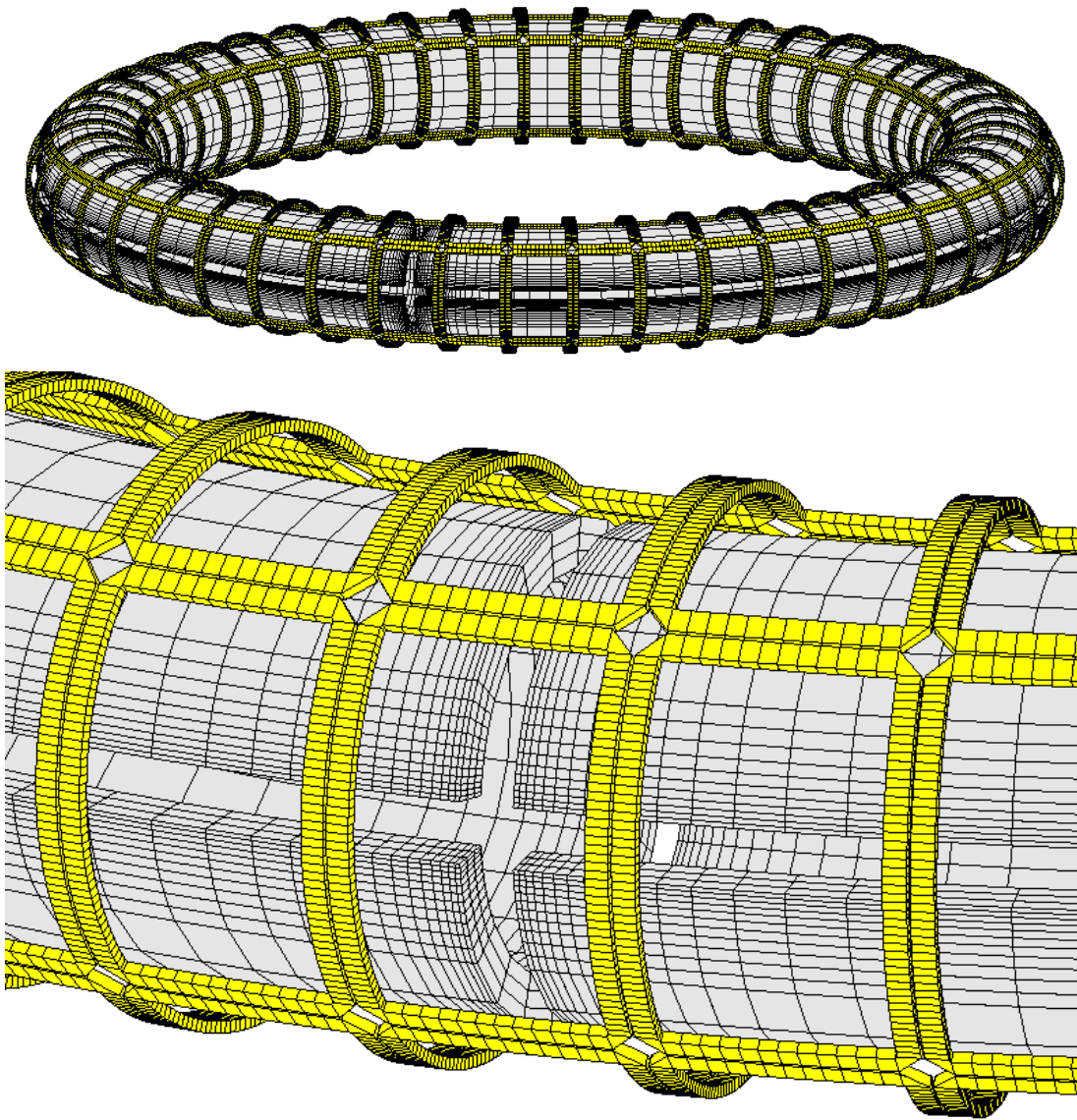


Figure 7.42: RFX-mod saddle coils: on the top the 192 coils, on the bottom a detail

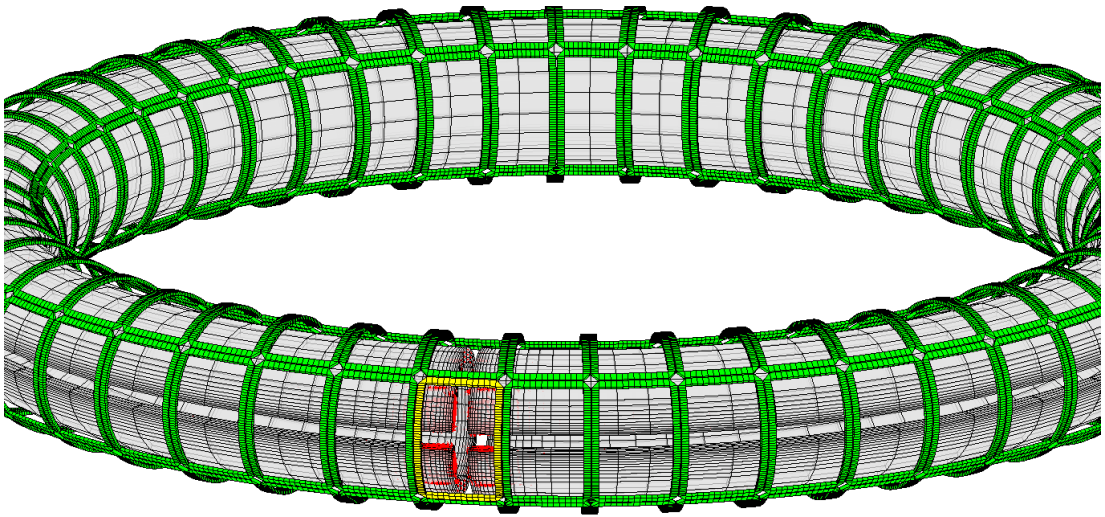


Figure 7.43: Saddle coils simulation: imaginary current density J_{im} results

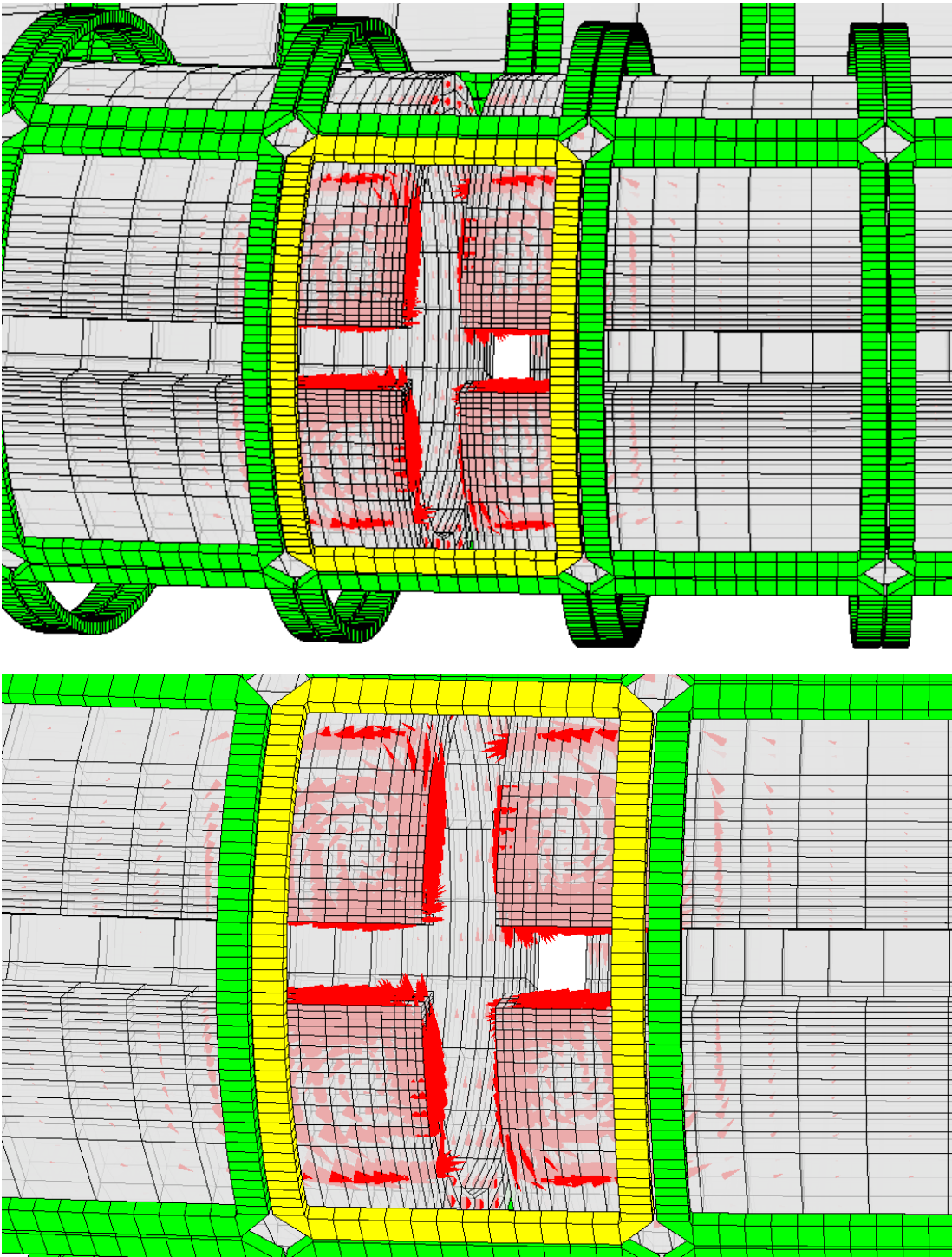


Figure 7.44: Saddle coils simulation: progressive enlargements in correspondence of fed coil

Chapter 8

Conclusions and future developments

In this thesis work we were able to develop a volume integral code for polyhedral meshes in every part composing it: solver, tools for the post processing phase and for the calculation of known magnetic induction sources.

Numerical simulations performed on a slab and on a sphere have proved the correct implementation of this volume integral method into the MATLAB codes since both the test problems have been correctly solved. As far as results accuracy is concerned, it is clear that the test on the slab has given more precise results than the test conducted on the sphere. This accuracy discrepancy has been obviously caused by the different complexity of this two involved domains which requires a different precision during the discretization. In fact, results on the slab have been precise despite the use of a coarse domain meshing, differently, for the sphere, we have registered higher errors in that domain points where the discretizing grid was too coarse with respect to fields variations. It follows that, in real applications, a careful domain discretization phase is necessary, otherwise only too approximate solutions may be accomplished. Nevertheless, despite these problems, a comparison between elements number used for FEM simulations and elements number involved in volume integral simulations has demonstrated that, even considering the use of more dense grids, the volume integral method allows to obtain accurate results with much less elements than a FEM code, maintaining its advantage respect to finite elements formulations.

Another issue observed during simulations was computation time consumption. Effectively, MATLAB version of this volume integral code was revealed not to be compatible with huge problems (i.e. with high elements number managing). In fact to perform simulations with more than one hundred elements, as required for sphere simulation, a hybrid FORTRAN-MATLAB version of the code has been conceived to avoid unacceptably long solving time.

Finally, the simulation on a real device was proficiently carried out, proving the code applicability on real problems with complex geometries.

From these considerations, some desirable developments are suggested. Improvements have to provide robust and general tools to generate refined meshes on arbitrarily shaped domains to enhance solution accuracy, and they have to reduce computation time thus improving code efficiency. In particular, from one hand it will be necessary to work on a complete code version in FORTRAN and integrate the code with new matrices sparsification techniques for time saving; on the other hand adaptive mesh refinement and subgridding techniques will have to be applied and improved to ensure solution accuracy.

Appendix A

Volume integral code: Matlab scripts

Hereafter the main MATLAB scripts which compose our volume integral code and that are mentioned in chapter 4 are displayed.

Particularly we will report:

- the main script `Main_eddyCurrent` containing all the codes represented in figure 4.1;
- R matrix assembly script `Rmatrix_Enh`;
- M matrix assembly script `Mmatrix_Enh`;
- the solver code `EddySolv_Enh`;
- the function that build the array of geometric entities `geoarr_Enh`.

Volume integral code main script “Main_eddyCurrent”

```
%% Main polyhedral volume integral simulator

clc
close all
clear all

%% VARIABLES TABLE
% P0: mesh nodes coordinates || size: tot n. of volumes
% P1: mesh edge baricenters || size: tot n. of edges
% P2: mesh face baricenters || size: tot. n. of faces
% P3: volume baricenters || size: n. of volumes
% L1: mesh edges vectors || size: n. of edges
% L2: face vectors || size: n. of faces
% L3: volume volumes || size: n. of volume

%% Geometry loading (P0,D,C,G + bcond)

% load 441_PB;
% load 881_PB;
load subl_modif_MP.mat;

%% Material parameters choices
```

```

m=1e-7; %resistivity value
disp(['Resistivity [Ohm m]: ',num2str(m)]);

mu=4.d-7*pi; %magn permeab
disp(['Magnetic Perm [H/m]:',num2str(mu)]);

% frequency choice
freq=50; %[Hz]
om=2*pi*freq; %[rad/s]
delta=sqrt(2*m/(mu*om)); %skin depth

disp(['Frequency [Hz]: ',num2str(freq)]);
disp(['skin depth [m]: ',num2str(delta)]);

%% R calculation
R_matrix_Enh;

%% M calculation

M_matrix_Enh; save('Msub_slab4x04MP.mat','M');

%% ALTERNATIVE: M loading

% load M881_Enh;
% load Msub_slab4x04MP;

%% A(r) Source Calculation and Problem solution

EddySolv_Enh;

save('Jsolution.mat','solJ');

R matrix assembly: Rmatrix_Enh

%% R_MATRIX CALCULATION

% Use of incident matrices D (vol,faces), C (faces,edges) and G (edges,nodes)

f_tot=size(C,1); %tot faces number
R=sparse(zeros(f_tot));

%volume-edge incidence matrix
VE=(double(D ~= 0) * double(C ~= 0)) ~= 0;
%volume-nodes inc. mtrx
VN=(double(VE ~= 0) * double(G ~= 0)) ~= 0;

for ivol=1:size(D,1);
[ p3 ] = volbar( P0,VN,ivol );
P3(ivol,:)=p3;
end

for ivol=1:size(D,1);
%% Plot geometry

%      geoplot;

%% Vector calc

```

```

[ p3 ] = volbar( P0,VN,ivol );
P3(ivol,:)=p3;

%function geoarr for face and dual edge calc
[ f,dual_e,vol,indG_f,p1,p2,l1,l3]=geoarr_Enh(P0,p3,C,D,G,ivol);

nf=numel(indG_f); %number of faces for a given volume

%dual edge bar are stored consecutively for each volume with the same face order
%reported in indG_f; for a given volume "ivol" there are "nf" dual edge
%bar, one for each face

%building edge(1) face(2) and volume(3) baricenters ("P") and vectors("L")

indx=find(VE(ivol,:));

P2(indG_f',:)=p2;

L2(indG_f',:)=f;
L3(ivol,:)=l3;
L1(indx',:)=l1(indx',:);
P1(indx',:)=p1(indx',:);

%% R calc

[ Rloc,aqw ] = Rloc_mtrx_En( ivol,f,dual_e,vol,m );

R(indG_f,indG_f)=R(indG_f,indG_f)+Rloc;

end

M matrix assembly: Mmatrix_Enh

%Global M-matrix calculation
%Script needs mesh geometry in a .mat file pre-loaded (here loaded in R_matrix.m)
%Script needs P3 vectors already built (built in R_matrix.m)

%Double cycle for each h and k mesh volume

M=zeros(size(D,2));
f_tot=size(M,1);

for ivolh=1:size(D,1)

disp(['*****STILL WORKING***** ivolh: ',num2str(ivolh)]);

[isr indG_fh val_fh]=find(D(ivolh,:));
[ fh,dual_eh,volh ] = localvect( ivolh,indG_fh,val_fh,L3,L2,P3,P2 );

for ivolk=1:size(D,1)

```

```

clear int1;
clear int4;

[iscr indG_fk val_fk]=find(D(ivolk,:));

[ fk,dual_ek,volk ] = localvect( ivolk,indG_fk,val_fk,L3,L2,P3,P2 );

if ivolh==ivolk

[ int4 ] = int4ordbis( P0,P2,P3,C,G,indG_fh,indG_fk,ivolh,ivolk);

[Mloc]=Mloc_mtrx_Enh(ivolh,ivolk,fh,fk,dual_ek,dual_ek,indG_fh,indG_fk,volh,volk,int4,mu);

else

[ int1,iop ] = int1ordbis( P0,P2,P3,C,G,indG_fh,indG_fk,ivolh,ivolk);

[Mloc]=Mloc_mtrx_Enh(ivolh,ivolk,fh,fk,dual_ek,dual_ek,indG_fh,indG_fk,volh,volk,int1,mu);

end
%local Mloc stored into global M matrix
M(indG_fh,indG_fk)=M(indG_fh,indG_fk)+Mloc;

end
end

```

Problem solver EddySolv_Enh

```

%% K matrix construction, for a given geometry

omega=2*pi*freq;

KR=C'*R*C;
KM=C'*M*C;

K=KR+1i*omega*KM;

%% b vector construction
% size(b)=edges x 1

% "bcond.mat" necessary for inner and outer edges identification

%Magnetic vector potential reconstr
B0=1;
ne=numel(bcond);

intAA=zeros(size(D,2),1);
for ivol=1:size(D,1)

[iscr indG_f val_f]=find(D(ivol,:));

[ f,dual_e,vol ] = localvect( ivol,indG_f,val_f,L3,L2,P3,P2 );

```

```

% magnetic vector potential calculation for uniform vertical B field
[ half_int ] = int_pot( P3,P2,ivol,indG_f,B0,dual_e );

intAA(indG_f',1)=intAA(indG_f',1)+half_int(indG_f',:);

end

% b calc values for all edges (then reduced only at edg_int!)
b_source=-li*omega.*(C'*intAA);

%% Boundary Conditions

% identifying inner/external edges-->"edg_int" || "edg_ext"
% if bcond(i,j)=1 then (i,j) edge is external, otherwise is internal (bcond=-1)

[edg_int]=find(bcond==-1); %edg_int IS A ROW!!!! (remember to transpose)

[edg_ext]=find(bcond==1);

% choosing only rows and columns from K referred to inner edges
K_red=K(edg_int',edg_int');

b_red=b_source(edg_int',:);

%% System solving

T_red=(K_red)\(b_red);

% full T reconstruction for each grid edge
T(edg_ext',:)=0;
T(edg_int',:)=T_red;

%% Flux vector calculation: flux = "I"

% size(I)=n. faces x 1, orderd with Global number, use D to obtain volumes fluxes

I=C*T;

%% Plot res and current density J calculation

plot_resEnh;

```

Geometric function geoarr_Enh

```

function [ f,dual_e,vol,indG_f,p1,p2,l1,l3]=geoarr_Enh(P0,p3,C,D,G,ivol)
% <<GeoArrow function>>: It calculates face, dual edge and volume vectors
% from geometry given incedent matrices C,D,G
% For more detailed comments about geometrical formulas
% see "hexaedral_test.m" script

% P0: node coordinates matrix
% indG_xx Global index referred to: f-ace;e-dge,node
% val_xx incidence matrices values (+/-1) referred as previus

```

```

%% Face Index calc

%iscr || global face ind || incidence face values
[iscr indG_f val_f]=find(D(ivol,:));

%% Geom calc

vol=zeros(3);

% questioning faces---->edges
for ii=1:numel(indG_f) %1:6

[iscr indG_e val_e]=find(C(indG_f(:,ii),:));

%questioning edges--->nodes
v=zeros(1,3);
v_face=zeros(1,3);

for kk=1:size(indG_e,2) %1:4

[iscr indG_nod val_nod]=find(G(indG_e(:,kk),:));

if val_nod(:,2)>val_nod(:,1)

v(2*kk-1,:)=P0(indG_nod(:,1),:); %ii-face vertices coordinates 1st node
v(2*kk,:)=P0(indG_nod(:,2),:); %ii-face vertices coordinates 2nd node
else
v(2*kk-1,:)=P0(indG_nod(:,2),:);
v(2*kk,:)=P0(indG_nod(:,1),:);
end
end

% Edge calculation

v_face=v;
alg_face_bar(ii,:)=sum(v_face)/size(v_face,1);

[area_t,bar_weighted,p1_loc,l1_loc,t]=facebar(indG_e,v_face,alg_face_bar,val_e,ii);

face_area=sum(area_t); %sum 4 triangles areas

p1(indG_e',:)=p1_loc(indG_e',:);
l1(indG_e',:)=l1_loc(indG_e',:);

%geometrical face baricenter
bar(ii,:)=(sum(bar_weighted)/face_area);

%face vectors calculation: then stored in l2--->L2
f(ii,:)=face_area.*(t(1,:)/norm(t(1,:)));

end

%volume calculation

```

```
for kk=1:numel(indG_f)
    %dual edge vectors
    dual_e(kk,:)=val_f(:,kk).*(bar(kk,:)-p3);
    vol_i=kron(f(kk,:),dual_e(kk,:));
    vol=vol+vol_i;

end

vol=vol(1,1);
l3=vol;
p2=bar;

end
```


Bibliography

- [1] L. Codecasa, R. Specogna, F. Trevisan, “*A geometric integral formulation for eddy-currents*”, International Journal for Numerical Methods in Engineering, John Wiley & Sons, pp. 1720-1736, Feb. 2010.
- [2] L. Codecasa, R. Specogna, F. Trevisan, “*A new set of basis functions for the discrete geometry approach*”, Journal of Computational Physics, pp. 7401-7410, 2010.
- [3] M. Fabbri, “*Magnetic flux density and vector potential of uniform polyhedral sources*”, IEEE Transaction On Magnetics, vol. 44, no. 1, pp. 32-36, 2008.
- [4] L. K. Urankar, “*Vector potential and magnetic field of current-carrying finite arc segment in analytical form. Part III: Exact computation for rectangular cross section*”, IEEE Transaction On Magnetics, vol. MAG-18, no. 6, pp. 1860-1867, Nov. 1982.
- [5] A. Capelluto, “*Magnetic calculation: pure numerical and semi-analytical procedures*”, Ph. D. Thesis, Università di Padova, Italy, Faculty of Electrical Engineering.
- [6] W. R. Smythe, “*Static and dynamic electricity*”, Toronto, Mc-Graw Hill, 1950, pp. 390-400.
- [7] G. Arfken, “*Mathematical methods for physicists*”, Orlando, Academic Press, 1985, pp. 102-104.
- [8] P. Bettini, L. Marrelli, R. Specogna, “*Calculation of 3-D magnetic fields produced by MHD active control systems in fusion devices*”, IEEE Transactions on Magnetics, vol. 50, no. 2, Feb. 2014.
- [9] P. Bettini, L. Grandò, G. Marchiori, “*Feasibility study of a local active correction system of magnetic field errors in RFX-mod*”, Fusion Engineering and Design, 2015.
- [10] J. D. Hanson, S. P. Hirshman, “*Compact expressions for the Biot-Savart fields of a filamentary segment*”, Physics of Plasmas, 2002.
- [11] J. Freidberg, “*Plasma physics and fusion energy*”, New York, Cambridge University Press, 2007.
- [12] E. Tonti, “*The mathematical structure of classical and relativistic physics*”, Basel, Switzerland: Birkhäuser, 2013.

- [13] P. Alotto, P. Bettini, R. Specogna, “*Sparsification of BEM matrices for large-scale eddy current problems*”, IEEE Transactions on Magnetics, vol. 52, 2015.
- [14] L. Codecasa, R. Specogna, F. Trevisan, “*Subgridding to solving magnetostatics within discrete geometric approach*”, IEEE Transactions on Magnetics, vol. 45, no. 3, March 2009.
- [15] M. Guarnieri, G. Malesani, “*Elettromagnetismo stazionario e quasi-stazionario: Elementi di elettrotecnica*”, Padova, Edizioni Progetto, 1999.
- [16] M. Abramowitz, I. A. Stegun, “*Handbook of mathematical functions with formulas, graphs and mathematical tables*”, Washington, National Bureau of Standards, 1964.
- [17] G. De Marco, “*Analisi due. Teoria ed esercizi*”, Bologna, Zanichelli, 1999.
- [18] Y. Jinyun, “*Symmetric Gaussian quadrature formulae for tetrahedral regions*”, Computer Methods in Applied Mechanics and Engineering, vol.43, pp. 349-353, 1986.
- [19] P. Keast, “*Moderate-degree tetrahedral quadrature formulas*”, Computer Methods in Applied Mechanics and Engineering, vol. 55, pp. 339-348, 1986.
- [20] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, “*The Finite Element Method: Its Basis And Fundamentals*”, Oxford, Butterworth-Heinemann, 2005.
- [21] P. Dular, “*Modélisation du champ magnétique et des courants induits dans des systèmes tridimensionnels non linéaires*”, Ph.D. Thesis 152, University of Liège, Belgium, Faculty of Applied Sciences, Nov. 1994.
- [22] The Economist Online, “*Stellar work*”, <http://www.economist.com/news/science-and-technology/21676752-research-fusion-has-gone-down-blind-alley-means-escape-may-now-be>, Oct. 2015.
- [23] “*Matlab software*”, <http://it.mathworks.com/>
- [24] “*Ansys software*”, <http://www.ansys.com/>
- [25] “*ITER, The way to new energy*”, <https://www.iter.org/>