



HAL
open science

Unstructured PEEC formulations considering resistive, inductive and capacitive effects for power electronics

Jonathan Siau

► To cite this version:

Jonathan Siau. Unstructured PEEC formulations considering resistive, inductive and capacitive effects for power electronics. Electric power. Université Grenoble Alpes, 2016. English. <NNT : >. <tel-01567899>

HAL Id: tel-01567899

<https://hal.science/tel-01567899v1>

Submitted on 24 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Génie Électrique**

Arrêté ministériel : 7 août 2006

Présentée par

Jonathan SIAU

Thèse dirigée par **Olivier CHADEBEC** et
codirigée par **Gérard MEUNIER** et
co-encadrée par **Jean-Michel GUICHON** et **Rémy PERRIN-BIT**

préparée au sein du **Laboratoire de Génie Électrique de
Grenoble**
dans l'**École Doctorale Électronique, Électrotechnique,
Automatique & Traitement du Signal**

Unstructured PEEC formulations considering resistive, inductive and capacitive effects for power electronics

Thèse soutenue publiquement le **15 décembre 2016**,
devant le jury composé de :

M. Marc BONNET

Directeur de recherche CNRS au POems à Palaiseau (Président)

M. Patrick DULAR

Directeur de recherche, F.R.S.-FNSRS à Liège (Rapporteur)

M. Lionel PICHON

Directeur de recherche CNRS au GeePs à Paris (Rapporteur)

M. Xavier BUNLON

Docteur-Ingénieur au Technocentre Renault à Guyancourt (Examineur)

M. Enrico VIALARDI

Docteur-Ingénieur à Altair Engineering France à Meylan (Examineur)

M. Olivier CHADEBEC

Directeur de recherche CNRS au G2Elab à Grenoble (Examineur)

M. Jean-Michel GUICHON

Maître de conférence au G2Elab à Grenoble (Examineur)

M. Gérard MEUNIER

Directeur de recherche CNRS au G2Elab à Grenoble (Examineur)

M. Ronan PERRUSSEL

Chargé de recherche CNRS au Laplace à Toulouse (Invité)



To my family and friends, true love does not suffer of distance or time.
I love you.

Acknowledgements

La reconnaissance est la mémoire du cœur.

Hans Christian Andersen

Je remercie tout d'abord les membres du jury pour leurs temps et l'intérêt qu'ils ont portés pour cette thèse :

M^r Marc BONNET, Directeur de recherche CNRS au laboratoire POems à Palaiseau, merci d'avoir présidé ce jury de thèse. Étant du domaine des Mathématiques appliquées, votre analyse, vos questions et commentaires ont apportés un point de vue fort intéressant.

M^r Patrick DULAR, Directeur de recherche F.R.S.-FNSRS à l'Institut de Montefiore à Liège and M^r Lionel PICHON, Directeur de recherche CNRS au laboratoire GeePs à Paris, je tiens également à vous remercier d'avoir accepté les postes de rapporteurs dans ce jury. Les questionnements et remarques posées lors de la soutenance étaient très intéressantes et pertinentes.

M^r Xavier BUNLON, Docteur-Ingénieur au Technocentre Renault à Guyancourt, un grand merci d'avoir participé à ce jury de thèse. Les commentaires que vous avez apportés, depuis un point de vue d'industriel, nous rassure sur la pertinence et capacité de ses travaux à répondre aux besoins des industriels.

Je tiens aussi à remercier très chaleureusement M^r Ronan PERRUSSEL, Chargé de recherche CNRS au laboratoire Laplace à Toulouse, étant dans mon jury en tant qu'invité. C'est en très grande partie grâce à toi si j'ai pu obtenir ce titre de docteur. Vous m'avez acceptés, avec Jean-René POIRIER, en tant que *jeune* stagiaire de M2 et fourni un cadre très agréable et propice à mon insertion dans le monde de la recherche. Donc un très grand merci à toi.

Cette thèse étant une collaboration entre le G2Elab et Altair (anciennement CE-DRAT), je voudrais remercier ensuite mon encadrement de thèse du côté académique et industriel, partiellement présent dans le jury.

En commençant par M^r Olivier CHADEBEC, Directeur de recherche CNRS au laboratoire G2Elab à Grenoble, le directeur de cette thèse. Il est dur de trouver les mots pour exprimer ma sincère gratitude, tu m'as donné ma chance en me proposant ce sujet de thèse et en m'acceptant en tant que doctorant. Étant un chercheur très occupé, tu as toujours pris le temps de m'aider et me guider lorsque j'en avais besoin pendant ces trois courtes années. Je tiens à ce que tu saches que je t'en suis éternellement reconnaissant. Merci.

Ensuite, je tiens à remercier très chaleureusement M^r Gérard MEUNIER, Directeur de recherche CNRS au laboratoire G2Elab à Grenoble, co-directeur de cette thèse. Un homme également avec d'innombrables responsabilités et toujours disponible pour ses doctorants. J'aimerais mettre en évidence l'importance que tu as dans ces travaux de thèse, puisque que tu es l'auteur original des formulations PEEC proposées durant cette thèse. Je te suis très reconnaissant de m'avoir permis de les étudier, implémenter et de les publier. Merci pour ta patience et pour le temps que tu as pris pour me transmettre une partie de ton savoir-faire indéniable sur les formulations. J'ai réellement apprécié les discussions de fins de journées sur les formulations, leur signification physique et philosophique. Infiniment merci à toi.

Le dernier encadrant académique: M^r Jean-Michel GUICHON, Maître de conférences au laboratoire G2Elab à Grenoble, je te remercie pour tout le temps que tu m'as consacré durant ces trois dernières années. À chaque fois que j'étais dos au mur, ta compréhension des outils numériques et tes connaissances en électromagnétisme nous ont permis de toujours avancer dans le bon sens et de tirer les bonnes conclusions.

Ensuite j'aimerais remercier mes encadrants d'Altair : M^r Rémy PERRIN-BIT et M^r Guy JÉRÔME. Merci pour votre accueil, votre sympathie et bienveillance. Vous fournissez un cadre de travail tellement agréable que cela encourage à la fois le développement de soi et numérique. De plus, je vous suis sincèrement reconnaissant pour la chance que vous m'offrez en me proposant de continuer l'aventure au sein d'Altair. Je sais désormais que mes travaux de recherches seront bien valorisés et ne tomberont pas dans l'oubli.

Et toujours du côté *Altairien*, je tiens à remercier chaleureusement M^r Enrico VIALARDI qui a accepté de faire partie du jury et d'y représenter Altair. Merci à toi, tu m'as tellement aidé dans les phases de la rédaction que je ne maîtrisais pas. Merci aussi pour le temps que tu as pris pour m'expliquer les aspects techniques, les besoins et le potentiel de ces travaux.

J'aimerais ensuite remercier ma famille qui vit presque entièrement à Tahiti, de

Acknowledgements

l'autre côté du globe. Il est dur de faire plus éloigné, mais comme la dédicace le dit, nous sommes peut être très éloignés, mais sachez que je vous aime. Malgré tout ce qui s'est passé depuis mon départ, vous m'avez toujours encouragé et soutenu.

À mon père Roger SIAU, qui n'a pas pu faire le déplacement pour soutenance, dû au décalage horaire et aux contraintes de chacun nous n'avons pas beaucoup l'occasion de nous *Skyp*, mais sache que je te remercie pour ton soutien sans failles et je t'aime de tout mon cœur.

À ma mère Michèle VIVISH, merci pour les agréables vacances que l'on a passé durant ta venue. Merci d'être venue pour me soutenir à ma soutenance, pour l'aide apportée à la mise en place du pot et plus que tout, merci d'être ma mère. Je t'aime maman.

Mes pensées vont également à mes petits frères, mes sœurs et au reste de ma famille pour leur soutien ♡.

Je n'oublie pas les incroyables amis dont j'ai croisé le chemin, la communauté de Brésilien(ne)s que je porte dans mon cœur : Lucas, Guilherme, Mateus, Sergio, Tamiris, Thiago, Poliana, Diego, Douglas, Angelica, Tiago, Fran, Rodrigo, Dyenny et les oubliés, malgré la distance et la séparation de nos chemins, je ne vous oublierai pas. Les amis "non brésiliens" également : Vincent, Limin, Lyes, Bertrand, Menad, Olivier, Quang, Trung, Vinh et les autres. Les Altairiens : Pauline, Anthony, Thomas et Quentin. Je suis sincèrement heureux de vous connaître et j'espère de tout cœur que nous pourrions passer de nouveaux bons moments ensemble.

Et je vais finir par remercier la personne la plus importante à mes yeux, celle qui partage mes jours et mes nuits, mon âme-sœur, ma *vahine*, ma chérie, Savana SINTES. Merci de m'avoir soutenu pendant toutes ces années, dans les meilleurs et les pires moments. Je t'aime.

CONTENTS

GENERAL INTRODUCTION	1
I \mathcal{H} -MATRICES	11
I.1 INTRODUCTION	13
I.1.a Compression Context	13
I.1.b Solution Context	15
I.2 CLUSTERING TECHNIQUE	16
I.2.a Cluster Tree	16
I.2.b Block Cluster Tree and Admissibility	20
I.3 ADAPTIVE CROSS APPROXIMATION (ACA)	23
I.3.a Cross Approximation	24
I.3.b Adaptive Cross Approximation	26
I.4 HYBRID CROSS APPROXIMATION (HCA)	27
I.4.a Analytical approach	27
I.4.b Hybrid Cross Approximation (II)	28
I.5 \mathcal{H} -MATRIX ARITHMETIC'S	31
I.5.a \mathcal{H} -matrix-vector product	31
I.5.b Agglomeration	32
I.5.c \mathcal{H} -LU decomposition, an efficient preconditioner	34
I.6 NUMERICAL RESULTS	38
I.6.a Electrostatic example	38
I.6.b Magnetostatic formulation	40
I.7 CONCLUSION	44
II ON THE PRECONDITIONING OF EDDY CURRENT FORMULATIONS	47
II.1 INTRODUCTION	49
II.2 INDUCTIVE PEEC FORMULATION	50
II.2.a PEEC Formulation	51

II.2.b Shell PEEC-formulation	59
II.2.c Preconditioning study	62
II.3 T FORMULATION	68
II.3.a Integral formulation	69
II.3.b Circuit approach	71
II.3.c Electrical circuit coupling	73
II.3.d Preconditioning	74
II.4 CONCLUSION	77
III ELECTROMAGNETIC MODELS CONSIDERING CAPACITIVE EF-	
FECTS	79
III.1 INTRODUCTION	81
III.2 CAPACITIVE PEEC FORMULATION WITHOUT DIELECTRICS	82
III.2.a PEEC Recalls	82
III.2.b Adding capacitive branches	83
III.2.c Formulation	84
III.2.d Example	85
III.3 MODEL FOR CONDUCTORS AND DIELECTRICS	86
III.3.a Maxwell's equations	87
III.3.b Discretization	88
III.3.c Practical Formulation	91
III.3.d Validation	94
III.3.d-i Case 1: Micro-coil	94
III.3.d-ii Case 2: Real device	96
III.3.d-iii Case 3: Volume vs. Thin shell formulation	98
III.4 PEEC-BIM FORMULATION	100
III.4.a BIM coupling	100
III.4.b Conductive-capacitive PEEC formulation	102
III.4.c Coupled BIM-PEEC formulation	103
III.4.d Validation	104
III.5 CONCLUSION	106
CONCLUSIONS AND OUTLOOKS	
	109
A FEATURE SELECTIVE VALIDATION OF EXAMPLES	
	113
A.1 CASE 1: MICRO-COIL	113
A.2 CASE 2: REAL DEVICE	113

CONTENTS

BIBLIOGRAPHY	117
PUBLICATIONS	127

General Introduction

Context

The electrical systems take a significant place in many industries now that the technologies provide a great efficiency for a lower cost. For example, the hybridization of cars requires the use of power converters from AC to DC and vice versa. The DC part is used to supply multiple on-board devices or to store the energy, whereas the electric motor supplies work in AC as well as the regenerative braking systems. The same goes for many applications, where the former hydraulic or thermal motors are replaced by electrical motors, as actuators.

The new technologies also allow us to product more compact devices, lowering the volume occupation. In return of the minimization and the increase of the operating frequency, the electrical circuits are more likely to interfere internally or with other nearby devices.

Engineers are then required to study these electromagnetic interferences (EMI), in order to ensure the electrical devices are complying with the constraints imposed by the standards on the electromagnetic compatibility (EMC). The standards usually depend on the country, the application or the business sector.

Electromagnetic Compatibility

The EMC phenomena can be divided in two categories: internal and external. The internal events occurring in a device are mainly related to the loss of signal integrity, due to parasitic coupling between the circuits (or the conductors).

The external effects, for which standards apply, are usually subdivided in:

- immunity, which is the ability of a system to resist the impact of the electromagnetic environment. The interferences coming from the environment can be natural, caused by atmospheric phenomena (as lightning) or extraterrestrial (as cosmic radiations), or they can be artificial. The artificial (or man-made)

interferences mostly come from the telecommunications, but, there are more impactful interferences caused by the power distribution, the static converters or other supplies.

- emissions, which are the interferences generated by the system and which disturb the surrounding areas.

Ensuring the electromagnetic compatibility of the system means for engineers increasing the immunity, while limiting the emissions. But, EMC studies are very complex since the interferences can appear through various couplings, as far field plane waves, capacitive or inductive crosstalks, or common-impedance ground loops. Moreover, EMI can occur on a very wide band of frequencies; in fact, for both immunity and emission phenomena. Typically, two bands of frequencies can be differentiated: the conducted, from 10 kHz to 30 MHz and the radiated from 30 MHz to 6 GHz. The interferences by conductions (suffered or generated) affect the devices via parasitic currents flowing through conductors or parasitic mutual couplings. The radiative interferences are EM waves represented by the fields (\mathbf{H} , \mathbf{E}) in free space.

Due to high levels of current flowing inside them, power electronics devices (the applications aimed by these works) need to be mostly studied from the emission point-of-view.

Power Electronics devices

The power electronics devices are very often mechanically supported by printed circuit boards, as in Figure 1, leading to multi-layers structures separated by dielectric blocks. These circuits usually have a length and a width of 15 to 30 cm, but their thickness can vary from a few millimeters to some centimeters. The printed conductors have commonly a thickness of 35 or 70 μm , but their other dimensions greatly vary. For example, the power/ground planes can be very long and wide, whereas the tracks are long with a fixed width and follow complex paths, due to the routing constraints. The geometric complexity of these devices is also increased by the presence of vias, i.e.: conductive hollow cylinders linking tracks and planes positioned on different layers.

There is a rule stating the wave propagation effects can be neglected while the greater dimension of the problem d_{\max} is significantly lower than the wavelength λ , a heuristic consists in ensuring $d_{\max} \leq \lambda/10$. As mentioned above, the power



Figure 1: Variable speed drive (Schneider-Electric).

electronics devices typically have a characteristic dimension between 15 to 30 cm, which leads to a frequency of about 100 to 200 MHz as limit threshold for quasi-static models (without propagation). However, the studies of these devices require crossing last limits since the standards consider frequencies up to few gigahertz, i.e.: about one decade above. The numerical methods then need to be able to cross a bit these limits.

On the other hand, the current density in the conductors is not uniformly flowing across a section: it is concentrated on the border due to the skin effects, non-negligible in this band of frequencies. Surface models can be adapted for high frequencies but not for the lower band, where volume approaches must be used. A mesh inside the conductive volumes has to be setup and therefore has to take into account the skin depth

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}},$$

the distance inside the conductors within the current density decays by 37% of its value on the surface. The skin depth for a conductor made of copper, at the previous frequencies, is between $2.09 \mu\text{m}$ and $1.48 \mu\text{m}$. So, even the thinnest conductors ($35 \mu\text{m}$) must have mesh subdivisions along their thickness.

Moreover, some ferromagnetic materials can also be found in these electrical devices, interacting strongly at the low and the middle frequencies. So, taking into account all these characteristics in a numerical model is a very big challenge. The choice of the method is therefore a critical point, as discussed in the following paragraph.

A quick survey of methods for low frequencies

In electrical engineering, the most popular numerical method was the Finite Element Method (FEM), or Finite Element Analysis (FEA) in the past years. It allows us to solve electromagnetic problems computing only the interactions of nearby degrees-of-freedom (DOF). The result is a low memory consumption and a fast assembly, since only a few interactions have to be computed and stored in a sparse matrix. Even with the advent of the first workstations and computers, the amount of memory available for computations remained very limited, thus pushing the FEM to gain even more interest for computational solutions. Researchers from multiple disciplines (electromagnetism, mechanic, etc.) focused their efforts in the study of the FEM's numerical aspects [1–3], making the FEM one of the most robust method.

Integral Equation Methods

At the end of the twentieth century, as the technology evolves, another method regains some interest: the integral equation method (IEM). Contrary to the FEM, the IEM computes all interactions between each DOF in order to solve a problem. It implies assembling dense matrices, which require a high amount of memory and suffer of quadratic complexities in terms of storage and basic arithmetic operations. Let's take for example the matrix-vector product (\mathbf{Ax}), for a square matrix $\mathbf{A} \in \mathbb{R}^n$: the complexity of the product is n^2 . Because of this, the use of an iterative solver could have become prohibitive if it had not converged in a few iterations.

On the other hand, the IEM needs to mesh only the active materials (see Fig. 2a) like conductors, dielectrics, etc., whereas the FEM also have to mesh the inactive regions (see Fig. 2b) like the air or a vacuum. We can see on the comparison of the figures 2a and 2b that the mesh used for the IEM contains significantly fewer elements than the FEM's. The IEM could be more interesting to solve problems containing a large amount of air. Since it usually has less DOFs, we could hope the assembly and solution to be computed faster than the FEM's.

However, the memory consumption was limiting its range of applications. Today's technologies make available a large amount of computational memory (RAM), for instance, 4 GBytes of RAM is nowadays the minimum amount available for laptops and high-end smartphones. This major evolution enables the IEM to enlarge its range of applications for larger and more complex applications. However, the IEM is still hardly applicable to industrial or compact/dense (in terms of active materials) problems, without the use of any compression techniques.

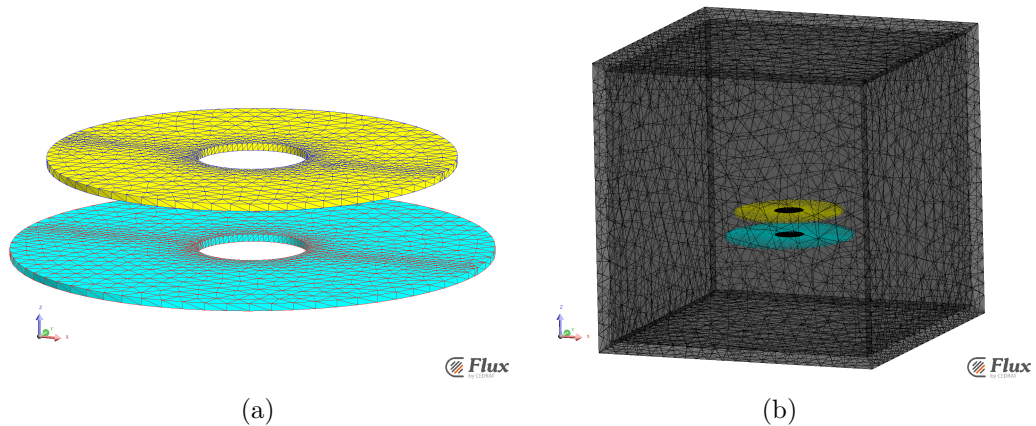


Figure 2: Comparison of meshes used by both methods on a power wireless transfer application. (a) Only the coils have to be meshed for the IEM. (b) A mesh of the coils and the air is required for the FEM.

Compressions

The most popular compression techniques in electromagnetism are the Fast Multipole Methods [4,5] (FMM), their Multi Level version (ML-FMM), and most recently the Cross Approximation based techniques. Their principle is to compute a tree structure to define near- and far-interactions and then approximate/compress the far-interactions. The FMM use multipole expansions to compute the approximations and usually an octree to define the interaction range (near or far). The expansion approach requires an explicit analytic formulæ for each kernel function, therefore limiting the method to known kernels. However, the FMM succeeded in reducing the complexity of the storage and the assembly to a linear complexity ($\mathcal{O}(n)$).

The Cross Approximation techniques take advantage of the rank deficiency of far-interactions to use a low-rank approximation and therefore reduce the storage. The low-rank approximation can be computed using the Adaptive Cross Approximation [6–8] (ACA), an algebraic method, or the Hybrid Cross Approximation [9] (HCA), a semi-analytical method. A hierarchical structure based on a binary-tree called a \mathcal{H} ierarchical matrix [10–12] is used to define the interactions and store the matrices. Moreover, the \mathcal{H} -matrix format also allows us to compute an efficient, fast and light preconditioner.

The \mathcal{H} -matrices are relatively new in the low-frequency community, so we have interests in studying and applying it to the low-frequency formulations.

Low Frequencies

The study of devices in the low frequencies mostly consists in focusing the impact of the resistive effects and the magnetic effects. The higher frequencies effects, as the capacitive or propagative effects are neglected. So, the solution of these problems can be done using eddy currents models.

The operating frequency of devices has now increased to a point that the models have to be improved to be able to work at the middle frequencies. Thus, the capacitive effects have to be added in the formulations. Taking for example the busbars in Figure 3, the conductors are separated by dielectrics. At low frequencies, the inductive couplings have to be modeled, but when the frequency increases the capacitive effects between conductors increase as well.



Figure 3: Example of busbars used on hybrid vehicles (source: Mersen).

The high-frequency community can provide formulations taking into account the capacitive effects with the use of the Method of Moments [13], for example, which suffers of the low-frequency breakdown [14–19]. Some cures exist for this issue, with the use of Loop-Star [20] decompositions. However, a more suitable method for the power electronics is the Partial Element Equivalent Circuit (PEEC) [21–23], which does not suffer of the frequency breakdown and enables external circuit couplings.

In the power electronics applications, the circuit approaches are crucial at low frequencies to study converters, printed circuit boards, etc., and to insure their design works and satisfies some specifications. The PEEC has been adapted for low frequencies [24–26] and the software InCa3D [27] (Inductance and Capacitance calculation for 3D geometries) has been created from these works.

PEEC method

The PEEC method has arisen from the inductance [28] and capacitance [21] calcula-

tion needs that EMC engineers had. The formulations based on the PEEC method are based on Maxwell's integral equations and an equivalent circuit representation of the domain. A simple example of an equivalent circuit is presented in the Figures 4a and 4b, we can see that an element (a hexahedron here) corresponds to one resistance, one self-inductance and mutuals (not represented here). Capacitive couplings between elements are also included in the model to take into account higher frequency effects. The values of these circuit elements are computed thanks to the integral equations, so the solution of the problem consists in solving the circuit equations (Kirchhoff's laws).

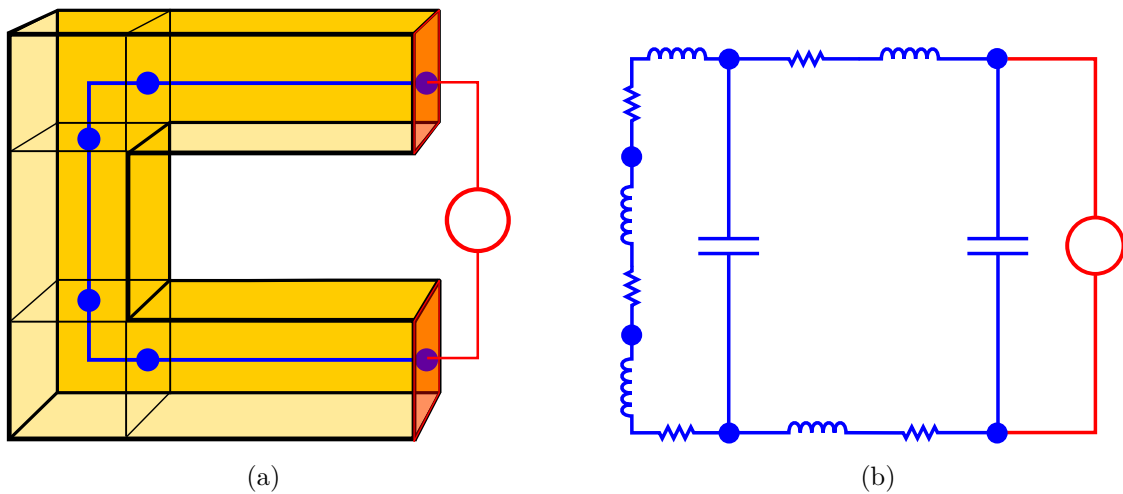


Figure 4: Simplified example of the equivalent circuit representation of a C-shape conductor. (a) The conductor meshed with hexahedra and an overview of the circuit. (b) A more detailed equivalent circuit, where the mutual inductances are hidden for a matter of understanding

Its first version suffered from few limitations, only structured (or Manhattan type) meshes were supported and it used 0-order (constant by element) shape functions for instance. Improvements have been developed to overcome the rectangular mesh limitation [29–33] and the most recent works [32, 33] use RWG [34] shape functions to also improve the accuracy.

The major and most problematic issues of the PEEC remain its ill-conditioning [35, 36] and the dense matrices it has to assemble since it is based on the integral equations. Some works tried to overcome the ill-conditioning using regularization techniques [36] or stating restricting assumptions with a DOF renumbering [37]. But we haven't found in the literature a general solution to this issue. On the other side, to significantly reduce the dense matrix storage and assembly time, we can use matrix compression techniques.

Objectives of this work

This PhD. takes place in the context of a collaboration between the G2Elab (Grenoble Electrical Engineering Laboratory) and the CEDRAT company (recently acquired and absorbed by Altair Engineering France). More specifically in the context of InCa3D, a software developed by both Parties.

InCa3D is a simulation software dedicated to power electronics applications for low and medium frequencies, as in Figure 5. Based on a classical inductive PEEC method, InCa3D provides models for EMC studies, electrical interconnection simulations and circuit element extraction. InCa3D's last version has arisen from V. Ardon's [38] PhD works consisting in a weak coupling of the inductive PEEC formulation with an electrostatic formulation using the Boundary Integral Method (BIM) to add into account capacitive effects and dielectrics.

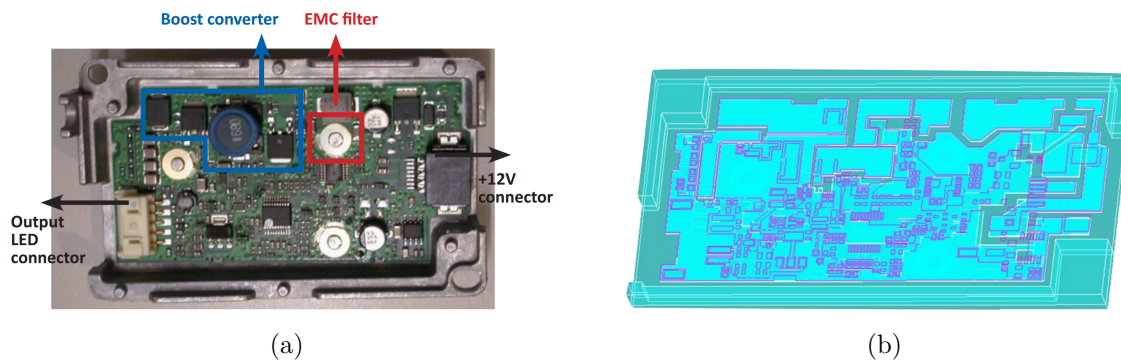


Figure 5: (a) A driver circuit. (b) The InCa3D model. (CEDRAT News N°67)

Today's version of InCa3D (3.1) is able to efficiently analyze structures made of linear conductors and dielectrics with resistive-inductive and capacitive behavior, through parasitic capacitance extraction or resonance frequency computation. Due to the fact that capacitive behavior is added a posteriori to inductive models, two different meshes have to be handled (boundary triangular elements for the electrostatic computation and volume hexahedra for classical PEEC) and their compatibility ensured for the coupling. Consequently, the electrostatic formulation is based on capacitive macro-elements (a set of mesh elements) that have to be defined and linked to a different set of hexahedra with Π - or T - connections by the user in order to solve the problem. The need to define and connect macro-elements significantly restraints the range of applications and the ease-of-use of InCa3D.

These limitations can be overcome by either trying to find a procedure to define and connect properly the macro-elements or using another formulation. This PhD

will propose a new formulation able to be easily used and without specifications or a deep knowledge of the device functioning to be simulated.

The objectives of this PhD are to develop a formulation allowing efficient circuit coupling of problems containing linear conductors, dielectrics and to provide a complete electromagnetic simulation with resistive, inductive and capacitive effects. As secondary objectives, we will focus on the formulation's numerical properties, as conditioning for the solution's efficiency and compression-compatibility to solve large scale problems. The most suited tool to reach these objectives is the \mathcal{H} -matrix allowing us to compress and construct an efficient preconditioner.

Chapter I

\mathcal{H} -matrices

*Life is like riding a bicycle. To keep your balance,
you must keep moving.*

Albert Einstein

CONTENTS

I.1	INTRODUCTION	13
I.1.a	Compression Context	13
I.1.b	Solution Context	15
I.2	CLUSTERING TECHNIQUE	16
I.2.a	Cluster Tree	16
I.2.b	Block Cluster Tree and Admissibility	20
I.3	ADAPTIVE CROSS APPROXIMATION (ACA)	23
I.3.a	Cross Approximation	24
I.3.b	Adaptive Cross Approximation	26
I.4	HYBRID CROSS APPROXIMATION (HCA)	27
I.4.a	Analytical approach	27
I.4.b	Hybrid Cross Approximation (II)	28
I.5	\mathcal{H} -MATRIX ARITHMETIC'S	31
I.5.a	\mathcal{H} -matrix-vector product	31
I.5.b	Agglomeration	32
I.5.c	\mathcal{H} -LU decomposition, an efficient preconditioner	34
I.6	NUMERICAL RESULTS	38
I.6.a	Electrostatic example	38
I.6.b	Magnetostatic formulation	40
I.7	CONCLUSION	44

Resume

In this chapter, we are presenting the \mathcal{H} -matrices through a brief overview of the theory. We aim to present the tool, so we won't provide a deep analysis of the theory. However, all the references will be provided for further details. The \mathcal{H} -matrices are then applied to an electrostatic and a magnetostatic formulation, in order to show their efficiency.

I.1 Introduction

In the integral equation methods context, the use of compression techniques is mandatory to avoid the high costs coming from dense matrices. Let us consider a classical integral equation [39, Chap.3]:

$$\int_{\Omega} \varphi_i(\mathbf{x}) \sum_{j=0}^{n-1} \int_{\Omega} \mathcal{G}(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) u_j d\mathbf{y} d\mathbf{x} = \int_{\Omega} \varphi_i(\mathbf{x}) \mathcal{F}(\mathbf{x}) d\mathbf{x} \quad 0 \leq i < n, \quad (\text{I.1})$$

arising from a standard Galerkin discretization and projected in the n -dimensional space $V_n = \text{span}\{\varphi_0, \dots, \varphi_{n-1}\}$, with \mathcal{G} a Green kernel function, \mathcal{F} the right-hand-side and the discrete solution $\mathcal{U}_n = \sum_{i=0}^{n-1} \varphi_i u_i$ projected in V_n . The equation (I.1) can be expressed as:

$$\mathbf{G}\mathbf{u} = \mathbf{f} \quad (\text{I.2})$$

where

$$\begin{aligned} \mathbf{G}_{i,j} &= \int_{\Omega} \varphi_i(\mathbf{x}) \int_{\Omega} \mathcal{G}(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) d\mathbf{y} d\mathbf{x}, \\ \mathbf{f}_i &= \int_{\Omega} \varphi_i(\mathbf{x}) \mathcal{F}(\mathbf{x}) d\mathbf{x}, \end{aligned}$$

The problematic dense matrices (i.e.: all entries are not zeros) that we will encounter can be written as \mathbf{G} , a classical form of integral equation matrices. To avoid the issues arising from the dense matrix \mathbf{G} , we have to reduce a maximum the number of entries to compute and store.

I.1.a Compression Context

An approach to reduce the costs is to take advantage of the rank deficiency of the far-interactions of \mathbf{G} . Let's take an example to illustrate the principle of this approach. First, we are considering a domain $\Omega = C_1 \cup C_2$ containing two distant cubes C_1 and C_2 , illustrated in Figure I.1. The interactions of \mathbf{G} can be separated as:

$$\mathbf{G} = \begin{array}{cc} & \begin{array}{cc} C_1 & C_2 \end{array} \\ \begin{array}{c} C_1 \\ \\ C_2 \end{array} & \begin{array}{|cc|} \hline \mathbf{G}|_{C_1 \times C_1} & \mathbf{G}|_{C_1 \times C_2} \\ \hline \mathbf{G}|_{C_2 \times C_1} & \mathbf{G}|_{C_2 \times C_2} \\ \hline \end{array} \end{array} \quad (\text{I.3})$$

The diagonal blocks correspond to near-interactions (i.e.: each cube against itself) and the extra-diagonal to far-interactions (one cube against the other). Let us take

for example a mesh where the cube's boundary is discretized with quadrangles (4 per faces \Rightarrow 24 by cubes) and using 0-order shape functions (φ_i element-wise constant), we have $\mathbf{G}|_{C_1 \times C_2} \in \mathbb{R}^{24 \times 24}$. It suffers of a rank deficiency $\text{rank}(\mathbf{G}|_{C_1 \times C_2}) = 16$ and usually a few ranks are enough to approximate it precisely. Computing the classical (non truncated) Singular Value Decomposition (SVD) of $\mathbf{G}|_{C_1 \times C_2}$:

$$\mathbf{G}|_{C_1 \times C_2} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{I.4})$$

we get the singular values (plotted in the figure I.2) stored in the diagonal of $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_n)$. The equation (I.4) is a low-rank matrix representation called the orthonormal outer-product form since \mathbf{U} and \mathbf{V} have orthonormal columns. The SVD allows us to define the minimal rank of the approximation for a given error criterion ϵ . For examples, if $\epsilon = 10^{-2}$ (1%) we can approximate $\mathbf{G}|_{C_1 \times C_2}$ using only a rank-1 approximation \mathbf{R}_1 defined as

$$\mathbf{R}_1 = \mathbf{U}(:, 1)\mathbf{\Sigma}(1, 1)\mathbf{V}(:, 1)^T, \quad (\text{I.5})$$

using the Matlab syntax. The error of this approximation is

$$\frac{\|\mathbf{R}_1 - \mathbf{G}|_{C_1 \times C_2}\|}{\|\mathbf{G}|_{C_1 \times C_2}\|} \sim 1.04 \times 10^{-4}, \quad (\text{I.6})$$

it could also be estimated using the singular values in Figure I.2 and knowing that the relative error of a rank- k can be computed using $\epsilon_r(k) = \frac{\sigma_{k+1}}{\sigma_1}$. If we need a greater precision $\epsilon = 10^{-6}$ for example, the rank-4 approximation \mathbf{R}_4 gives a relative error of 2.44×10^{-8} . The memory requirement of $\mathbf{R}_4 = \mathbf{U}(:, 1:4)\mathbf{\Sigma}(1:4, 1:4)\mathbf{V}(:, 1:4)^T$ is of $mk + k + kn = 196$ entries against $mn = 576$ for $\mathbf{G}|_{C_1 \times C_2}$, dividing by almost

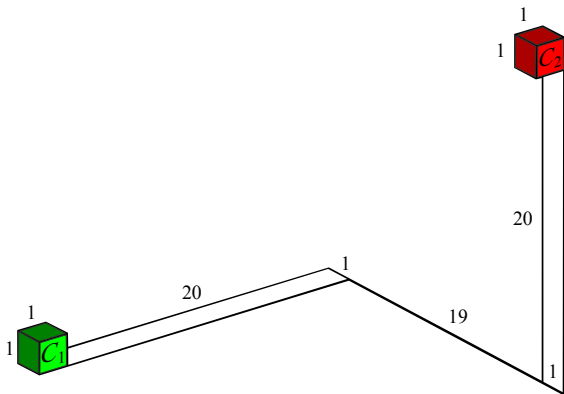


Figure I.1: The geometry of the example considered. The dimensions are in mm units

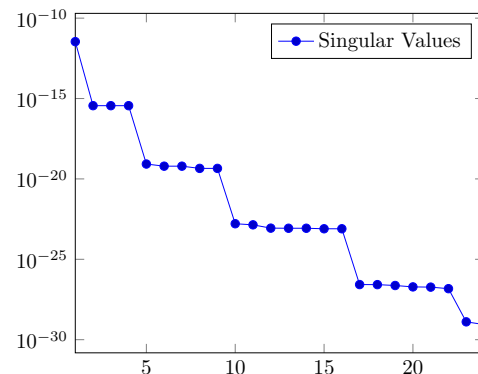


Figure I.2: Plot of the singular values of $\mathbf{G}|_{C_1 \times C_2}$.

3 the amount of memory required. Note that, in this manuscript, m and n is used to denote the number of rows and columns of a matrix, respectively, and k is usually used for the rank's approximation. The low-rank matrix representation also reduces the complexity of basic arithmetic operations, for instance, the product matrix-vector $\mathbf{R}_k \mathbf{b}$ have a complexity of $\mathcal{O}(k(m+n))$ instead of $\mathcal{O}(mn)$. We can see, thank to this example, the gains when using low-rank matrix approximations in term of memory consumption and complexity.

I.1.b Solution Context

Once the assembly is done (i.e.: the compressions and the full assemblies), the previous matrix \mathbf{G} can be approximated by:

$$\mathbf{G} \approx \tilde{\mathbf{G}} = \begin{array}{|c|c|} \hline \mathbf{G}_1 & \mathbf{R}_k \\ \hline \mathbf{R}_k^T & \mathbf{G}_2 \\ \hline \end{array} \quad (\text{I.7})$$

where $\mathbf{G}_i = \mathbf{G}|_{C_i \times C_i}$ for $i = 1, 2$, the green blocks are compressed and the red ones are dense matrices. We can note that this format is the simplest form of a \mathcal{H} -matrix.

The problem (I.2) can then be solved very efficiently using the approximation $\tilde{\mathbf{G}}$ and taking advantage of its format. A hierarchical LU decomposition, similar to a block-LU factorization, can be applied to $\tilde{\mathbf{G}}$:

$$\begin{array}{|c|c|} \hline \mathbf{G}_1 & \mathbf{R}_k \\ \hline \mathbf{R}_k^T & \mathbf{G}_2 \\ \hline \end{array} = \begin{array}{|c|c|} \hline \begin{array}{|c|c|} \hline 0 & 0 \\ \hline \mathbf{L}_1 & \end{array} & \begin{array}{|c|} \hline 0 \\ \hline \end{array} \\ \hline \begin{array}{|c|} \hline \mathbf{L}_2 \\ \hline \end{array} & \begin{array}{|c|c|} \hline 0 & \\ \hline \mathbf{L}_3 & \end{array} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \begin{array}{|c|c|} \hline \mathbf{U}_1 & \\ \hline 0 & \end{array} & \begin{array}{|c|} \hline \mathbf{U}_2 \\ \hline \end{array} \\ \hline \begin{array}{|c|} \hline 0 \\ \hline \end{array} & \begin{array}{|c|c|} \hline & \mathbf{U}_3 \\ \hline 0 & \end{array} \\ \hline \end{array} \quad (\text{I.8})$$

where its construction comes down in solving the next linear system

$$\begin{cases} \mathbf{G}_1 = \mathbf{L}_1 \mathbf{U}_1 & \text{gives } \mathbf{L}_1 \text{ and } \mathbf{U}_1, \\ \mathbf{R}_k = \mathbf{L}_1 \mathbf{U}_2 & - \mathbf{U}_2, \\ \mathbf{R}_k^T = \mathbf{L}_2 \mathbf{U}_1 & - \mathbf{L}_2, \\ \mathbf{G}_2 = \mathbf{L}_2 \mathbf{U}_2 + \mathbf{L}_3 \mathbf{U}_3 & - \mathbf{L}_3 \text{ and } \mathbf{U}_3. \end{cases}$$

In the previous system, we have to compute two LU decompositions (first and last steps) and two forward substitutions. This LU factorization has a lower cost than

one applied directly on \mathbf{G} , since such decomposition scales with $\mathcal{O}(n^3)$ and the last system requires two smaller decompositions. Moreover, we can note that the hierarchical structure of $\tilde{\mathbf{G}}$ is preserved, and therefore the memory consumption is also reduced.

The problem can finally be solved using forward and back block-substitutions, as the block LU-decompositions. Any examples as light as this one can be easily solved using direct solvers, but for real examples, a direct solution cannot always be achieved. An iterative solution therefore has to be employed. We will see later on that this hierarchical LU decomposition can also be used as a preconditioner to accelerate an iterative solver.

We have shown an overview of what the \mathcal{H} -matrices can bring to the table with the low-rank approximations as a compression technique, and how we can take advantage of it to compute efficient solutions.

On this particular example the identification of far-interactions is trivial, but for more general 3D geometries, we will have to use a clustering technique. The next section of this chapter will be dedicated to the clustering techniques used to define the near- and far-interactions for any geometry. We have computed the approximations using the SVD, unfortunately, it is a very expensive method with a high complexity, and so cheaper methods have to be used. Two low-rank approximation techniques with a small cost and complexity will be presented in the sections [I.3](#) and [I.4](#). In [Section I.5](#), we give an overview of the arithmetic operations essential for an efficient preconditioner construction. Some numerical tests are finally given in [Section I.6](#) in order to show the efficiency of the \mathcal{H} -matrices.

I.2 Clustering Technique

As previously mentioned, the \mathcal{H} -matrices are based on a hierarchical structure in order to define blocks of interactions. The near- and far-interactions can be easily defined for trivial examples, but for more general problems a *cluster tree* has to be used.

I.2.a Cluster Tree

Let us define $\mathcal{I} = \{0, \dots, n-1\} \subset \mathbb{N}$ the set of indices corresponding to the indices of the shape functions φ_i . We can therefore say that \mathcal{I} is also representing the DOFs,

so in order to define blocks of interactions and their admissibility, we want to create a partition of the product $\mathcal{I} \times \mathcal{I}$. A general approach to create a partition of $\mathcal{I} \times \mathcal{I}$ is to construct a cluster tree of \mathcal{I} first.

Definition I.2.1 (Cluster Tree). *The rooted tree T is a **cluster tree** $T_{\mathcal{I}}$ for \mathcal{I} , if there is a mapping function $\hat{\cdot} : t \rightarrow \hat{t}$ which links each node t of T to a subset of \mathcal{I} called the label of t , and satisfies:*

- the label of the root $r \in T$ is defined as $\hat{r} = \mathcal{I}$,
- every node passes on to his sons a disjoint partition of its label, i.e.:

$$\hat{t} = \bigcup_{s_i \in \text{sons}(t)} \hat{s}_i \quad \text{and} \quad \hat{s}_i \cap \hat{s}_j = \emptyset, \forall i \neq j.$$

The nodes t of $T_{\mathcal{I}}$ are called clusters.

For a matter of pedagogy, let us consider an example of cluster tree in Figure I.3a with its rooted tree associated in Figure I.3b. This example consists in constructing a cluster tree for a 1-D geometry: a segment containing 10 points of discretization. In the first level (the higher), the node $r = 1$ is the root of the tree and the set of indices associated is $\hat{r} = \mathcal{I} = \{0, \dots, 9\}$. In the second level we have cut the geometry in the middle of the segment, and then created two subsets $\hat{2} = \{0, \dots, 4\}$

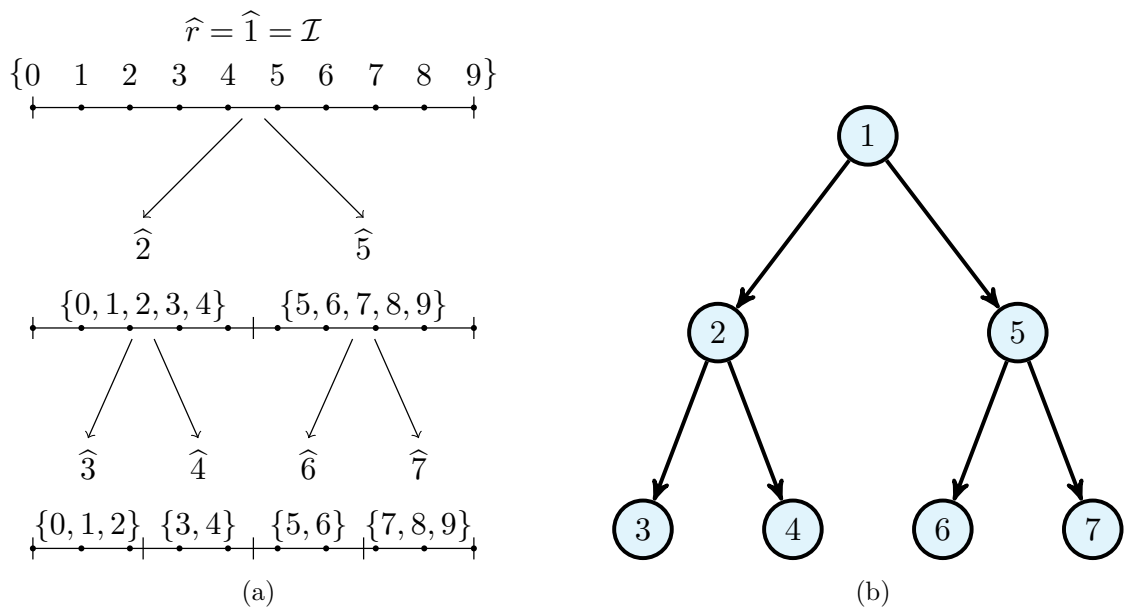


Figure I.3: (a) An example of tree with 7 nodes. (b) The cluster tree, with each cluster associated to a tree's node.

and $\widehat{\mathfrak{S}} = \{5, \dots, 9\}$ respecting the last two conditions of the Definition I.2.1. We repeat the clustering process for each son once more.

The clustering of 1D examples can be trivial, but for n-dimensional problems more general clustering techniques have to be used in order to construct the cluster tree.

We have mentioned earlier that the indices $i \in \mathcal{I}$ correspond to the shape function φ_i , but in the clustering context, a shape function is represented by its support $\Omega_i = \text{supp}(\varphi_i)$. However, considering the support itself is very complicated and might be costly. To simplify the problems, we can choose a point $\mathbf{x}_i \in \Omega_i, \forall i \in \mathcal{I}$ to represent the support of φ_i . For example, 0-order shape functions (constant by element) can be represented by the barycenter of its supporting element.

The general approach starts by setting the root r as the whole index set $\widehat{r} = \mathcal{I}$ in order to satisfy the first condition of the cluster tree definition. Then, considering all the points $\mathbf{x}_i \in \mathbb{R}^d$ (d the dimension, usually 2 or 3) of the cluster $t = r$, we want to split them and create two non-overlapping clusters in respect to the other conditions of Definition I.2.1. To achieve this, we compute a bounding box B_t of the cluster t , defined by:

$$B_t = [a_1, b_1] \times \dots \times [a_d, b_d], \quad \text{with } a_i, b_i \in \mathbb{R}. \quad (\text{I.9})$$

Then, we split B_t defining the splitting direction $j \in \{1, \dots, d\}$ and $c = (a_j + b_j)/2$. We define a partition of t as $\{t_0, t_1\}$ such as:

$$\widehat{t}_0 = \{i \in \widehat{t} \mid (\mathbf{x}_i)_j \leq c\} \quad \text{and} \quad \widehat{t}_1 = \{i \in \widehat{t} \mid (\mathbf{x}_i)_j > c\} \quad (\text{I.10})$$

with $(\mathbf{x}_i)_j$ the j^{th} dimension of \mathbf{x}_i . Finally, we repeat this process with the sons $t = t_0$ and $t = t_1$ until the clusters reach a minimum number of indices n_{\min} .

The definition of the bounding boxes, i.e a_i and b_i , depends on the method used.

Two main methods can be used: The Regular and the Adaptive.

The regular subdivision method stands on a regular splitting of B_t at each step, defining the son's bounding boxes B_{t_0} and B_{t_1} as:

$$B_{t_0} = [a_1, b_1] \times \dots \times [a_j, c] \times \dots \times [a_d, b_d] \quad \text{and} \quad B_{t_1} = [a_1, b_1] \times \dots \times [c, b_j] \times \dots \times [a_d, b_d].$$

At each step the bounding boxes are split in half following the dimension j . This regular construction is the most naive and fast method constructing bounding boxes with the same sizes, but it may construct clusters that are empty.

A method to avoid empty clusters is the adaptive subdivision method, also called geometric bisection. Contrary to the regular method, this one defines boxes shrinking

to their minimal size while containing all the points. The bounding boxes are defined at the beginning of each step as in (I.9) with

$$a_i = \min_{k \in \ell} ((\mathbf{x}_k)_i) \quad \text{and} \quad b_i = \max_{k \in \ell} ((\mathbf{x}_k)_i) \quad \forall i \in \{1, \dots, d\},$$

constructing smaller and more distant clusters.

We can see the difference of the methods on a 2D example in the figures I.4. The first row of figures (I.4a, I.4b, I.4c) shows the splits done at the first two steps of the regular subdivision, whereas the next row (I.4d, I.4e, I.4f) shows the partitions of the adaptive subdivision. We clearly see that the adaptive method gives smaller and more distant clusters at the level 1 and 2. Since we want a maximum of far-interaction blocks, in order to compress more, we choose to use the adaptive subdivision method for our works.

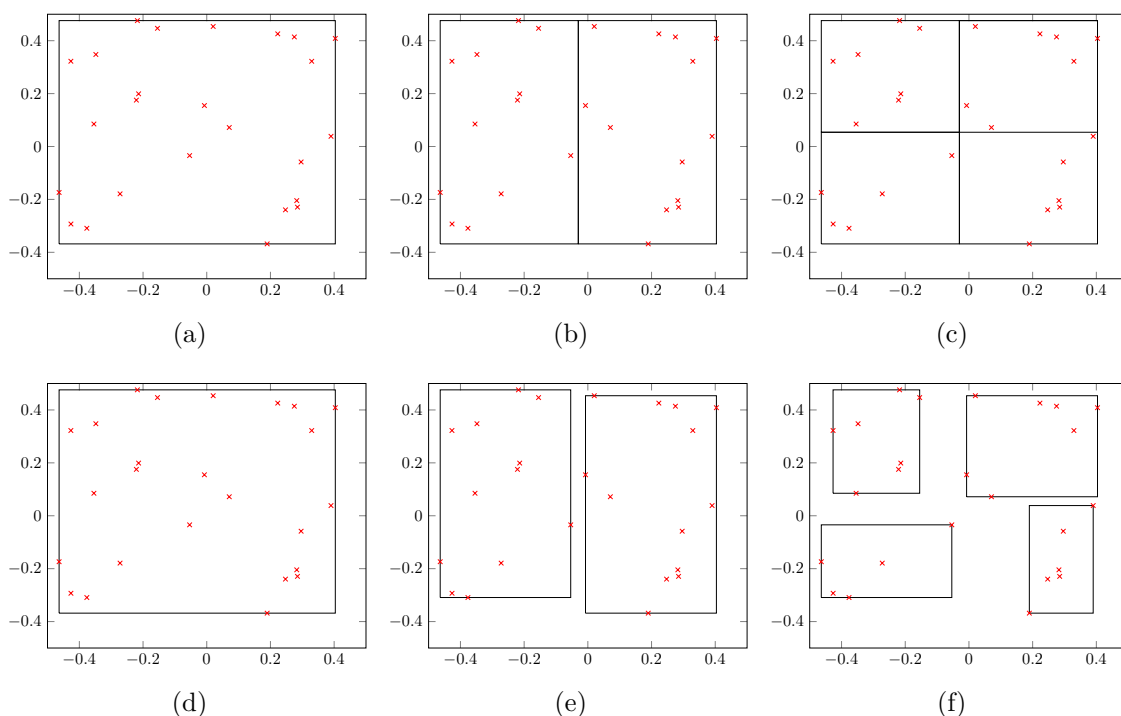


Figure I.4: Comparison of two subdivision methods. The left figures are the lvl 0 tree, the middle ones the lvl 1 and the right ones are the lvl 2. (a - c) Regular subdivision method. (d - f) Adaptive Subdivision method.

The choice of the direction/dimension j to split is not trivial, we choose to split perpendicularly to the dimension where the bounding boxes are the largest. Another approach may consist in choosing an initial direction j and incrementing it at each level.

We can find pedagogical examples of implementation in the Lecture note of Steffen Börm and al. [39] using the C programming.

We now know how to split our geometry and create clusters of DOFs. The next step is to define the blocks of interactions and their admissibility to the compression or not.

I.2.b Block Cluster Tree and Admissibility

Given two index sets \mathcal{I} and \mathcal{J} , we want to compute the interactions $\mathcal{I} \times \mathcal{J}$ efficiently and without quadratic costs. Computing directly the product $\mathcal{I} \times \mathcal{J}$ leads to the dense matrix \mathbf{G} of (I.2). In order to reduce the costs, we have to split the matrix in near- and far-blocks of interactions, like in (I.3). The definition of these blocks can be done using the cluster tree $T_{\mathcal{I}}$, $T_{\mathcal{J}}$ and testing the clusters of each tree against the other. The most-suited structure enabling such approach is called a *block cluster tree*.

Definition I.2.2 (Block Cluster Tree). *Given two cluster trees $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$, the block cluster tree $T_{\mathcal{I} \times \mathcal{J}}$ is a rooted tree where the nodes $b \in T_{\mathcal{I} \times \mathcal{J}}$ are defined by a pair of clusters, as $b = (t, s)$ with $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{J}}$. Moreover, the root of $T_{\mathcal{I} \times \mathcal{J}}$ must be defined as $\text{root}(T_{\mathcal{I} \times \mathcal{J}}) = (\text{root}(T_{\mathcal{I}}), \text{root}(T_{\mathcal{J}}))$ (as in the level 0 of Figure I.5) and the sons of each node $b = (t, s)$ are the combination of each son of t and s . Usually leading to a succession of four sons, as in the next levels of Figure I.5.*

*The label of $b = (t, s)$ is then given by the index product $\hat{b} = \hat{t} \times \hat{s} \subseteq \mathcal{I} \times \mathcal{J}$. We will call the nodes b **block clusters**, whose label corresponds to a subpart of the whole matrix.*

We have presented a general definition of a block cluster tree for two different cluster trees \mathcal{I} and \mathcal{J} , but in fact, we usually have $\mathcal{I} = \mathcal{J}$ and $T_{\mathcal{I}} = T_{\mathcal{J}}$.

The main purpose of such object in the compression context is to partition the matrix \mathbf{G} in sub-blocks and compress as many blocks as possible in order to lower the costs. We have mentioned earlier that far-interactions blocks can be efficiently compressed, so to define the admissibility condition of a block (t, s) to the compression, we want the clusters t and s to be distant. The following Definition provides a rigorous statement of this last condition.

Definition I.2.3 (Admissibility condition). *Let $b = (t, s) \in T_{\mathcal{I} \times \mathcal{J}}$, $\eta \in \mathbb{R}$, b is an admissible block cluster if it verifies*

$$\min \{ \text{diam}(\Omega_t), \text{diam}(\Omega_s) \} \leq \eta \text{dist}(\Omega_t, \Omega_s) \quad (\text{I.11})$$

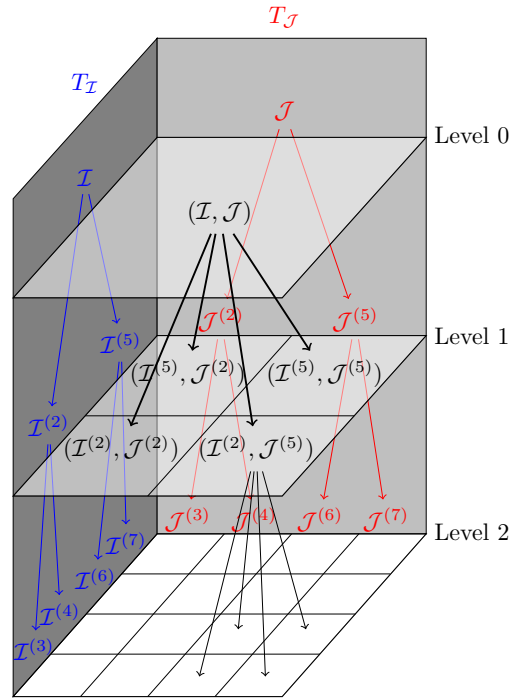


Figure I.5: Example of a block cluster tree level-wise.

with

$$\Omega_\tau = \bigcup_{i \in \hat{\tau}} \Omega_i,$$

the support of the cluster τ .

The condition I.11 may seem easy to check, but for complex geometries, the computation of the diameters and distances of the clusters can be expensive. A simple way to compute the diameter of a support is using once more the bounding box approach. A support usually contains multiple elements, so computing the bounding boxes will simply consist in gathering all the mesh's points used by these elements and looking for the minimum and maximum values in each dimension. Figure I.6 shows the elements (filled in gray) used by different types of DOF as well as the set of nodes (the black circles) tested to get the resulting boxes (in green).

The diameter of a bounding box $B_t = [a_1, b_1] \times \cdots \times [a_d, b_d]$ is then computed as

$$\text{diam}(\Omega_t) = \text{diam}(B_t) = \left(\sum_{\delta=1}^d (b_\delta - a_\delta)^2 \right)^{1/2}, \quad (\text{I.12})$$

representing the length of the box's diagonal. The distance between two bounding boxes B_t and $B_s = [c_1, d_1] \times \cdots \times [c_d, d_d]$ can then be easily computed using:

$$\text{dist}(B_t, B_s) = \left(\sum_{\delta=1}^d \text{dist}([a_\delta, b_\delta], [c_\delta, d_\delta])^2 \right)^{1/2}. \quad (\text{I.13})$$

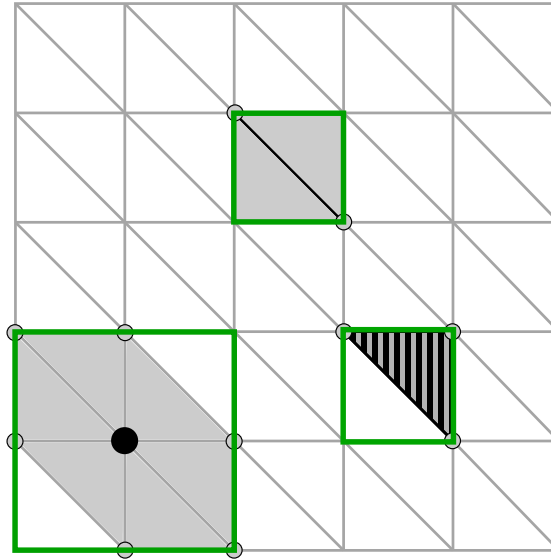


Figure I.6: Example of supports for DOFs of type (from left to right) nodal, edge and element. The elements filled with grey are the elements used by the DOF in black. The nodes used to compute the bounding boxes (in green) are represented by black circles (not filled).

This approach allows us to compute the distances and the diameters of the supports with low and constant complexity $\mathcal{O}(1)$.

We can note that there are other approaches to do such computation as Chebyshev's [40]. However, we choose to use the box approach due to its ease of implementation, whereas the Chebyshev's is more difficult to implement.

Once the bounding boxes of each cluster of the cluster trees are computed and an admissibility condition chosen, we can construct the block cluster tree following the algorithm 1.

The initial call of this recursive algorithm is with $(\text{root}(T_{\mathcal{I}}), \text{root}(T_{\mathcal{J}}))$ the roots of both trees, in order to satisfy the first condition of the Definition 1.2.2. The next conditions of this Definition are satisfied by construction.

Let us consider the previous 1D example showed in Figure 1.3a with its cluster tree, and use it to illustrate the block cluster tree construction. We choose the admissibility condition (1.11) with η a very large number, so that any disjoint clusters would be admissible. Using the cluster tree $T_{\mathcal{I}}$ in Figures 1.3a (or 1.7a), we are going to construct the block cluster $T_{\mathcal{I} \times \mathcal{I}}$. So, applying the Algorithm 1 with the last admissibility condition, we obtain the hierarchy of the block cluster tree in Figure 1.7b for the first two recursive steps of the construction. Note that this hierarchy

Algorithm 1 Construction of the Block Cluster Tree $T_{\mathcal{I} \times \mathcal{J}}$, given $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$.

```

1: procedure BLOCKCLUSTERTREECONSTRUCTION( $t \in T_{\mathcal{I}}, s \in T_{\mathcal{J}}$ )
2:   if  $(t, s)$  satisfies the admissibility condition then
3:     // Stop the recursive process
4:     sons( $t \times s$ ) =  $\emptyset$ 
5:     return
6:   else
7:     // Set the hierarchy
8:     sons( $t \times s$ ) =  $\{t' \times s' \mid t' \in \text{sons}(t) \text{ and } s' \in \text{sons}(s)\}$ 
9:     // Check if any sons are admissible
10:    for each son:  $(t', s') \in \text{sons}(t \times s)$  do
11:      BLOCKCLUSTERTREECONSTRUCTION( $t', s'$ )
12:    end for
13:  end if
14: end procedure

```

directly corresponds to the matrix structure.

At the 0^{th} level (initial step), we have the block cluster (r, r) which is not admissible, so we initiate the recursive process with the sons 2 and 5. At the level 1, there are four block clusters $\{(2, 2), (5, 5), (2, 5), (5, 2)\}$. Since the cluster 2 and 5 are disjoint, the block clusters $(2, 5)$ and $(5, 2)$ are admissible (filled with green) and we can stop the recursion. The other two block clusters (filled with red) do not satisfy the condition, so we repeat the process once more at the level 2, and we get the same results one level deeper.

Now, we have presented all the tools to define the far- (admissible) and near-interactions (non-admissible), so we have the hierarchical structure to compress a matrix. The next sections will present the most popular compression techniques based on low-rank approximations.

I.3 Adaptive Cross Approximation (ACA)

A brief demonstration of the low-rank approximation efficiency using the SVD has been shown in [section I.1 Introduction](#) at the beginning of this chapter. The approach shown is very expensive in time and memory due to the SVD which is intrinsically a very expensive decomposition and requires all the entries of the matrix. So, we have to find another way to compute low-rank approximations.

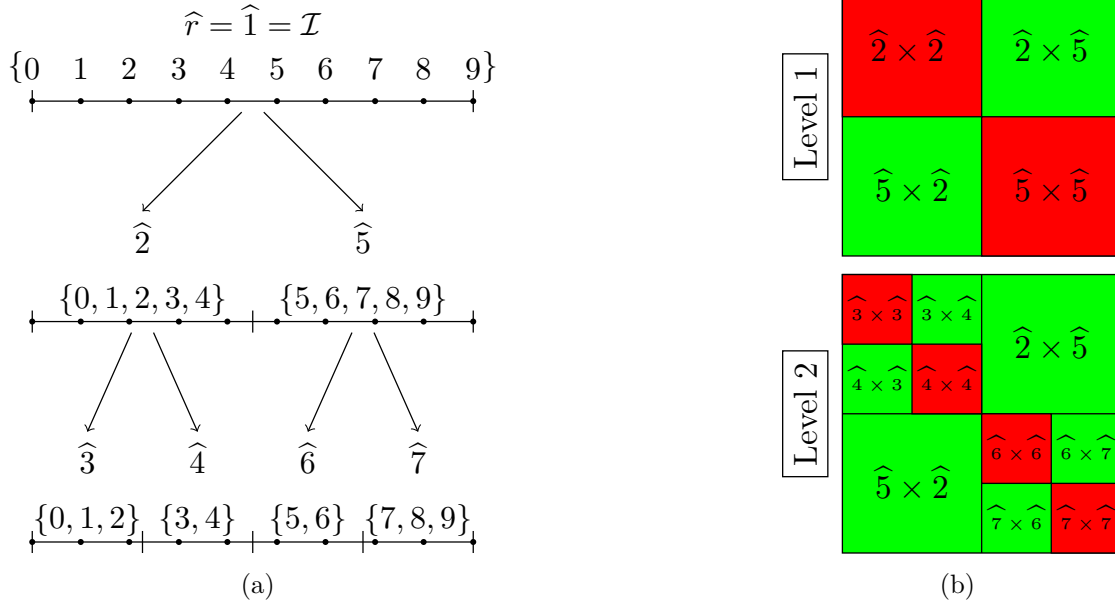


Figure I.7: (a) Duplicate of the Figure I.3a for a matter of understanding representing a cluster tree $T_{\mathcal{I}}$. (b) The matrix splits according to the block cluster tree construction at level 1 and 2.

I.3.a Cross Approximation

We can find in the recent literature the papers [8,41–43] presenting a purely algebraic approach called Pseudoskeleton approximations (or cross approximations) using low-rank matrices. This algorithm requires only a few entries of the matrix, so we don't have to assembly the whole matrix. This allow us to save memory and quadrature costs for all admissible block clusters.

The main idea is to construct a low-rank approximation using only a few rows and columns of the matrix. Let us consider an admissible block cluster (t, s) and its associated matrix \mathbf{M} (a submatrix of \mathbf{G}). A cross approximation consists in choosing pivots (τ_i, σ_i) (or crosses, like in figure I.8 on the matrix \mathbf{M}), forming a subset of k rows and columns $t^* = \{\tau_i, \forall i = \{0, \dots, k-1\}\} \subset t$ and $s^* = \{\sigma_i, \forall i = \{0, \dots, k-1\}\} \subset s$, respectively. The approximation of rank k is then defined as the following low-rank matrix \mathbf{R} :

$$\mathbf{M} \approx \mathbf{R} = \mathbf{A}\mathbf{B}^T \quad (\text{I.14})$$

with $\mathbf{A} = \mathbf{G}(:, s^*)$ and $\mathbf{B} = \mathbf{G}(t^*, :)$ the matrices storing in their columns the chosen columns s^* and rows t^* , respectively (see the figure I.8 for example).

The demonstration of the existence of this kind of cross approximations can be found

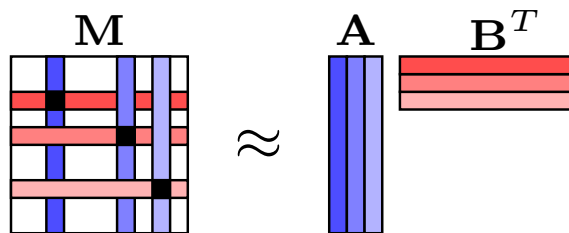


Figure I.8: Illustrative example of cross approximations.

in [41, Th. 3.1].

The pending point is to define how to choose the crosses. Since we don't want to compute all the entries of the matrix, we don't have any information to choose the cross, a priori. Therefore, we are using a heuristic to choose an arbitrary index of row τ_0 at first. We usually pick the first index of \hat{t} or randomly. Once the first row's index of the cross is chosen, we set the cross at the column's index σ_0 as

$$\sigma_0 = \operatorname{argmax}_{j \in \hat{s}} |\mathbf{M}_{\tau_0, j}|$$

the index of the maximal entry in modulus on the row τ_0 . We can note that this procedure can be started by choosing the column's index first. However, we have chosen the cross (τ_0, σ_0) with the need to compute only one row, and one column to complete the cross. The entire method is presented in Algorithm 2 and consists in repeating almost the previous procedure k times.

This algorithm is efficient only if we are able to assembly sub-parts of the global matrix as rows and columns parts, for instance. So, one should be ensured that its assembling functions enable such feature. Adopting this *on-the-go* procedure for all the admissible block clusters will allow us to significantly reduce the costs since it avoids direct assemblies of all the sub-matrices. The same goes for the complexity, this procedure can be computed with a linear complexity $\mathcal{O}(k^2(\#\hat{t} + \#\hat{s}))$, whereas the usual approach (direct assembly) has a quadratic complexity.

The construction of the approximation requires just the prescribed rank k as parameter. The accuracy of the approximation is therefore defined very approximately by the user experience. Moreover, all the blocks do not give the same accuracy for a same rank. The global accuracy is finally very hard to estimate *a priori*. A method has been published soon after the last one based on the same principles by M. Bebendorf in [7]: the Adaptive Cross Approximation (ACA). Probably the most popular low-rank approximation method nowadays.

Algorithm 2 Cross approximation algorithm with a Matlab syntax.

```

1: procedure CROSSAPPROXIMATION(block cluster  $b = (t, s)$ , rank  $k \in \mathbb{N}^*$ )
2:   Set  $\tau_0 = \min\{i \in \hat{t}\}$  or  $\tau_0 = \text{random}(i \in \hat{t})$ 
3:   for  $r = 0, \dots, k - 1$  do
4:     Seek  $\sigma_r = \text{argmax}_{j \in \hat{s}} |\mathbf{M}(\tau_r, j)|$  ▷ Compute the row  $\tau_r$ 
5:     Compute  $\delta = \mathbf{M}(\tau_r, \sigma_r) - \mathbf{A}(\tau_r, :) \mathbf{B}(\sigma_r, :)^T$ 
6:     if  $\delta = 0$  then
7:       Terminate with a rank  $r$  approximation OR look for another cross
8:     else
9:       Compute and store the cross in  $\mathbf{A}$  and  $\mathbf{B}$ :
10:       $\mathbf{A}(:, r + 1) = \mathbf{M}(:, \sigma_r) - \mathbf{A}(:, 1 : r) \mathbf{B}(\sigma_r, 1 : r)^T$  ▷ Comp. the col.  $\sigma_r$ 
11:       $\mathbf{B}(:, r + 1) = \left( \mathbf{M}(\tau_r, :) - \mathbf{A}(\tau_r, 1 : r) \mathbf{B}(:, 1 : r)^T \right)^T$ 
12:    end if
13:    Choose the next pivot's row:  $\tau_{r+1} = \text{argmax}_{i \in \hat{t} \setminus t^*} |\mathbf{B}(:, r)|$ 
14:  end for
15:  return the rank  $k$  approximation  $\mathbf{R} = \mathbf{A} \mathbf{B}^T$ 
16: end procedure

```

I.3.b Adaptive Cross Approximation

Given an approximation accuracy $\epsilon \in \mathbb{R}^+$ (usually $0 < \epsilon \ll 1$), the ACA aims the construction of a low-rank matrix $\mathbf{R} = \mathbf{A} \mathbf{B}^T$ of rank $k(\epsilon)$ determined adaptively according to the accuracy which satisfies:

$$\frac{\|\mathbf{M} - \mathbf{R}\|}{\|\mathbf{M}\|} \leq \epsilon, \quad (\text{I.15})$$

the relative error criterion. In fact, checking directly this criterion contradicts what we want to achieve, because the evaluation of a norm requires all the entries of the matrix, so another way to estimate the accuracy is needed.

Another heuristic is used to estimate the last criterion. Assuming the approximation's accuracy decays at each iteration, we can estimate

$$\|\mathbf{M} - \mathbf{R}^{(r)}\| \lesssim \|\mathbf{M} - \mathbf{R}^{(r-1)}\| \approx \|\mathbf{R}^{(r)} - \mathbf{R}^{(r-1)}\| = \|\mathbf{A}(:, r) \mathbf{B}(:, r)^T\| \quad (\text{I.16})$$

where $\mathbf{R}^{(r)}$ is the approximation at iteration r . This estimation allows us to define the new criterion:

$$\epsilon_{rel}(r) = \frac{\|\mathbf{A}(:, r)\|_2 \|\mathbf{B}(:, r)\|_2}{\|\mathbf{A}(:, 1)\|_2 \|\mathbf{B}(:, 1)\|_2} \leq \epsilon, \quad (\text{I.17})$$

approximating (I.15) and requiring only the columns of the low-rank matrix already computed. Finally, the algorithm of the ACA is the same as Algorithm 2, except at

line 3: the loop **for** is replaced by the loop “**while** $\epsilon_{rel}(r) > \epsilon$ ”. A more detailed and concrete algorithm can be found in [44, Algo. 3.9].

The ACA is a very general method since it is purely algebraic and no restrictive assumptions have been stated. Many publications show his great efficiency and capacity in solving large scale electromagnetic problems. A counterpart of its efficiency is the fact that it is based on two heuristics (error criterion and pivot choice). Using this pivot choice heuristic do not allow any proof of the convergence for the general case, in fact, there are simple examples where the ACA fails (for example: [39, Ex. 4.8 and 4.9]).

There exists a way to choose the pivots with which the convergence of a cross approximation can be proved: The full pivoting approach (see in [39, Sec. 4.3]). It requires to know all entries of the matrix and looks for the maximal entry in absolute value in order to set the cross. However, the costs and complexity of this approach are too high and quadratic. A way for such method to have a low complexity and a proven converge is to sacrifice its generality (the “purely algebraic” criterion), and finally have some kind of *hybrid* approach.

I.4 Hybrid Cross Approximation (HCA)

The low-rank matrix approximations can also be computed using an interpolation approach, usually aiming the separation of variables of the kernel function \mathcal{G} . The interpolation approach is very expensive compared to the ACA, in addition, the resulting approximation is usually not as accurate (except for cases where the ACA fails).

The HCA is a hybridization taking advantages of the ACA and an analytical method, from which results an efficient and robust approximation algorithm. Let us start by introducing the analytical approach.

I.4.a Analytical approach

As a reminder, we are considering the next matrix

$$\mathbf{G}_{i,j} = \int_{\Omega} \varphi_i(\mathbf{x}) \int_{\Omega} \mathcal{G}(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) d\mathbf{y} d\mathbf{x}$$

with \mathcal{G} the kernel function depending on the formulation. Considering far-interactions, the classical kernel functions can be approximated by degenerated expansions (as

(Taylor or Lagrange expansions) allowing the separation of variables as,

$$\mathcal{G}(\mathbf{x}, \mathbf{y}) \approx \tilde{\mathcal{G}}(\mathbf{x}, \mathbf{y}) = \sum_{r=0}^{k-1} u_r(\mathbf{x})v_r(\mathbf{y}). \quad (\text{I.18})$$

with u_r and v_r functions given by the expansion and the kernel. Achieving this allows to approximate the matrix \mathbf{G} as well with $\tilde{\mathcal{G}}$.

Using the approximation $\tilde{\mathcal{G}}$ instead of \mathcal{G} in the matrix \mathbf{G} gives us

$$\begin{aligned} \tilde{\mathbf{G}}_{i,j} &= \int_{\Omega} \varphi_i(\mathbf{x}) \int_{\Omega} \tilde{\mathcal{G}}(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) d\mathbf{y} d\mathbf{x}, \\ &= \sum_{r=0}^{k-1} \left(\int_{\Omega} \varphi_i(\mathbf{x}) u_r(\mathbf{x}) d\mathbf{x} \right) \left(\int_{\Omega} \varphi_j(\mathbf{y}) v_r(\mathbf{y}) d\mathbf{y} \right), \\ &= \mathbf{A}\mathbf{B}^T, \end{aligned}$$

the low-rank matrix approximation of \mathbf{G} , defined analytically and depending on the expansion of the kernel.

The HCA stands on a similar approach to separate the variables and uses the ACA to choose the proper pivot. In fact, there are two variants presented in [9], the first HCA(I) uses a Lagrange interpolation and the second HCA(II) is based on cross approximations. The HCA(II) showed better results on academic examples, so we will present only this one.

I.4.b Hybrid Cross Approximation (II)

Let assume that the kernel can be written as

$$\mathcal{G} = D_{\mathbf{x}} D_{\mathbf{y}} \gamma, \quad (\text{I.19})$$

with $\gamma : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$ a smooth generator function, $D_{\mathbf{x}}$ and $D_{\mathbf{y}}$ two partial differential operators defined as :

$$\begin{aligned} D_{\mathbf{x}} &= \langle c^{\mathbf{x}}, \nabla_{\mathbf{x}} \rangle + c_0^{\mathbf{x}}, \\ D_{\mathbf{y}} &= \langle c^{\mathbf{y}}, \nabla_{\mathbf{y}} \rangle + c_0^{\mathbf{y}}, \end{aligned}$$

with $c^{\mathbf{x}}, c^{\mathbf{y}} : \Omega \rightarrow \mathbb{R}^3$ and $c_0^{\mathbf{x}}, c_0^{\mathbf{y}} : \Omega \rightarrow \mathbb{R}$. Taking for example the single layer potential (SLP) kernel $\mathcal{G}_{SLP}(\mathbf{x}, \mathbf{y}) = (4\pi\|\mathbf{x} - \mathbf{y}\|_2)^{-1}$, satisfying (I.19) is straightforward with $\gamma = \mathcal{G}_{SLP}$ and $D_{\mathbf{x}} = D_{\mathbf{y}} = 1$.

It may seem limiting to write the kernel function as an application of two operators, but in fact, most (if not all) of the kernels can. For examples, we are able to treat electrostatic, magnetostatic and magnetodynamic formulations.

Once the decomposition (I.19) is achieved, we want to approximate γ by a skeleton function [42, 2. There is a skeleton...] of the form:

$$\tilde{\gamma}_1(\mathbf{x}, \mathbf{y}) = \frac{\gamma(\mathbf{x}, \mathbf{y}_{j_1})\gamma(\mathbf{x}_{i_1}, \mathbf{y})}{\gamma(\mathbf{x}_{i_1}, \mathbf{y}_{j_1})} \quad (\text{I.20})$$

in order to separate the variables. Since the ACA is really efficient to choose the right pivots $(\mathbf{x}_{i_q}, \mathbf{y}_{j_q})$, we want to apply it to γ . However, applying it directly to γ on a general mesh won't allow to prove the convergence since we will fall back to the previous issues. So, we get rid of the mesh and use interpolation points instead. For instance, we use the $(m+1)^3$ points \mathbf{x}_{i_p} and \mathbf{y}_{j_q} from a m^{th} tensor-product Chebyshev interpolation scheme of an admissible bounding box $B_t \times B_s$.

Then we apply the ACA on $\gamma(\mathbf{x}_{i_p}, \mathbf{y}_{j_q})$ in order to choose the pivots in the interpolation points set. Once the ACA is done and the pivots chosen, we get the approximation given by:

$$\begin{aligned} \tilde{\gamma}(\mathbf{x}, \mathbf{y}) &= \sum_{p=1}^k \sum_{q=1}^k [\gamma(\mathbf{x}, \mathbf{y}_{j_q})] [\gamma(\mathbf{x}_{i_p}, \mathbf{y}_{j_q})]^{-1} [\gamma(\mathbf{x}_{i_p}, \mathbf{y})] \\ &= \mathbf{AS}^{-1}\mathbf{B}^T \end{aligned} \quad (\text{I.21})$$

We note that we need to inverse the matrix \mathbf{S} , but a direct invert is very expensive to compute. A new variant of the ACA has been proposed in [45] allowing to construct a LU-decomposition of $\mathbf{S} = \mathbf{LU}$ *on-the-go*, where \mathbf{L} and \mathbf{U} are lower and upper triangular matrices, respectively. The final approximation is therefore given by

$$\tilde{\gamma}(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^k \left(\sum_{q=1}^l \gamma(\mathbf{x}, \mathbf{y}_{j_q}) \mathbf{U}_{l,q}^{-1} \right) \left(\sum_{q=1}^l \gamma(\mathbf{x}_{i_q}, \mathbf{y}) \mathbf{L}_{l,q}^{-1} \right), \quad (\text{I.22})$$

where \mathbf{U}^{-1} and \mathbf{L}^{-1} can be computed very efficiently.

Finally, the kernel's approximation is

$$\tilde{\mathcal{G}}(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^k \left(\sum_{q=1}^l D_{\mathbf{x}} \gamma(\mathbf{x}, \mathbf{y}_{j_q}) \mathbf{U}_{l,q}^{-1} \right) \left(\sum_{q=1}^l D_{\mathbf{y}} \gamma(\mathbf{x}_{i_q}, \mathbf{y}) \mathbf{L}_{l,q}^{-1} \right) \quad (\text{I.23})$$

and at the matrix level, we get:

$$\tilde{\mathbf{G}}_{i,j} = (\tilde{\mathbf{A}}\mathbf{U}^{-1}) (\tilde{\mathbf{B}}\mathbf{L}^{-1})^T \quad (\text{I.24})$$

with

$$\tilde{\mathbf{A}}_{i,q} = \int_{\Omega} \varphi_i(\mathbf{x}) D_{\mathbf{x}} \gamma(\mathbf{x}, \mathbf{y}_{j_q}) d\mathbf{x} \quad \text{and} \quad \tilde{\mathbf{B}}_{j,q} = \int_{\Omega} \varphi_j(\mathbf{y}) D_{\mathbf{y}} \gamma(\mathbf{x}_{i_q}, \mathbf{y}) d\mathbf{y}.$$

Algorithm 3 Hybrid Cross Approximation algorithm.

- 1: **procedure** HCA(block cluster $b = (t, s)$, error criterion ϵ , interpol. order m)
- 2: // Compute the bounding boxes of t and s
- 3: $B_t = \text{ComputeBoundingBox}(t)$; $B_s = \text{ComputeBoundingBox}(s)$
- 4: // Compute the Chebyshev's interpolation nodes of both boxes
- 5: $\mathbf{x}_{i_p} = \text{CompChebyshevNodes}(B_t)$; $\mathbf{y}_{j_q} = \text{CompChebyshevNodes}(B_s)$
- 6: // Compute the ACA for the pivots and the LU-decomposition
- 7: {Pivots $(\tilde{\mathbf{x}}_{i_r}, \tilde{\mathbf{y}}_{j_r}), r = 1, \dots, k$; Matrices \mathbf{L}, \mathbf{U} } = ACAlu($\gamma(\mathbf{x}_{i_p}, \mathbf{y}_{j_q}), \epsilon$)
- 8: // Compute the single integrals

$$\tilde{\mathbf{A}}_{i,q} = \int_{\Omega} \varphi_i(\mathbf{x}) D_{\mathbf{x}} \gamma(\mathbf{x}, \tilde{\mathbf{y}}_{j_q}) d\mathbf{x} \quad \text{and} \quad \tilde{\mathbf{B}}_{j,q} = \int_{\Omega} \varphi_i(\mathbf{y}) D_{\mathbf{y}} \gamma(\tilde{\mathbf{x}}_{i_q}, \mathbf{y}) d\mathbf{y}$$

- 9: **return** the low-rank matrix $(\tilde{\mathbf{A}}\mathbf{U}^{-1})(\tilde{\mathbf{B}}\mathbf{L}^{-1})^T$

- 10: **end procedure**

The construction of the approximation (I.24) requires the assembly of two single integrals \mathbf{A} and \mathbf{B} , instead of a twofold integral for \mathbf{G} , and point-wise evaluations of γ for the pivots' choices. The algorithm of the HCA described previously is defined in Algorithm 3.

The order of the interpolation has to be defined by the user, since it's an input of the algorithm. Setting a too high order will lead into an excess of computations, whereas a lower order may lead into inaccuracies. Some tests have been made in [39, Sec. 4.8.4] to determine approximately the accuracy ϵ in function of m , leading to $\epsilon = 10^{-m-1}$. However, it assumes m to be chosen first when we have the accuracy criterion first. Therefore, we use the same equation but backward, i.e.: assuming $\epsilon = 10^{-\tilde{p}}$ we set $m = \tilde{p} - 1$. These tests were made using Laplace single and double layer potentials, but we use the same criterion for other kernels by default.

The HCA succeeds in mixing an interpolation approach, very expensive but robust, and a cross approximation technique, really efficient with a lack of robustness. This hybrid approach has a linear complexity of $\mathcal{O}(k^2(\#\hat{t} + \#\hat{s}) + k^2m)$, depending on the interpolation order m , and allows proving its convergence [9].

Now that we have presented the tools to define the hierarchical structure and how to construct low-rank approximations, we know how to assembly a \mathcal{H} -matrix. In the next section, we introduce the essential of the \mathcal{H} -matrix arithmetic's.

I.5 \mathcal{H} -matrix arithmetic's

Putting together all the pieces, a \mathcal{H} -matrix is a block cluster tree where the non-recursive block clusters store either a classical or a low-rank matrix, depending on their admissibility. In this section, we will see an overview of all basic arithmetic operations and the LU decomposition for \mathcal{H} -matrices, the \mathcal{H} -LU decomposition.

The \mathcal{H} -matrices are usually used in the context of an iterative solution, since they significantly lower the computational costs and are able to construct efficient preconditioners. Note that it is also possible to compute a direct solution with a certain accuracy. Let us start by introducing the most intensively used operation for iterative solutions: the \mathcal{H} -matrix-vector product.

I.5.a \mathcal{H} -matrix-vector product

Let us consider the recursive block cluster $b = (t, s)$ with 4 sons which, at the matrix level, can be represented as by 2×2 block-matrix \mathbf{B} . So, the \mathcal{H} -matrix-vector product $\mathbf{B}\mathbf{x} = \mathbf{y}$ comes down to compute

$$\left[\begin{array}{c|c} \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{B}_3 & \mathbf{B}_4 \end{array} \right] \left[\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array} \right] = \left[\begin{array}{c} \mathbf{y}_1 \\ \mathbf{y}_2 \end{array} \right], \quad (\text{I.25})$$

where \mathbf{B}_i represents b_i , the i^{th} son of b . The equation (I.25) is equivalent to:

$$\begin{cases} \mathbf{y}_1 = \mathbf{B}_1\mathbf{x}_1 + \mathbf{B}_2\mathbf{x}_2 \\ \mathbf{y}_2 = \mathbf{B}_3\mathbf{x}_1 + \mathbf{B}_4\mathbf{x}_2 \end{cases}. \quad (\text{I.26})$$

The products $\mathbf{B}_i\mathbf{x}_j$ of the sons are computed depending on the admissibility or the recursion of b_i . We can find three cases:

- a) b_i is recursive, we repeat the same process for b_i , one level deeper in the hierarchy.
- b) b_i is admissible, \mathbf{B}_i is a low-rank matrix and the product can be computed directly.
- c) b_i is non-admissible, \mathbf{B}_i is a dense matrix and the product can also be computed directly.

The Algorithm 4 is straightforward, it consists in exploring the hierarchy and computing products with sub-vectors. This product has a complexity of $\mathcal{O}(n \log n)$ and for an optimized implementation, one should work directly with \mathbf{x} and \mathbf{y} sub-parts, instead of using temporary vectors at each level of recursion.

Algorithm 4 \mathcal{H} -matrix-vector product algorithm.

```

1: procedure HPROD(block cluster  $b = (t, s)$ , vector  $\mathbf{x}$ , vector  $\mathbf{y}$ )
2:   if  $b$  is recursive then
3:     for all  $b_i \in \text{sons}(b)$  do
4:       HPROD( $b_i$ ,  $\mathbf{x}$ ,  $\mathbf{y}$ )
5:     end for
6:   else
7:     if  $b$  is admissible then
8:       Compute  $\mathbf{B}\mathbf{x}|_{\hat{s}} = \mathbf{U}\mathbf{V}^T\mathbf{x}|_{\hat{s}} = \mathbf{y}|_{\hat{t}}$ 
9:     else  $\triangleright b$  is non-admissible
10:      Compute  $\mathbf{B}\mathbf{x}|_{\hat{s}} = \mathbf{y}|_{\hat{t}}$ 
11:    end if
12:  end if
13: end procedure

```

The cost of the product, as well as other arithmetic operations, obviously depends on the complexity of the hierarchy. We can find in the literature a technique to simplify the hierarchical structure using the SVD: the Agglomeration [46] (also called Coarsening).

I.5.b Agglomeration

The assembly of a \mathcal{H} -matrix can lead to a complex hierarchical structure, which significantly reduces the efficiency of such storage. A way to simplify the hierarchy is to merge/agglomerate some blocks into one non-recursive block.

The idea is to approximate (or agglomerate), with a prescribed accuracy ε_{aggl} , a recursive block cluster by one low-rank matrix as follows

$$\left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{A}_2 \\ \hline \mathbf{A}_3 & \mathbf{A}_4 \end{array} \right] \approx \mathbf{U}\mathbf{X}\mathbf{V}^T, \quad (\text{I.27})$$

under the assumption that the \mathbf{A}_i are low-rank matrices under the orthonormal outer-product form, i.e.: $\mathbf{A}_i = \mathbf{U}_i\mathbf{X}_i\mathbf{V}_i^T$. The block clusters resulting from the \mathcal{H} -matrix assembly are not under the orthonormal outer-product form, so preliminary conversions have to be computed first.

To convert non-admissible blocks into admissible blocks with a low-rank under the right form, we use a reduced (or truncated) SVD decomposition $\mathbf{M} \approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$ satisfying the prescribed accuracy $\|\mathbf{M} - \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T\| \leq \|\mathbf{M}\|_{\varepsilon_{aggl}}$. Note that we have

already presented this procedure in the [section I.1 Introduction](#).

The low-rank matrices resulting from the ACA (or the HCA) are under the outer-product form, i.e.: $\mathbf{R} = \mathbf{A}\mathbf{B}^T$. Their conversion consists in applying the SVD of low-rank matrices to \mathbf{R} , so that $\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. This SVD (sometimes called rkSVD) is described in [46, Sec. 1.1.4].

This procedure can also be seen as a recompression procedure if we truncate last SVD with the accuracy ε_{aggl} .

Once the conversions completed, we can write

$$\left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{A}_2 \\ \hline \mathbf{A}_3 & \mathbf{A}_4 \end{array} \right] = \hat{\mathbf{U}}\hat{\mathbf{X}}\hat{\mathbf{V}}^T \quad (\text{I.28})$$

with

$$\hat{\mathbf{U}} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 & & \\ & & \mathbf{U}_3 & \mathbf{U}_4 \end{bmatrix}, \hat{\mathbf{X}} = \begin{bmatrix} \cdot & \cdot & & 0 \\ & & \mathbf{X}_i & \\ & & & \cdot \\ 0 & & & \cdot \end{bmatrix} \text{ and } \hat{\mathbf{V}} = \begin{bmatrix} \mathbf{V}_1 & & \mathbf{V}_3 & \\ & \mathbf{V}_2 & & \mathbf{V}_4 \end{bmatrix}.$$

Then, we compute the following QR decompositions

$$[\mathbf{U}_1, \mathbf{U}_2] = \mathbf{Q}_1\mathbf{R}_1, \quad [\mathbf{U}_3, \mathbf{U}_4] = \mathbf{Q}_2\mathbf{R}_2, \quad [\mathbf{V}_1, \mathbf{V}_3] = \mathbf{Q}_3\mathbf{R}_3 \quad \text{and} \quad [\mathbf{V}_2, \mathbf{V}_4] = \mathbf{Q}_4\mathbf{R}_4$$

and partitioning $\mathbf{R}_i = [\mathbf{R}'_i, \mathbf{R}''_i]$ according to the ranks lead us to:

$$\hat{\mathbf{U}}\hat{\mathbf{X}}\hat{\mathbf{V}}^T = \begin{bmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{R}'_1\mathbf{X}_1\mathbf{R}'_3{}^T & \mathbf{R}''_1\mathbf{X}_2\mathbf{R}'_4{}^T \\ \mathbf{R}'_2\mathbf{X}_3\mathbf{R}''_3{}^T & \mathbf{R}''_2\mathbf{X}_4\mathbf{R}''_4{}^T \end{bmatrix}}_{\stackrel{\text{rSVD}}{\approx} \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T} \begin{bmatrix} \mathbf{Q}_3^T & \\ & \mathbf{Q}_4^T \end{bmatrix}.$$

Computing the reduced SVD on the middle matrix, with the accuracy of ε_{aggl} , finally allows us to verify (I.27) with

$$\hat{\mathbf{U}} = \begin{bmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{bmatrix} \tilde{\mathbf{U}}, \quad \hat{\mathbf{X}} = \tilde{\mathbf{\Sigma}} \quad \text{and} \quad \hat{\mathbf{V}} = \begin{bmatrix} \mathbf{Q}_3 & \\ & \mathbf{Q}_4 \end{bmatrix} \tilde{\mathbf{V}}.$$

If the approximation $\hat{\mathbf{U}}\hat{\mathbf{X}}\hat{\mathbf{V}}^T$ uses less memory than the recursive structure, we overwrite it by an admissible block cluster which contains the approximation, as shown in [Figure I.9](#). If not, we keep the actual structure and therefore stop the agglomeration process of the current branch of the hierarchy.

The agglomeration is a recursive procedure (as illustrated in [Figure I.9](#)) starting at the deepest level possible of the hierarchy, and trying to go back up to the higher level while agglomerating as many block clusters as possible.

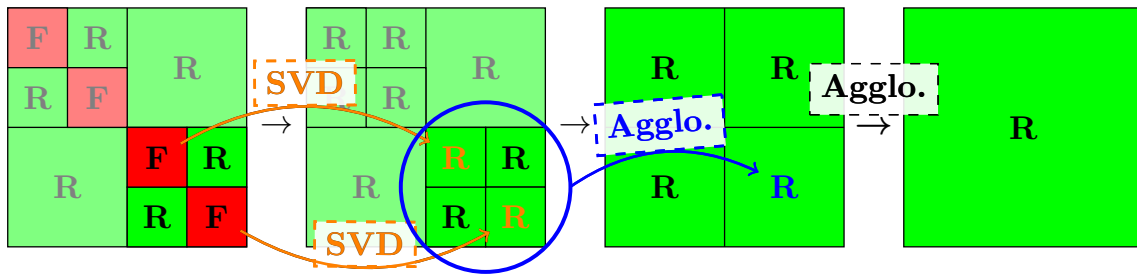


Figure I.9: Example of the agglomeration procedure on a recursive block cluster. The green blocks “**R**” and red blocks “**F**” represent admissible and non-admissible block clusters, respectively.

Let us illustrate the efficiency of the agglomeration by considering a simple example: the single layer potential on a sphere discretized with 572 triangles. The initial \mathcal{H} -matrix $\mathbf{M}_{\mathcal{H}}$ assembled using the HCA with an error $\epsilon = 10^{-4}$ is shown in Figure I.10a. Note that the numbers inside the green blocks correspond to their rank. On its right-hand side, we can see Figure I.10b showing the agglomerated \mathcal{H} -matrix $\tilde{\mathbf{M}}_{\mathcal{H}}$ with the same accuracy $\varepsilon_{aggl} = \epsilon$. The structure has obviously been greatly simplified, and the storage has reduced from 3.23 kB/dof to 2.36 kB/dof. In the figures I.10c and I.10d, we can see the evolution of the \mathcal{H} -matrix when the accuracy is deteriorated to $\varepsilon_{aggl} = 10^{-3}$ and 10^{-2} , respectively.

The agglomeration is very useful to lower the costs of the arithmetic operations, but it also plays a key role for the preconditioning, since it allows us to control the accuracy of a \mathcal{H} -matrix. We will introduce the preconditioner and see the role of the agglomeration in the next subsection.

I.5.c \mathcal{H} -LU decomposition, an efficient preconditioner

The LU decompositions (or LU factorizations) are well-known in linear algebra and have proven to be really efficient and robust for direct solution of problems, especially with multiple right-hand-side (RHS) vectors. A LU decomposition consists in factoring a matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ as

$$\begin{bmatrix} & & & \\ & & & \\ \mathbf{M} & & & \\ & & & \end{bmatrix} = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & \ddots & \\ \mathbf{L}_{i,j} & & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{U}_{1,1} & & & \mathbf{U}_{i,j} \\ & \ddots & & \\ 0 & & & \mathbf{U}_{n,n} \end{bmatrix}, \quad (\text{I.29})$$

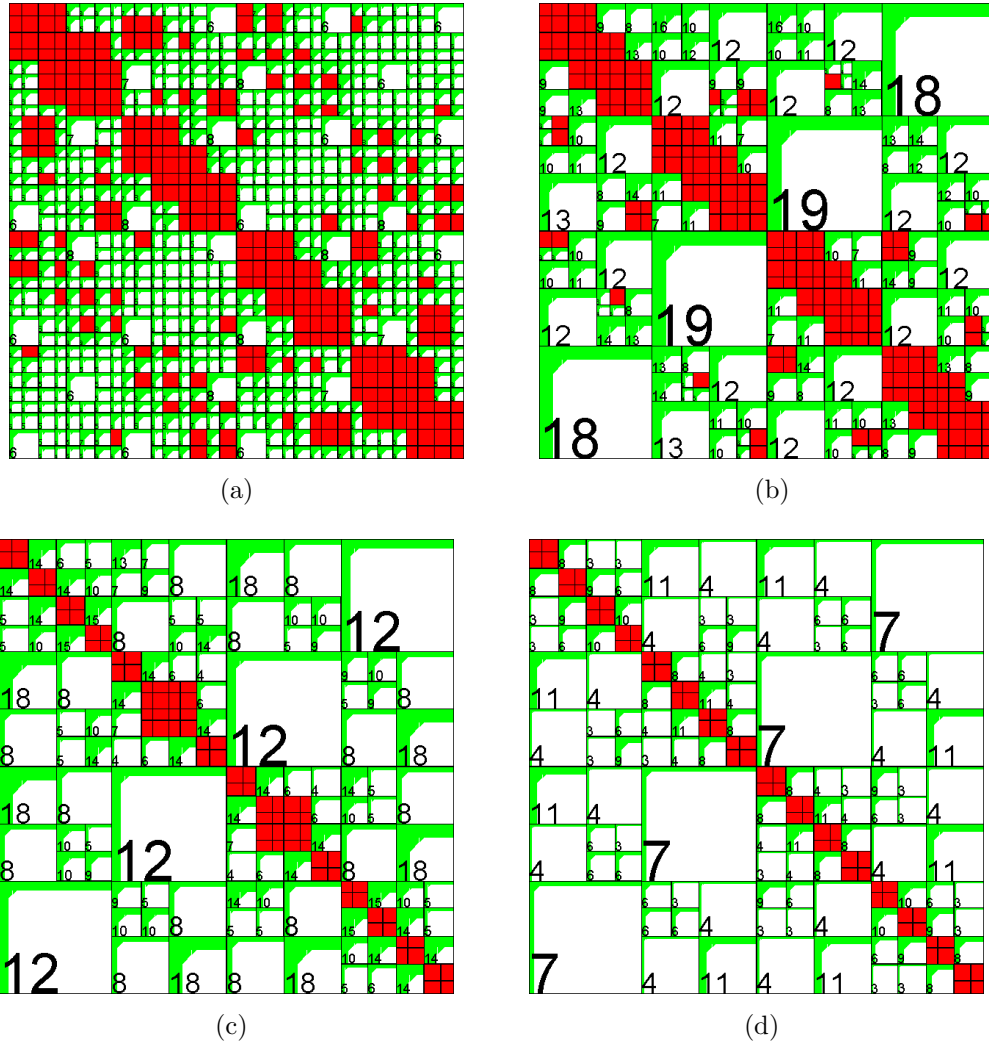


Figure I.10: (a) The \mathcal{H} -matrix resulting from the assembly with a prescribed error $\epsilon = 10^{-4}$. (b) The previous \mathcal{H} -matrix agglomerated with the same accuracy $\epsilon_{aggl} = \epsilon$. (c) Agglomeration with $\epsilon_{aggl} = 10^{-3}$. (d) Agglomeration with $\epsilon_{aggl} = 10^{-2}$.

the product of a lower and an upper triangular matrices \mathbf{L} and \mathbf{U} , respectively. The linear problem $\mathbf{M}\mathbf{x} = \mathbf{b}$ is then equivalent to

$$\begin{cases} \mathbf{L}\mathbf{y} = \mathbf{b} \\ \mathbf{U}\mathbf{x} = \mathbf{y} \end{cases}$$

which can be easily solved using forward and backward substitutions. Note that there are some variants which use partial or full pivoting in order to improve their robustness.

As most of the factorizations, the LU factorizations suffer of a very high complexity of $\mathcal{O}(n^3)$ and require all entries of the matrix. Even though this factorization is one of the most efficient and fast method to compute a direct solution, its computational

costs are still too high.

The alternative to avoid these costs is the iterative solution. We can find a few options in the literature, but the most competitive methods usually are the Krylov subspace based methods as: the Conjugate gradient (CG), the Bi-Conjugate gradient stabilized (BiCGStab) and the (Flexible-) Generalized minimum residual ((F-)GMRes). Their main asset is to be able to solve a problem using matrix-vector products only.

Keeping in mind that a classical matrix-vector product has a quadratic complexity ($\mathcal{O}(n^2)$), the gain of an iterative solver therefore depends on the number of iterations required to reach the solution. The estimation of the convergence speed is given by the conditioning of the problem, defined via the condition number of the matrix: $\kappa(\mathbf{G})$. A problem is *well-conditioned* if its condition number is low, and *ill-conditioned* if it is high. So, a problem with large condition number would converge after more iterations, which may lead to a very high cost and could even make the solution impossible to reach. That is the main reason why the conditioning of a formulation is very important.

In order to cure an ill-conditioned formulation, a preconditioning technique has to be used. The preconditioning consists in constructing a *preconditioner* \mathbf{P} , an object (matrix, decomposition, etc) to approximate the inverse of \mathbf{G} . The accuracy of the preconditioner defines its efficiency in lowering the number of iterations, in fact, if $\mathbf{P} = \mathbf{G}^{-1}$ then the solution has to be obtained instantaneously. However, constructing a very accurate inverse is very expensive in time and memory.

We have a few options when it comes to choose a preconditioner, we can find analytical preconditioners, which consist in curing the conditioning issue at the operator level (a nice recipe for operator preconditioning is presented in [47]). There are also algebraic preconditioners, which usually are decompositions. Both approaches are robust but the operator preconditioning has shown little efficiency, therefore, we will focus on the second kind of preconditioning.

As mentioned above, the algebraic preconditioner usually are complete or partial decompositions. We have started this section introducing the LU decomposition, a direct solver, but it also could be used as preconditioner if its costs were not too high. Based on the same idea, we have the incomplete LU (iLU) decompositions $\mathbf{G} \approx \tilde{\mathbf{L}}\tilde{\mathbf{U}}$ which approximate \mathbf{G} using the lower and upper triangular matrices $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{U}}$, respectively. The iLU decomposition therefore allows to approximate the inverse \mathbf{G}^{-1} using the same process as for the LU decompositions. Many variants of iLU decompositions are available, as the threshold (iLUT) or the iLUT with per-

mutations (iLUTP). They are usually applied on sparse matrices, but they can also be efficient for truncated dense matrices. Here “truncated” means that only few entries of the matrix have been wisely chosen and used for the factorization. However, the iLU factorizations have proved to be very useful since some of their parameters are used to control/limit their sparsity, and therefore their accuracy and memory consumption. The sparsity control is a powerful feature, it allows to significantly reduce the complexity and the computational costs of such decomposition.

The \mathcal{H} -matrix format also provides its LU decomposition, the \mathcal{H} -LU:

$$\begin{array}{ccc}
 \mathbf{H} & & \mathbf{L}_{\mathcal{H}} & & \mathbf{U}_{\mathcal{H}} \\
 \begin{array}{|c|c|c|} \hline \color{red}\blacksquare & \color{green}\blacksquare & \color{green}\blacksquare \\ \hline \color{green}\blacksquare & \color{red}\blacksquare & \color{green}\blacksquare \\ \hline \color{green}\blacksquare & \color{green}\blacksquare & \color{red}\blacksquare \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline \color{red}\blacksquare & & \\ \hline \color{green}\blacksquare & \color{red}\blacksquare & \\ \hline \color{green}\blacksquare & \color{green}\blacksquare & \color{red}\blacksquare \\ \hline \end{array} & & \begin{array}{|c|c|c|} \hline \color{red}\blacksquare & \color{green}\blacksquare & \color{green}\blacksquare \\ \hline & \color{red}\blacksquare & \color{green}\blacksquare \\ \hline & & \color{red}\blacksquare & \color{green}\blacksquare \\ \hline & & & \color{red}\blacksquare \\ \hline \end{array} \\
 \end{array} \tag{I.30}$$

which preserves the hierarchical structure. Its algorithm is very similar to the block-LU decomposition at the root level, the equations to solve are

$$\left[\begin{array}{c|c} \mathbf{H}_1^{(0)} & \mathbf{H}_2^{(0)} \\ \hline \mathbf{H}_3^{(0)} & \mathbf{H}_4^{(0)} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{L}_1^{(0)} & 0 \\ \hline \mathbf{L}_2^{(0)} & \mathbf{L}_3^{(0)} \end{array} \right] \left[\begin{array}{c|c} \mathbf{U}_1^{(0)} & \mathbf{U}_2^{(0)} \\ \hline 0 & \mathbf{U}_3^{(0)} \end{array} \right], \tag{I.31}$$

equivalent to:

$$\left\{ \begin{array}{l} \mathbf{H}_1^{(0)} = \mathbf{L}_1^{(0)} \mathbf{U}_1^{(0)} \\ \mathbf{H}_2^{(0)} = \mathbf{L}_1^{(0)} \mathbf{U}_2 \\ \mathbf{H}_3^{(0)} = \mathbf{L}_2^{(0)} \mathbf{U}_1 \\ \mathbf{H}_4^{(0)} = \mathbf{L}_2^{(0)} \mathbf{U}_2 + \mathbf{L}_3^{(0)} \mathbf{U}_3^{(0)} \end{array} \right. \quad \begin{array}{l} \text{gives } \mathbf{L}_1^{(0)} \text{ and } \mathbf{U}_1^{(0)} \text{ via } \mathcal{H}\text{-LU,} \\ \text{--- } \mathbf{U}_2^{(0)} \text{ --- } \mathcal{H}\text{-forward substitution,} \\ \text{--- } \mathbf{L}_2^{(0)} \text{ --- transposed ---,} \\ \text{--- } \mathbf{L}_3^{(0)} \text{ and } \mathbf{U}_3^{(0)} \text{ --- } \mathcal{H}\text{-LU.} \end{array}$$

This last system is solved using a recursive algorithm, which constructs the factorization starting at the top left corner block to the bottom right corner, following a Z-like path. The major difficulty is to preserve the hierarchical structure and admissibility of all the blocks (as shown in I.30), since we are dealing with three different formats (hierarchical, low-rank and classical matrices). The lecture note [39] provides a great support for a deeper understanding of this factorization.

One may want to compute the \mathcal{H} -LU decomposition of the \mathcal{H} -matrix resulting from the assembly directly, but the complexity of its hierarchical structure will reflect in a very high cost. The idea to reduce the costs is to compute a deteriorated, sparser decomposition as the incomplete-LU. The only tool we have to reduce the complexity of the structure and sparsify a \mathcal{H} -matrix is the agglomeration.

We saw in [subsection I.5.b Agglomeration](#), the efficiency of such technique in simplifying/sparsifying a \mathcal{H} -matrix according to a prescribed tolerance ε_{aggl} . Taking advantage of this, we want to agglomerate a copy of the \mathcal{H} -matrix which results from the assembly and then apply the \mathcal{H} -LU to this copy. The deteriorated copy will therefore be the preconditioner. We can point out that, the more we deteriorate the copy, the lower the preconditioner efficiency is. So, there is a balance to find between the deterioration/savings and the efficiency.

The demonstration of the efficiency of this preconditioner is shown in the next section, considering some examples dealing with matrices used in the next Chapters. The key role of the accuracy control will also be highlighted.

I.6 Numerical results

In this section, we are considering two examples of formulations, the first is an electrostatic formulation, where we will compare the \mathcal{H} -matrices and the Adaptive Multi-Level Fast Multipole Method (AMLFMM). The second example is a magnetostatic formulation with a spherical shell and we will see what the agglomeration brings to the solution. The efficiency of the \mathcal{H} -LU preconditioner will also be demonstrated during the previous example.

The numerical results are obtained using a laptop DELL PRECISION M4800 with an Intel(R) Core(TM) i7-4800MQ CPU @2.70GHz.

I.6.a Electrostatic example

Let us consider an electrostatic equation on a sphere S (see [Figure I.11](#)) with a standard Galerkin discretization of the SLP

$$\frac{1}{4\pi} \sum_j \left(\int_S \int_S \frac{\varphi_i(\mathbf{x})\varphi_j(\mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|} d\mathbf{y}d\mathbf{x} \right) q_j = \int_S \varphi_i(\mathbf{x})V_0(\mathbf{x})d\mathbf{x}, \quad \forall i, \quad (\text{I.32})$$

where φ_j are piece-wise constant functions, q_j are the charge density unknowns and V_0 a real constant.

To compare the FMM and the \mathcal{H} -matrices, in the assembly step we set ϵ , the relative error in Froebenius norm for each block, to 10^{-4} so that both methods provide an equivalent accuracy. The relative norm of the residual in the iterative solver GMRes has to be lower than 10^{-6} . The histogram in [Figure I.12](#) shows the assembly and solution times for both methods.

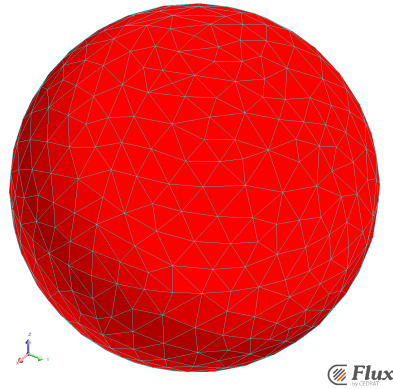
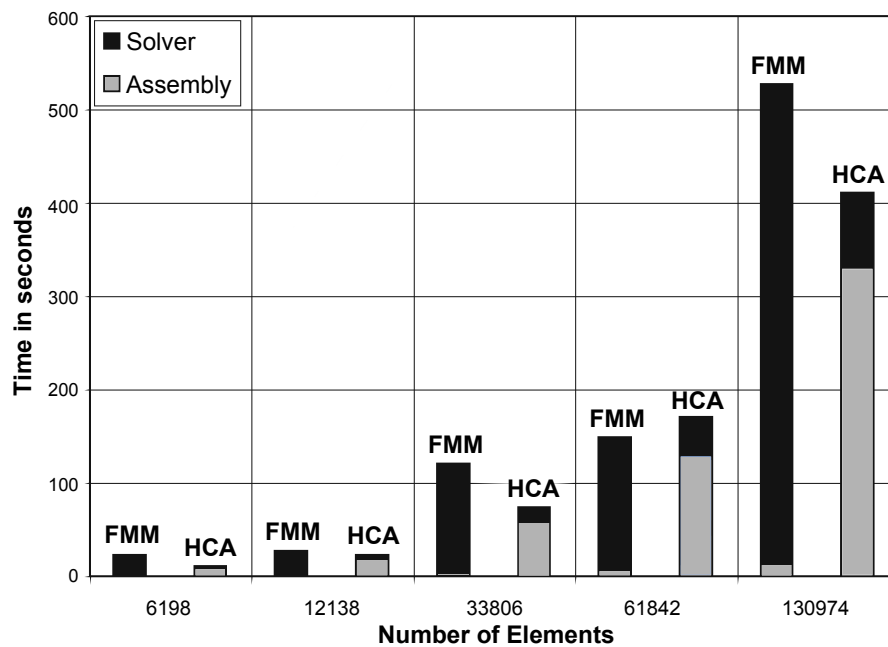


Figure I.11: Example of a meshed sphere.

Figure I.12: Comparison of the FMM and \mathcal{H} -matrix times.

We can see that the \mathcal{H} -matrix is faster than the FMM on almost all cases. The assembly of \mathcal{H} -matrices takes more time. In return, the FMM's iterations are about 10 times slower than the \mathcal{H} -matrix's, because the chosen FMM variant consists in storing only the near-interactions and computing the far-interactions on-the-go, during each product. This variant allows us to use little memory as shown in Figure I.13. Note also that the \mathcal{H} -matrix storage (here in kB/DOF) is close to the theoretical logarithmic complexity.

For these results we did not use the \mathcal{H} -LU preconditioning of the \mathcal{H} -matrices because it is a well-conditioned problem. The next example will highlight the efficiency

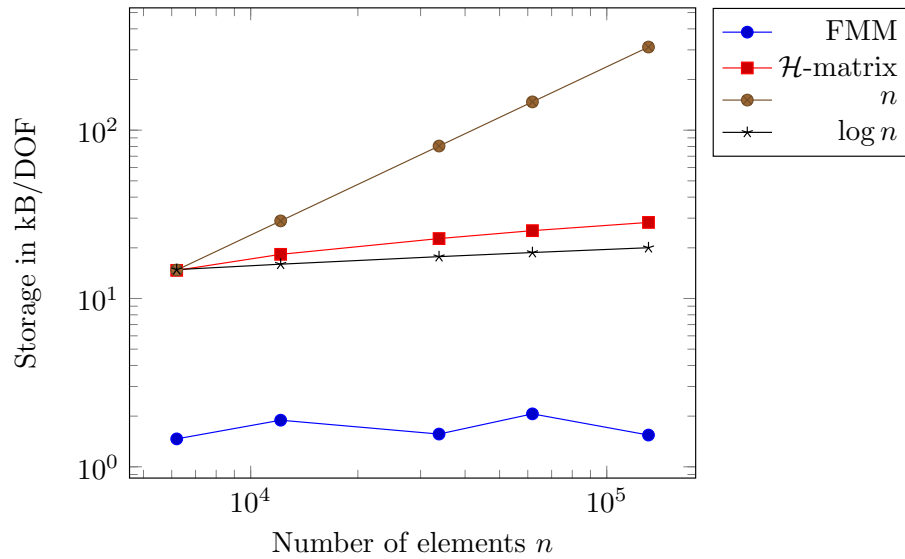


Figure I.13: Storage requirements.

of the \mathcal{H} -LU preconditioner.

I.6.b Magnetostatic formulation

Let us consider the ferromagnetic behavior law

$$\mathbf{M}(\mathbf{x}) = \chi(\mathbf{x}, \mathbf{H})\mathbf{H}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad (\text{I.33})$$

with \mathbf{M} the magnetization, \mathbf{H} the magnetic field and Ω an isotropic ferromagnetic material with χ its magnetic susceptibility. We define χ as a constant for the next experiments. The magnetic field can be written as

$$\mathbf{H} = \mathbf{H}_{\text{red}} + \mathbf{H}_0$$

with \mathbf{H}_{red} the reduced magnetic field created by the ferromagnetic material and \mathbf{H}_0 the magnetic source field created by current flows.

We consider the following integral equation [48] using the total scalar potential Φ

$$\Phi(\mathbf{x}) + \frac{1}{4\pi} \int_{\Omega} \chi(\mathbf{y}, \mathbf{H}) \frac{\nabla \Phi(\mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|^3} d\mathbf{y} = \Phi_0(\mathbf{x}) \quad (\text{I.34})$$

with Φ_0 the scalar potential deriving from the magnetic field \mathbf{H}_0 . Using a P_1 finite element approximation for Φ and the Galerkin discretization leads to the linear system

$$(\mathbf{M} + \mathbf{G}) \Phi = \mathbf{D} \quad (\text{I.35})$$

with the mass matrix and the right-hand-side vector

$$\mathbf{M}_{i,j} = \int_{\Omega} \varphi_i(\mathbf{x})\varphi_j(\mathbf{x})d\mathbf{x}, \quad \mathbf{D}_i = \int_{\Omega} \varphi_i(\mathbf{x})\Phi_0(\mathbf{x})d\mathbf{x}, \quad (\text{I.36})$$

and the volume integral

$$\mathbf{G}_{i,j} = \frac{1}{4\pi} \int_{\Omega} \varphi_i(\mathbf{x}) \int_{\Omega} \chi \nabla \varphi_j(\mathbf{y}) \cdot \nabla_{\mathbf{y}} \frac{1}{\|\mathbf{x} - \mathbf{y}\|} d\mathbf{y} d\mathbf{x}. \quad (\text{I.37})$$

Solving (I.35) permits to compute the magnetic field \mathbf{H} in the air [49]. The matrix \mathbf{M} is a finite element matrix with a sparse storage, so the limitation here is the dense matrix \mathbf{G} .

HCA degenerate expansion

In order to apply the HCA to \mathbf{G} , we need the degenerate expansion of its kernel. In (I.37) the kernel function $\mathcal{G}(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{y}} \frac{1}{\|\mathbf{x} - \mathbf{y}\|}$ enables the identifications

$$\gamma(\mathbf{x}, \mathbf{y}) = \frac{1}{\|\mathbf{x} - \mathbf{y}\|}, \quad D_{\mathbf{x}} = \text{Id}, \quad D_{\mathbf{y}} = \nabla_{\mathbf{y}},$$

in order to satisfy (I.19). The two single integrals to compute are then

$$\int_{\Omega} \frac{\varphi_i(\mathbf{x})}{\|\mathbf{x} - \tilde{\mathbf{y}}_{ji}\|} d\mathbf{x} \quad \text{and} \quad \int_{\Omega} \chi \nabla \varphi_j(\mathbf{y}) \cdot \nabla_{\mathbf{y}} \frac{1}{\|\tilde{\mathbf{x}}_{iq} - \mathbf{y}\|} d\mathbf{y}.$$

The computational cost of these integrals is proportional to the number of quadrature points per element instead of the quadratic cost of the two-fold integral, allowing us to have a lower complexity than the ACA for the same rank.

Numerical results

Let us consider a spherical shell meshed with tetrahedra (a cut of the solution is shown in Figure I.14), we will solve (I.35) and focus our attention on two different points. We will be focusing on the impact factor of the agglomeration, for assemblies using ACA and HCA. Then, we will focus on the \mathcal{H} -LU performances.

Study case 1: Keeping the same parameters as for the electrostatic example, we are assembling \mathcal{H} -matrices using the ACA and HCA on 3 meshes of different size. The computational data are presented in Table I.1, as well as the accuracy of the results. The accuracy (column “Error”) is computed comparing the results of each method against those obtained with the classical/full approach. We can see that the HCA performs faster than the ACA regarding the assembly of an \mathcal{H} -matrix,

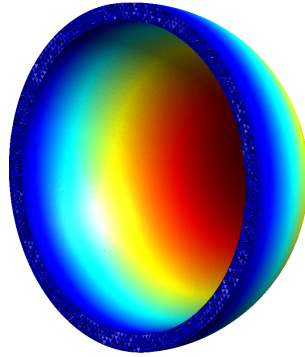


Figure I.14: Isovalues of the solution on a spherical shell.

but in counterpart, the HCA requires more memory. These results follow the same behavior to those found in the literature, which also treat spherical geometries for other integral equations.

Table I.1: Computing times, storage and error without agglomeration.

	NbDOF	Assembly (s)	Solver (s)	Storage (kb/DOF)	Error
A	3,846	718	0.5	15.4	1.53%
C	12,938	7,321	10	24.3	0.85%
A	48,154	56,112	89	37.6	0.37%
H	3,846	278	0.6	18.3	5.23%
C	12,938	2,333	12	29.4	1.33%
A	48,154	18,154	109	44.9	1.42%

These memory requirements can be reduced using the algebraic re-compression technique previously presented, the agglomeration, while keeping the same accuracy. This technique does not change the asymptotic complexity of the assembly step, and it allows us to reduce the complexity of the arithmetic operations.

The same procedure is repeated with, in addition, the agglomeration directly applied after the assembly. The new computational data are presented in Table I.2. The assembly times have only slightly increased (about 3-4%) compared to Table I.1, but in counterpart, both methods now have an equivalent and reduced storage. Therefore, the efficiency of the HCA rose significantly compared to the ACA, providing an assembly about three times faster for the same memory consumption and about the same accuracy. Note that the initial block compression is required to be slightly more accurate before recompression compared to experiments in Table I.1. According to [50], in order to ensure the whole assembly satisfies a tolerance ϵ , the ACA (or HCA) has to be computed with an accuracy of $\epsilon/20$ and the recompression with

a tolerance of $(2 + \epsilon)^{-1}$. This is why the error can be lower in Table I.2 compared to Table I.1. We can also note very small improvements in the column “Solver”, because the solutions were already very fast.

Table I.2: Computing times, storage and error with agglomeration.

	NbDOF	Assembly (s)	Solver (s)	Storage (kB/DOF)	Error
A	3,846	952	0.5	7.0	1.66%
C	12,938	9,774	10	10.3	0.27%
A	48,154	74,604	95	14.9	0.45%
H	3,846	287	0.5	7.0	1.63%
C	12,938	2,586	12	10.4	0.54%
A	48,154	18,876	98	14.9	0.72%

To conclude this analysis, the agglomeration provides a great combination with the HCA, reducing drastically the storage for a very insignificant cost.

Study case 2: Based on the last results, we will be using the HCA for this case and now focusing on the preconditioning. Using the same meshes, we will compare the solutions’ data of multiple cases using the \mathcal{H} -LU preconditioners with different tolerances.

The computational data are presented in Table I.3, we have considered three cases using the \mathcal{H} -LU decomposition with the next accuracies $\varepsilon_{LU} = 0.3, 0.1, 0.01$. We want to specifically look at the number of iterations taken to reach the solution “NbIt”, and the time taken by the solver “T_{Solv}”, the agglomerations “T_{Agglo}” and the \mathcal{H} -LU decomposition “T_{HLU}”. We will compare these data to those from a solution without preconditioner, in the columns “No Prec.”.

Looking at the results, the balance between the efficiency, i.e.: the number of iteration reduction, and the preconditioner’s accuracy is obvious. The more agglomerated is the preconditioner, the more iterations are required. On the other hand, we clearly see the impact of the agglomeration on the \mathcal{H} -LU decompositions. As we previously mentioned, the agglomeration reduces the costs of arithmetic operations, thus, the results clearly show a significant decrease of the \mathcal{H} -LU’s costs.

Focusing on the case with 12,938 DOFs, the agglomeration took 23.46 seconds which led to a \mathcal{H} -LU taking 19.47 sec. for an accuracy of 0.01; whereas, for a tolerance of 0.3, the agglomeration is more expensive (34.96 sec.) but the \mathcal{H} -LU time has decreased by a factor 10 (1.91 sec.). As a result, the preconditioner construction is

usually faster for a low accuracy (0.3 is often used) than for a high accuracy.

On the other hand, the solver will require more iterations, so, the choice of the accuracy also depends on the number of right-hand-side vectors, directly connected to the number of solutions required to solve the whole problem. Increasing the accuracy reduces the number of iterations, therefore, it should shorten the time taken by the solver. So, a problem with multiple right-hand-side vectors should use a higher accuracy than one with a single RHS vector.

To conclude this study, we summed the times of each case and gathered them in Table I.4. We have pointed out the balance with which we have to play, between the agglomeration and the \mathcal{H} -LU cost, let now focus on the time taken to solve the problem. First of all, we can emphasize the fact that the solution without preconditioner already converges very fast, so this problem may not highlight the efficiency of the \mathcal{H} -LU. However, the column “0.1” shows an improvement for the larger mesh, whereas for the other meshes, the time taken by the solution is pretty similar to those from the non-preconditioned column. The other two cases (0.3 and 0.01) do not bring any improvements, the case 0.3 took too much time to agglomerate the preconditioner, whereas the case 0.01 spent too much time computing the \mathcal{H} -LU. So, for this example, a preconditioner with an accuracy of 10% shows the best balance.

I.7 Conclusion

In this chapter, we wanted to present a powerful matrix compression technique, the \mathcal{H} -matrices, that we will be using afterward in the thesis, as a compression and preconditioning technique. We didn’t bring any contribution to this great tool, so we intended to give a brief view of the tool and the theory behind, so the reader could understand the computational challenges we will be facing in the next chapters.

We have started with a presentation of the essential components to a \mathcal{H} -matrix, starting with the hierarchy construction for general n-D geometries and the compression techniques. We have introduced two existing compression techniques based on low-rank approximations: the ACA, an efficient algebraic method suffering of a lack of robustness, and the HCA, a hybrid technique which combines the ACA and an analytical approach, in order to cure the issues of the ACA.

Once the main two components are presented, the assembly of a \mathcal{H} -matrix is straightforward. We could therefore present the indispensable arithmetic operations to solve a problem. With the aim of a preconditioned iterative solutions, we have

Table I.3: Computational times (in seconds) and the number of iterations are gathered for 3 preconditioned cases with different accuracy and one non-preconditioned case.

NbDOF	0.3			0.1			0.01			No Prec.					
	NbIt	T _{Solv}	T _{Agglo}	T _{HLU}	NbIt	T _{Solv}	T _{Agglo}	T _{HLU}	NbIt	T _{Solv}	T _{Agglo}	T _{HLU}	NbIt	T _{Solv}	T _{Agglo}
3,846	17	0.47	5.24	2.1	9	0.32	5.48	1.37	5	0.21	7.53	3.09	47	0.82	3.56
12,938	24	2.47	34.96	1.91	16	1.35	27.75	4.9	12	1.19	23.46	19.47	214	11.72	20.62
45,154	33	14.82	226.68	9.45	23	16.15	120.88	33.4	42	20.9	150.75	115.77	340	114.88	89

Table I.4: Total time in seconds taken by the solution of each case.

NbDOF	0.3	0.1	0.01	NoPrec.
3,846	7.81	7.17	10.83	4.38
12,938	39.34	34.0	44.12	32.34
45,154	250.95	170.43	287.42	203.88

presented the \mathcal{H} -matrix-vector product and the crucial elements for the preconditioner construction. The preconditioner stands on the combination of the \mathcal{H} -LU decomposition and the agglomeration, allowing to deteriorate the preconditioner with a control on the accuracy. The numerical tests have highlighted the pertinence of such control.

The first example consisted in comparing the efficiency of the \mathcal{H} -matrix and the FMM to solve an electrostatic problem. The results showed that the \mathcal{H} -matrices are usually faster for the whole solution, even without preconditioner, but the FMM requires less memory.

Then, we focused on the construction of the low-rank approximations and we compared the ACA and HCA considering two cases. The ACA did not fail for this formulation and the considered examples. Comparing the ACA and HCA, we showed that the HCA is faster but requires a larger storage. Adding an agglomeration step, the storages become equivalent for both methods. At the end the HCA, seems more efficient and reliable than the ACA for this formulation.

Finally, we studied the same examples and we focused our attention on the construction of the preconditioner. Three different accuracies of preconditioner have been computed and we compared the results against the case without preconditioner. We clearly saw the efficiency of the preconditioner, via the reduction of number of iterations. We have also monitored the time spent in the agglomeration, and we highlighted a correlation between the agglomeration and the \mathcal{H} -LU costs.

Next, we will study similar formulations to the ones we are aiming and we will focus our attention on their preconditioning, based on \mathcal{H} -matrices.

Chapter II

On the preconditioning of eddy current formulations

On ne fait jamais attention à ce qui a été fait ; on ne voit que ce qui reste à faire.

Marie Curie Sklodowska

CONTENTS

II.1 INTRODUCTION	49
II.2 INDUCTIVE PEEC FORMULATION	50
II.2.a PEEC Formulation	51
II.2.b Shell PEEC-formulation	59
II.2.c Preconditioning study	62
II.3 \mathbf{T} FORMULATION	68
II.3.a Integral formulation	69
II.3.b Circuit approach	71
II.3.c Electrical circuit coupling	73
II.3.d Preconditioning	74
II.4 CONCLUSION	77

Resume

In this chapter, we will focus on two different types of “simple” formulations, in the sense that they consider conductors with resistive and inductive effects only. These formulations emerged from the different approaches we spoke about in the section [A quick survey of methods for low frequencies](#). First, we will present the formulation from the circuit approach: the inductive PEEC, and then the \mathbf{T} formulation, from the “pure” integral equations approach. We will see that they are very similar formulations, and taking advantage of that,

we will see how to add circuit elements in the last formulation. In the meantime, we will study the potential of preconditioning for each formulation.

II.1 Introduction

As a remainder, we aim to model problems containing conductors and dielectrics, taking in consideration the resistive, inductive and capacitive effects with a circuit coupling. Therefore, at low-frequencies, where the propagation is negligible, we provide a full electromagnetic solution, i.e.: the electric and magnetic effects are taken into account. But for a matter of understanding, we decided to introduce simpler formulations first, where we can directly use the \mathcal{H} -matrices as compression and hopefully as preconditioner.

This chapter arbitrarily takes place in the context of very low-frequencies applications, where the propagation effects can be neglected as well as the capacitive effects. The electromagnetic problems are then resumed in modeling resistive and inductive effects only. Since the capacitive effects are neglected, we can also neglect any dielectrics. So, the problems are reduced to eddy current problems where we consider volume and/or thin conductors only, see Figure II.1.

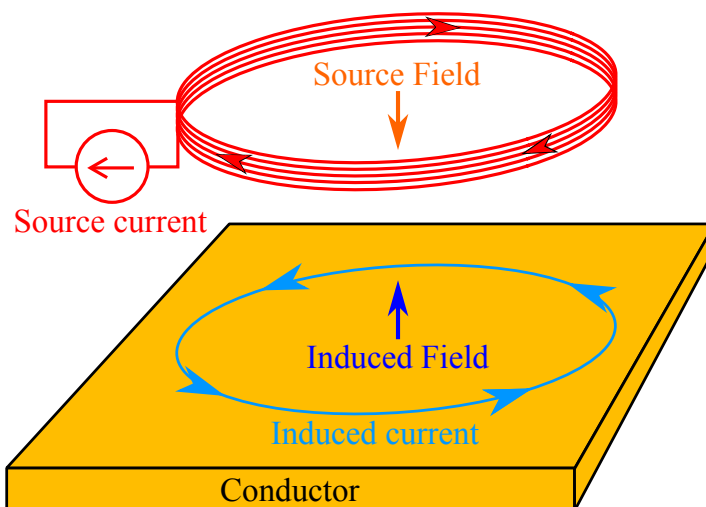


Figure II.1: Eddy current problem with an inductor and a conductive plate.

These kinds of problems have been widely studied in the past, so various formulations are available in the literature. They are usually employed for applications as eddy-current testings for nondestructive analysis, wireless power transfer by induction, electromagnetic braking, etc.

The method that we intend to use for the more complicated formulation with dielectrics is based on the PEEC method. We will therefore be studying an inductive PEEC formulation based on the current density \mathbf{J} , as a first step, in order to provide a better understanding of the difficulties encountered during the PhD.

Secondly, we will present the \mathbf{T} formulation and see if it could allow us to overcome previous difficulties.

II.2 Inductive PEEC Formulation

The Partial Element Equivalent Circuit method has proven to be very efficient solving interconnected problems with a need of external circuit coupling. The circuit coupling is essential for numerous applications, to model the supplies or to add passive circuit elements (RLC). The efficiency of the PEEC to solve problems with a circuit approach mainly comes from the fact that it uses an equivalent circuit representation of the problem.

We have previously discussed in the [section A quick survey of methods for low frequencies](#) about the classical PEEC [21, 22, 28], which suffers of important limitations, such as the exclusive use of hexahedral meshes and the current's interpolation of low-order (piece-wise constant). The last limitation implies having 1-directional elements as in Figure II.2, which allow the current to flow along only one direction (usually the dimension of the larger size). There are 2D elements as well, combining two 1D elements to allow the current to follow two directions, along the larger and longer dimension.

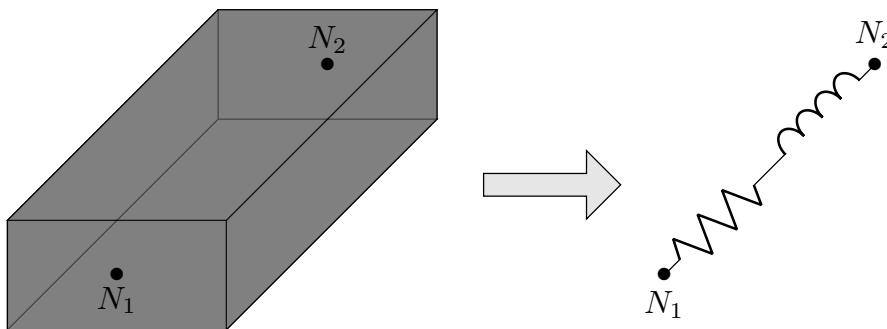


Figure II.2: Equivalent circuit representation of a 1D inductive PEEC-element.

For the eddy current problems, using these 1D or 2D PEEC-elements may lead to inaccurate solution, since the induced currents are more likely to form circular shapes that are not uniform across any plane. Another PEEC approach, using different PEEC elements, therefore has to be used. We are briefly presenting previous works done in the G2Elab that overcome these limitations [51–54].

II.2.a PEEC Formulation

In reality the current flows through conductors following a continuous path, but in a discretized space, we can't exactly represent such current. The most reliable and natural way to approach this reality would be to let the current flows in or out of an element of the mesh. Using the faces of the mesh as a support of the current seems the most natural way to represent the current flowing. After all, the supply injections are usually connected through some faces of the device.

The next formulation uses face element to achieve such representation. This work comes from the Ph.D. of T.T. Nguyen [51].

Maxwell's equations

Let us start by the development of the Maxwell's equations in order to reach the PEEC integral equation corresponding to face element discretization.

In our context the Maxwell's equations [55] are

$$\begin{cases} \mathbf{curl}\mathbf{H} = \mathbf{J}, & \text{(II.1)} \\ \mathbf{curl}\mathbf{E} = -j\omega\mathbf{B}, & \text{(II.2)} \\ \mathbf{div}\mathbf{B} = 0, & \text{(II.3)} \end{cases}$$

with the following constitutive laws

$$\begin{cases} \mathbf{J} = \sigma\mathbf{E}, & \text{(II.4)} \\ \mathbf{B} = \mu_0\mathbf{H}. & \text{(II.5)} \end{cases}$$

From (II.3) we can define a vector potential \mathbf{A} such as $\mathbf{B} = \mathbf{curl}\mathbf{A}$. Injecting the last equation in (II.2) also allows us to define a scalar potential V satisfying:

$$\mathbf{E} = -j\omega\mathbf{A} - \mathbf{grad}V. \quad \text{(II.6)}$$

Without any ferromagnetic matter, *Biot et Savart* law gives

$$\mathbf{A} = \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}}{r} d\Omega, \quad \text{(II.7)}$$

with r the distance between the testing (where \mathbf{A} is expressed) and integration points. So, (II.6) becomes:

$$\mathbf{E} = -j\omega \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}}{r} d\Omega - \mathbf{grad}V. \quad \text{(II.8)}$$

The problem finally consists in matching the last equation with the first constitutive law (II.4), leading to

$$\frac{\mathbf{J}}{\sigma} + j\omega \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}}{r} d\Omega = -\mathbf{grad}V. \quad \text{(II.9)}$$

Discretization

As mentioned previously, we want to use the face elements to discretize the currents, our unknowns, for a better approximation of the reality. But, this also allows us to overcome the other limitation of the classical PEEC, enabling the use of a general mesh (triangular, quadrangular, tetrahedral, etc).

We are then discretizing the current density vector using the \mathbf{w}_i face shape functions (see Figure II.3), which are Whitney elements of second order [56] (also called face elements), as follows:

$$\mathbf{J} = \sum_k \mathbf{w}_k I_k, \quad (\text{II.10})$$

with I_k the current flowing through the face k . The face elements are well-suited to interpolate current density, since they preserve the normal components of the \mathbf{J} between elements, therefore allowing to naturally represent the current flowing.

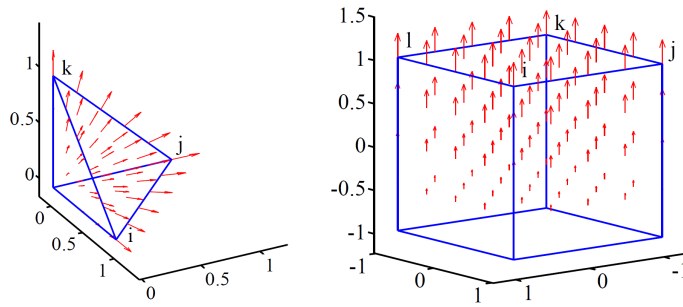


Figure II.3: Face shape functions

We finally apply a standard Galerkin discretization scheme to (II.9) using the face elements, which leads to the next system under its matrix form:

$$(\mathbf{R} + j\omega\mathbf{L}) \mathbf{I} = \mathbf{U}, \quad (\text{II.11})$$

with

$$\begin{cases} \mathbf{R}_{i,j} = \int_{\Omega} \frac{\mathbf{w}_i \mathbf{w}_j}{\sigma} d\Omega, & (\text{II.12}) \\ \mathbf{L}_{i,j} = \int_{\Omega} \mathbf{w}_i \int_{\Omega} \frac{\mathbf{w}_j}{r} d\Omega d\Omega, & (\text{II.13}) \\ \mathbf{U}_i = - \int_{\Omega} \mathbf{w}_i \mathbf{grad} V d\Omega. & (\text{II.14}) \end{cases}$$

In addition, we impose the current not to flow outside of the conductors imposing:

$$\mathbf{J} \cdot \mathbf{n} = 0$$

on the border faces that are not connected to the external circuit. The problem is finally defined as a basic circuit equation: $\mathbf{Z}\mathbf{I} = \mathbf{U}$, composed of resistances and

inductances. Now, we have to define the circuit representation associated with such discretization.

Equivalent circuit representation

In order to know how to construct the equivalent circuit, we are investigating the meaning of \mathbf{U} in (II.11). Let us start by using the Divergence Theorem:

$$-\mathbf{U}_i = \int_{\Omega} \mathbf{w}_i \mathbf{grad} V d\Omega = \int_{\Gamma} (\mathbf{w}_i \cdot \mathbf{n}) V d\Gamma - \int_{\Omega} (\text{div} \mathbf{w}_i) V d\Omega \quad (\text{II.15})$$

Considering an inner face, we have

$$-\mathbf{U}_i = \int_{\Omega} (\text{div} \mathbf{w}_i) V d\Omega \quad (\text{II.16})$$

and we know that the support of the face element \mathbf{w}_i is shared by two elements (e_1 and e_2), so we have

$$\mathbf{U}_i = \int_{e_1} \frac{V}{\text{Vol}(e_1)} d\Omega - \int_{e_2} \frac{V}{\text{Vol}(e_2)} d\Omega \quad (\text{II.17})$$

knowing that $\text{div} \mathbf{w}_i = \text{Vol}(e_i)^{-1}$, with $\text{Vol}(e_i)$ giving the volume of the element e_i . We can see that (II.17) represents the difference of averaged potential between two inner elements. So, from a circuit approach, we can assimilate this as the voltage of a branch linking two elements, as the black lines in Figure II.4.

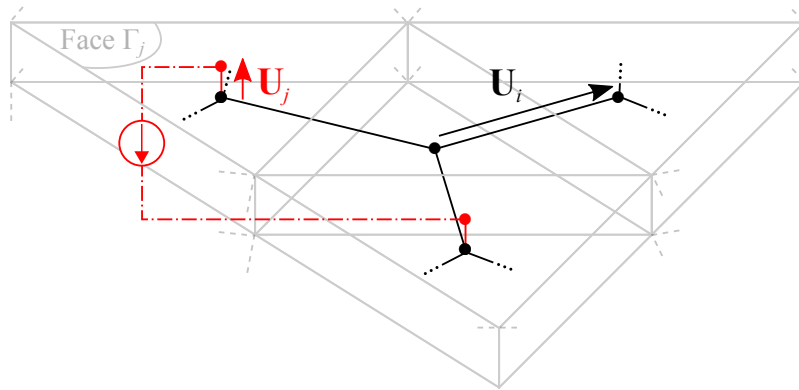


Figure II.4: Shows the difference of averaged potential between elements (in black) and between a border face and its inner volume element (in red). The external circuit is in red and discontinuous lines.

Let's now consider a face j on the border. Its support is only on one element (e_1 for example), so (II.15) gives

$$-\mathbf{U}_j = \int_{\Gamma} (\mathbf{w}_j \cdot \mathbf{n}) V d\Gamma - \int_{e_1} \frac{V}{\text{Vol}(e_1)} d\Omega \quad (\text{II.18})$$

and knowing that $(\mathbf{w}_j \cdot \mathbf{n}) = S_j^{-1}$, with S_j the surface of the face j , we have

$$-\mathbf{U}_j = \int_{\Gamma_j} \frac{V}{S_j} d\Gamma - \int_{\epsilon_1} \frac{V}{\text{Vol}(e_1)} d\Omega \quad (\text{II.19})$$

also representing the difference of potential between a border face and its inner support, as the red lines in Figure II.4.

So, the right-hand-side vector is the potential difference (or voltage) on the circuit branches, in black and red continuous lines in Figure II.4, the circuit topology is in fact the dual mesh.

Now that the branches are placed and the topology is set, we have to investigate of what the branches are composed. In (II.11), the unknowns are the currents flowing through the faces, which are represented by the circuit branches. When looking at the equation (II.11), more precisely at the matrices, we see that the model of each face is composed of one resistance and one inductance. The equivalent circuit representation therefore consists of branches, containing R and L lumped elements, which link one element of the mesh to its neighbors throughout the faces they share. For example, let us take the last mesh (in Figure II.4) and focus the middle prism. In Figure II.5, the middle prism P_0 (in black) shares three of its faces with its neighbors (P_1 , P_2 and P_3 in gray), whereas the top and bottom faces are on the boundary. So, three RL-branches connect N_0 , the barycenter of P_0 , to the barycenters of its neighbors. Note that we chose this example for a matter of understanding, but this principle can be easily and directly transposed for other meshes and types of elements. Like, for example, the rectangles that are now used in InCa3D software.

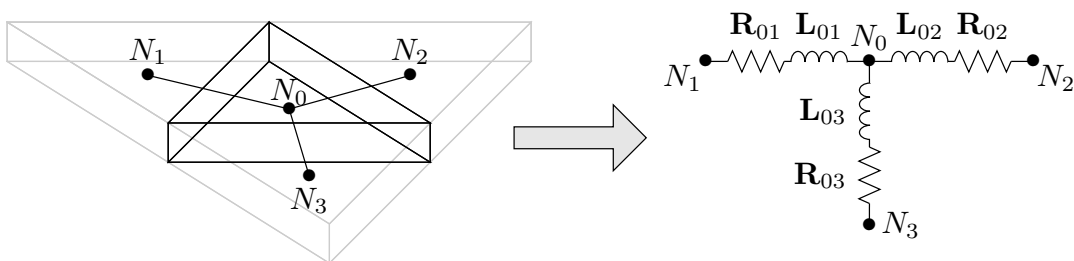


Figure II.5: Equivalent circuit of a mesh-element using a ‘face discretization’.

In the general case, the circuit topology is equivalent to the dual mesh, see Figure II.6 for example, where the dual edges are RL-branches and the dual vertices are the circuit nodes.

Circuit solver

We have presented the PEEC integral equations and its equivalent circuit represen-

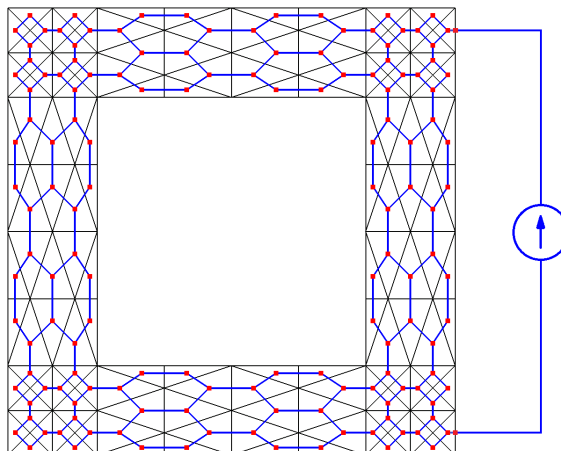


Figure II.6: The primal mesh (in black) is a conductive loop meshed with triangles and the dual mesh is in blue. Each blue edge corresponds to one RL-branch.

tation, but we want to solve problems with external electrical circuits. The solution therefore can be computed using a circuit solver, which role's is to solve electrical circuit problems and thus ensure the current density solenoidality. Such solver only need two inputs: the circuit topology, providing the informations about the connections of the branches and their circuit nature (R, L, C, sources, etc.). The second input is the set of matrices which defines the value of each branch, for our inductive case, we have to provide the complex matrix: $\mathbf{R} + j\omega\mathbf{L}$, from (II.11). Once the circuit solver has these informations, the solution can be computed.

The circuit solver consists in reformulating the problem, which is in the branches basis, into another basis in order to satisfy Kirchhoff's laws.

We can typically find two types of circuit solvers, which use different basis. There also exists hybrid approaches that will not be considered here. The first type of solver uses the first Kirchhoff's law (the Kirchhoff's current law), also called the nodal approach/analysis because it transfers the equation (II.11) into an independent node basis. This approach is probably the most popular in the literature [57]. The last type of solver uses the second Kirchhoff's law (Kirchhoff's voltage law), which uses the independent loop basis. We are presenting briefly both approaches:

(a) Nodal analysis.

Based on the circuit topology, given by the equivalent circuit representation, the nodal analysis consists in writing the circuit equations in an independent node basis. The independent nodes are a subset of the initial circuit nodes, where one node by independent circuit has been removed to be consider as the reference. The change-of-basis is represented via an incidence matrix \mathbf{M}^\bullet , transferring the equations written

in the branches basis into the independent node basis. The incidence matrix \mathbf{M}^\bullet has to fulfill the next condition:

$$\mathbf{M}^\bullet_{i,j} = \begin{cases} 0 & \text{if the branch } j \text{ is not linked to the node } i, \\ 1 & \text{if the branch } j \text{ is oriented towards the node } i, \\ -1 & \text{if the branch } j \text{ is not oriented towards the node } i. \end{cases}$$

Note that the incidence matrix is a very light sparse-matrix. The equivalent circuit is composed of R and L elements (as in Figure II.5), but to add the external circuit we have to take into account the new branches that can be composed of current and voltage supplies and RLC passive elements. Taking in consideration these external branches, the incidence matrix can be written as:

$$\mathbf{M}^\bullet = \begin{array}{c} \text{loops} \\ \left[\begin{array}{c|c|c} \Omega + \text{RLC} & \text{I Src.} & \text{U Src.} \\ \hline \mathbf{M}^\bullet_{\mathbf{Z}} & \mathbf{M}^\bullet_{\mathbf{I}} & \mathbf{M}^\bullet_{\mathbf{U}} \\ \hline \end{array} \right] \end{array} \quad (\text{II.20})$$

← branches →

if the branches are ordered properly. The matrix $\mathbf{M}^\bullet_{\mathbf{Z}}$ corresponds to the branches containing the equivalent circuit and the passive elements (R,L or C) added from the external circuit. Whereas $\mathbf{M}^\bullet_{\mathbf{I}}$, $\mathbf{M}^\bullet_{\mathbf{U}}$ are the incidence matrices restricted to the branches that contain the current and voltage supplies, respectively.

The ordering used here is entirely arbitrary. One can use another strategy, but for a matter of ease, the branches should be grouped by types.

Using the first Kirchhoff's law can lead to the next system of equations:

$$\begin{bmatrix} \mathbf{Z}_{\text{bz}} & \mathbf{M}^{\bullet T}_{\mathbf{Z}} & 0 & 0 \\ \mathbf{M}^\bullet_{\mathbf{Z}} & 0 & \mathbf{M}^\bullet_{\mathbf{U}} & 0 \\ 0 & \mathbf{M}^{\bullet T}_{\mathbf{U}} & 0 & 0 \\ 0 & \mathbf{M}^{\bullet T}_{\mathbf{I}} & 0 & \mathbf{I}_d \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ V_N \\ \mathbf{U}_{\mathbf{I}} \\ \mathbf{I}_{\mathbf{U}} \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{M}^\bullet_{\mathbf{I}} \mathbf{I}_{\mathbf{I}} \\ -\mathbf{U}_{\mathbf{U}} \\ 0 \end{bmatrix} \quad (\text{II.21})$$

where V_N is the potential of the independent nodes, \mathbf{I} is the current of the branches related to passive elements. $\mathbf{I}_{\mathbf{I}}$ the currents imposed by the current sources, $\mathbf{U}_{\mathbf{I}}$ the unknown voltages of these sources. The same goes for the voltage sources, where $\mathbf{U}_{\mathbf{U}}$ are the applied voltages and $\mathbf{I}_{\mathbf{U}}$ the unknown currents of these sources. The matrix \mathbf{Z}_{bz} corresponds to the contribution of the passive components, so considering a circuit coupling gives

$$\mathbf{Z}_{\text{bz}} = \begin{bmatrix} \mathbf{Z} & 0 \\ 0 & \mathbf{Z}_c \end{bmatrix} \quad (\text{II.22})$$

where \mathbf{Z} is given by the formulation, here $\mathbf{Z} = \mathbf{R} + j\omega\mathbf{L}$, and \mathbf{Z}_c is the impedance matrix given by the electrical circuit.

This system is not under its usual form, to obtain the classical and smaller (in terms of size) system one just has to combine the first two equations (or rows). The combination gives the equation $\mathbf{M}_\mathbf{Z}^\circ \mathbf{Y} \mathbf{M}_\mathbf{Z}^{\circ T} V_N = \mathbf{M}_\mathbf{I}^\circ \mathbf{I}_{bI}$, with \mathbf{Y} the admittance matrix. Solving this admittance-equation is very problematical in our context, since we know how to assemble \mathbf{Z} but not $\mathbf{Y} = \mathbf{Z}^{-1}$, without inverting \mathbf{Z} . There also are other “modified” or “augmented” variants of the basic nodal approach [57], but they are also based on the admittance matrix. So, we will be working with the first and larger (in terms of number of unknowns but not necessarily in terms of memory) equation (II.21).

(b) Loop Analysis.

A circuit loop is a continuous set of branches which forms a closed loop, and any node is used only once (no self-intersection), see Figure II.7 for example.

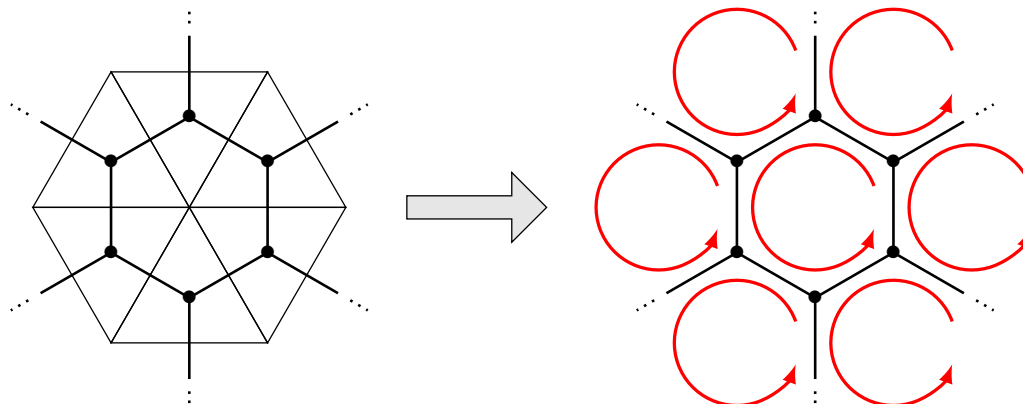


Figure II.7: Example of a set of loops (in red) associated to an equivalent circuit.

We normally use the independent loop approach because it uses fewer unknowns than the nodal approach. Based on the circuit topology, the solver seeks a set of loops which form a system of independent equations. The independent loop search algorithm we are using is published in [58].

The result of the search is the set of independent loops, under the form of an incidence matrix \mathbf{M}° , satisfying:

$$\mathbf{M}^\circ_{i,j} = \begin{cases} 0 & \text{if the branch } j \text{ is not in the loop } i, \\ 1 & \text{if the branch } j \text{ is in the loop } i \text{ with the same orientation,} \\ -1 & \text{if the branch } j \text{ is in the loop } i \text{ with the opposite orientation.} \end{cases}$$

The problem $\mathbf{Z}\mathbf{I} = \mathbf{U}$ in the independent loop basis is then defined as

$$\mathbf{M}^\circ \mathbf{Z} \mathbf{M}^{\circ T} \mathbf{I}_m = \mathbf{M}^\circ \mathbf{U}, \quad (\text{II.23})$$

with \mathbf{I}_m the loop currents ($\mathbf{M}^{\circ T} \mathbf{I}_m = \mathbf{I}$) and \mathbf{Z} an impedance matrix. The Kirchhoff's voltage law gives: $\mathbf{M}^\circ \mathbf{U} = 0$. So, for problems with no external coupling, the equation to solve is:

$$\mathbf{M}^\circ \mathbf{Z} \mathbf{M}^{\circ T} \mathbf{I}_m = 0, \quad (\text{II.24})$$

with $\mathbf{Z} = \mathbf{R} + j\omega\mathbf{L}$. But with a circuit coupling, we have to construct the independent loop set taking in consideration the additional branches, leading to the incidence matrix with the following form/split:

$$\mathbf{M}^\circ = \begin{array}{c} \text{loops} \\ \left[\begin{array}{c|c|c} \Omega + \text{RLC} & \text{I Src.} & \text{U Src.} \\ \hline \mathbf{M}_Z^\circ & \mathbf{M}_I^\circ & \mathbf{M}_U^\circ \\ \hline \end{array} \right] \end{array} \quad (\text{II.25})$$

← branches →

with \mathbf{M}_Z° the submatrix corresponding to the passive elements. \mathbf{M}_I° and \mathbf{M}_U° are the matrices for the current and voltage source branches, respectively.

The circuit equations defining a problem with sources can be written as the next system:

$$\begin{cases} \mathbf{M}_Z^\circ \mathbf{Z}_{b_Z} \mathbf{M}_Z^{\circ T} \mathbf{I}_m - \mathbf{M}_I^\circ \mathbf{U}_I = \mathbf{M}_U^\circ \mathbf{U}_U, \\ \mathbf{M}_I^{\circ T} \mathbf{I}_m = -\mathbf{I}_I, \\ \mathbf{M}_U^{\circ T} \mathbf{I}_m + \mathbf{I}_U = 0. \end{cases} \quad (\text{II.26})$$

The equivalent matrix equation of the system (II.26) is

$$\begin{bmatrix} \mathbf{M}_Z^\circ \mathbf{Z}_{b_Z} \mathbf{M}_Z^{\circ T} & -\mathbf{M}_I^\circ & 0 \\ \mathbf{M}_I^{\circ T} & 0 & 0 \\ \mathbf{M}_U^{\circ T} & 0 & \mathbf{I}_d \end{bmatrix} \begin{bmatrix} \mathbf{I}_m \\ \mathbf{U}_I \\ \mathbf{I}_U \end{bmatrix} = \begin{bmatrix} \mathbf{M}_U^\circ \mathbf{U}_U \\ -\mathbf{I}_I \\ 0 \end{bmatrix}. \quad (\text{II.27})$$

There we have the general circuit equations using the second Kirchhoff's law. The solution of the problem finally comes down in solving (II.27).

The circuit solver is an independent tool compatible with any PEEC formulations, as long as the problem can be represented using a lumped element equivalent circuit. So, a comparison of the conditioning of both approaches will be done later on.

We have developed the previous PEEC formulation in the general case (with volume conductors), but we also aim for PCB-like applications which contains very thin conductor. The last formulation may suffer of inaccuracies coming from the discretization, so we will present next another formulation which allows to solve efficiently such problems.

II.2.b Shell PEEC-formulation

In the context of these works, we are also targeting PCB-like applications. Typically, we are facing problems with conductors as PCB tracks, i.e.: very thin layers which have a thickness $e \leq 70\mu\text{m}$ and with the other dimensions that are much greater than the thickness. Using very thin (volume) mesh-elements might cause inaccuracy issues during the integration process if one uses numerical integral computation techniques. Moreover, the frequency range of these type of application imposes us to take into account the skin effects. Therefore, we may have to add more mesh-subdivisions along the thickness, in consequence, we will have even thinner elements.

One solution is to use thin shell approximations in order to use surface elements instead. We will quickly see how to obtain a thin shell PEEC-formulation [53].

The distribution of current density across the thickness can be estimated analytically, so we are using this information to approximate the current density in the volume using an average current density supported by a surface. For example, considering an extruded element, as in Figure II.8, we will use the median rectangle S to approximate the current density flowing through the left hexahedron.

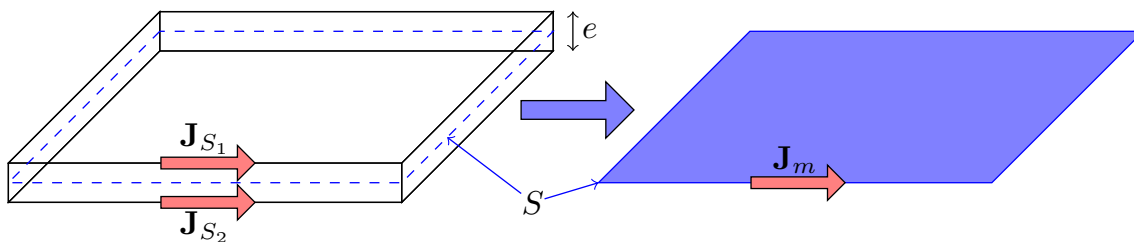


Figure II.8: Very thin hexahedron approximated using its median rectangle on the right.

We typically distinguish two cases to compute the average current density, which depend on the thickness e and the skin depth δ :

- a) $e \ll \delta$: the current density is uniform across the thickness, as in Figure II.9, where the hexahedron of Figure II.8 is shown from the side.

Assuming,

$$\int_{\Omega} \frac{\mathbf{J}}{r} d\Omega \approx \int_S \frac{1}{r} \left\{ \int_0^e \mathbf{J}(\mathbf{y}) d\mathbf{y} \right\} dS = e \int_S \frac{\mathbf{J}_m}{r} dS, \quad (\text{II.28})$$

and injecting it in (II.9), the previous integral equation can be approximated by:

$$\frac{\mathbf{J}}{\sigma} + j\omega \frac{\mu_0}{4\pi} e \int_S \frac{\mathbf{J}_m}{r} dS = -\mathbf{grad}V. \quad (\text{II.29})$$

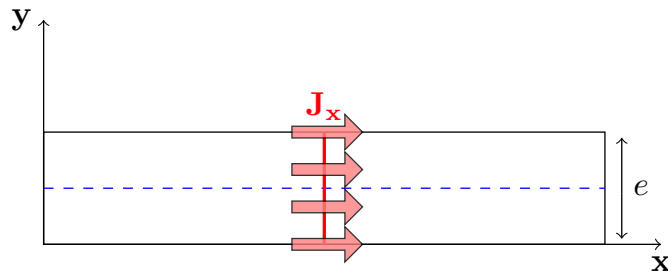


Figure II.9: Side view of a case where the current density is flowing through an element uniformly on y (the thickness).

Since the problem is still using volume quantities, \mathbf{J} and V for instance, we are writing the last equation on the top and bottom sides of the conductor (S_1 and S_2) and averaging them gives

$$\frac{\mathbf{J}_m}{\sigma} + j\omega \frac{\mu_0}{4\pi} e \int_S \frac{\mathbf{J}_m}{r} dS = -\mathbf{grad}V_m \quad (\text{II.30})$$

with $V_m = (V_{S1} + V_{S2})/2$.

The discretization procedure is similar to the volume's, the only change is the use of surface face functions $\mathbf{w}_i^{(s)}$ also known as RWG functions [34], see Figures II.10. Because the face in 3D-meshes degenerates into edges in 2D-meshes, the face elements are in fact special edge functions.

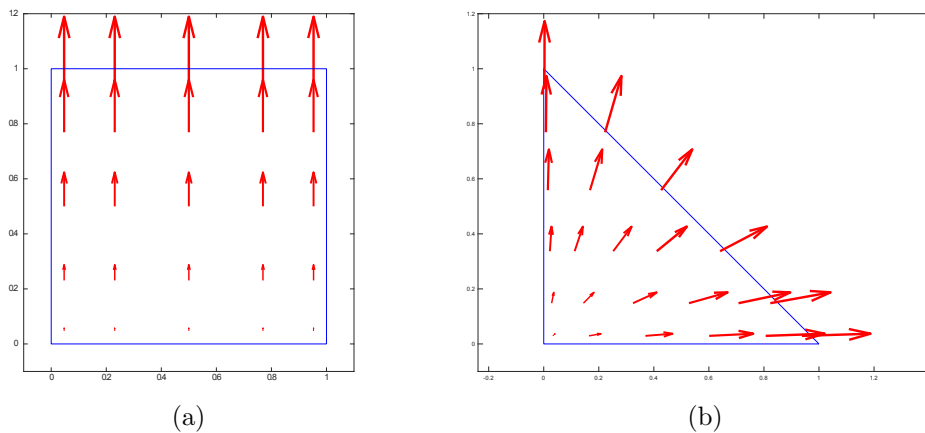


Figure II.10: (a-b) Example of a face shape function on a quadrangle and a triangle.

The average current density is discretized using the surface face elements as follows,

$$\mathbf{J}_m = \frac{1}{e} \sum_j \mathbf{w}_j^{(s)} \mathbf{I}_j^{(s)} \quad (\text{II.31})$$

where $\mathbf{I}_j^{(s)}$ is the current flowing through the degenerated face j . Following the same procedure as previously, we apply a standard Galerkin discretization with the surface face shape functions and the result is

$$\left(\mathbf{R}^{(s)} + j\omega\mathbf{L}^{(s)}\right)\mathbf{I}^{(s)} = \mathbf{U}^{(s)}, \quad (\text{II.32})$$

with

$$\begin{cases} \mathbf{R}_{i,j}^{(s)} = \int_S \frac{\mathbf{w}_i^{(s)} \mathbf{w}_j^{(s)}}{e\sigma} dS, & (\text{II.33}) \\ \mathbf{L}_{i,j}^{(s)} = \int_S \mathbf{w}_i^{(s)} \int_S \frac{\mathbf{w}_j^{(s)}}{r} dS dS, & (\text{II.34}) \\ \mathbf{U}_i^{(s)} = - \int_S \mathbf{w}_i^{(s)} \mathbf{grad} V_m dS. & (\text{II.35}) \end{cases}$$

Putting aside the changes from volume to surface, we can note that this system is very similar to the volume's, with only the conductivity that is scaled by the thickness here. We can see it like an equivalent conductivity that represents the simple phenomena occurring in the volume.

The case where the current has a uniform distribution may seem trivial and/or irrelevant for some applications, so we generalize this approximation in the second case.

- b) $e \gtrsim \delta$: The skin effect cannot be neglected, therefore, we can't assume the current to be uniform across the thickness. However, we know that the current has a tendency to be more concentrate on the top and bottom faces of the volume in case of thin shell geometries (as in Figure II.11), due to the skin effect.

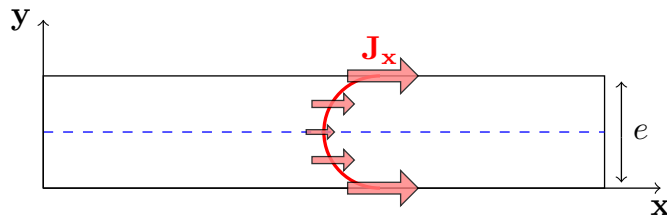


Figure II.11: Side view of a case where the current density is flowing through an element with a hyperbolic evolution along \mathbf{y} (the thickness).

Previous works provide an analytical expression of the current density distribution while taking into account the skin effect and the thickness of the conductor [59–61]. The average current density expression is given by

$$\mathbf{J}_m = \frac{1}{e} \int_0^e \mathbf{J}(\mathbf{y}) d\mathbf{y} = G \frac{\mathbf{J}_{S_1} + \mathbf{J}_{S_2}}{2}, \quad (\text{II.36})$$

with G a complex number defined as

$$G = \frac{2\delta \tanh\left(\frac{e(1+j)}{2\delta}\right)}{e(1+j)}, \quad (\text{II.37})$$

which takes into account the skin depth. The factor G was obtained considering infinite planes, so inaccuracies may occur when dealing with geometries having corners or complex shapes.

Writing (II.29) on both sides and averaging it here leads to the next equation

$$\frac{\mathbf{J}_m}{G\sigma} + j\omega \frac{\mu_0}{4\pi} e \int_S \frac{\mathbf{J}_m}{r} dS = -\mathbf{grad}V_m, \quad (\text{II.38})$$

and using the same discretization with (II.31) gives:

$$\left(\tilde{\mathbf{R}}^{(s)} + j\omega \mathbf{L}^{(s)}\right) \mathbf{I}^{(s)} = \mathbf{U}^{(s)}, \quad (\text{II.39})$$

where

$$\tilde{\mathbf{R}}_{i,j}^{(s)} = \int_S \frac{1}{e} \frac{\mathbf{w}_i^{(s)} \mathbf{w}_j^{(s)}}{G\sigma} dS. \quad (\text{II.40})$$

As for the last case, the phenomena happening in the volume are taken into account using an equivalent complex conductivity ($\sigma^* = G\sigma$).

The integral equations to solve are very similar for all the cases we have presented, but the equivalent circuit should be simplified, since no discretization has to be considered across the thickness. The independent loop set therefore should be smaller, which may lead to a lighter problem to solve since (II.23)'s size is the number of independent loops (without sources). Treating very thin conductors with one of the last shell formulations reduces the complexity of the problem, the integral computation times (since we have fewer elements), and as a result, it provides a more efficient solution.

We have presented two formulations to solve magnetoharmonic problems with volume or thin shell conductors and a circuit coupling. Both formulations consist in solving the circuit equations under the form of (II.27). As a remainder, we have presented these formulations in order to study their conditioning and more specifically, if we can construct an efficient preconditioner.

II.2.c Preconditioning study

We are investigating the capability to construct an efficient preconditioner for a low cost, using the \mathcal{H} -matrices for a solution using the nodal or the loop analysis.

Circuit solver using the nodal approach

In a first step, we assume that the external circuit does not contain any passive elements for a matter of understanding, so $\mathbf{Z}_{\mathbf{b}\mathbf{z}} = \mathbf{Z}$. The matrix to precondition, defined in (II.21), is large with a block-structure, however, most of the matrix is very sparse since only \mathbf{Z} is dense. We want to use the \mathcal{H} -matrices to compress and compute the preconditioner of \mathbf{Z} , thus we decided to split the matrix as:

$$\left[\begin{array}{c|c} \mathbf{Z}_{1,1} & \mathbf{Z}_{1,2} \\ \hline \mathbf{Z}_{2,1} & \mathbf{Z}_{2,2} \end{array} \right] = \left[\begin{array}{c|cc} \mathbf{Z} & \mathbf{M}^{\bullet T} & 0 & 0 \\ \hline \mathbf{M}^{\bullet} & 0 & \mathbf{M}_{\mathbf{U}}^{\bullet} & 0 \\ 0 & \mathbf{M}_{\mathbf{U}}^{\bullet T} & 0 & 0 \\ 0 & \mathbf{M}_{\mathbf{I}}^{\bullet T} & 0 & \mathbf{I}_d \end{array} \right], \quad (\text{II.41})$$

to isolate \mathbf{Z} . The incidence matrices are usually very sparse by construction and as we can compute an efficient LU decomposition of $\mathbf{Z}_{1,1}$ (via the \mathcal{H} -LU), we decided to use a block-LU which takes in consideration the sparse storages. The block-LU procedure is actually the same as computing a \mathcal{H} -LU with a very simple \mathcal{H} -matrix, so the reader can refer to the first chapter for more details about a 2×2 block-LU.

The efficiency of this preconditioner will be studied in the next test case, but let us first analyze the loop approach.

Loop approach

We have to solve the circuit equations arising from the second Kirchhoff's law (II.23):

$$\mathbf{Z}_{\mathbf{m}} \mathbf{I}_m = \mathbf{M}_{\mathbf{Z}}^{\circ} \mathbf{Z} \mathbf{M}_{\mathbf{Z}}^{\circ T} \mathbf{I}_m = 0, \quad (\text{II.42})$$

for a problem without sources. To study how to construct a preconditioner, we have to analyze $\mathbf{Z}_{\mathbf{m}}$ closely.

The matrix $\mathbf{Z}_{\mathbf{m}} \in \mathbb{C}^{n_l \times n_l}$, with n_l the number of independent loops, is a square matrix that corresponds to the change from the branches basis to the independent loops basis. This change-of-basis is supported by $\mathbf{M}_{\mathbf{Z}}^{\circ} \in \mathbb{R}^{n_l \times n_b}$, with n_b the number of branches, the rectangular incidence matrix which stores the informations of the loops' path. Our aim here is to find a way to approximate the inverse of $\mathbf{Z}_{\mathbf{m}}$ at low costs (in time and memory) using the \mathcal{H} -matrices.

The PEEC integral equations of all previous formulations are directly compatible with the \mathcal{H} -matrices, since the matrices \mathbf{L} and $\mathbf{L}^{(s)}$ are very classic integrals. Note that the inductance matrices are very similar to the electrostatic matrix in [subsection I.6.a Electrostatic example](#), with the only difference that they use a vector kernel function here. The use of the \mathcal{H} -matrices provides a twofold benefit, since we

want to use it for the compression, in order to lower the costs of the assembly and storage, and for the \mathcal{H} -LU preconditioner.

Keeping these conditions in mind, we started to think about two naive approaches to invert \mathbf{Z}_m : the direct and the substitution.

Directly trying to invert \mathbf{Z}_m leads to assemble some or all of its entries. Nevertheless, \mathbf{Z}_m is never assembled directly because we did not have the numerical tools for it and implementing such integration in the loops basis would be very time consuming. So, instead we assemble and compress \mathbf{Z} using the \mathcal{H} -matrices, and we have to handle an explicit change-of-basis. Explicitly assembling \mathbf{Z}_m is achieved computing two matrix-products, which requires $(n_b + n_l)n_b n_l$ operations, and storing the result into a dense matrix of size $n_l \times n_l$. These costs are obviously too important to consider assembling totally \mathbf{Z}_m , alternatively we could afford to assemble it partially and use a sparse storage for the result. This now leads to the choice of which part of \mathbf{Z}_m should be computed. A former PhD student proposed to compute an incomplete-LU preconditioner using only the near-interactions [62, II.5], but the results were not satisfactory. Other classical preconditioners have been studied in these works, so we decided to try to construct a preconditioner in the branches basis, i.e.: based on \mathbf{Z} .

Another naive approach consists in splitting the solution of (II.23) into a system of three equations obtained by substitution:

$$\begin{cases} \mathbf{M}\mathbf{y} = 0, & \text{(II.43)} \\ \mathbf{Z}\mathbf{x} = \mathbf{y}, & \text{(II.44)} \\ \mathbf{M}^T \mathbf{I}_m = \mathbf{x}. & \text{(II.45)} \end{cases}$$

Solving this system requires solving rectangular systems, so non-classical solvers has to be employed. The SuiteSparse provides a sparse QR decomposition (SPQR) [63], directly usable via Matlab, to solve (II.43) and (II.45) very efficiently. At first sight, solving the system should not be a problem since we can efficiently solve each equations. However, the resulting solution does not satisfy the original equation (II.23), although each of the three solutions gives a correct result. In order to understand what went wrong, we have to look at these three equations in details. The second equation is not a problem since it is the \mathcal{H} -matrix part and the last equation is over-determined (more rows/equations than columns/unknowns), therefore the correct solution should be reached. So, (II.43) is problematical since it is an under-determined equation, there are an infinite number of solution \mathbf{y} , therefore, even though the solution \mathbf{y} satisfies (II.43), it is not the correct one. In fact, when we

pay more attention to the linear algebra theory and more specially to the conditions of the substitution, in order to solve (II.43) correctly the next condition have to be ensured in addition: $\mathbf{y} \in \text{span}\{\mathbf{Z}\}$. Finally, solving (II.43) with the last constraint leads to solve the whole problem, so this approach shows no advantages.

No reliable and efficient way to precondition a change-of-basis, or the PEEC formulations using the second Kirchhoff's law, have been found. The two most naive and direct approaches have been explored, since we have not found any publications on the matter, one has to investigate either more evolved approaches or change the starting point.

We are comparing the nodal and loop analysis in the next test case, with the nodal approach that is preconditioned by a block-LU factorization.

Test case

Let us consider an example to evaluate the preconditioner presented for the nodal analysis and compare it with the loop analysis without preconditioner. The device we are studying is made of two thin conductive layers with a zigzags shape, a picture of it is presented in Figure II.12. It is supplied by a current source of 1 A connecting the two layers by their lower left corners and a short-circuit is linking the layers on the opposite side (lower right corners). The current density resulting at 1MHz from the PEEC solver is shown in Figure II.13. Note that the dielectric block separating the conductive layers was not taken in consideration here.

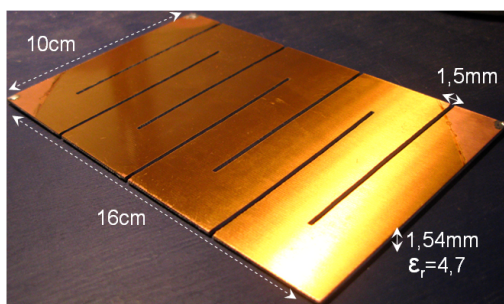


Figure II.12: Picture of the device.

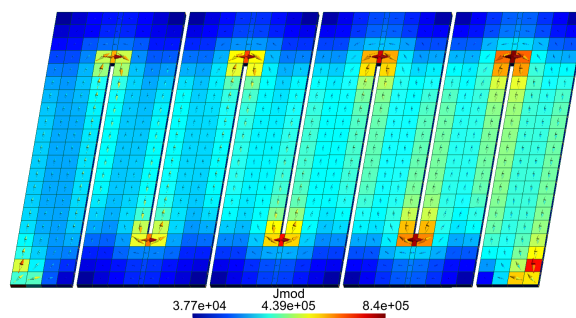


Figure II.13: Isovalues of the current density magnitude.

We have solved this problem using the independent node (IN) approach with and without preconditioner (respectively denoted by “IN w/ prec.” and “IN w/o prec.”) and loop approach (denoted by “IL w/o prec.”), in order to position their performances. The residual evolution during the FGMRes of the last three approaches is presented in Figure II.14.

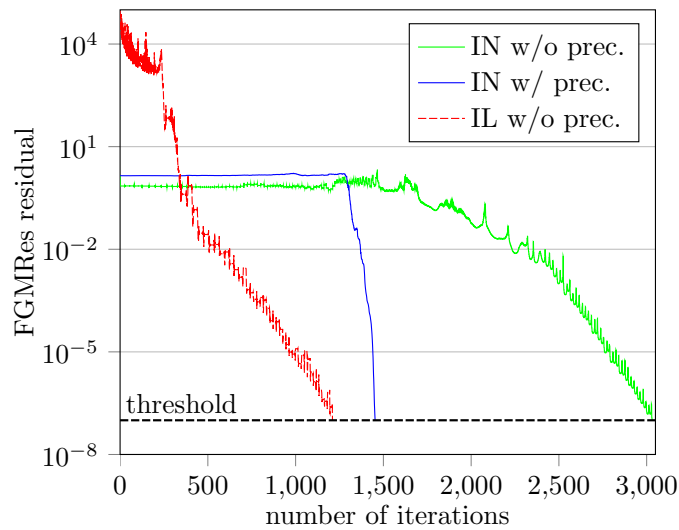


Figure II.14: Evolution of the FGMRes residual against the iterations. “IN” (Independent Nodes) denotes the nodal analysis and “IL” (Independent Loops) denotes the loop analysis.

The solution using the IN without preconditioner is naturally taking more iterations to converge. Constructing the block-LU preconditioner almost divides by two the number of iterations, however, the independent loop (IL) solution took less iterations without preconditioner.

Focusing now on the computational costs of each approach, Figure II.15 and Figure II.16 contain the elapsed time and the memory used for the solution, respectively. We can point out that the solution with preconditioner allowed to reduce the solution time, but as previously, we also clearly see that the IL solution is much faster. We observe almost the same behavior for the memory consumption, except that the preconditioned solution requires a large amount of memory compared to the others.

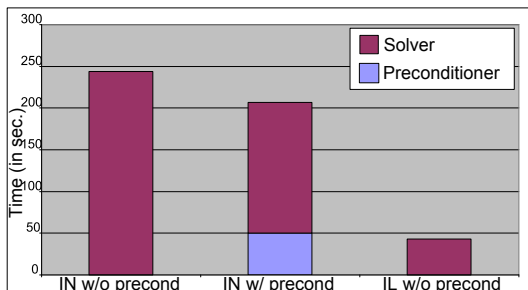


Figure II.15: Elapsed time taken by the solutions.

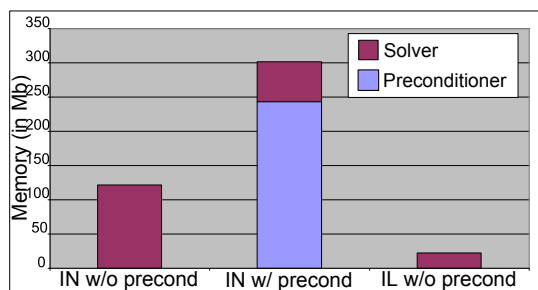


Figure II.16: Memory requirements of the solutions.

To understand better why the preconditioner is requiring that much memory, we are looking at the blocks of the block-LU factorization. The decomposition is constructed as:

$$\left[\begin{array}{c|c} \mathbf{Z}_{1,1} & \mathbf{Z}_{1,2} \\ \hline \mathbf{Z}_{2,1} & \mathbf{Z}_{2,2} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{L}_{1,1} & 0 \\ \hline \mathbf{L}_{2,1} & \mathbf{L}_{2,2} \end{array} \right] \left[\begin{array}{c|c} \mathbf{U}_{1,1} & \mathbf{U}_{1,2} \\ \hline 0 & \mathbf{U}_{2,2} \end{array} \right] \quad (\text{II.46})$$

with $\mathbf{L}_{1,1}$ and $\mathbf{U}_{1,1}$ given by the \mathcal{H} -LU of $\mathbf{Z}_{1,1}$. Keeping the sparse storage of the extra-diagonal matrices, we want to solve

$$\begin{cases} \mathbf{Z}_{1,2} = \mathbf{L}_{1,1} \mathbf{U}_{1,2} & (\text{II.47}) \\ \mathbf{Z}_{2,1} = \mathbf{L}_{2,1} \mathbf{U}_{1,1} & (\text{II.48}) \end{cases}$$

by forward substitutions and get $\mathbf{U}_{1,2}$ and $\mathbf{L}_{2,1}$ with a sparse storage. For this example, the pattern of $\mathbf{U}_{1,2}$ and $\mathbf{L}_{2,1}$ are shown in Figure II.17 and II.18, respectively. Taking advantage of the sparse storage led to solutions ($\mathbf{U}_{1,2}$ and $\mathbf{L}_{2,1}$) filled at 50%, so it allowed to halved to normal storage requirement. However, since they are very large blocks, they are consuming too much memory compared to a classical iLU which can keep the initial matrix sparsity. Moreover, constructing the last matrices consists in applying an iLU factorization to $\mathbf{Z}_{2,2} - \mathbf{L}_{2,1} \mathbf{U}_{1,2} = \mathbf{L}_{2,2} \mathbf{U}_{2,2}$, even though $\mathbf{Z}_{2,2}$ is very sparse, the product $\mathbf{L}_{2,1} \mathbf{U}_{1,2}$ is more likely to give a dense matrix. So, in the end the bigger blocks (i.e.: $\mathbf{L}_{2,2}$ and $\mathbf{U}_{2,2}$) should be dense, which explains the high needs in memory for this preconditioner.

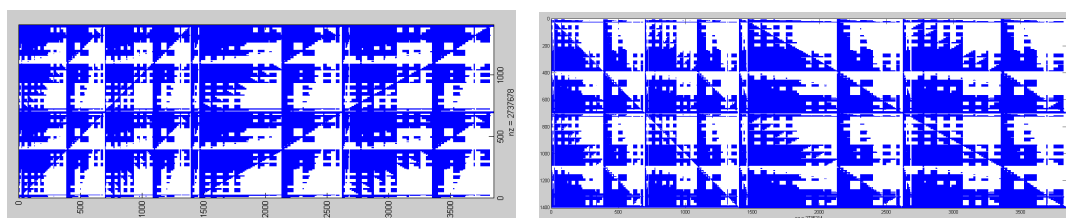


Figure II.17: Sparsity pattern of $\mathbf{U}_{1,2}^T$. Figure II.18: Sparsity pattern of $\mathbf{L}_{2,1}$.

The block-LU preconditioner might be improved using ordering techniques that are well-known to reduce the filling factor of factorizations. Or the extra-diagonal blocks can be neglected to compute only the iLU of $\mathbf{Z}_{2,2}$ and thus avoid the filling of the blocks. Such approach leads to a poor preconditioner, since $\mathbf{Z}_{2,2}$ is an extremely sparse block. Moreover, the extra-diagonals are very important since they represent all the connections and more importantly the supplies coupling with the device. Other splits of the matrix have been tested, but the conclusions were the same.

Other approaches have been studied and tested, we mainly have implemented and experimented the block solvers as the Successive Over Relaxation (SOR). Thus, we could keep using the \mathcal{H} -LU of \mathbf{Z} for its solution. Looking carefully to the block-structure of the matrix to solve in (II.21), we can note that the diagonal contains two blocks of 0's. There is one appearing only when the problem has voltage sources, while the other is very large, constantly there and very problematic since we can't approach its inverse numerically. So, even though the other blocks can be preconditioned efficiently, this block of 0's does not enable us to compute an efficient block solution.

We have found that our problem without supplies can in fact be written as a (generalized) saddle point problem [64], but we did not have the time to investigate seriously the opportunities. It might be an interesting way to follow since it is a problem shared with other communities and many iterative schemes have been developed and published.

In conclusion, we did not succeed in computing an efficient preconditioner for either the nodal or the loop approach. We have only tried algebraic-based techniques using factorizations or inverses, so one possible continuation is to investigate iterative scheme to approach the invert of the nodal problems.

However, we could see that the loop analysis provides the most efficient and less expensive solution, even without preconditioner. So, we have investigated a similar formulation using the vector potential \mathbf{T} as unknown.

II.3 T Formulation

We want to study a formulation which allows the construction of an efficient preconditioner and an electrical circuit coupling. We have found the next formulation to be very similar to the PEEC method and the loop approach, so, even though the \mathbf{T} formulation does not allow a natural circuit coupling, we will see how to recycle the previous works in order to improve this one.

We are starting by presenting the formulation, showing that the \mathcal{H} -matrices can be applied directly. In the same time, we will present how the circuit couplings can be taken in consideration. Finally, we will see if we are able to precondition a problem using this formulation.

II.3.a Integral formulation

Let us start as the PEEC formulations, with the Ampère-Maxwell equation

$$\frac{\mathbf{J}}{\sigma} + j\omega \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}}{r} d\Omega = -\mathbf{grad}V - j\omega \mathbf{A}_0, \quad (\text{II.49})$$

with \mathbf{A}_0 a magnetic source field. The current density's solenoidality: $\text{div}\mathbf{J} = 0$, gives the existence of an electric vector potential \mathbf{T} that satisfies

$$\mathbf{curl}\mathbf{T} = \mathbf{J}, \quad (\text{II.50})$$

and using \mathbf{T} as unknown leads to the next integral equation:

$$\frac{\mathbf{curl}\mathbf{T}}{\sigma} + j\omega \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{curl}\mathbf{T}}{r} d\Omega = -\mathbf{grad}V - j\omega \mathbf{A}_0. \quad (\text{II.51})$$

The potential \mathbf{T} is discretized using edge elements \mathbf{w}^e ,

$$\mathbf{T} = \sum_j \mathbf{w}_j^e T_j \quad (\text{II.52})$$

and applying a standard Galerkin discretization method leads to the following equation

$$(\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}}) \mathbf{T} = -\tilde{\mathbf{U}} - j\omega\tilde{\mathbf{A}}_0 \quad (\text{II.53})$$

where the quantities are defined as below

$$\left\{ \begin{array}{l} \tilde{\mathbf{R}}_{i,j} = \int_{\Omega} \frac{\mathbf{curl}\mathbf{w}_i^e \mathbf{curl}\mathbf{w}_j^e}{\sigma} d\Omega, \\ \tilde{\mathbf{L}}_{i,j} = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \int_{\Omega} \frac{\mathbf{curl}\mathbf{w}_j^e}{r} d\Omega d\Omega, \\ \tilde{\mathbf{U}}_i = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \mathbf{grad}V d\Omega. \\ \tilde{\mathbf{A}}_0 = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \mathbf{A}_0 d\Omega. \end{array} \right. \quad (\text{II.54})$$

$$\tilde{\mathbf{L}}_{i,j} = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \int_{\Omega} \frac{\mathbf{curl}\mathbf{w}_j^e}{r} d\Omega d\Omega, \quad (\text{II.55})$$

$$\tilde{\mathbf{U}}_i = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \mathbf{grad}V d\Omega. \quad (\text{II.56})$$

$$\tilde{\mathbf{A}}_0 = \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \mathbf{A}_0 d\Omega. \quad (\text{II.57})$$

with the edge shape functions $\mathbf{curl}\mathbf{w}_i^e$, as in Figure II.19. Focusing the RHS vector $\tilde{\mathbf{U}}$, we apply the Divergence theorem and get

$$\begin{aligned} \tilde{\mathbf{U}}_i &= \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \mathbf{grad}V d\Omega = \int_{\Omega} \text{div}(\mathbf{curl}\mathbf{w}_i^e V) d\Omega - \underbrace{\int_{\Omega} \text{div}(\mathbf{curl}\mathbf{w}_i^e) V d\Omega}_{=0} \\ &= \int_{\Gamma} V (\mathbf{curl}\mathbf{w}_i^e) \cdot \mathbf{n} d\Gamma. \end{aligned}$$

We have previously set $\mathbf{J} \cdot \mathbf{n} = 0$ on the boundary in the PEEC formulations, here it comes down to $(\mathbf{curl}\mathbf{w}_i^e) \cdot \mathbf{n} = 0$. Finally, we have $\tilde{\mathbf{U}} = 0$ and thus

$$\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}} = -j\omega\tilde{\mathbf{A}}_0. \quad (\text{II.58})$$

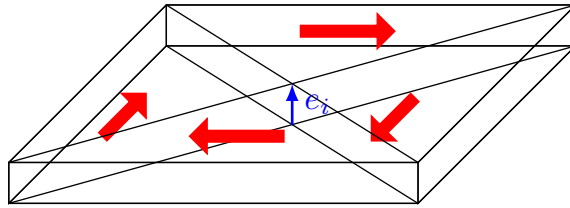


Figure II.19: Overview of the edge element \mathbf{curlw}_i^e on the edge e_i .

The shape function \mathbf{curlw}_i^e forms a current loop which turns around the edge e_i , as in Figure II.19. Taking a circuit point-of-view, we can perceive and assimilate the current loops formed by all the edge shape functions as a specific set of loops, as in Figure II.20. This is very similar to the independent loops approach we presented in the previous subsection II.2.a PEEC Formulation to solve the circuit equations. But one major difference is that we are able to directly and easily apply the \mathcal{H} -matrices to (II.53), since we do not have to handle any change-of-basis.

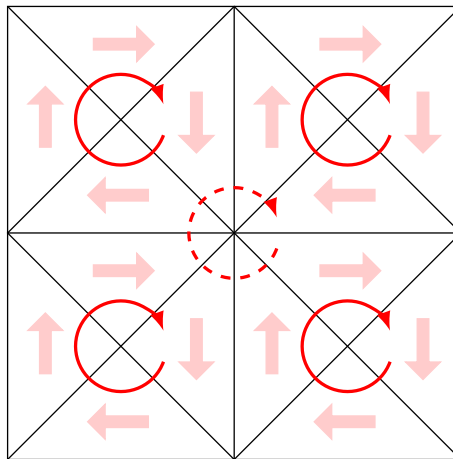


Figure II.20: Top view of a mesh and its set of loops formed by the edge shape functions.

The assembly of the matrices can be done using the proper curl edge elements or one can use a discretized form of the curl operator, represented by a sparse *curl matrix* \mathbf{M}_{curl} . This matrix is also called incidence matrix, which links the current flowing through the faces and the circulation of \mathbf{T} along the edges, since $\mathbf{J} = \mathbf{curl}\mathbf{T}$. In fact, the currents and the vector potential are linked via $\mathbf{I} = \mathbf{M}_{\text{curl}}^T \mathbf{T}$, so, applying the curl operator on (II.11) and using \mathbf{T} as unknown leads to the next equation

$$\left(\mathbf{M}_{\text{curl}} \mathbf{R} \mathbf{M}_{\text{curl}}^T + j\omega \mathbf{M}_{\text{curl}} \mathbf{L} \mathbf{M}_{\text{curl}}^T \right) \mathbf{T} = -j\omega \mathbf{M}_{\text{curl}} \tilde{\mathbf{A}}_0, \quad (\text{II.59})$$

that is equivalent to (II.53), with the matrices \mathbf{R} and \mathbf{L} defined as in (II.11). In fact, one of the first \mathbf{T} formulations for eddy-current problems [65] uses this approach.

The last equation is very similar to the circuit equations obtained using the second Kirchhoff's law, in fact, the only difference is the use of \mathbf{M}_{curl} instead of \mathbf{M}° . As a short recall, the incidence matrix \mathbf{M}° is a change-of-basis matrix (from the branches to the loops) computed by the circuit solver and using graphs theory. Whereas the discrete curl matrix \mathbf{M}_{curl} can be seen, from a circuit point-of-view, as a representation of the small loops, as in Figure II.20. So, \mathbf{M}_{curl} can be seen as a specific case of \mathbf{M}° if the small loops are independents.

In general the \mathbf{T} formulation does not ensure the uniqueness of the solution, let us differentiate two cases:

- Simply-connected regions. The system of equations requires a gauge, typically $\mathbf{T} \cdot \mathbf{u} = 0$ (using graphs theory [65,66]), and we want to impose $\mathbf{J} \cdot \mathbf{n} = 0$ on the boundary additionally. One way to do it is imposing $\mathbf{T} = 0$ on the boundary (so $\mathbf{J} \cdot \mathbf{n} = 0$), and then annihilating inner edges using graphs theory [67, Chap.2 Part.B]. In fact, even if the inner edges are not annihilated, the system can be solved, since the system of equations is compatible [68], and thus auto-gauging [69] when using iterative solvers (as FGMRes or ICCG for example).
- Non-simply-connected regions, typically geometries with holes. The previous method is not sufficient in this case, in fact, insuring last gauge requires special treatments [66]. A classical approach consists in introducing a number of additional unknowns, corresponding to the number of holes [70].

We are introducing an alternate way to ensure the uniqueness of the solution while keeping the loops formed by the incidence matrix \mathbf{M}_{curl} , based on a circuit approach below.

II.3.b Circuit approach

We have briefly mentioned that using \mathbf{M}_{curl} to compute the curl operator, is equivalent to assemble directly with the curl edge functions $\mathbf{curl}\mathbf{w}_i^e$. We also saw that \mathbf{M}_{curl} is a specific case of the incidence matrix \mathbf{M}° , arising from the PEEC formulations. The hybridization we want to achieve consists in assembling directly the matrices in the small loops of the \mathbf{T} formulation, that we know how to precondition, and use a special \mathbf{M}° to cure the connectivity issues.

Looking at the \mathbf{T} formulation with a circuit point-of-view, we can extract a particular set of small loops (as in Figure II.20), under the form of a matrix \mathbf{M}_{curl} . We can enforce this set of loops as an initial set in the construction process of \mathbf{M}° , which

will look for the exceeding loops (contained in the initial set) and the missing loops as in Figure II.21. Searching these missing loops is mandatory to solve the problem and to cure the connectivity issue. This gives us a partially arbitrary matrix:

$$\mathbf{M}^\circ = \begin{bmatrix} \tilde{\mathbf{M}}_{\text{curl}} \\ \mathbf{M}_{ml} \end{bmatrix} \quad (\text{II.60})$$

with \mathbf{M}_{ml} representing the missing loops and $\tilde{\mathbf{M}}_{\text{curl}} \subseteq \mathbf{M}_{\text{curl}}$. The initial set of small loops can contain too many loops, in the sense that some are a linear combination of other loops, so we should delete these loops (or rows of \mathbf{M}_{curl}) in order to get an independent set $\tilde{\mathbf{M}}_{\text{curl}}$. However, one can keep these loops (so $\mathbf{M}_{\text{curl}} = \tilde{\mathbf{M}}_{\text{curl}}$), leading to a compatible system of equations. Some tests showed that the problem tends to converge faster when the exceeding equations are kept [71]. As we are aiming for preconditioned iterative solutions, it can be advantageous to keep the exceeding loops.

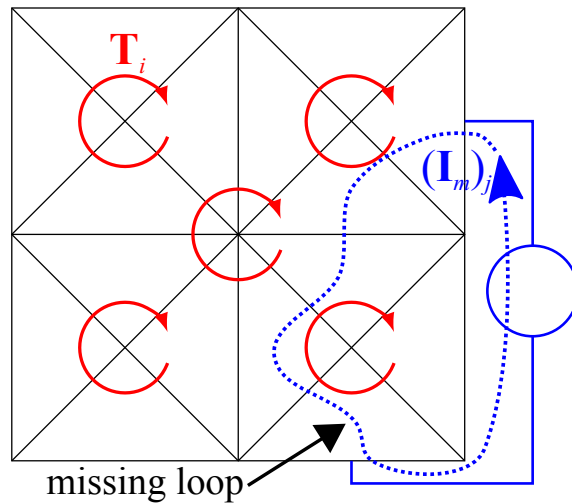


Figure II.21: Example of an initial set of small loops (in red) with a missing loop (in blue and dashed).

Applying \mathbf{M}° to the integral equation in the faces (or branches) basis, we get the same equation $\mathbf{M}^\circ \mathbf{Z} \mathbf{M}^{\circ T} \mathbf{I}_m = \mathbf{M}^\circ \mathbf{U}$ as before which gives the next block structure:

$$\mathbf{M}^\circ \mathbf{Z} \mathbf{M}^{\circ T} = \left[\begin{array}{c|c} \mathbf{M}_{\text{curl}} \mathbf{Z} \mathbf{M}_{\text{curl}}^T & \mathbf{M}_{\text{curl}} \mathbf{Z} \mathbf{M}_{ml}^T \\ \hline \mathbf{M}_{ml} \mathbf{Z} \mathbf{M}_{\text{curl}}^T & \mathbf{M}_{ml} \mathbf{Z} \mathbf{M}_{ml}^T \end{array} \right] \quad (\text{II.61})$$

with $\mathbf{Z} = \mathbf{R} + j\omega \mathbf{L}$. It could seem like we are facing the same issues as the PEEC formulation, but we know that applying the matrix \mathbf{M}_{curl} to a branch basis is equivalent to apply the curl operator. So, we have

$$\mathbf{M}_{\text{curl}} \mathbf{Z} \mathbf{M}_{\text{curl}}^T = \tilde{\mathbf{Z}} = \tilde{\mathbf{R}} + j\omega \tilde{\mathbf{L}}, \quad (\text{II.62})$$

the exact equation (II.53) and also

$$[\mathbf{M}_{\text{curl}}\mathbf{Z}] \mathbf{M}_{ml}^T = \left[\int_{\Omega} \frac{\mathbf{curl}\mathbf{w}_i^e \mathbf{w}_j}{\sigma} d\Omega + j\omega \int_{\Omega} \mathbf{curl}\mathbf{w}_i^e \int_{\Omega} \frac{\mathbf{w}_j}{r} d\Omega d\Omega \right] \mathbf{M}_{ml}^T. \quad (\text{II.63})$$

The matrix \mathbf{Z} thus does not have to be assembled. Note that the extra-diagonal terms are symmetrical, so the previous equations can be transposed to get the opposite term. Finally, we have the next equation to solve:

$$\left[\begin{array}{c|c} \tilde{\mathbf{Z}} & \mathbf{M}_{\text{curl}}\mathbf{Z}\mathbf{M}_{ml}^T \\ \hline \mathbf{M}_{ml}\mathbf{Z}\mathbf{M}_{\text{curl}}^T & \mathbf{M}_{ml}\mathbf{Z}\mathbf{M}_{ml}^T \end{array} \right] \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_m \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{M}_{ml}\mathbf{U} \end{bmatrix} \quad (\text{II.64})$$

with the top left block that we know how to precondition and the additional blocks to ensure the uniqueness of the solution, in a general purpose.

This hybrid formulation was achieved starting in a circuit context, so we can also add a circuit coupling to this formulation without much effort.

II.3.c Electrical circuit coupling

The formulation we have reached in fact stands on the construction of \mathbf{M}° , with an initial set of loops (\mathbf{M}_{curl}). So, we can add other branches without difficulties, leading to an incidence matrix of the form:

$$\mathbf{M}^\circ = \begin{array}{c} \uparrow \\ \text{loops} \\ \downarrow \end{array} \begin{array}{cc} & \begin{array}{c} \Omega \\ \text{Circuit} \end{array} \\ \begin{array}{|c|c|} \hline \mathbf{M}_{\text{curl}} & 0 \\ \hline \mathbf{M}_{ml}^\Omega & \mathbf{M}_{ml}^{\text{cir}} \\ \hline \end{array} \\ \leftarrow \text{branches} \rightarrow \end{array} \quad (\text{II.65})$$

where the first block-row is imposed as previously and the second $[\mathbf{M}_{ml}^\Omega, \mathbf{M}_{ml}^{\text{cir}}]$ is constructed as \mathbf{M}_{ml} , looking for the missing loops. Keeping the same ordering as in (II.25), we can split the circuit branches and get $\mathbf{M}_{ml}^{\text{cir}} = [\mathbf{M}_{ml}^{\text{RLC}}, \mathbf{M}_{ml}^{\text{I}}, \mathbf{M}_{ml}^{\text{U}}]$. Again, the branch ordering showed here is completely arbitrary and can be changed, however, one should group the branches wisely for a matter of ease.

Once the “new” \mathbf{M}° computed in (II.65), we repeat the same procedure used in the previous subsection with $\mathbf{M}^\circ|_{\Omega} = [\mathbf{M}_{\text{curl}}; \mathbf{M}_{ml}^\Omega]$ (the first block-column) and we get (II.64) with \mathbf{M}_{ml} replaced by \mathbf{M}_{ml}^Ω .

Adding the circuit coupling simply consists in injecting (II.61) in (II.27) and replacing change-of-basis as previously. The circuit equations resulting are finally:

$$\left[\begin{array}{cc|cc} \tilde{\mathbf{Z}} & \mathbf{M}_{\text{curl}} \mathbf{Z} \mathbf{M}_{ml}^{\Omega T} & 0 & 0 \\ \hline \mathbf{M}_{ml}^{\Omega} \mathbf{Z} \mathbf{M}_{\text{curl}}^T & \mathbf{M}_{ml}^{\Omega} \mathbf{Z} \mathbf{M}_{ml}^{\Omega T} + \mathbf{M}_{ml}^{\text{cir}} \mathbf{Z}_c \mathbf{M}_{ml}^{\text{cir} T} & -\mathbf{M}_{ml}^{\text{I}} & 0 \\ \hline 0 & \mathbf{M}_{ml}^{\text{I} T} & 0 & 0 \\ 0 & \mathbf{M}_{ml}^{\text{U} T} & 0 & \mathbf{I}_d \end{array} \right] \begin{bmatrix} \mathbf{T} \\ \mathbf{I}_m \\ \mathbf{U}_\text{I} \\ \mathbf{I}_\text{U} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{U}} \\ \mathbf{M}_{ml}^{\text{U}} \mathbf{U}_\text{U} \\ -\mathbf{I}_\text{I} \\ 0 \end{bmatrix}. \quad (\text{II.66})$$

This formulation will be referred as the $\mathbf{T} - \mathbf{I}_m$ formulation.

The problem is now resumed in solving a system with a 4×4 blocks matrix. As we repeatedly said previously, we know how to treat efficiently the top left block, but the other blocks are composed of problematical change-of-basis and ‘0’. These additional blocks correspond to the missing loops we had to add. The number of missing loops depends on the geometry of the device. Usually, there are as many missing loops as holes in the geometry. The point is that the number of missing loops plus the number of supplies is usually very small and negligible compared to the number of small loops, i.e.: $\#\mathbf{I}_m + \#\mathbf{U}_\text{I} + \#\mathbf{I}_\text{U} \ll \#\mathbf{T}$.

Therefore, the preconditioning procedure we intend to use consists in applying the \mathcal{H} -matrices to the matrix $\tilde{\mathbf{Z}}$ (any other efficient preconditioner could also be used actually) and using a trivial and light preconditioner for the other diagonal blocks (as Jacobi or identities).

II.3.d Preconditioning

In this subsection, we have to put the works in the PhD context, i.e.: we have investigated this formulation at the end of the PhD, so due to the time constraints, we could not apply the \mathcal{H} -LU. We could however use the classical LU decomposition, giving as a result the maximum reduction of iterations we could get. The computational times are not presented since they are not relevant in that case.

We have considered two cases, one torus and one loop conductor meshed with surface and volume elements, respectively. These are academic examples, but they permit to see the loops completion easily, as well as the maximal efficiency of the preconditioner.

Surface torus example

Let us consider the torus meshed using quadrangles in Figure II.22, with an inner radius of 40mm and the section's radius of 10mm. A current source is plugged on opposite sides, on the inner and outer sides as in Figure II.23. The missing loops found by the circuit solver are also presented in the previous Figure.

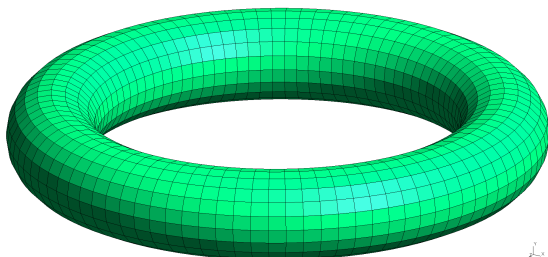


Figure II.22: Torus' surface mesh.

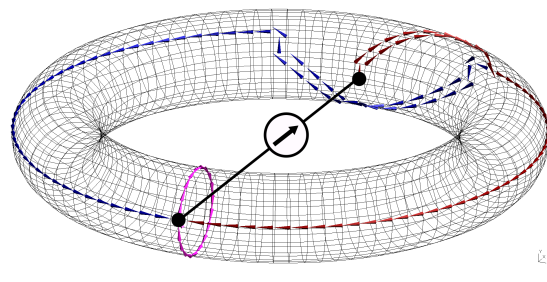


Figure II.23: Supply connection and missing loops.

A torus alone requires two additional loops: one for the middle hole formed by the torus (in blue) and one turning around the section (in magenta). The supply adds another cut (in red) to link the electrical circuit. The circuit solver has found all three, even though their path is not optimal.

We have run a test at a frequency of 100Hz and the resulting current density magnitude is presented in Figure II.24, with the real part of \mathbf{J} in superposition. The validation of the formulation is done comparing the losses obtained with those of the inductive PEEC, the relative error of the $\mathbf{T} - \mathbf{I}_m$ formulation is of about 8%.

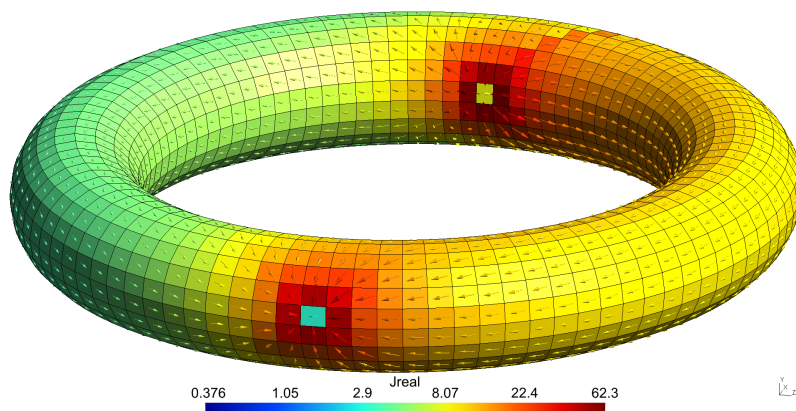


Figure II.24: Magnitude and real part of the current density at $f=100\text{Hz}$.

The preconditioning results are in Table II.1. Using the FGMRes without preconditioner, the problem is solved after 191 iterations. Whereas, when the preconditioner is employed, only 9 iterations are required for the convergence. In comparison, the

inductive PEEC formulation is taking 170 iterations when using the loop analysis without preconditioner. This example proves the efficiency of this approach: preconditioning only the matrix \mathbf{Z} is sufficient to get a significant reduction of iterations.

Table II.1: Number of iterations taken by each approach.

	$\mathbf{T} - \mathbf{I}_m$ formulation		Inductive PEEC
	no precondition.	with precondition.	Loop analysis no precondition.
Nb Iterations	191	9	170

Volume loop conductor example

The second example is a loop conductor with a rectangular section discretized using tetrahedra, see Figure II.25. One current source is connected on the two bottom corners.

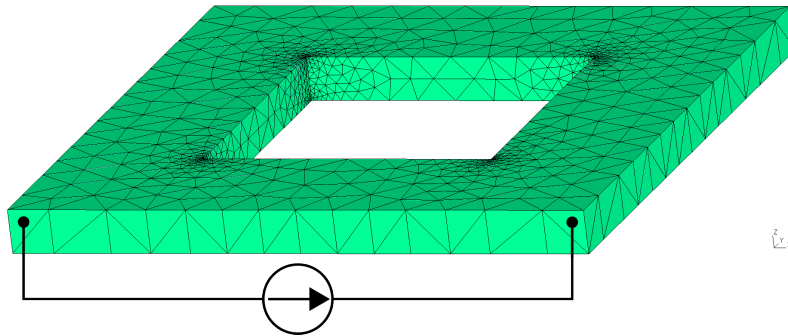


Figure II.25: The loop conductor volume mesh.

Only two missing loops have been found here, one for the hole and the other for the supply connection (see Figure II.26). A solution has been computed at the same frequency, the real part of the result is shown in Figure II.27 (the imaginary part is negligible).

The results are presented in Table II.2. The preconditioning is behaving even better than previously, the number of iterations without preconditioner is of 212, whereas the preconditioned solution took only 4 iterations. This efficiency is increased here most probably due to two factors: a) this problem has more unknowns, so the preconditioned block is bigger, and b) only two missing loops are required here, so the ratio is in favor of the preconditioning.

Finally, we have found a formulation that we are able to precondition efficiently and using the \mathcal{H} -matrices. We were not able to show any demonstration using the \mathcal{H} -matrices, but based on the last results, we can be very optimistic. This

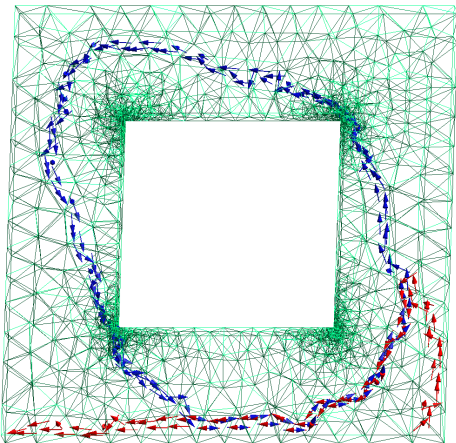


Figure II.26: Cuts resulting from the circuit solver completion.

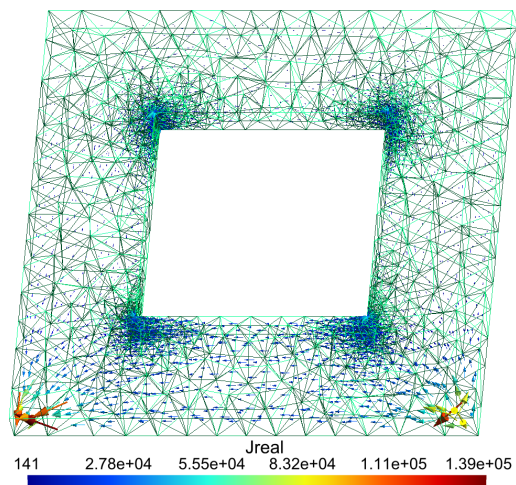


Figure II.27: Real part of the current density vector.

Table II.2: Number of iterations taken by each approach.

	$\mathbf{T} - \mathbf{I}_m$ formulation		Inductive PEEC
	no precondition.	with precondition.	Loop analysis no precondition.
Nb Iterations	212	4	400

formulation is still limited to eddy's currents problems, however, studying the Loop-Star formulations might allow us to add in consideration the capacitive effects.

II.4 Conclusion

In this chapter, we have focused the studies toward eddy's current formulations in order to show the difficulties we have encountered and the solutions we have chosen all along this PhD.

Starting by the inductive PEEC formulations for volume and thin conductors, we introduced the linear systems we have to deal with. Using the loop analysis led to an equation with a change-of-basis, which we did not succeed in inverting.

The nodal approach does not have any change-of-basis, however, a larger block-structure has to be handled. We have tried to reuse our knowledge in block-LU factorization, but the constructed preconditioner has a too high filling factor and thus requires too much memory. Other block solvers have been tested, but the 0's blocks proved to be very problematic.

At the end, we failed to precondition the PEEC formulations. However, we have

not found any content on this subject, so these works could hopefully be useful for further works toward the preconditioning of PEEC formulations, or more precisely of circuit equations.

The last works of this PhD consisted in investigating the \mathbf{T} formulation to try and see if an efficient preconditioner could be constructed. Using the experience acquired while studying the loop approach, we could add a circuit coupling without many difficulties. Finally, we proved that this formulation can be easily preconditioned, since the most important block, in terms of size and physic interpretation, is the one being preconditioned.

Future works might consist in improving this formulation to take in consideration the capacitive effects and the dielectrics, or trying to lower the memory consumption of the block-LU preconditioner used for the nodal approach. An extension of this formulation has been published in [71], taking into account the magnetic materials.

As a remainder, we aim at a formulation taking into account volume and/or thin shell conductors and dielectrics with a solution coupling strongly the magnetic and electric effects. We were able to meet these objectives using the PEEC method during the first year of this PhD. These formulations are presented in the next chapter.

Chapter III

Electromagnetic models considering capacitive effects

Scientific research is one of the most exciting and rewarding of occupations.

Frederick Sanger

CONTENTS

III.1 INTRODUCTION	81
III.2 CAPACITIVE PEEC FORMULATION WITHOUT DIELECTRICS	82
III.2.a PEEC Recalls	82
III.2.b Adding capacitive branches	83
III.2.c Formulation	84
III.2.d Example	85
III.3 MODEL FOR CONDUCTORS AND DIELECTRICS	86
III.3.a Maxwell's equations	87
III.3.b Discretization	88
III.3.c Practical Formulation	91
III.3.d Validation	94
III.4 PEEC-BIM FORMULATION	100
III.4.a BIM coupling	100
III.4.b Conductive-capacitive PEEC formulation	102
III.4.c Coupled BIM-PEEC formulation	103
III.4.d Validation	104
III.5 CONCLUSION	106

Resume

This last chapter introduces the formulations we have developed to model devices made of conductors and dielectrics, while taking into account the resistive, inductive and capacitive effects.

III.1 Introduction

We have presented in the previous chapter three inductive formulations to consider problems with conductors only, but our aim is to treat real devices. The actual technologies are rarely entirely composed of conductors, dielectric materials are often present as insulation and/or as a mechanical support. Considering the constant increase of the operational frequencies, the electromagnetic formulations can no longer neglect the dielectrics. A complete electromagnetic solution thus has to be computed, taking in consideration the strong coupling between the magnetic and the electric effects.

Starting with the inductive PEEC formulation, we will present a way to take into account the capacitive (or electrostatic) effects. In the first capacitive formulation the dielectrics will be neglected, nevertheless, it can be used for problems where the dielectric does not interact significantly. In the case that the dielectric is just a mechanical support for example. Adding the capacitive effects permits to find the resonance frequencies of a device (see Figure III.1), and to study the disturbance of interconnected problems for example.

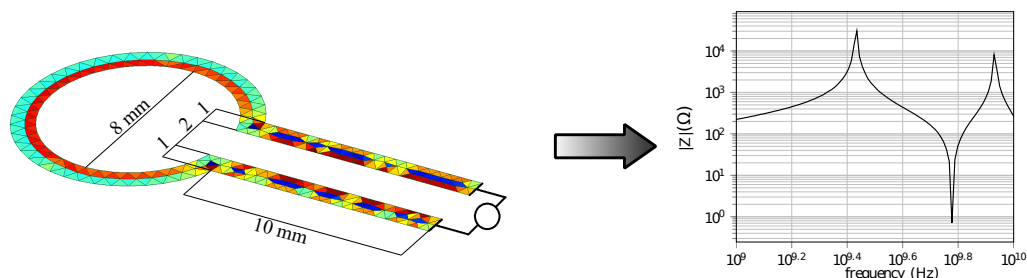


Figure III.1: Example of a loop antenna and its impedance magnitude curve with some resonance frequencies.

In order to satisfy the previous specifications, the dielectric materials consideration will be added in the capacitive formulation, leading to the main formulation of this PhD. Modeling the conductors and dielectrics allows treating problems with a compact geometry which leads to strong interactions between different materials. In particular, the PCB-like applications are very compact and the interactions often occur through the dielectrics, since the conductors are either buried in or separated by a dielectric (an example is in Figure III.2).

A particularity of a PCB is its conductive tracks thickness, for the power electronics devices the printed tracks usually have a thickness lower than $70\mu\text{m}$. Compared to the other dimensions, the thickness is very small. As mentioned in the previous

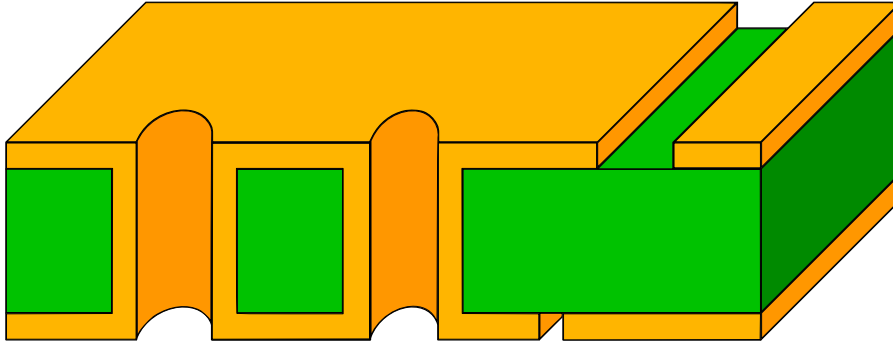


Figure III.2: Example of PCB with vias and tracks in yellow and dielectric insulation in green.

chapter, this kind of geometry gives a mesh with also very thin volume elements and thus, numerical issues may occur during the integral computations. We will see that the thin shell approximation can be applied to the last formulation, so a thin shell formulation can be deduced to treat thin conductors and volume dielectrics, as PCB applications.

The dielectric materials can be very dominant in terms of volume occupation, leading to large problems. To avoid such problems, another formulation will be presented: it mixes a PEEC method and a boundary integral method to model the linear dielectrics using only the boundaries.

III.2 Capacitive PEEC formulation without dielectrics

With the increase of operational frequencies, the eddy currents formulations do not suffice to model the devices at their operating frequencies, in consequence, the electrostatic (or capacitive) effects have to be considered. The inductive PEEC formulations previously introduced can be improved to do so.

III.2.a PEEC Recalls

As a reminder, the inductive PEEC integral equation is the Ampère-Maxwell equation:

$$\frac{\mathbf{J}}{\sigma} + j\omega \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J}}{r} d\Omega = -\mathbf{grad}V. \quad (\text{III.1})$$

Using the face discretization leads to the next circuit equation

$$(\mathbf{R} + j\omega\mathbf{L})\mathbf{I} = \mathbf{U}, \quad (\text{III.2})$$

with the equivalent circuit representation in Figure III.3.

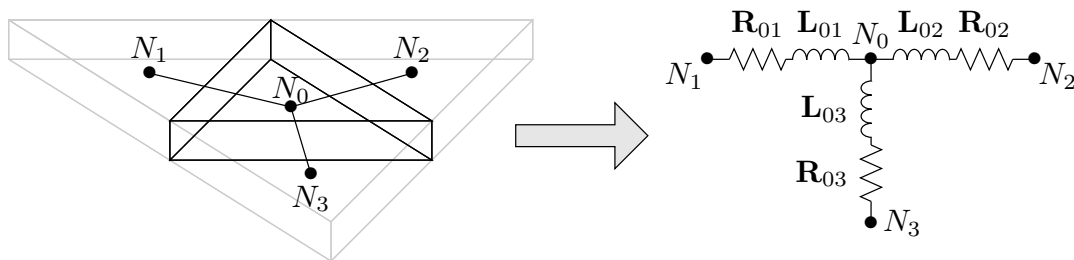


Figure III.3: Equivalent circuit of a mesh-element using a ‘face discretization’.

III.2.b Adding capacitive branches

We can note that the current is not allowed to go out of the volumes, all the branches are contained in the volume elements by construction. Whereas the electrostatic charges are located on the boundaries and can move all around the conductors. Representing such behavior with a circuit approach consists in adding external branches to allow a “current” to flow in and out of the matter, as shown in Figure III.4. The external branches in orange (and dashed) are connected from the boundary faces to a common node “ ∞ ” to allow a current to move onto the border of the conductors, as the charges would do. Note that there are additional inner branches linking the previous inner electric nodes to the border nodes in red.

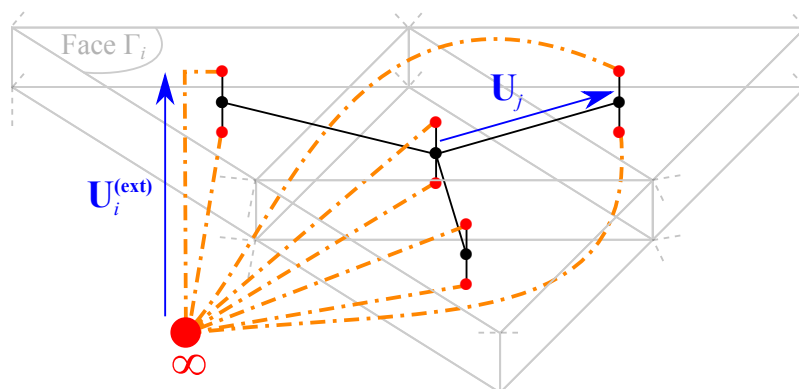


Figure III.4: Example of capacitive PEEC circuit.

Assuming the last circuit representation, the contribution of the new branches has to be expressed. The external branches are capacitive, so we have to look for their contribution.

III.2.c Formulation

In the paragraph [Equivalent circuit representation](#) of [subsection II.2.a](#), we have shown that the RHS vector \mathbf{U} of [\(III.2\)](#) represents the voltage (or the difference of potential) of all branches. We have added new branches linking the external faces to a common point ∞ (see [Figure III.4](#)), so we have to express the difference of potential $\mathbf{U}^{(\text{ext})}$ of these external branches in order to complete the equations. Since they are linked to a border face (thus a circuit node) and the node ∞ , their potential is the difference of voltages between these nodes:

$$\mathbf{U}_i^{(\text{ext})} = V_{\Gamma_i} - V_{\infty}, \quad (\text{III.3})$$

with V_{Γ_i} the potential of the border face i . Setting arbitrarily the voltage of the node ∞ to 0, we get $\mathbf{U}_i^{(\text{ext})} = V_{\Gamma_i}$. Actually, we have already expressed V_{Γ_i} in [\(II.19\)](#):

$$V_{\Gamma_i} = \int_{\Gamma} \frac{V}{S_i} d\Gamma, \quad (\text{III.4})$$

depending on the electric scalar potential V , which can be written as

$$j\omega V = \frac{1}{4\pi\epsilon_0} \int_{\Omega} \mathbf{J} \cdot \mathbf{grad} \left(\frac{1}{r} \right) d\Omega, \quad (\text{III.5})$$

in cases with conductors and no dielectrics. Using the Divergence theorem and knowing that $\text{div}\mathbf{J} = 0$, we get:

$$V = \frac{1}{j\omega\epsilon_0} \int_{\Gamma} \frac{\mathbf{J} \cdot \mathbf{n}}{4\pi r} d\Gamma, \quad (\text{III.6})$$

where \mathbf{n} is the normal to $\Gamma = \partial\Omega$, the boundary of the geometry. Injecting [\(III.6\)](#) in [\(III.4\)](#) and discretizing \mathbf{J} with the face elements gives

$$\mathbf{U}_i^{(\text{ext})} = \frac{1}{j\omega\epsilon_0} \int_{\Gamma} \frac{1}{S_i} \sum_j \left(\int_{\Gamma} \frac{\mathbf{w}_j \cdot \mathbf{n}}{4\pi r} d\Gamma \right) \mathbf{I}_j^C d\Gamma, \quad (\text{III.7})$$

with \mathbf{I}_i^C the current on the external branches. So, [\(III.7\)](#) can be written as:

$$\mathbf{U}^{(\text{ext})} = \frac{1}{j\omega} \mathbf{C} \mathbf{I}^C \quad (\text{III.8})$$

with

$$\mathbf{C}_{i,j} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma} \int_{\Gamma} \frac{1}{S_i S_j} \frac{1}{r} d\Gamma d\Gamma. \quad (\text{III.9})$$

Note that the matrix \mathbf{C} is not a capacitance matrix, it is in fact an elastance matrix (the inverse of \mathbf{C}).

Finally, to complete the equation (III.2) with the capacitive contributions, we just have to add $\mathbf{U}^{(\text{ext})}$:

$$(\mathbf{R} + j\omega\mathbf{L})\mathbf{I} + \frac{1}{j\omega}\mathbf{C}\mathbf{I}^C = \mathbf{U} + \mathbf{U}^{(\text{ext})}. \quad (\text{III.10})$$

The last equation in fact is under the next block-form:

$$\begin{bmatrix} \mathbf{R} + j\omega\mathbf{L} & 0 \\ 0 & \frac{1}{j\omega}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I}^C \end{bmatrix} = \begin{bmatrix} \mathbf{U} \\ \mathbf{U}^{(\text{ext})} \end{bmatrix}. \quad (\text{III.11})$$

For a matter of ease, the concatenated current and voltage vectors will be denoted as \mathbf{I} and \mathbf{U} , respectively, with no differentiation between the inner and outer branches, thus (III.11) can be written as:

$$\left(\mathbf{R} + j\omega\mathbf{L} + \frac{1}{j\omega}\mathbf{C}\right)\mathbf{I} = \mathbf{U}. \quad (\text{III.12})$$

The solution of (III.12) is done using the circuit solver, exactly as in [subsection II.2.a PEEC Formulation](#).

In addition, the thin shell approximation is directly applicable to this formulation, only the circuit representation changes.

III.2.d Example

Considering the example published in [72]: a very thin loop antenna, shown in Figure III.5, we want to validate the capacitive PEEC formulation.

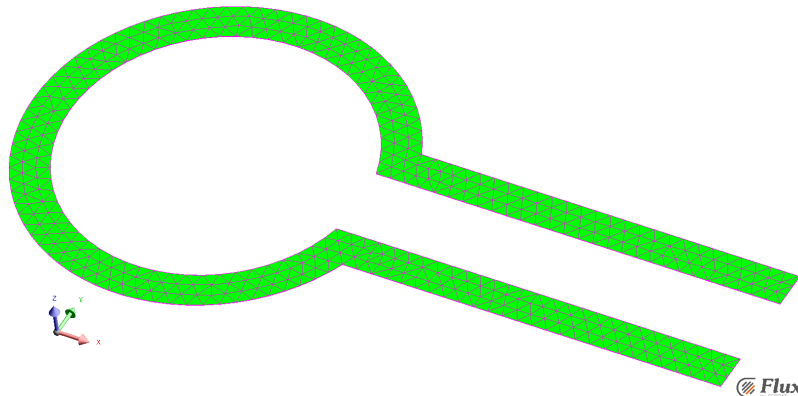


Figure III.5: Mesh of the loop antenna.

This example consists in computing the impedance of the device in the frequency range of 1 to 10 GHz. So, we have computed the impedance magnitude curve using

the thin shell variant of the last formulation, which is denoted as ‘‘Capa. PEEC’’ in Figure III.6. Then, we compare the results to those obtained with another method of the paper. The two curves are greatly matching, but the peaks are approximative. The magnitude of the peaks depends on the resistance of the device, so it could reveal an inaccuracy in the resistive computations, maybe due to the mesh or integral computation or the thin shell approximation.

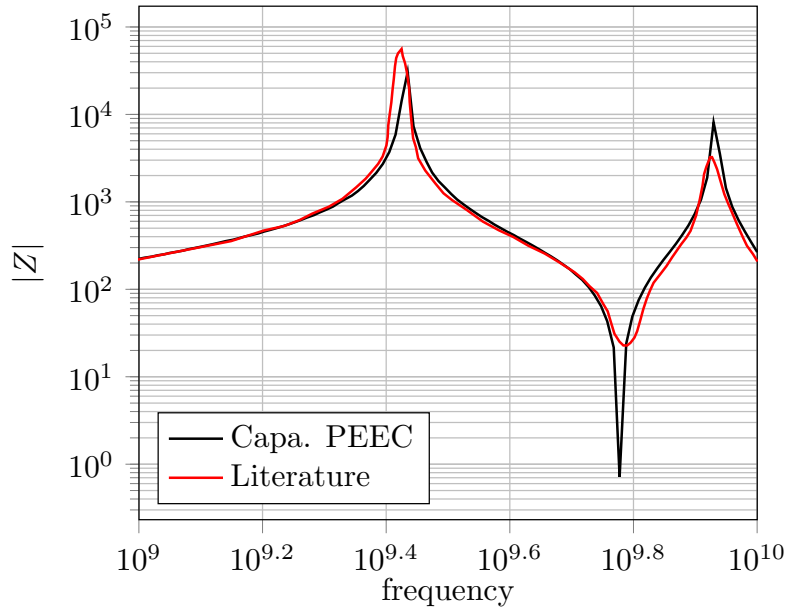


Figure III.6: Impedance curves of the approaches.

However, the results are very similar, so the last formulation is considered as validated.

Here, the capacitive formulation is limited to conductors, thus the next step is to add the dielectric materials in the formulation.

III.3 Model for conductors and dielectrics

To add the dielectrics to the formulation, we first have to start by defining the following regions: Ω_J and Ω_D contain, respectively, the conductors and the dielectrics, and $\Omega = \Omega_J \cup \Omega_D$. Their borders are defined as $\Gamma_J = \partial \Omega_J$, $\Gamma_D = \partial \Omega_D \setminus \Gamma_J$ and $\Gamma = \Gamma_J \cup \Gamma_D$, see Figure III.7.

For a matter of understanding, the dielectric formulation has to be detailed starting from the Maxwell’s equations.

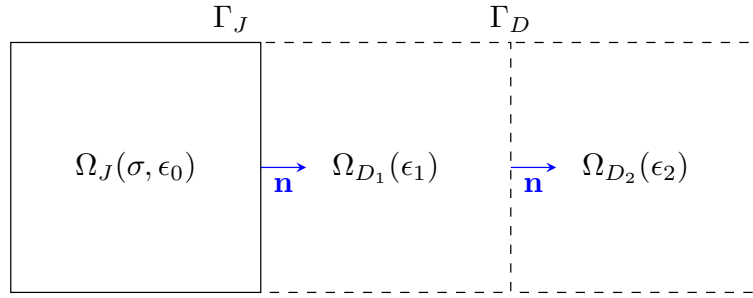


Figure III.7: Interfaces with multiple materials. Γ_J in black continuous lines and Γ_D in dashed lines.

III.3.a Maxwell's equations

In a problem with conductors, dielectrics and without any magnetic material, we have the following constitutive laws

$$\begin{cases} \mathbf{J} = \sigma \mathbf{E} & \text{in } \Omega_J, & \text{(III.13)} \\ \mathbf{D} = \epsilon \mathbf{E} = \epsilon_0 \mathbf{E} + \mathbf{P} & \text{in } \Omega_D, & \text{(III.14)} \\ \mathbf{B} = \mu_0 \mathbf{H} & & \text{(III.15)} \end{cases}$$

where \mathbf{D} the displacement current, \mathbf{P} the polarization density and ϵ the permittivity. According to Maxwell's equations, we get

$$\mathbf{curl} \mathbf{E} = -j\omega \mathbf{B} = -\mu_0 j\omega \mathbf{H}, \quad \text{(III.16)}$$

$$\mathbf{curl} \mathbf{H} = \mathbf{J} + j\omega \mathbf{D} = (\mathbf{J} + j\omega \mathbf{P}) + \epsilon_0 j\omega \mathbf{E}. \quad \text{(III.17)}$$

As previously, we can write the potentials in the presence of dielectrics and without propagation effects using the Lorentz gauge as follows,

$$\mathbf{A} = \frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J} + j\omega \mathbf{P}}{r} d\Omega \quad \text{(III.18)}$$

$$j\omega \mathbf{V} = \frac{1}{4\pi\epsilon_0} \int_{\Omega} (\mathbf{J} + j\omega \mathbf{P}) \mathbf{grad} \left(\frac{1}{r} \right) d\Omega \quad \text{(III.19)}$$

where only the term $j\omega \mathbf{P}$ has been added from the original expression.

The conductive formulations use the current density \mathbf{J} as the unknown. The normal component of the current density is preserved between two elements, so it seems natural to use the face element discretization. This choice also permits overcoming the structured mesh limitation of the classical PEEC.

Now, we want to take into account the dielectric materials, so we have to use another unknown having the same conservative property in order to keep the same benefits.

We choose the total current density \mathbf{J}_{tot} as the new unknown:

$$\mathbf{J}_{tot} = \mathbf{J} + j\omega\mathbf{D} = \sigma^*\mathbf{E} \quad (\text{III.20})$$

with $\sigma^* = \sigma + j\omega\epsilon$, which can be seen as a complex conductivity varying in the matter. \mathbf{J}_{tot} obviously keeps the same benefits in the conductors, and in the dielectric $\mathbf{J}_{tot} = j\omega\mathbf{D}$ verifies the same property. In addition, $\text{div}\mathbf{J}_{tot} = 0$ everywhere by definition, thanks to (III.17).

Thus, this integral equation formulation consists in matching the electrical constitutive law (III.20) everywhere in the materials (conductors and dielectrics) with

$$\mathbf{E} = -j\omega\mathbf{A} - \text{grad}V \quad (\text{III.21})$$

resulting in

$$\frac{\mathbf{J}_{tot}}{\sigma^*} = -j\omega\mathbf{A} - \text{grad}V. \quad (\text{III.22})$$

To express the potential based on \mathbf{J}_{tot} , we use the second constitutive law (III.14) and (III.20) to write

$$\mathbf{J}_{tot} = \mathbf{J} + j\omega\mathbf{P} + j\omega\epsilon_0\mathbf{E} \quad (\text{III.23})$$

so we can deduce

$$\mathbf{J} + j\omega\mathbf{P} = \frac{\sigma^* - j\omega\epsilon_0}{\sigma^*}\mathbf{J}_{tot}. \quad (\text{III.24})$$

And by substituting (III.24) into (III.18) we can write (III.22) as

$$\frac{\mathbf{J}_{tot}}{\sigma^*} + j\omega\frac{\mu_0}{4\pi} \int_{\Omega} \frac{\sigma^* - j\omega\epsilon_0}{\sigma^*} \frac{\mathbf{J}_{tot}}{r} d\Omega = -\text{grad}V. \quad (\text{III.25})$$

III.3.b Discretization

To be able to treat unstructured meshes, we use the same test and trial functions, so, we discretize \mathbf{J}_{tot} using the face elements \mathbf{w}_j as follows:

$$\mathbf{J}_{tot} = \sum_j \mathbf{w}_j I_j \quad (\text{III.26})$$

with I_j the current flowing through the face j .

Finally, applying a standard Galerkin projection to (III.25) with the same functions leads to the matrix system

$$(\mathbf{R}_t + j\omega\mathbf{L}_t) \mathbf{I} = \mathbf{U} \quad (\text{III.27})$$

$$\left\{ \begin{array}{l} \mathbf{R}_{t,i,j} = \int_{\Omega} \frac{\mathbf{w}_i \mathbf{w}_j}{\sigma^*} d\Omega \end{array} \right. \quad (\text{III.28})$$

$$\left\{ \begin{array}{l} \mathbf{L}_{t,i,j} = j\omega \int_{\Omega} \mathbf{w}_i \int_{\Omega} \frac{\sigma^* - j\omega\epsilon_0}{\sigma^*} \frac{\mathbf{w}_j}{r} d\Omega d\Omega \end{array} \right. \quad (\text{III.29})$$

$$\left\{ \begin{array}{l} \mathbf{U}_i = - \int_{\Omega} \mathbf{w}_i \text{grad}V d\Omega \end{array} \right. \quad (\text{III.30})$$

with \mathbf{R}_t a sparse matrix which represents the resistive effect in the conductor and capacitive effects in the dielectrics, and \mathbf{L}_t is the inductive dense matrix. We can see an example of an equivalent circuit representation in Figure III.9 of the mesh in Figure III.8. Note that Figure III.9 shows only the self-element contribution (not the mutuals), but the complete circuit can be easily constructed using [32, Fig. 1 and 2].

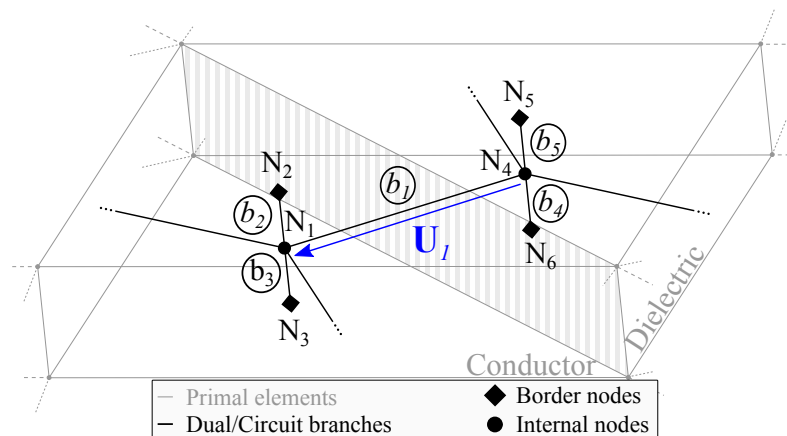


Figure III.8: Example of a topology at the border between a conductor and a dielectric. Each face of the primal mesh corresponds to a dual branch and the black (circuit) nodes correspond to volume elements and border faces.

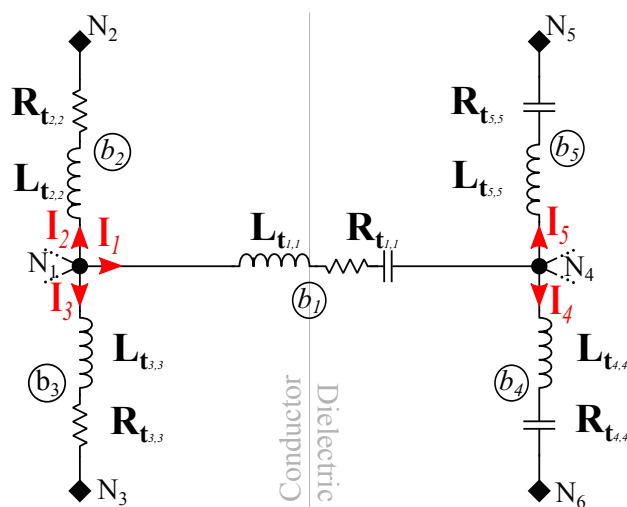


Figure III.9: The simplified equivalent circuit (the mutuals are hidden) of the geometry shown in Figure III.8.

Equation (III.27) corresponds to the incomplete circuit equations ($\mathbf{U} = \mathbf{Z}\mathbf{I}$): the circuit is not closed (see Figure III.8). So, we will have to add new branches on the

border faces and to express their contributions, as the previous formulation. For example, we have to add four capacitive branches in Figure III.9 (linking the nodes N_2 , N_3 , N_5 and N_6 to the common node ∞) in order to close the circuit, as shown in Figure III.10.

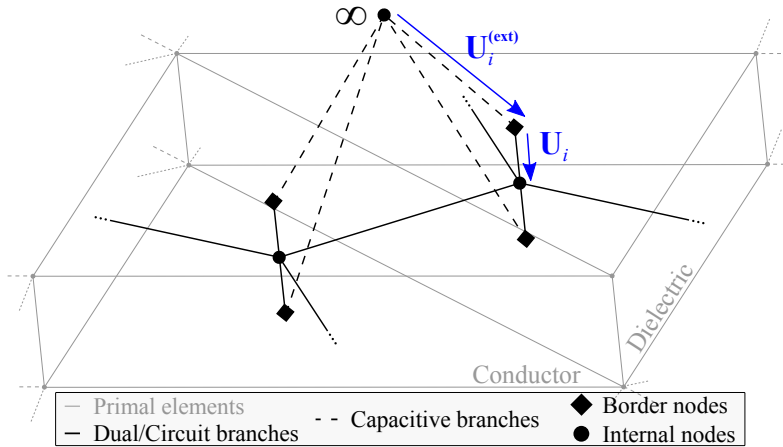


Figure III.10: Adding of the capacitive branches.

To complete the equations, we introduce the common node ∞ with a null potential, so we get

$$\mathbf{U}_j^{(\text{ext})} = V_{\Gamma_j} \quad (\text{III.31})$$

with

$$V_{\Gamma_j} = \int_{\Gamma_j} \frac{V}{S_j} d\Gamma_j. \quad (\text{III.32})$$

In presence of dielectrics, the potential V is given in (III.19), so we get here

$$\mathbf{U}_j^{(\text{ext})} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma} \frac{1}{S_j} \int_{\Omega} \frac{\sigma^* - j\omega\epsilon_0}{j\omega\sigma^*} \mathbf{J}_{\text{tot}} \mathbf{grad} \left(\frac{1}{r} \right) d\Omega d\Gamma \quad (\text{III.33})$$

and using the same discretization procedure, we can write

$$\mathbf{U}^{(\text{ext})} = \frac{1}{j\omega} \mathbf{P}_t \mathbf{I} \quad (\text{III.34})$$

with

$$\mathbf{P}_{t,i,j} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma} \frac{1}{S_i} \int_{\Omega} \frac{\sigma^* - j\omega\epsilon_0}{\sigma^*} \mathbf{w}_j \mathbf{grad} \left(\frac{1}{r} \right) d\Omega d\Gamma \quad (\text{III.35})$$

Here we have managed to express the difference of potential on the border faces, which can be assimilated as capacitive effects. Lastly, we have to solve the equation

$$\left(\mathbf{R}_t + j\omega\mathbf{L}_t + \frac{1}{j\omega} \mathbf{P}_t \right) \mathbf{I} = \mathbf{U}_t. \quad (\text{III.36})$$

with $\mathbf{U}_t = \mathbf{U} + \mathbf{U}^{(\text{ext})}$ the difference of potential of all the branches.

At this point, we have presented a general approach to include dielectric materials and the capacitive effects. This formulation contains matrices with a dependency on the frequency (σ^* and $\frac{\sigma^* - j\omega\epsilon_0}{\sigma^*}$ for instance). This requires assembling each matrix once for each frequency we want to study. So, for a frequency study, this is a critical disadvantage, and necessitates using a more practical formulation.

III.3.c Practical Formulation

In this section, we will present a more practical formulation using the same approach, but making a few more assumptions to overcome the last critical point. The next formulation is fundamentally very similar to the previous one, however, its implementation is significantly improved.

First, we assume that

$$\sigma \gg \omega\epsilon_0 \quad (\text{III.37})$$

so the previous unknown \mathbf{J}_{tot} becomes

$$\tilde{\mathbf{J}}_{tot} = \begin{cases} \mathbf{J} & \text{in } \Omega_J, \\ j\omega\mathbf{D} & \text{in } \Omega_D. \end{cases}$$

In other words, the displacement currents are completely neglected in the conductors.

Following the same protocol as in [subsection III.3.a Maxwell's equations](#), we write the Ampère–Maxwell equation ([III.21](#)) as

$$\mathbf{E} + j\omega\frac{\mu_0}{4\pi} \int_{\Omega} \frac{\mathbf{J} + j\omega\mathbf{P}}{r} d\Omega = -\mathbf{grad}V \quad (\text{III.38})$$

and we can also write

$$\mathbf{J} + j\omega\mathbf{P} = \begin{cases} \mathbf{J} = \tilde{\mathbf{J}}_{tot} & \text{in } \Omega_J \\ j\omega\mathbf{P} = \frac{\epsilon - \epsilon_0}{\epsilon} \tilde{\mathbf{J}}_{tot} & \text{in } \Omega_D \end{cases} \quad (\text{III.39})$$

using the behavior laws. In consequence, ([III.38](#)) can be rewritten as

$$\mathbf{E} + j\omega\frac{\mu_0}{4\pi} \left(\int_{\Omega_J} \frac{\tilde{\mathbf{J}}_{tot}}{r} d\Omega + \int_{\Omega_D} \frac{\epsilon - \epsilon_0}{\epsilon} \frac{\tilde{\mathbf{J}}_{tot}}{r} d\Omega \right) = -\mathbf{grad}V. \quad (\text{III.40})$$

Now, we apply the Galerkin projection with the interpolation functions \mathbf{w}_i :

$$\int_{\Omega} \mathbf{w}_i \mathbf{E} d\Omega + j\omega\frac{\mu_0}{4\pi} \int_{\Omega} \mathbf{w}_i \left(\int_{\Omega_J} \frac{\tilde{\mathbf{J}}_{tot}}{r} d\Omega + \int_{\Omega_D} \frac{\epsilon - \epsilon_0}{\epsilon} \frac{\tilde{\mathbf{J}}_{tot}}{r} d\Omega \right) d\Omega = - \int_{\Omega} \mathbf{w}_i \cdot \mathbf{grad}V d\Omega \quad (\text{III.41})$$

and we can easily see from the first two constitutive laws that

$$\mathbf{E} = \begin{cases} \frac{\tilde{\mathbf{J}}_{tot}}{\sigma} & \text{in } \Omega_J, \\ \frac{\tilde{\mathbf{J}}_{tot}}{j\omega\epsilon} & \text{in } \Omega_D. \end{cases}$$

Using the same discretization as (III.26), we finally have

$$\left(\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}} + \frac{1}{j\omega}\tilde{\mathbf{C}} \right) \mathbf{I} = \mathbf{U}. \quad (\text{III.42})$$

with

$$\tilde{\mathbf{R}}_{i,j} = \int_{\Omega_J} \frac{\mathbf{w}_i \mathbf{w}_j}{\sigma} d\Omega \quad (\text{III.43})$$

$$\tilde{\mathbf{C}}_{i,j} = \int_{\Omega_D} \frac{\mathbf{w}_i \mathbf{w}_j}{\epsilon} d\Omega \quad (\text{III.44})$$

$$\tilde{\mathbf{L}}_{i,j} = \frac{\mu_0}{4\pi} \int_{\Omega} \mathbf{w}_i \left(\int_{\Omega_J} \frac{\mathbf{w}_j}{r} d\Omega + \int_{\Omega_D} \frac{\epsilon - \epsilon_0}{\epsilon} \frac{\mathbf{w}_j}{r} d\Omega \right) d\Omega \quad (\text{III.45})$$

We can see that we have one more matrix than for (III.27) ($\tilde{\mathbf{C}}$). In fact, the previous matrix \mathbf{R}_t has been simplified to $\tilde{\mathbf{R}} + \frac{1}{j\omega}\tilde{\mathbf{C}}$. So, even if we have added a matrix, the actual amount of memory needed has been greatly reduced, because \mathbf{R}_t is complex while $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{C}}$ are real matrices, and \mathbf{R}_t computes coupling terms between conductors and dielectrics. We can make the same remark about $\tilde{\mathbf{L}}$ and \mathbf{L}_t , so the whole storage has been globally divided by 4 (if the number of dielectric and conductor unknowns are equal).

More importantly, we have succeeded in overcoming the previous issue, indeed, the matrices are no longer depending on the frequency.

At this point, we still have to complete the circuit equation to close the equivalent circuit, and to take into account the capacitive effects. The protocol is exactly the same, so let's begin with (III.31):

$$\mathbf{U}_i^{(\text{ext})} = V_{\Gamma_i} = \int_{\Gamma_i} \frac{V}{S_i} d\Gamma_i \quad (\text{III.46})$$

Using (III.39) in (III.19) leads to a separation of the regions as

$$\mathbf{U}_i^{(\text{ext})} = \frac{1}{4\pi\epsilon_0} \frac{1}{j\omega} \int_{\Gamma} \frac{1}{S_i} \left(\int_{\Omega_J} \tilde{\mathbf{J}}_{tot} \nabla \frac{1}{r} d\Omega + \int_{\Omega_D} \frac{\epsilon - \epsilon_0}{\epsilon} \tilde{\mathbf{J}}_{tot} \nabla \frac{1}{r} d\Omega \right) d\Gamma. \quad (\text{III.47})$$

In this case, we can simplify the first twofold integral:

$$\int_{\Gamma} \frac{1}{S_i} \int_{\Omega_J} \tilde{\mathbf{J}}_{tot} \nabla \frac{1}{r} d\Omega d\Gamma = \int_{\Gamma} \frac{1}{S_i} \int_{\Gamma_J} \frac{\epsilon_0}{\epsilon_j} (\tilde{\mathbf{J}}_{tot} \cdot \mathbf{n}) \frac{1}{r} d\Gamma d\Gamma \quad (\text{III.48})$$

using the Divergence Theorem and knowing that $\text{div} \tilde{\mathbf{J}}_{tot} = 0$. The permittivity constant ϵ_j corresponds to the permittivity there might have on the opposite side

of the border face j , i.e: if the face j is shared between a conductor and a dielectric with the permittivity ϵ_1 , then $\epsilon_j = \epsilon_1$.

In addition, assuming that the dielectrics are linear (which implies $\text{div}\mathbf{P} = 0$ in each dielectric of Ω_D), we can reapply the Divergence Theorem to simplify the second twofold integral:

$$\int_{\Gamma} \frac{1}{S_i} \int_{\Omega_D} \frac{\epsilon - \epsilon_0}{\epsilon} \tilde{\mathbf{J}}_{tot} \nabla \frac{1}{r} d\Omega d\Gamma = \int_{\Gamma} \frac{1}{S_i} \int_{\Gamma_D} \frac{(\epsilon_1 - \epsilon_2)\epsilon_0}{\epsilon_1\epsilon_2} (\tilde{\mathbf{J}}_{tot} \cdot \mathbf{n}) \frac{1}{r} d\Gamma d\Gamma \quad (\text{III.49})$$

with ϵ_1 and ϵ_2 correspond to the permittivity of the material \mathbf{n} comes out from and goes into, respectively, as in Figure III.7. For example, if \mathbf{n} is going out of a FR4 dielectric to go in the air, then $\epsilon_1 = \epsilon_{FR4}$ and $\epsilon_2 = \epsilon_0$.

Now we use the same discretization on $\tilde{\mathbf{J}}_{tot}$ and we can write

$$\mathbf{U}^{(\text{ext})} = \frac{1}{j\omega} \tilde{\mathbf{P}}\mathbf{I} \quad (\text{III.50})$$

with

$$\tilde{\mathbf{P}}_{i,j} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma} \frac{1}{S_i} \left(\int_{\Gamma_J} \frac{1}{S_j} \frac{\epsilon_0}{\epsilon_j} \frac{1}{r} d\Gamma + \int_{\Gamma_D} \frac{(\epsilon_1 - \epsilon_2)\epsilon_0}{\epsilon_1\epsilon_2} \frac{1}{S_j} \frac{1}{r} d\Gamma \right) d\Gamma. \quad (\text{III.51})$$

Here, we have managed to complete the circuit equations with a matrix considering only boundaries. This matrix is much smaller than \mathbf{P}_t , so it requires less memory and can be assembled faster.

At the end, we have the next equation

$$\left(\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}} + \frac{1}{j\omega} (\tilde{\mathbf{C}} + \tilde{\mathbf{P}}) \right) \mathbf{I} = \mathbf{U}_t \quad (\text{III.52})$$

to solve. We can note that the matrices do not have any dependency on the frequency, so we have overcome the last critical point. Moreover, we managed to save more a significant amount of memory. Note that the matrices are purely real, so, using real-assembly codes permits lowering the assembly time.

Moreover, the full matrices ($\tilde{\mathbf{L}}$ and $\tilde{\mathbf{P}}$) are classic in the theory of integral equations and can be compressed, which could allow us to solve bigger problems.

The shell approximation is also directly applicable to this formulation. As previously, the only change (apart from the processes related to the mesh) is the conductivity that is either scaled by the thickness e or eG (defined in [subsection II.2.b Shell PEEC-formulation](#)) depending on the case.

The validation of the volume and thin shell variant of the practical formulation is done next, considering two devices.

III.3.d Validation

The validation of the practical formulation is done comparing the impedances obtained for two examples: a micro-coil and a real device. The first example will compare our results to those from an industrial software, and the second will consist in comparing our results against measurements.

III.3.d-i Case 1: Micro-coil

The considered micro-coil (see Figure III.11) is made of three conductive spires with a dielectric between the I/O. This validation is done comparing the impedances from the formulation and a development version of an industrial finite element software: Flux®3D [73]. Flux3D takes advantage of the symmetry plane on XY , while our integral code can't. So, the results will show only a half of the device on Figure III.12a and Figure III.12b, even though the practical formulation used the entire mesh.

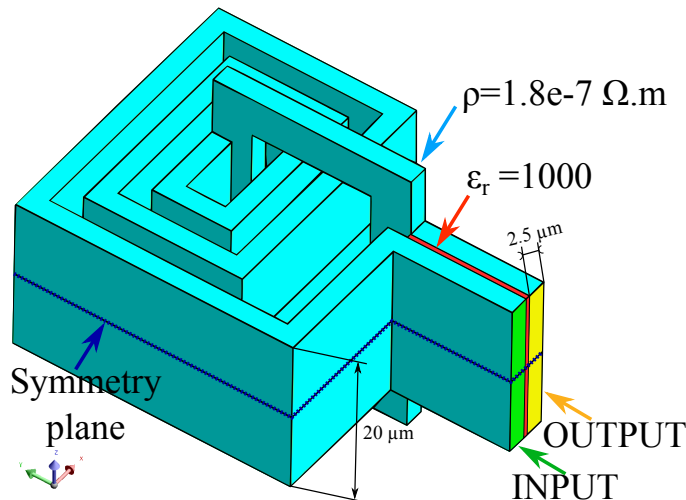


Figure III.11: Studied device: Micro-coil with 3 spires.

For a matter of ease, the Flux3D uses a structured mesh for the active regions (conductor and dielectric) composed of 9,579 hexahedra and an usual mesh is used in the air with 92,049 elements. For the proposed formulation, we used a mesh of 4,822 tetrahedra.

The solution of this formulation was done using a FGMRes solver and a Block LU preconditioner. The FEM solutions used an ICCG solver, so the computational time cannot be fairly compared.

The numerical results are presented in Figure III.13a for the impedance magnitude and in Figure III.13b for the impedance phase. In the last two figures, “FEM”

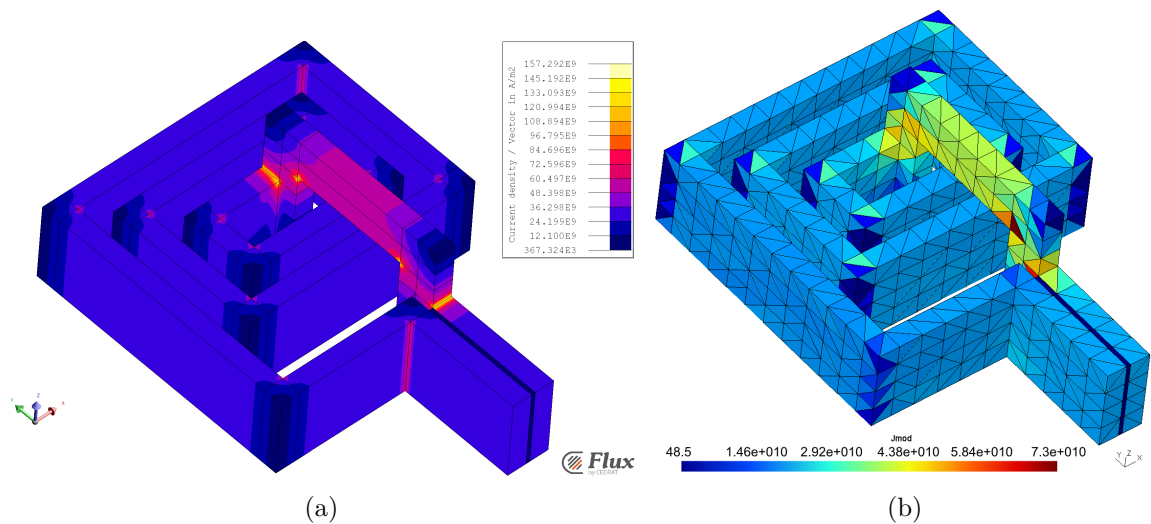


Figure III.12: (a) $|\tilde{\mathbf{J}}_{tot}|$ isovalues at 100 MHz computed using the FEM. (b) $|\tilde{\mathbf{J}}_{tot}|$ using the “practical formulation”.

stands for the results provided by Flux®3D, which are published in [74].

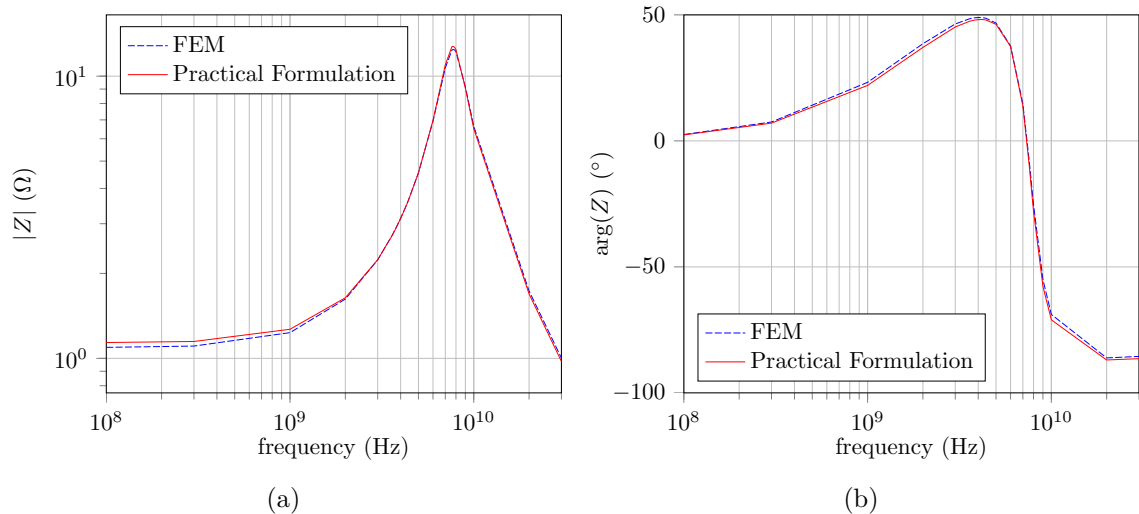


Figure III.13: (a) Impedance magnitude versus frequency. (b) Impedance phase versus frequency. In blue (dashed): The reference (FEM), in red: the simulation results.

The curves look very similar but according to the IEEE standard [75], the Feature Selective Validation (FSV) techniques have to be used to compare impartially two curves. These curves have been qualified as an excellent matching. The details of the validation of the curves are presented in Appendix A.

III.3.d-ii Case 2: Real device

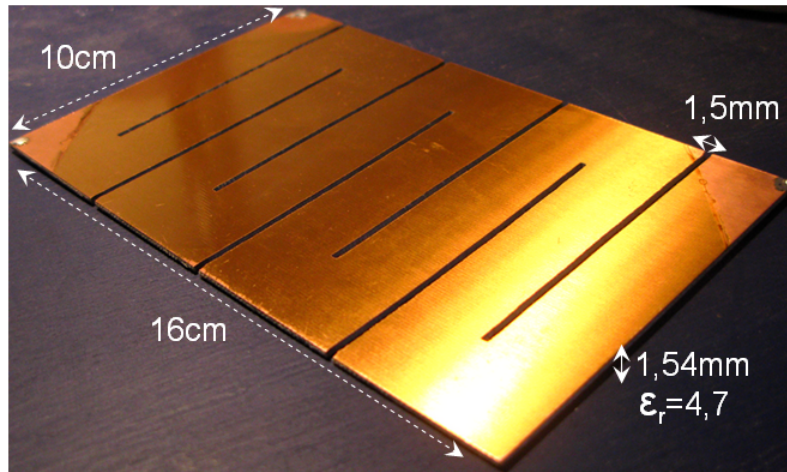


Figure III.14: Picture and dimensions of the studied device.

Let's now consider a real device made of two layers of copper 35 μm thick separated by a layer of dielectric FR4 with a thickness of 1.47 mm (Figure III.14). Two cases have been considered to do the measurements: the short-circuit case consisting of connecting each layer by a corner (see Figure III.15) and the open-circuit case which consists in leaving the layers without a connection between them (see Figure III.16). Both cases are considered from 1 MHz to 100 MHz.

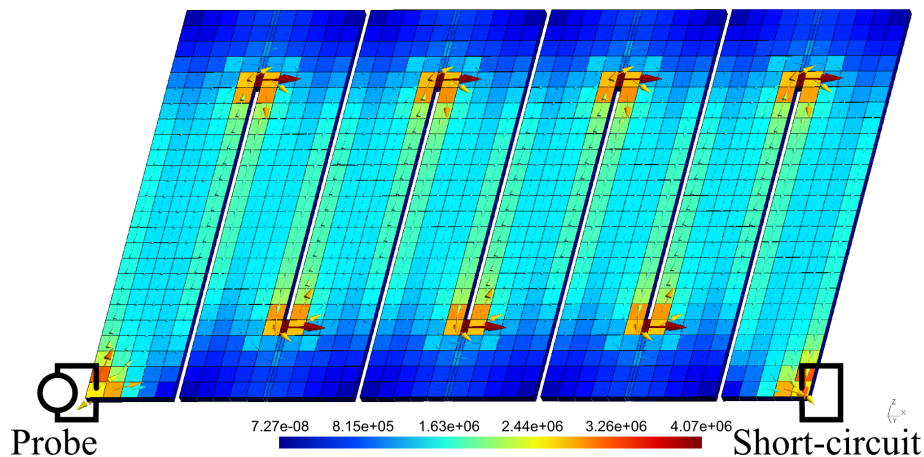


Figure III.15: The current density isovalues of the short-circuit case at 1 MHz, with the circuit connections drawn. The magnitude $|\tilde{\mathbf{J}}_{tot}|$ and the real part $\Re(\tilde{\mathbf{J}}_{tot})$ of the current density are represented by the colored elements and vectors, respectively.

The same mesh has been used for all the studies and is made of 3,210 hexahedra which leads to 16,951 faces and 8,148 independent loops (the degrees of freedom).

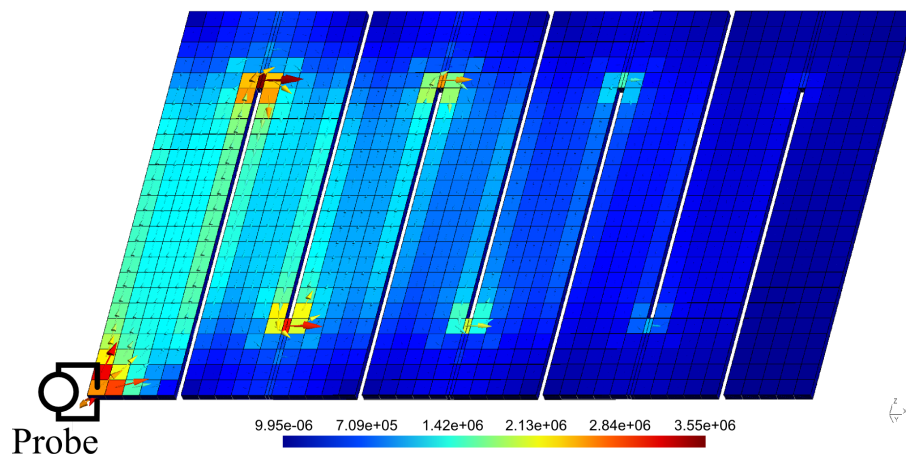


Figure III.16: The current density isovalues of the open-circuit case at 1 MHz, with the circuit connections drawn.

We are using a structured mesh because the geometry is well-suited and the conductors are very thin, but an unstructured mesh could also have been used. The thickness of each layer is discretized with only one element, because the skin depth is greater than the thickness for the studied frequencies.

The solutions were done using a classical Lower–Upper (LU) factorization, due to the ill-conditioning of the problem. The computations were done on the same laptop as in the previous subsection. The numerical results are presented in Figure III.17a and Figure III.17b for the short-circuit and the open circuit case, respectively.

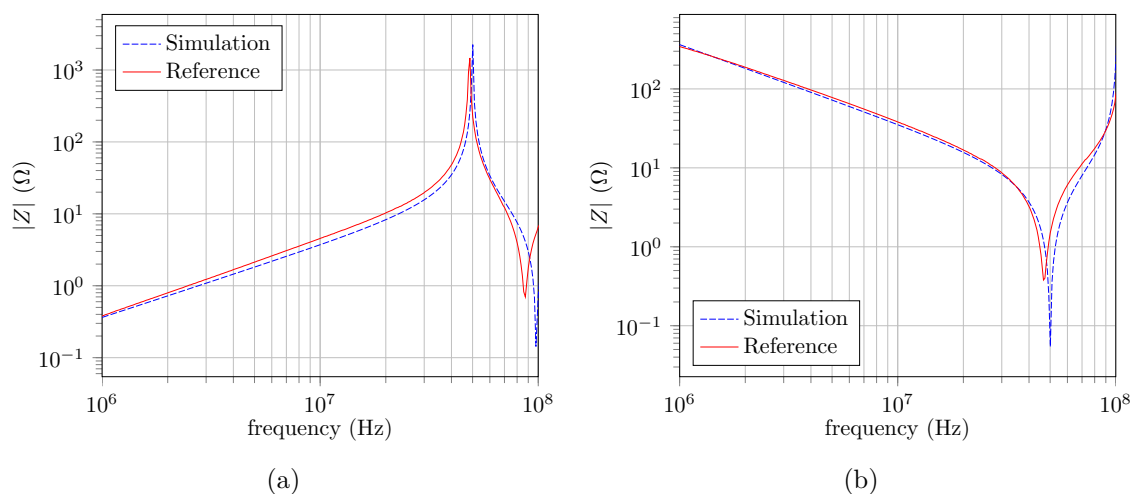


Figure III.17: (a) Impedance magnitude versus frequency for the short-circuit case. (b) Impedance magnitude versus frequency for the open-circuit case. In blue (dashed): the reference (measurements), in red: the simulation results.

On the one hand, both cases show a good accuracy regarding the first peak

frequency. On the other hand, the amplitude of the peaks are underestimated. But overall, we see a good match between the curves. As for the previous example, the FSV techniques have been applied to these curves, and their match have been rated as good to excellent. The details are also in [Appendix A](#).

This device has the properties to be considered as a PCB application. The two layers of conductors are very thin compared to the other dimensions. The next study case will focus on comparing the volume and thin shell formulations.

III.3.d-iii Case 3: Volume vs. Thin shell formulation

Considering the same device as for the previous case, the thin shell variant of the dielectric formulation has been used. This time the range of frequency considered is from 1MHz to 1GHz, so the formulations are tested to their limits since the propagation effects should appear beyond 200 MHz. The short-circuit and the open-circuit case results are presented in [Figure III.18](#) and [Figure III.19](#), respectively.

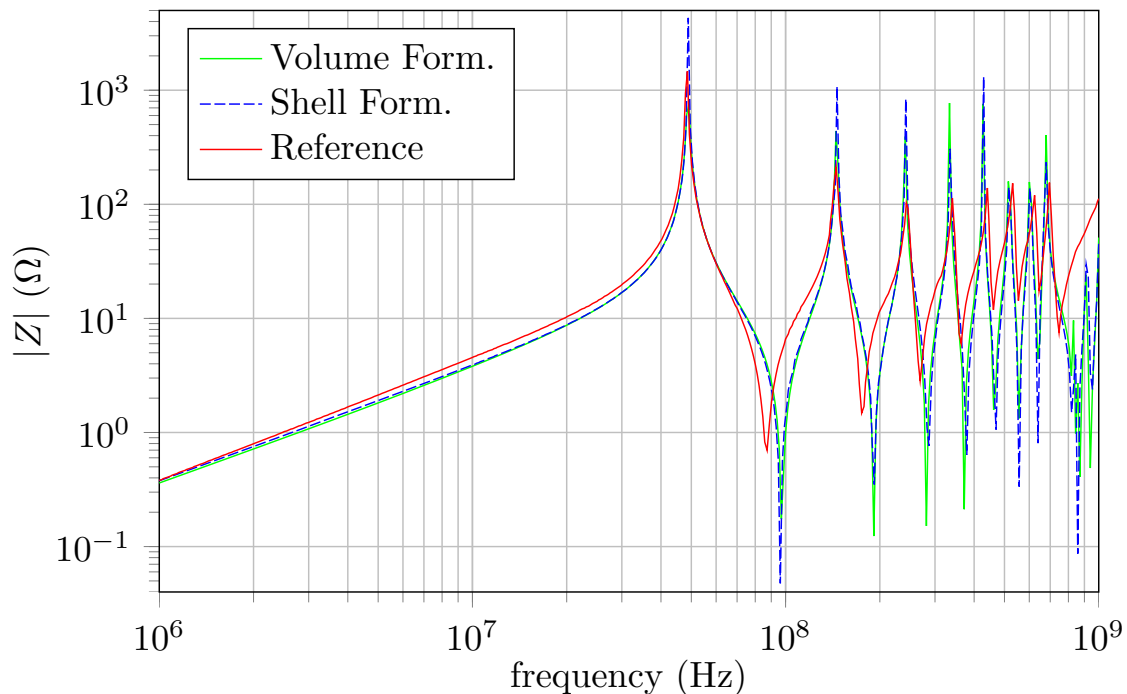


Figure III.18: Impedance magnitude curves for the short-circuit case.

The aim is to study the thin shell approximation and compare it with the volume approach. Both cases show an almost perfect match between the volume and the thin shell formulation, except on the peaks' magnitude. Thus, the approximation does not suffer of significant losses of accuracy here.

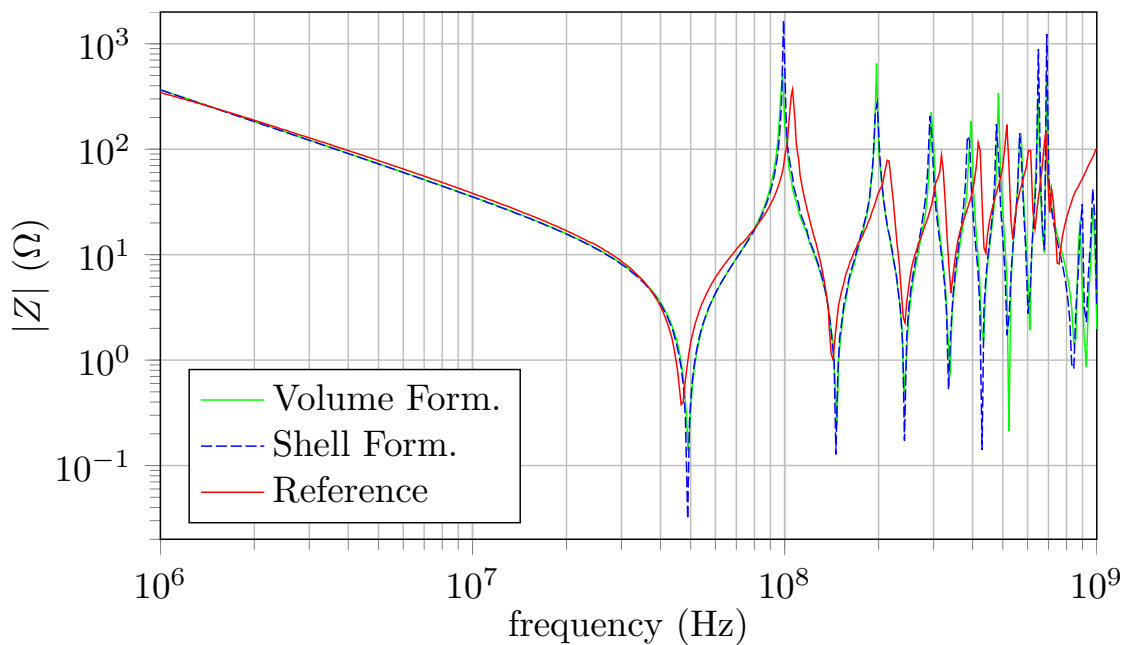


Figure III.19: Impedance magnitude curves for the open-circuit case.

The same discretization as for the volume results has been used, leading here to 10,920 branches or faces. Comparing to the volume approach, there is a great reduction: 10,920 instead of 16,951. Regarding the number of unknowns, there are 7,271 independent loops here, instead of 8,148 for the volume approach. The reduction is lower for the loops due to the circuit representation, the surface approach mainly takes off the branches corresponding to the thickness.

The solution times are presented in Table III.1. The time to compute the solution (for one frequency) is reduced to about 378.3 seconds, instead of 474.8 sec. for the volume approach. The thin shell formulation allows us to lower computational time by $\sim 20\%$, but, we can also point out that the memory usage greatly decreased, since the number of branches decreased of about 35%.

Table III.1: Time taken to compute the solution by approach.

Formulation	Volume	Thin Shell
Solution Time per frequency (in sec.)	474.8	378.3

The main formulation of this PhD has been validated through the study of the last two devices. The thin shell variant of the practical formulation showed a great accuracy and improvements on the PCB device.

Another particularity of such device is that the dielectrics are very dominant in terms of volume, thus their consideration can become problematical. The next formulation allows to significantly reduce the number of dielectric-unknowns.

III.4 PEEC-BIM formulation

To reduce the unknowns linked to the dielectrics, this formulation provides a solution to model the dielectric without needing its volume, thanks to a coupling between the conductive PEEC and a boundary integral method (BIM).

Let us start with the inductive PEEC presented in [subsection II.2.a](#), consisting in matching the constitutive law $\mathbf{J} = \sigma\mathbf{E}$ with the Ampère-Maxwell equation:

$$\mathbf{E} = -j\omega\mathbf{A} - \text{grad}V. \quad (\text{III.53})$$

The coupling models the conductive materials thank to the PEEC method, whereas the dielectrics are taken into account through a boundary integral method. The last method here consists in expressing the contribution of the dielectric volume using only its boundary, so we want to write the Ampère-Maxwell equation on a dielectric boundary.

III.4.a BIM coupling

Considering an interface Γ_D between two materials with linear permittivities ϵ_1 and ϵ_2 , as in [Figure III.7](#). This interface can correspond to a separation between a dielectric ($\epsilon_1 > \epsilon_0$) and either another dielectric ($\epsilon_2 > \epsilon_0$) or the air ($\epsilon_2 = \epsilon_0$).

The electric field normal component $E_n = \mathbf{E} \cdot \mathbf{n}$ is not continuous through interfaces, only its tangential component is preserved. To write the Ampère-Maxwell on Γ_D , we will have to use an average value of the normal field E_n . From the conservation of D_n through Γ_D , we deduce

$$\epsilon_1 E_{n_1} = \epsilon_2 E_{n_2}. \quad (\text{III.54})$$

Knowing that $\mathbf{D} = \epsilon_0\mathbf{E} + \mathbf{P}$, we can also deduce that

$$\partial P_n = P_{n_1} - P_{n_2} = \epsilon_0 (E_{n_2} - E_{n_1}) \quad (\text{III.55})$$

using [\(III.54\)](#) in the last equation gives:

$$E_{n_2} \left(1 - \frac{\epsilon_2}{\epsilon_1}\right) = \frac{\partial P_n}{\epsilon_0}. \quad (\text{III.56})$$

The average electric field (denoted by E_n) is then given by:

$$E_n = \frac{E_{n_2} + E_{n_1}}{2} = \frac{1}{2} \left(\frac{\epsilon_1}{\epsilon_2} + 1 \right) E_{n_2} = \frac{1}{2\epsilon_0} \frac{\epsilon_1 + \epsilon_2}{\epsilon_1 - \epsilon_2} \partial P_n. \quad (\text{III.57})$$

The Ampère-Maxwell equation can be averaged on the boundary as:

$$E_n = -j\omega A_n - (\mathbf{grad}V) \cdot \mathbf{n}. \quad (\text{III.58})$$

Here, an important hypothesis is stated: the vector potential \mathbf{A} contribution is neglected in the dielectric regions. The magnetic effects are negligible compared to the capacitive effects, since

$$\frac{\frac{1}{j\omega} \mathbf{C}_{i,j}}{j\omega \mathbf{L}_{i,j}} = \mathcal{O} \left(\frac{1}{\omega^2 \epsilon \mu_0 d^2} \right) \gg 1$$

and $\epsilon_0 \mu_0 \approx 1.79 \times 10^{-19}$, with d a characteristic dimension. This hypothesis is therefore acceptable in a limited range of frequencies.

The equation (III.58) then comes down to:

$$E_n = -(\mathbf{grad}V) \cdot \mathbf{n}. \quad (\text{III.59})$$

The scalar potential V is continuous but $\mathbf{grad}V$ is not, so we also have to compute its average value. As a remainder, the potential V is defined in (III.19) and in subsection III.3.c Practical Formulation, we saw that the potential can be expressed using only the boundaries, thanks to the material linearity, as:

$$j\omega V = \frac{1}{4\pi\epsilon_0} \left[\int_{\Gamma_J} \frac{\epsilon_0}{\epsilon} \frac{J_n}{r} d\Gamma + j\omega \int_{\Gamma_D} \frac{\partial P_n}{r} d\Gamma \right]. \quad (\text{III.60})$$

Only the term ‘ r ’ (with $r = \|\mathbf{x} - \mathbf{y}\|$) depends on the testing point of V , so the average value at the testing point \mathbf{x} is given by

$$\begin{aligned} (\mathbf{grad}_x V(\mathbf{x})) \cdot \mathbf{n}_x &= \frac{1}{4\pi\epsilon_0} \left[\frac{1}{j\omega} \int_{\Gamma_J} \frac{\epsilon_0}{\epsilon} J_n(\mathbf{y}) \mathbf{grad}_x \left(\frac{1}{r} \right) \cdot \mathbf{n}_x d\Gamma \right. \\ &\quad \left. + \int_{\Gamma_D} \partial P_n(\mathbf{y}) \mathbf{grad}_x \left(\frac{1}{r} \right) \cdot \mathbf{n}_x d\Gamma \right] \quad (\text{III.61}) \end{aligned}$$

with \mathbf{y} the integration point. J_n is discretized using the face elements (as previously) $J_n = \sum_j (\mathbf{w}_j \cdot \mathbf{n}) I_j$, with $(\mathbf{w}_j \cdot \mathbf{n}) = 1/S_j$. The polarization density is here only supported by the elements of Γ_D and we use 0-order interpolation functions (constant by elements) φ_i for its discretization: $(\partial \mathbf{P} \cdot \mathbf{n}) = \sum_j \varphi_j (\delta P_{n_j})$. We are using the next functions:

$$\varphi_j = \begin{cases} 1 & \text{on } \Gamma_{Dj}, \\ 0 & \text{elsewhere.} \end{cases}$$

Projecting the Ampère-Maxwell equation on Γ_D using the last shape functions finally gives the next integral equation:

$$\int_{\Gamma_D} \frac{1}{2\epsilon_0} \frac{\epsilon_1 + \epsilon_2}{\epsilon_1 - \epsilon_2} \frac{1}{S_i} d\Gamma Q_i = \int_{\Gamma_D} \frac{\mathbf{n}_i}{4\pi\epsilon_0} \sum_j \left[\int_{\Gamma_D} \mathbf{grad}_y \left(\frac{1}{r} \right) \frac{1}{S_j} d\Gamma \right] Q_j d\Gamma + \frac{1}{j\omega} \int_{\Gamma_D} \frac{\mathbf{n}_i}{4\pi\epsilon_0} \sum_j \left[\int_{\Gamma_j} \frac{\epsilon_0}{\epsilon} \mathbf{grad}_y \left(\frac{1}{r} \right) \frac{1}{S_j} d\Gamma \right] \mathbf{I}_j d\Gamma \quad (\text{III.62})$$

with $\delta P_{n_j} = Q_j/S_j$. Note that we used $\mathbf{grad}_y \left(\frac{1}{r} \right) = -\mathbf{grad}_x \left(\frac{1}{r} \right)$ to invert the sign and to correspond to the integral computation.

Here the contributions of the dielectrics are computed using only the conductive and dielectric boundaries, adding new surface unknowns for the dielectrics (i.e: the charges Q). The boundary integral equation (III.62) can also be written as:

$$(\mathbf{F} + \mathbf{A}) \mathbf{Q} + \frac{1}{j\omega} \mathbf{B} \mathbf{I} = 0, \quad (\text{III.63})$$

with

$$\left\{ \begin{array}{l} \mathbf{F}_{i,j} = \frac{-\delta_{i,j} \epsilon_1 + \epsilon_2}{2\epsilon_0 \epsilon_1 - \epsilon_2}, \end{array} \right. \quad (\text{III.64})$$

$$\left\{ \begin{array}{l} \mathbf{A}_{i,j} = \int_{\Gamma_D} \frac{\mathbf{n}_i}{4\pi\epsilon_0} \int_{\Gamma_D} \mathbf{grad}_y \left(\frac{1}{r} \right) \frac{1}{S_j} d\Gamma d\Gamma, \end{array} \right. \quad (\text{III.65})$$

$$\left\{ \begin{array}{l} \mathbf{B}_{i,j} = \int_{\Gamma_D} \frac{\mathbf{n}_i}{4\pi\epsilon_0} \int_{\Gamma_j} \frac{\epsilon_0}{\epsilon} \mathbf{grad}_y \left(\frac{1}{r} \right) \frac{1}{S_j} d\Gamma d\Gamma. \end{array} \right. \quad (\text{III.66})$$

We have mentioned at the beginning that the conductors are modeled using the PEEC method, so we are investigating the changes this coupling brings.

III.4.b Conductive-capacitive PEEC formulation

As a remainder, in order to add into account the capacitive effects into the inductive PEEC formulation, we had to add capacitive branches and express their contribution. To determine the capacitive contributions we have investigated the meaning of the RHS vector and concluded that they are given by the average potential of the external faces $\mathbf{U}^{(\text{ext})}$. In presence of dielectrics, these capacitive contributions change, so, we are investigating the other contributions we have to add.

The average potential on the conductors' boundary is given in (III.46) by

$$\mathbf{U}_i^{(\text{ext})} = \int_{\Gamma_i} \frac{V}{S_i} d\Gamma_i. \quad (\text{III.67})$$

The potential V is defined in (III.60), so we have the capacitive contributions expressed by:

$$\mathbf{U}_i^{(\text{ext})} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma_j} \frac{1}{S_i} \left[\frac{1}{j\omega} \int_{\Gamma_j} \frac{\epsilon_0}{\epsilon} \frac{J_n}{r} d\Gamma + \int_{\Gamma_D} \frac{\delta P_n}{r} d\Gamma \right] d\Gamma. \quad (\text{III.68})$$

Using the same discretization for J_n and δP_n , we get the next matrix equation:

$$\mathbf{U}^{(\text{ext})} = \frac{1}{j\omega} \mathbf{P}_c \mathbf{I} + \mathbf{P}_d \mathbf{Q} \quad (\text{III.69})$$

representing the capacitive effects, with:

$$\left\{ \begin{array}{l} \mathbf{P}_{c_{i,j}} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma_J} \frac{1}{S_i} \int_{\Gamma_J} \frac{\epsilon_0}{\epsilon S_j} \frac{1}{r} d\Gamma d\Gamma, \\ \mathbf{P}_{d_{i,j}} = \frac{1}{4\pi\epsilon_0} \int_{\Gamma_J} \frac{1}{S_i} \int_{\Gamma_D} \frac{1}{S_j} \frac{1}{r} d\Gamma d\Gamma. \end{array} \right. \quad (\text{III.70})$$

$$\quad (\text{III.71})$$

The capacitive effects coming from the conductors are represented by the matrix \mathbf{P}_c , which is very similar to the matrix \mathbf{C} from the [subsection III.2.c Formulation](#), except that the external permittivity is taken into account with the term ‘ ϵ_0/ϵ ’. Note the additional matrix \mathbf{P}_d , representing the contribution of the dielectric boundaries to the capacitive effects.

The PEEC integral equation here is

$$\left(\mathbf{R} + j\omega\mathbf{L} + \frac{1}{j\omega} \mathbf{P}_c \right) \mathbf{I} + \mathbf{P}_d \mathbf{Q} = \mathbf{U}, \quad (\text{III.72})$$

giving the next equation to solve when using the loop approach:

$$\mathbf{M}^\circ \left(\mathbf{R} + j\omega\mathbf{L} + \frac{1}{j\omega} \mathbf{P}_c \right) \mathbf{M}^{\circ T} \mathbf{I}_m + \mathbf{M}^\circ \mathbf{P}_d \mathbf{Q} = 0. \quad (\text{III.73})$$

Note that the nodal approach could also have been used here, as well as the thin shell formulation since the integrals related to the conductors only did not change.

III.4.c Coupled BIM-PEEC formulation

Finally, the formulation coupling the boundary integral equation ([III.62](#)) and the PEEC formulation ([III.73](#)) consists in solving the following system:

$$\begin{bmatrix} \mathbf{Z}_m & \mathbf{M}^\circ \mathbf{P}_d \\ \mathbf{B} \mathbf{M}^{\circ T} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{I}_m \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (\text{III.74})$$

with $\mathbf{Z}_m = \mathbf{M}^\circ \left(\mathbf{R} + j\omega\mathbf{L} + (j\omega)^{-1} \mathbf{P}_c \right) \mathbf{M}^{\circ T}$ and $\mathbf{M}^{\circ T} \mathbf{I}_m = \mathbf{I}$.

This formulation succeeds in modeling the dielectrics using only their boundaries thanks to a boundary integral equation and the assumption that the dielectrics have a linear permittivity. On the other hand, the conductors are modeled with a PEEC method, allowing to keep the same benefits as previously, but in counterpart, no preconditioner can be efficiently constructed *a priori* due to the top left PEEC-block in ([III.74](#)).

Compared to the matrix in the dielectric formulation (defined in (III.45)), only the inductances inside the conductors are computed. Therefore, if the volume of dielectric is very big, very big savings can be expected here since the dense matrix assembling is very expensive (with a quadratic complexity, without compression).

Next, the validation and computational savings of this coupled formulation are investigated considering the last real device.

III.4.d Validation

Considering the last real device (introduced in subsection III.3.d-ii), the validation of this coupling is done comparing the numerical results with the measurements of two different operating cases. The volume of the dielectrics is bigger than that of the conductors, so the improvements should be significant. The dielectric mesh is shown in Figure III.20, only the dielectric/air interface is required here. The volume approach is using a mesh where the inner volume is filled with hexahedra, whereas only the boundary is partially meshed with rectangular elements here. So, the number of unknowns linked to the dielectrics is most likely to reduce significantly.

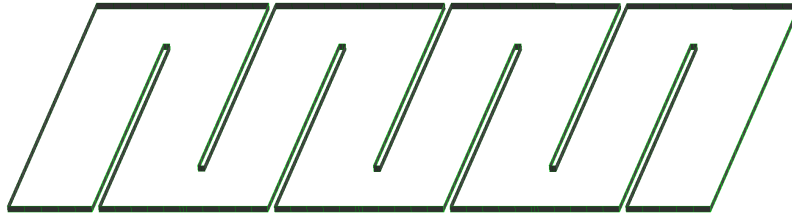


Figure III.20: Dielectric mesh Γ_D , corresponding to the interface dielectric/air.

The impedance curves are presented in Figure III.21 and Figure III.22 for the short-circuit and open-circuit case, respectively. A loss of accuracy can be seen at about 300 MHz, the PEEC-BIM is suffering of the assumption stated. Indeed, neglecting the displacement currents is limiting the upper threshold of validity of the formulation here. The rest of the results are however very satisfying and show a great match. A difference regarding the magnitude of the peaks can be noted. The accuracy of the magnitude could be increased by considering the losses of dielectrics (with a complex ϵ).

The same mesh discretization is used to run the tests, leading to 3,843 independent loops and 438 elements on the dielectric border, thus 4,281 DOFs. It is about the half of the volume approach. The computing times are presented in Table III.2.

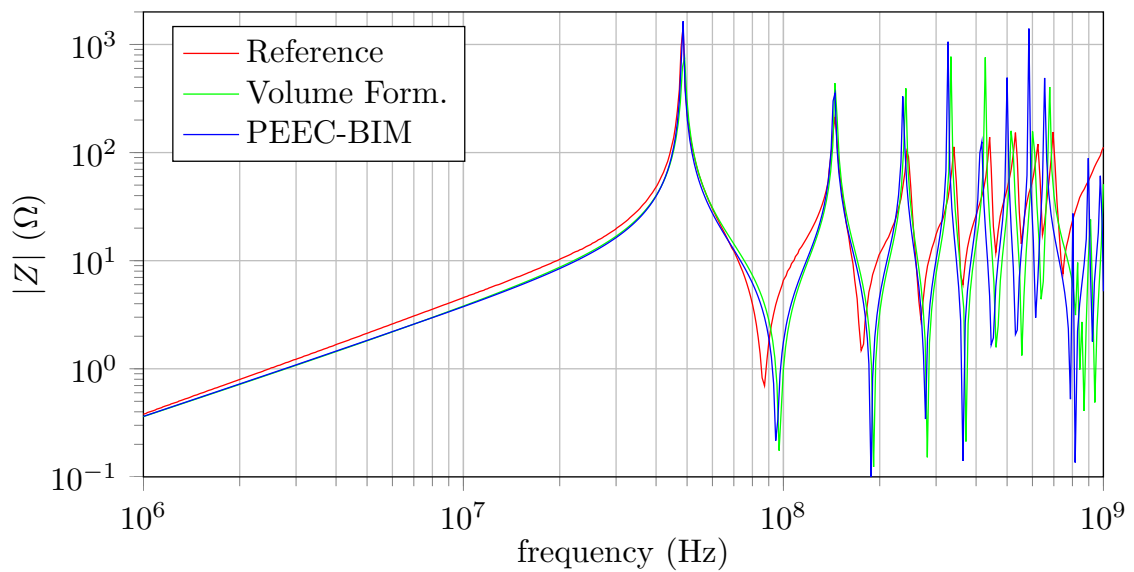


Figure III.21: Impedance magnitude curves for the short-circuit case.

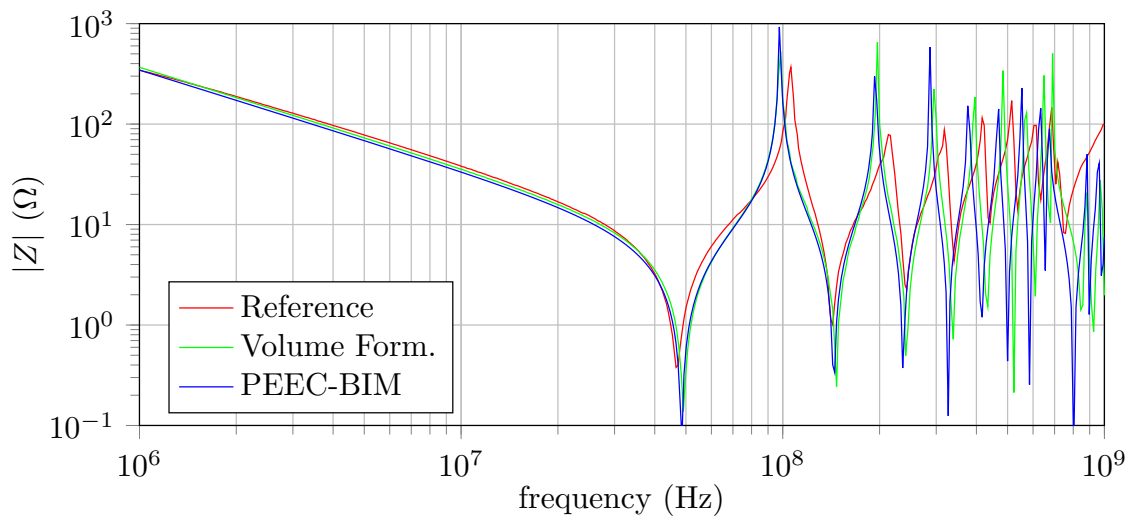


Figure III.22: Impedance magnitude curves for the open-circuit case.

The average time taken by a direct solution is of 104.5 seconds per frequency, dividing the solution time of the volume formulation by a factor 4.5.

Table III.2: Time taken to compute the solution by approach.

Formulation	Volume	Thin Shell	PEEC-BIM
Solution Time per frequency (in sec.)	474.8	378.3	104.5

This coupled formulation showed significant reductions here, regarding the num-

ber of unknowns and the computational costs. We have tested it on only one device and not compared to other non-PEEC formulations; nevertheless, we can clearly see that this formulation can be very efficient to treat devices with a big volume of dielectrics.

No preconditioning studies were made for this formulation since the matrix to solve contains a PEEC-block. However, the goal here was to significantly reduce the number of unknowns added to consider the dielectric materials, which in some cases can be prohibitively too important.

III.5 Conclusion

In this chapter, the formulations based on the PEEC method and considering dielectric materials have been presented. At the very beginning, a purely conductive and capacitive formulation was introduced in order to show how to add the capacitive effects occurring in the air.

The capacitive effects linked to the dielectrics were taken into account in the general formulation, in which no restrictive assumptions were stated. Nevertheless, it suffers of a critical implementation issue. As a remainder, it requires assembling all the matrices for each considered frequency. This problem can be avoided in exchange of its generality.

The practical formulation was deduced from the general formulation stating few assumptions. For instance, the materials are supposed to have linear properties here. This is not restricting in our context, these works indeed consider only linear conductors and dielectrics.

The validation of this formulation was done considering two different devices, one academic (i.e.: not real) and a real device. In the range of frequencies where the formulation is valid, the numerical results were excellently matching.

In the last case considered (the PCB device), the thin shell variant of the practical formulation was compared against the measurements and the volume (previous) results on a wider range of frequencies. The results of the thin shell formulation did not show any significant loss of accuracy here, while the computational costs were significantly reduced. So, the thin shell proved to be very useful for this kind of applications.

Focusing even more on the PCB application, i.e: with thin conductors and a large amount of dielectric volume, modeling the volume of the dielectrics can be prohibitive. Considering the whole volume of dielectrics indeed can significantly

increase the number of unknowns. So, a formulation coupling the PEEC method and a boundary integral equation was presented to avoid using the volume of dielectrics to model them.

The validation of this coupling was achieved thanks to the real device again. Significant improvements were observed regarding the number of unknowns and the solution times. The counterpart is that the frequency range of validity of the formulation is a bit reduced, however at the middle frequencies, the formulation was still accurate.

In the end, two formulations using the PEEC method are provided to consider devices made of linear conductors and dielectrics. The entire set of formulations presented here is also compatible with the thin shell approximation, presented in the last chapter. The last coupled formulation gives a special treatment for applications like PCBs, reducing significantly the computational costs.

The proposed formulations should be positioned, comparing the efficiency and the computation costs against other classical formulations.

Conclusions and Outlooks

This PhD was aiming for new general PEEC formulations able to consider conductors and dielectrics with a complete electromagnetic simulation. The choice of the PEEC method was done due to its efficiency in solving circuit coupled problems in a wide range of frequency. Thus, we considered it as one of the most suitable method to treat power electronic devices.

These works are a continuation to the PhD of T.-T. Nguyen [51], who implemented an inductive unstructured-PEEC formulation, and to the PhD of V. Ardon [38], who developed a coupling between a classical (inductive) PEEC formulation and an electrostatic formulation to add into account the dielectrics and the capacitive effects. Both of these former PhD candidates studied at the G2Elab and the works of the last gave birth to the last version of InCa3D.

The implemented formulations are an extension to the inductive unstructured-PEEC formulation, providing the consideration of dielectric materials and a full electromagnetic modeling for low- to middle-frequencies. The formulations implemented are still using the face elements, allowing to overcome the most limiting issues of the classical PEEC, as the low order interpolation and the requirement of structured meshes only. As the new formulations keep the same benefits as the former, the thin shell approximations are directly applicable, providing a variant to these formulations able to model thin conductors very efficiently. These last features also permit to improve the user-experience of a future InCa3D version, since the actual is limited to Manhattan-type meshes and requires a deep knowledge of the circuit theory.

Another asset of the PEEC formulations is that they are compatible with the use of a compression technique, mandatory to treat complex and industrial devices nowadays, and in our industrial context, being able to consider large-scale problems is critical. The actual version of InCa3D uses the adaptive multi-level fast multipole method, which proved to be very efficient in enabling the consideration of large-scale problems. The AMLFMM assembles matrices with low costs, however, the solution

through a circuit solver (inheriting from the PEEC method) is very problematic. The ill-conditioning of the PEEC formulation is well-known, but very few interests were found to cure such formulations in the literature.

The \mathcal{H} -matrices were implemented during the author's internship to provide a twofold compression technique also able to construct an efficient preconditioner: the \mathcal{H} -LU factorization. The use of the \mathcal{H} -matrices is very recent in the community of low-frequencies, thus we have great interests in using it to cure these conditioning issues.

We have presented the equations arising from the circuit theory, using the nodal and the loop approach, and we explored a few ways to construct a preconditioner within the \mathcal{H} -matrices are used.

The independent loop approach consists in solving an explicit change-of-basis equation. No literature was found on the matter, so we have started by trying naive methods, however, we did not succeed in approaching the inverse of the change-of-basis while using the \mathcal{H} -matrices.

The second approach consists in solving very large system with a 4×4 -blocks matrix, mostly populated with zeros. A naive approach to construct a preconditioner consisted in computing a block-LU factorization, but even if the sparse storages were taken into account in the factorization to optimize the memory consumption, the required amount of memory was too important for a preconditioner. At last, we have noted that the equation could be written as a singular saddle point problem, we found that there is a dedicated class of iterative solvers for such problems as Uzawa-SOR (Successive Over-Relaxation) methods. No time was taken to investigate deeply their potential, instead, we have chosen to study another type of formulation, very similar to the loop analysis.

At the very end of this PhD, we decided to take a step back and try another formulation, coming from the pure integral equation community. The \mathbf{T} formulation is a well-known approach to consider eddy currents problems in the low frequencies. From a circuit perspective, the unknown \mathbf{T} is similar to a loop. Taking advantage of the previous works done on the independent loops approach, we were able to re-use this knowledge and couple the classical \mathbf{T} formulation to an external electrical circuit. In addition, the \mathcal{H} -matrices can be directly applied to this formulation, with no difficulties. We succeeded in preconditioning this formulation, considering only conductors with resistive and inductive effects, and a circuit coupling in addition. The last formulation is not complying with the specifications of the PEEC formulations, nevertheless, it might be the way to follow.

Indeed, we can find in the high frequencies community formulations based on the Loop-Star decomposition. These formulations are an extension to the \mathbf{T} (or Loop) formulation, allowing to take into account the capacitive effects “analytically ” (in the sense of without a circuit approach). A further work may consists in adding the circuit coupling to the Loop-Star, to be at equal footing compared to the capacitive PEEC formulation without dielectrics. To go even further, one could try to add the dielectrics in the formulation in order to perfectly comply with the specifications.

During this PhD, we have explored some ways to precondition a circuit solver, using the nodal or the loop approach, without success. However, there are still a few paths to explore, as using the class of Uzawa-SOR methods for the nodal approach or enabling the circuit coupling in the Loop-Star formulations.

The PEEC formulations were validated on an academic example (three spires inductor) and a real device, with a simple geometry. So, the formulations should be tested on industrial and complex devices.

From a research perspective, the magnetic materials could be taken into account to provide one very complete tool, since the power electronics devices are likely to use ferrite cores to concentrate the induction flux. Another very important improvement to the PEEC is the consideration of the thermal aspect, allowing to study the thermal behavior of a device in operation.

Appendix A

Feature Selective Validation of examples

We are assessing the accuracy of the results using the FSV techniques, accordingly to the IEEE standard [75]. The stand-alone software FSV tool [76] is employed to compare and qualify the next curves. In fact, the amplitude difference measure (ADM) and the feature difference measure (FDM) techniques will be applied, which give the matching level of two curves.

A.1 Case 1: Micro-coil

In III.3.d-i, a micro-coil has been studied. Its impedance magnitude and phase have been computed using a FEM software and the proposed “practical” formulation.

The ADM and the FDM have been applied to the impedance magnitude curves (see Figures A.1a and A.1b) and to the impedance phase curves (see Figures A.2a and A.2b).

We can see in Figure III.13a and Figure III.13b that the results of both methods are very similar. This is also supported by the FSV technique results (Figures A.1a, A.1b, A.2a and A.2b) which rate the match of the curves as excellent.

A.2 Case 2: Real device

In III.3.d-ii, a *real device* has been studied. We wanted to compare the results obtained with the “practical formulation” and the experimental data acquired by V. Ardon, a former PhD Student. To quantify the numerical results, we have used the same FSV technique as before (using the FSV Tool). The results are shown in Figs A.3a and A.3b for the open-circuit case and in Figs A.4a and A.4b for the short-circuit case. The ADM technique rates both cases mostly from good to excellent, whereas the FDM rates them mostly as excellent.

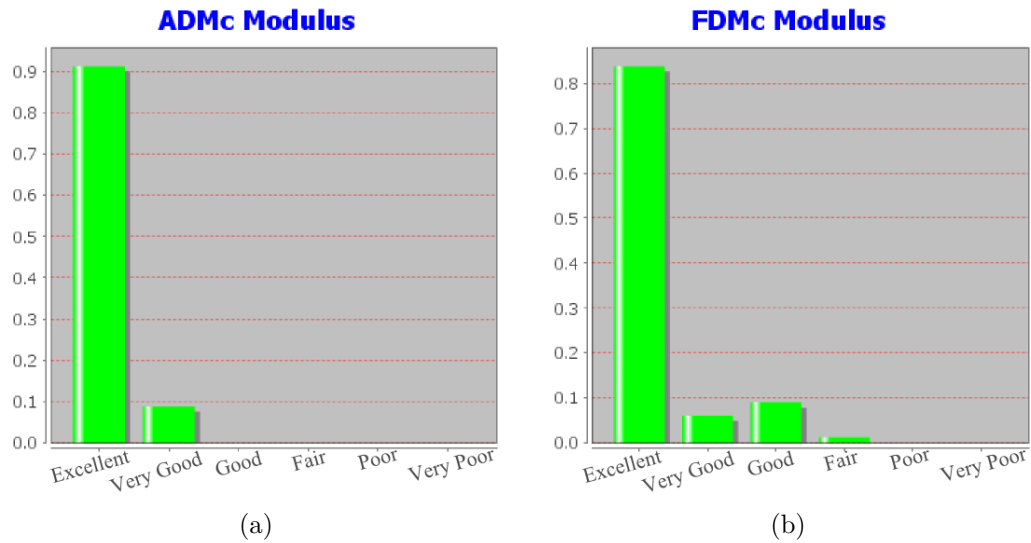


Figure A.1: (a) Results of the ADM applied to the impedance magnitude curves. (b) Results of the FDM applied to the impedance magnitude curves.

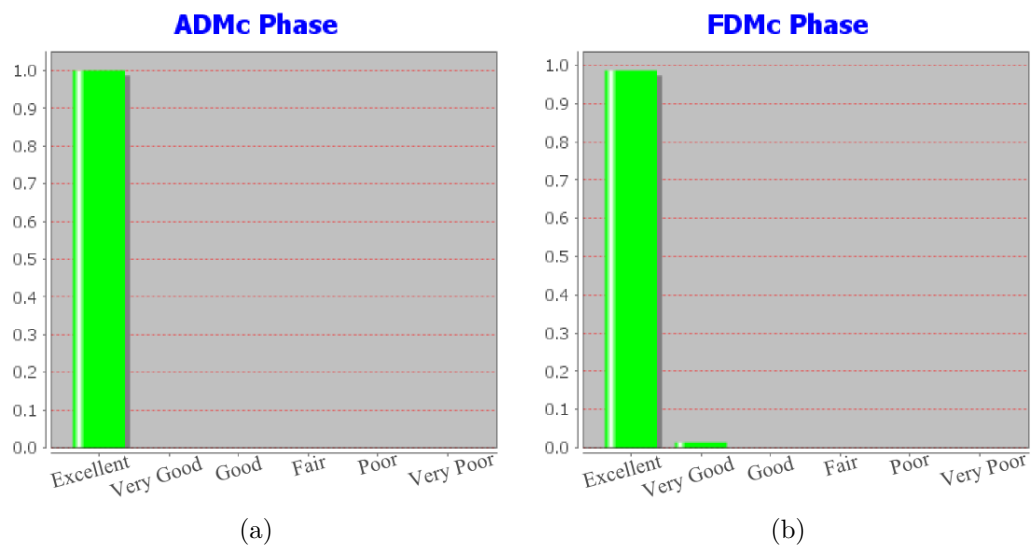


Figure A.2: (a) Results of the ADM applied to the impedance phase curves. (b) Results of the FDM applied to the impedance phase curves.

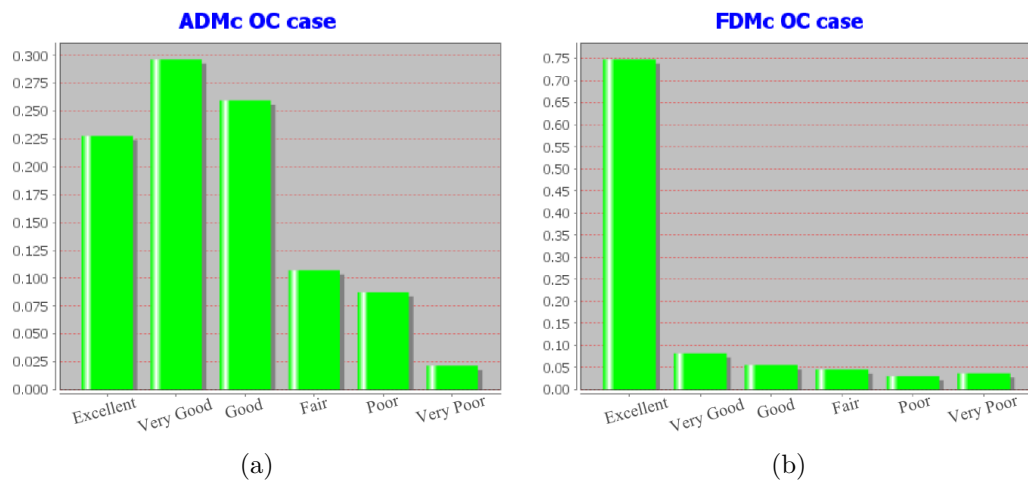


Figure A.3: (a) Results of the ADM applied to the impedance magnitude curves for the Open-Circuit case. (b) Results of the FDM applied to the impedance magnitude curves for the Open-Circuit case.

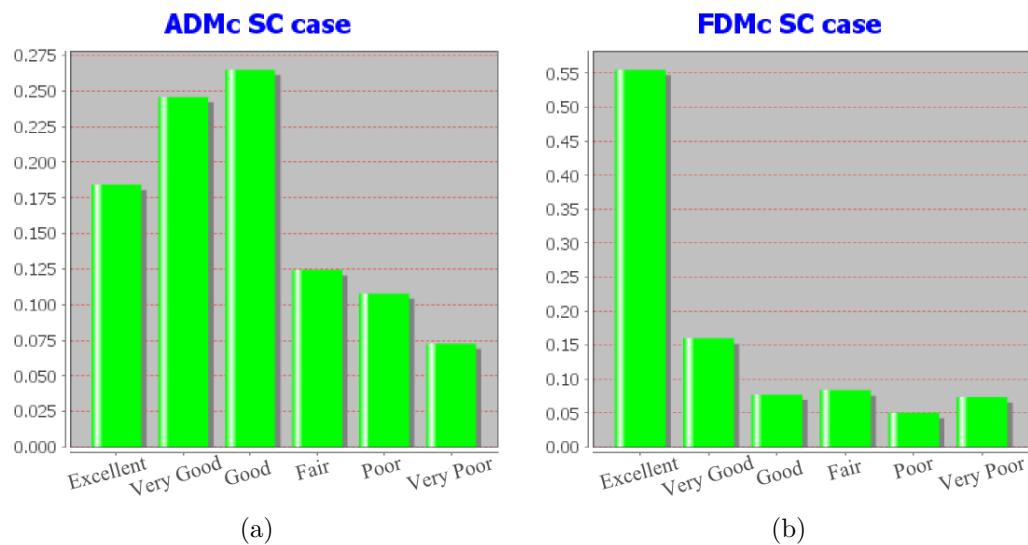


Figure A.4: (a) Results of the ADM applied to the impedance magnitude curves for the Short-Circuit case. (b) Results of the FDM applied to the impedance magnitude curves for the Short-Circuit case.

Bibliography

- [1] G. Strang and G. J. Fix, An Analysis of the Finite Element Method. Prentice-Hall, Inc., 1973.
- [2] G. Strang, Introduction to Applied Mathematics. Wellesley-Cambridge Press, 1986.
- [3] R. BARNHILL, The Mathematics of Finite Elements and Applications. Elsevier, 1973, pp. 83–112. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780127472508500084>
- [4] V. Rokhlin, “Rapid solution of integral equations of classical potential theory,” Journal of Computational Physics, vol. 60, no. 2, pp. 187–207, sep 1985. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/0021999185900026>
- [5] L. Greengard and V. Rokhlin, “A fast algorithm for particle simulations,” Journal of Computational Physics, vol. 73, no. 2, pp. 325–348, dec 1987. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/0021999187901409>
- [6] M. Bebendorf, “Approximation of boundary element matrices,” Numerische Mathematik, vol. 86, no. 4, pp. 565–589, oct 2000. [Online]. Available: <http://link.springer.com/10.1007/PL00005410>
- [7] M. Bebendorf and S. Rjasanow, “Adaptive Low-Rank Approximation of Collocation Matrices,” Computing, vol. 70, no. 1, pp. 1–24, feb 2003. [Online]. Available: <http://link.springer.com/10.1007/s00607-002-1469-6>

-
- [8] E. Tyrtyshnikov, "Incomplete Cross Approximation in the Mosaic-Skeleton Method," Computing, vol. 64, no. 4, pp. 367–380, jun 2000. [Online]. Available: <http://link.springer.com/10.1007/s006070070031>
- [9] S. Börm and L. Grasedyck, "Hybrid cross approximation of integral operators," Numerische Mathematik, vol. 101, no. 2, pp. 221–249, jun 2005. [Online]. Available: <http://link.springer.com/10.1007/s00211-005-0618-1>
- [10] W. Hackbusch, "A Sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices," Computing (Vienna/New York), vol. 62, no. 2, pp. 89–108, apr 1999. [Online]. Available: <http://link.springer.com/10.1007/s006070050015>
- [11] —, "A Sparse H-Matrix Arithmetic," Computing, vol. 62, no. 1, pp. 21–47, 2000. [Online]. Available: <http://link.springer.com.gaelnomade.ujf-grenoble.fr/article/10.1007/PL00021408>
- [12] L. Grasedyck and W. Hackbusch, "Construction and Arithmetics of H-Matrices," Computing, vol. 70, no. 4, pp. 295–334, aug 2003. [Online]. Available: <http://link.springer.com/10.1007/s00607-003-0019-1>
- [13] R. F. Harrington, Field Computation by Moment Methods. Wiley-IEEE Press, 1993.
- [14] W.-L. Wu, A. W. Glisson, and D. Kajfez, "A study of two numerical solution procedures for the electric field integral equation at low frequency," Applied Computational Electromagnetics Society Journal, vol. 10, no. 3, pp. 69–80, 1995.
- [15] M. Burton and S. Kashyap, "A study of a recent moment method algorithm that is accurate to very low frequencies," Applied Computational Electromagnetics Society Journal, vol. 10, no. 3, pp. 58–68, 1995.
- [16] F. P. Andriulli, F. Vipiana, G. Vecchi, and E. Michielssen, "Analysis and Regularization of the TD-EFIE Low-Frequency Breakdown," IEEE Transactions on Antennas and Propagation, vol. 57, no. 7, pp. 2034–2046, 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5159560>

- [17] X. Y. Z. Xiong, L. J. Jiang, W. E. I. Sha, and Y. H. Lo, "A New EFIE Method Based On Coulomb Gauge," Progress In Electromagnetics Research, vol. 140, pp. 613–631, 2013. [Online]. Available: <http://www.jpier.org/PIER/pier.php?paper=13040303>
- [18] J. Zhu and D. Jiao, "A Rigorous Solution to the Low-Frequency Breakdown in Full-Wave Finite-Element-Based Analysis of General Problems Involving Inhomogeneous Lossless/Lossy Dielectrics and Nonideal Conductors," IEEE Transactions on Microwave Theory and Techniques, vol. 59, no. 12, pp. 3294–3306, dec 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6082419>
- [19] J. Zhu, S. Omar, W. Chai, and D. Jiao, "A rigorous solution to the low-frequency breakdown in the electric field integral equation," in 2011 IEEE International Symposium on Antennas and Propagation (APSURSI). IEEE, jul 2011, pp. 3214–3217. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5997218>
- [20] J. Lim, S. Rao, and D. Wilton, "A novel technique to calculate the electromagnetic scattering by surfaces of arbitrary shape," in 1993 URSI Radio Science Meeting Digest, 1993, p. 322.
- [21] A. Ruehli and P. Brennan, "Efficient Capacitance Calculations for Three-Dimensional Multiconductor Systems," IEEE Transactions on Microwave Theory and Techniques, vol. 21, no. 2, pp. 76–82, feb 1973. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1127927>
- [22] A. E. Ruehli, "Equivalent Circuit Models for Three-Dimensional Multiconductor Systems," IEEE Transactions on Microwave Theory Techniques, vol. 22, no. 3, pp. 216–221, 1974. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1128204>
- [23] A. Ruehli and P. Brennan, "Capacitance models for integrated circuit metallization wires," IEEE Journal of Solid-State Circuits, vol. 10, no. 6, pp. 530–536, dec 1975. [Online]. Available: <http://dx.doi.org/10.1109/JSSC.1975.1050654http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1050654>
- [24] J.-L. Schanen, C. Guerin, J. Roudet, and G. Meunier, "Modeling of Printed Circuit Board loop inductance," IEEE Transactions on Magnetics, vol. 30,

- no. 5, pp. 3590–3593, 1994. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=312715>
- [25] J. L. Schanen, E. Clavel, and J. Roudet, “Modeling of low inductive connections: the planar busbar structure,” in Industry Applications Society Annual Meeting, 1994., Conference Record of the 1994 IEEE, no. Ura 355, 1994, pp. 1246–1250 vol.2. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=377556>
- [26] E. Clavel, J.-L. Schanen, and J. Roudet, “Electromagnetic modeling of a power module case,” in Euro. Electromagnetic Conf., 1994.
- [27] InCa3D. [Online]. Available: <http://www.altairhyperworks.com/product/flux/InCa3D>
- [28] A. E. Ruehli, “Inductance Calculations in a Complex Integrated Circuit Environment,” IBM Journal of Research and Development, vol. 16, no. 5, pp. 470–481, sep 1972. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5391464>
- [29] Hao Shi, Jun Fan, J. Drewniak, T. Hubing, and T. Van Doren, “Modeling multilayered PCB power-bus designs using an MPIE based circuit extraction technique,” in 1998 IEEE EMC Symposium. International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.98CH36253), vol. 2. IEEE, 1998, pp. 647–651. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=750273>
- [30] A. Ruehli, G. Antonini, and A. Orlandi, “Extension of the partial element equivalent circuit method to non-rectangular geometries,” in 1999 IEEE International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.99CH36261), vol. 2. IEEE, 1999, pp. 728–733. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=810108>
- [31] G. Antonini, A. Ruehli, and J. Esch, “Nonorthogonal PEEC formulation for time and frequency domain modeling,” in IEEE International Symposium on Electromagnetic Compatibility, vol. 1. IEEE, 2002, pp. 452–456. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1032521>

- [32] F. Freschi, G. Gruosso, and M. Repetto, “Unstructured PEEC formulation by dual discretization,” IEEE Microwave and Wireless Components Letters, vol. 16, no. 10, pp. 531–533, oct 2006. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1704853>
- [33] T.-T. Nguyen, G. Meunier, and al., “An Integral Formulation for the Computation of 3-D Eddy Current Using Facet Elements,” IEEE Transactions on Magnetics, vol. 50, no. 2, pp. 549–552, Feb. 2014.
- [34] S. Rao, D. Wilton, and A. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” IEEE Transactions on Antennas and Propagation, vol. 30, no. 3, pp. 409–418, may 1982. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1142818>
- [35] D. Gope, A. Ruehli, and V. Jandhyala, “Solving Low-Frequency EM-CKT Problems Using the PEEC Method,” IEEE Transactions on Advanced Packaging, vol. 30, no. 2, pp. 313–320, may 2007. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4214905>
- [36] D. Daroui and J. Ekman, “PEEC-Based Simulations Using Iterative Method and Regularization Technique for Power Electronic Applications,” IEEE Transactions on Electromagnetic Compatibility, vol. 56, no. 6, pp. 1448–1456, dec 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6822528>
- [37] F. Freschi, “Fast block-solution of PEEC equations,” IEEE Transactions on Magnetics, vol. 49, no. 5, pp. 1753–1756, May 2013.
- [38] V. ARDON, “Méthodes Numériques et outils logiciels pour la prise en compte des effets capacitifs dans la modélisation CEM de dispositifs d’électronique de puissance,” Ph.D. dissertation, University of Grenoble Alpes, 2010.
- [39] S. Börm, L. Grasedyck, and W. Hackbusch, “Hierarchical Matrices,” June 2006, lecture note no.: 21.
- [40] J. J. Sylvester, “A question in the geometry of situation,” Quarterly Journal of Pure and Applied Mathematics, no. 1, p. 79, 1857. [Online]. Available: [https://babel.hathitrust.org/cgi/pt?id=uc1.\\$b417523;view=1up;seq=97](https://babel.hathitrust.org/cgi/pt?id=uc1.$b417523;view=1up;seq=97)

- [41] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, “A theory of pseudoskeleton approximations,” Linear Algebra and its Applications, vol. 261, no. 1-3, pp. 1–21, aug 1997. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0024379596003011>
- [42] E. Tyrtyshnikov, “Mosaic-Skeleton approximations,” Calcolo, vol. 33, no. 1-2, pp. 47–57, jun 1996. [Online]. Available: <http://link.springer.com/10.1007/BF02575706>
- [43] E. E. Tyrtyshnikov, “Mosaic ranks and skeletons,” in Numerical Analysis and Its Applications, 1997, vol. 1196, pp. 505–516.
- [44] S. Rjasanow and O. Steinbach, The Fast Solution of Boundary Integral Equations, ser. Mathematical and Analytical Techniques with Applications to Engineering, Springer, Ed. Springer US, 2007. [Online]. Available: <http://link.springer.com/10.1007/0-387-34042-4>
- [45] M. Bebendorf and S. Kunis, “Recompression techniques for adaptive cross approximation,” Journal of Integral Equations and Applications, vol. 21, no. 3, pp. 331–357, sep 2009. [Online]. Available: <http://projecteuclid.org/euclid.jiea/1248269700>
- [46] M. Bebendorf, Hierarchical Matrices, ser. Lecture Notes in Computational Science and Engineering, T. J. Barth, D. E. Keyes, and D. Roose, Eds. Springer Berlin Heidelberg, 2008, vol. 63. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-77147-0>
- [47] R. Hiptmair, “Operator Preconditioning,” Computers and Mathematics with Applications, vol. 52, pp. 699–706, 2006.
- [48] A. Kalimov, “Application of a hybrid integrodifferential method for analysis of thin magnetic shields,” IEEE Trans. on Mag., (Volume:34 , Issue: 5), pp. 2453–2456, Sep 1998.
- [49] A. Carpentier and al., “Resolution of Nonlinear Magnetostatic Problems With a Volume Integral Method Using the Magnetic Scalar Potential,” IEEE Trans. on Mag., vol. 49, no. 5, pp. 1685–1688, May 2013.
- [50] A. M. Bradley, “H-Matrix and Block Error Tolerances,” ArXiv e-prints, Oct. 2011.

- [51] T. T. Nguyen, “Méthode PEEC inductive par élément de facette pour la modélisation des régions conductrices volumiques et minces,” Ph.D. dissertation, UNIVERSITE DE GRENOBLE, 2014.
- [52] T.-T. Nguyen, G. Meunier, J.-M. Guichon, O. Chadebec, and T.-S. Nguyen, “An Integral Formulation for the Computation of 3-D Eddy Current Using Facet Elements,” IEEE Transactions on Magnetics, vol. 50, no. 2, pp. 549–552, 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6749118>
- [53] T.-T. Nguyen, G. Meunier, J.-M. Guichon, and O. Chadebec, “3-D Integral Formulation Using Facet Elements for Thin Conductive Shells Coupled With an External Circuit,” IEEE Transactions on Magnetics, doi: 10.1109/TMAG.2014.2363365.
- [54] G. Meunier, O. Chadebec, and J. M. Guichon, “A magnetic flux - electric current volume integral formulation based on facet elements for solving electromagnetic problems,” IEEE Transactions on Magnetics, vol. 51, no. 3, pp. 1–4, March 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7093533/>
- [55] C. Paul, Introduction to Electromagnetic Fields, international editions ed., Mc Graw-Hill, Ed., 1987.
- [56] A. Bossavit, “Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism,” IEE Proceedings A Physical Science, Measurement and Instrumentation, Management and Education, Reviews, vol. 135, no. 8, p. 493, 1988. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/ip-a-1.1988.0077>
- [57] Chung-Wen Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” IEEE Transactions on Circuits and Systems, vol. 22, no. 6, pp. 504–509, jun 1975. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1084079>
- [58] T.-S. Nguyen, J.-M. Guichon, O. Chadebec, G. Meunier, and B. Vincent, “AN INDEPENDENT LOOPS SEARCH ALGORITHM FOR SOLVING INDUCTIVE PEEC LARGE PROBLEMS,” Progress In Electromagnetics Research M, vol. 23, no. November 2011, pp. 53–63, 2012. [Online]. Available: <http://www.jpier.org/PIERM/pier.php?paper=11111503>

- [59] L. Krahenbuhl and D. Muller, “Thin layers in electrical engineering-example of shell models in analysing eddy-currents by boundary and finite element methods,” IEEE Transactions on Magnetics, vol. 29, no. 2, pp. 1450–1455, mar 1993. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=250676>
- [60] C. Guerin and G. Meunier, “3-D Magnetic Scalar Potential Finite Element Formulation for Conducting Shells Coupled With an External Circuit,” IEEE Transactions on Magnetics, vol. 48, no. 2, pp. 323–326, feb 2012. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6136627>
- [61] T. Le-Duc, G. Meunier, O. Chadebec, J.-M. Guichon, and J. Bastos, “General integral formulation for the 3d thin shell modeling,” IEEE Transactions on Magnetics, vol. 49, no. 5, pp. 1989–1992, May 2013.
- [62] T. S. Nguyen, “Réduction de modèles issus de la méthode PEEC pour la modélisation électromagnétique des interconnexions électriques,” Ph.D. dissertation, University of Grenoble Alpes, 2012.
- [63] SuiteSparse. [Online]. Available: <http://faculty.cse.tamu.edu/davis/suitesparse.html>
- [64] M. Benzi, G. H. Golub, and J. Liesen, “Numerical solution of saddle point problems,” Acta Numerica, vol. 14, pp. 1–137, may 2005. [Online]. Available: <http://www.journals.cambridge.org/abstract{ }S0962492904000212>
- [65] R. Albanese, R. Martone, G. Miano, and G. Rubinacci, “A T formulation for 3D finite element Eddy current computation,” IEEE Transactions on Magnetics, vol. 21, no. 6, pp. 2299–2302, nov 1985. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1064186>
- [66] R. Albanese and G. Rubinacci, “Integral formulation for 3d eddy-current computation using edge elements,” Physical Science, Measurement and Instrumentation, Management and Education-Reviews, IEE Proceedings A, vol. 135, no. 7, pp. 457–462, Sep 1988.
- [67] P. Dular, “Modélisation du champ magnétique et des courants induits dans des systèmes tridimensionnels non linéaires,” Docteur en Sciences appliquées,

- Université de Liège, Liège, Belgique, 1994. [Online]. Available: <https://orbi.ulg.ac.be/bitstream/2268/191441/1/1994-PhD-PDular-All-22.pdf>
- [68] Zhuoxiang Ren, “Influence of the RHS on the convergence behaviour of the curl-curl equation,” IEEE Transactions on Magnetism, vol. 32, no. 3, pp. 655–658, may 1996. [Online]. Available: <http://ieeexplore.ieee.org/document/497323/>
- [69] —, “Autogauging of vector potential by iterative solver-Numerical evidence,” in 3d Int. Workshop on Electric and Magnetic Fields (Proc., Liège, 6-9 May 1996), A.I.M. (31 Rue St-Gilles, Liège), 1996, pp. 119–24.
- [70] G. Meunier, Y. Le Floch, and C. Guerin, “A nonlinear circuit coupled $t - t/\text{sub } 0/ - \phi$ formulation for solid conductors,” IEEE Transactions on Magnetism, vol. 39, no. 3, pp. 1729–1732, may 2003. [Online]. Available: <http://ieeexplore.ieee.org/document/1198567/>
- [71] G. Meunier, O. Chadebec, J. M. Guichon, V. Le-Van, J. Siau, B. Bannwarth, and F. Sirois, “A - T volume integral formulations for solving electromagnetic problems in the frequency domain,” IEEE Transactions on Magnetism, vol. 52, no. 3, pp. 1–4, March 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7312981/>
- [72] P. Leroy, G. Akoun, B. Essakhi, L. Santandrea, L. Pichon, and C. Guyot, “An efficient global analysis of a rectenna using the combination of a full-wave model and a rational approximation,” The European Physical Journal Applied Physics, vol. 29, no. 1, pp. 39–43, nov 2004. [Online]. Available: <http://www.epjap.org/10.1051/epjap:2004201>
- [73] Flux. [Online]. Available: <http://www.altairhyperworks.com/product/flux>
- [74] P. Pham Quang, C. Guérin, G. Meunier, and Y. Lambeye, “Validation of a 3d decoupled magnetodynamic-electric model by determining the impedance of microcoils,” in Volume 34: Computer Field Models of Electromagnetic Devices, ser. Studies in Applied Electromagnetics and Mechanics, S. Wiak and E. Napieralska-Juszczak, Eds. Amsterdam: IOS Press, Nov. 2010, pp. 599–608.
- [75] IEEE Standard P1597, Standard for Validation of Computational Electromagnetics Computer Modeling and Simulation—Part 1, 2 2008.
- [76] A. Orlandi. (2006). Free stand-alone FSV application. [Online]. Available: http://ing.univaq.it/uaqemc/FSV_4_0_3L/

Publications

Articles published in international journals

Published

[IEEE Transactions on Electromagnetic Compatibility](#) [Volume Integral Formulation Using Face Elements For Electromagnetic Problem Considering Conductors And Dielectrics](#)

J. Siau, G. Meunier, O. Chadebec, J.-M. Guichon and R. Perrin-Bit - Volume 58 Issue 5 Pages 1587 – 1594.
Oct. 2016.

Abstract: A new integral formulation is presented, enabling the computation of resistive, inductive, and capacitive effects considering both conductors and dielectrics in the frequency domain. The considered application allows us to neglect any propagation effects and magnetic materials. In this paper, we will show how to improve the unstructured-partial element equivalent circuit approach to consider dielectric materials, keeping the same benefits. Results obtained with this formulation are compared to results from an industrial finite-element method software and measurements.

Publications in international conferences

Presented and Published

CEFC 2014 - [Hybrid Cross Approximation for a Magnetostatic Formulation](#)
Annecy,
FRANCE

J. Siau, O. Chadebec, R. Perrussel and J.-R. Poirier -
Oral Session. May 2014.

Published in [IEEE Transactions on Magnetics](#)

CompuMag [A Mixed Surface Volume Integral Formulation for the](#)
2015 - Montréal, [Modeling of High-Frequency Coreless Inductors](#)
CANADA

Z. De Grève, J. Siau, G. Meunier, J.-M. Guichon and
O. Chadebec - Poster Session. June-July 2015.

Published in [IEEE Transactions on Magnetics](#)

[A-T Volume Integral Formulations for Solving
Electromagnetic Problems in the Frequency Domain](#)

G. Meunier, O. Chadebec, J.-M. Guichon, V. Le-Van,
J. Siau, B. Bannwarth and F. Sirois - Oral Session.
June-July 2015.

Published in [IEEE Transactions on Magnetics](#)

[Generalized PEEC Analysis of Inductive Coupling
Phenomena in a Transmission Line Right-of-Way](#)

L. Blattner Martinho, J. Siau, B. Bannwarth, J. -M.
Guichon, O. Chadebec, G. Meunier and V. C. Silva -
Poster Session. June-July 2015.

Published in [IEEE Transactions on Magnetics](#)

EMF 2016 - [Coupling of PEEC Volume Integral Formulation and
Lyon, FRANCE Boundary Element Method to Consider Thin Shell
Conductors and Dielectrics](#)

J. Siau, G. Meunier, O. Chadebec, J.-M. Guichon and
R. Perrin-Bit - Poster Session. April 2016.

[A Efficient Integral Formulation for Electric Thin
Region](#)

G. Meunier, J.-M. Guichon, O. Chadebec and J. Siau -
Poster Session. April 2016.

Publications in national conferences

Presented

Numelec 2015 [Couplage entre formulations intégrale de volume et
- St Nazaire intégrale de frontière pour la modélisation de problèmes
FRANCE électromagnétiques en présence de diélectriques](#)

J Siau, G Meunier, O Chadebec, JM Guichon and R
Perrin-Bit - Poster Session. June 2015.

[Formulations intégrales de volume en électromagnétisme
basées sur une approche circuit](#)

G. Meunier, O. Chadebec, J.-M. Guichon, V. Le Van, T.
T. Nguyen, J. Siau, B. Bannwarth and P. Labie - Oral
Session. June 2015.

UNSTRUCTURED PEEC FORMULATIONS CONSIDERING RESISTIVE, INDUCTIVE AND CAPACITIVE EFFECTS FOR POWER ELECTRONICS

Abstract - The classical method PEEC is based on a semi-analytical integral method to construct an equivalent electric circuit using lumped components. This method is particularly well-suited to model filiform conductors. It is actually possible to consider thin conductive regions with this method, but it's still limited and sometimes unsatisfactory. In fact, the meshes that can be used are very constrained (geometrically discretized by quadrangles) and the frequency approach is limited (capacitive effect is neglected). The aim of this thesis is to introduce the capacitive and magnetic effects into the method PEEC to get a general tool, efficient and usable at the industry level. Particularly, the generality of the formulation and its flexibility should enable a simple use of the software InCa3D for non-expert users on numerical methods. The work consists in: Improving the last works by introducing the capacitive and magnetic effects in the formulations. Suggesting some methods of matrix compression to improve the efficiency of the computation, and to lower the needed memory.

Keywords: electromagnetic formulation, integral methods, \mathcal{H} -matrix, power electronics.