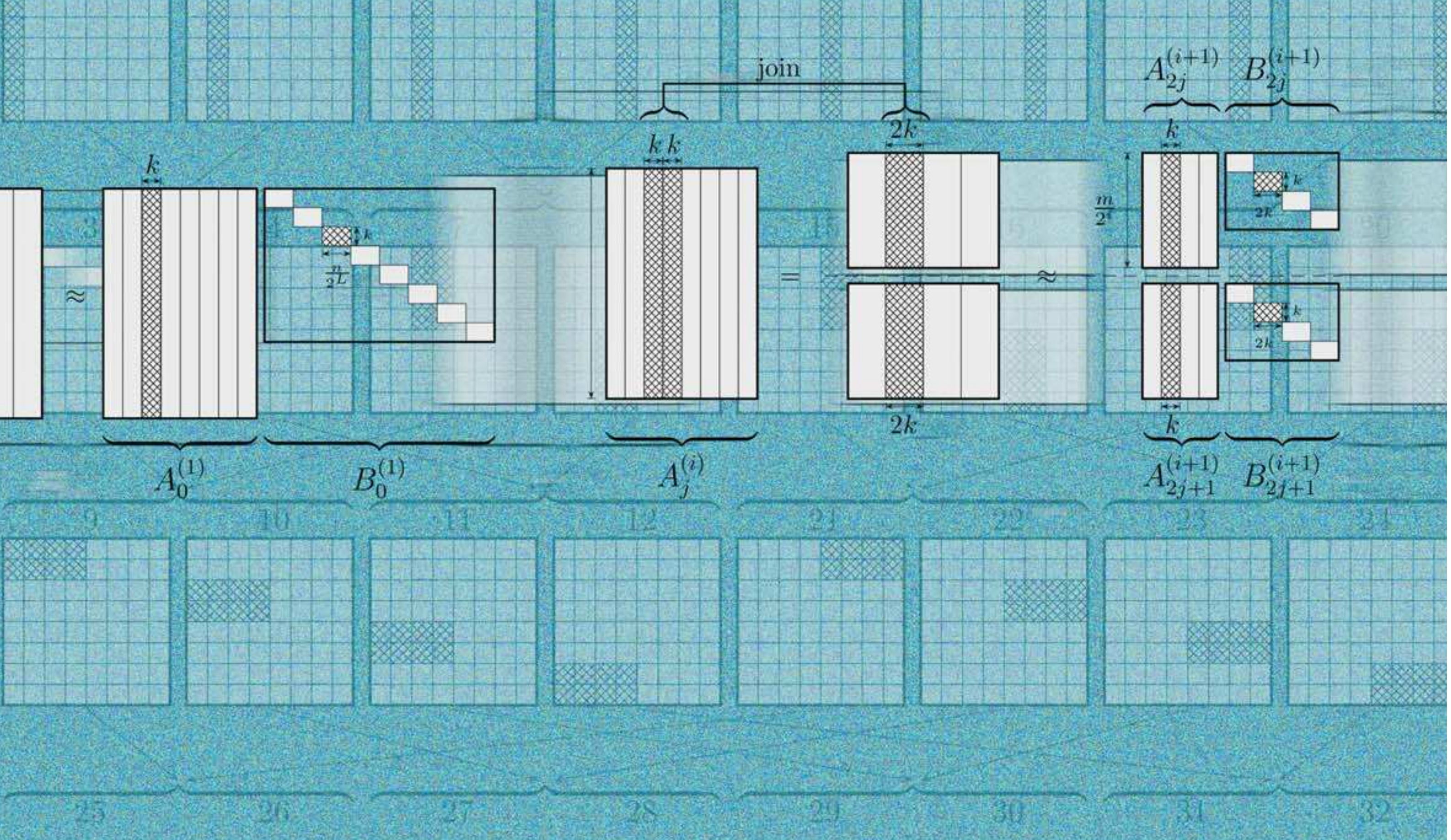


Author

José M. Tamayo

Multilevel Adaptive Cross Approximation and Direct Evaluation Method for Fast and Accurate Discretization of Electromagnetic Integral Equations

Thesis Advisors:

Juan M. Rius
Alexander Heldring

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Multilevel Adaptive Cross Approximation and Direct Evaluation Method for Fast and Accurate Discretization of Electromagnetic Integral Equations

Author:
José M. Tamayo

Thesis Advisors:
Juan M. Rius
Alexander Heldring

AntennaLab
Universitat Politècnica de Catalunya
Departament de Teoria del Senyal i Comunicacions

Tesis presentada per obtenir el títol de Doctor
per la Universitat Politècnica de Catalunya

January 17, 2011

Cover by Francesc Morera (a.bís)

Abstract

This thesis dissertation presents new approaches to accelerate or increase the accuracy of integral equations discretized by the Method of Moments (MoM) in computational electromagnetics.

Firstly, a fast iterative solver, the Multilevel Adaptive Cross Approximation (MLACA), for accelerating the solution of the MoM linear system is detailed. In the quest for a general-purpose scheme, the MLACA is a method independent of the kernel of the integral equation and is purely algebraic. It improves both efficiency and compression rate with respect to the previously existing single-level version, the ACA. Therefore, it represents an excellent alternative for the solution of the MoM system of large-scale electromagnetic problems.

Secondly, the direct evaluation method, which has proved to be the main reference in terms of efficiency and accuracy, is extended to overcome the computation of the challenging 4-D hyper-singular integrals arising in the Magnetic Field Integral Equation (MFIE) and Combined Field Integral Equation (CFIE) formulations. The maximum affordable accuracy –machine precision– is obtained in a more than reasonable computation time, surpassing any other existing technique in literature.

Thirdly, the aforementioned hyper-singular integrals become near-singular when the discretized elements are very closely placed but not touching. It is shown how traditional integration rules fail to converge also in this case, and a possible solution is proposed.

Finally, an effort to facilitate the usability of any antenna simulation software based on the MoM has led to the development of a general mathematical model of an excitation port in the discretized space. With this new model, it is no longer necessary to adapt the mesh edges to the port.

***Keywords*:** Method of Moments (MoM), Fast Solvers, Surface Integral Equations, Singular Integrals, Numerical Simulation, Electromagnetism, Impedance Matrix Compression, Adaptive Cross Approximation, Edge Adjacent Integration , Vertex Adjacent Integration.

Preface

This thesis is based upon studies conducted during September 2006 to December 2010 at the Department of Teoria del Senyal i Comunicacions, at Universitat Politècnica de Catalunya, Barcelona, Spain. Part of it was done during a short stay from September 2009 to December 2009 at the Laboratory of Electromagnetics and Acoustics, at École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

I would like to express my sincere gratitude to my supervisors, Juan M. Rius and Alex Heldring. Without their advise and unique support this thesis would never had become a reality. Apart from the purely technical support, there are a number of experiences worth mentioning, some of which are expressed in the following. From the pleasant Juan's introduction to Switzerland, through the nice time at Cádiz with baby Aline to the enjoying summer barbecue. But most importantly, for being there not only at the best moments but also at the worst ones, giving me the necessary patience to face them.

I would like to thank also the other member of our research family, Eduard Ubeda with whom I have had the pleasure of working and with whom I have shared a number of amazing trips, California and Algonquin Park among others.

I would like to show an affectionate acknowledgement to my advisors during my stay at LEMA, who took care of me at the EPFL and with whom I had the pleasure to closely collaborate, Juan R. Mosig, Athanasios Polimeridis and Michael Mattes. Not to mention the great time we had at Ovronnaz, as well as in several pubs in Lausanne.

I had the opportunity to attend several conferences and courses around the world, where I met plenty of interesting people, who I have now the pleasure of consider friends. Without being exhaustive, I come to mention and thank many of them in the following lines. Guido and our pints, whom I might consider to be my forth brother, Simone with whom I had a very nice time in Edinburgh, Andrea, Katherine, Milica, Mónica, Oscar, Remi, Rodney, Vikram and Felipe. Javi, Ferran, Jordi and Eloy from our particular and very special society. Fernando, Sebas, María, Javi, José Luis and José María from Cartegena. Paco Medina, Paco Mesa and Rafael from Sevilla. And all the ones which I have not mentioned due to my deficient memory.

My stay at Lausanne would not have been the same without the contribution of Eden and his cosy flat and nice chat, Francesco and his "anchoas", Ioannis, Rafal, Gabriele, David, Gabriela, Eulalia, Sergio, Roberto, Laleh, Ruzica, Maddalena, Marco, Stephan, Xilos. Of course, the excursions to excellent places like Zermatt with Alejandro, Alonso, Toni, Carlos, Felix, Javo, Marta, Neus, Oriol, Pau, Xavi, Neus and Núria really made a difference.

I cannot forget my dear friends at the TSC department. With them I have shared plenty of refreshing coffee breaks, lunches, dinners, parties, trips and the best of laughs. Benji, my right hand, Roberto, Dani, Sergio, Pere, Alberto, Giuseppe, Alba, Rubén, Navarrete, Gemma, Hugo, Isaac, Javi, Jordi B., Juan Carlos, Luca, Julien, Marco, Mariví, Omar, Oscar, Raquel, Ricardo, Benji, Edgar, Bea, Enrique, Jordi A., Aline, Angel, Juanfer, Gerard, Sandra, Pablo, Santi, Marta, Dani, José Antonio and Edu. I would like to thanks also Aynie and Teresa for making paperwork easier.

I would like to thank Roger, Sergio, Javi, Ana Belén, Quim, Esther, Susana and David who, although we meet very seldom, have been there for the last twenty years.

Countless experiences can be mentioned, omitted here for the sake of brevity, with Arnau, Gerard, Pau, Marc, Eugènia, Laia, Julia, Rai, Ana, Tobi, Homs, Pamias, Pepe, Roger, Oriol, Bea, Thomas and Neli. You made me a better person. Of course, my football team mates have given me a good reason for practising some sport, Carles, David, Gabri, Juanlu, Jordi, Pablo, Buru, Albert and Jorge.

Someone said that university friends are forever, so here is my most sincere greeting to Ignasi, Adrià, Agnès, Borja, BdB, Carla, Pino, Javi, Vero, Gemma B., Gemmita, Dome, Judith, Marcel, Roger, Ricard, Xous, Juanjo...

During the last years, holiday has been synonym of trip, trip synonym of friendship, and friendship synonym of Benjamin, Sigrid, Joaquin, Nilufar, Min, Seung-ku, Kim, Nhung, Thierry, Marie-Pierre, Robinson, Roco and Ael among many others.

I would like to thank Francesc and Cristina for treating me as their own son and for countless dinner invitations.

I wish to express my greatest thanks to Laura, with whom I have shared everything wholeheartedly in the last eight years and the years to come.

I would like to express a special thanks to my family, especially my father and my brothers, Ignasi and Alex.

And last but not the least, I would like to dedicate this thesis to my mother (deceased) who gave me life and love to the extreme. To you.

José M. Tamayo, January 2010.

Contents

Fast Solvers

Singular Integrals

Multi-port Systems

Thesis Conclusions

List of Figures

List of Tables

Acronyms

1-D	One dimensional
2-D	Two dimensional
3-D	Three dimensional
4-D	Forth dimensional
ACA	Adaptive Cross Approximation
ACE	Antenna Centre of Excellence
ACES	Applied Computational Electromagnetics Society
AIM	Adaptive Integral Method
AP-S	Antennas and Propagation Symposium
APC	Author Publication in Conference
APJ	Author Publication in Journal
BiCG	Bi-Conjugate Gradient
BiCGSTAB	Bi-Conjugate Gradient Stabilized
CCIE	Current and Charge Integral Equation
CEM	Computational Electromagnetics International Workshop
CFIE	Combined Field Integral Equation
CG	Conjugate Gradient
CGS	Conjugate Gradient Squared
CN/LT	Constant Normal / Linear Tangential
CPU	Central Processing Unit
CPV	Cauchy Principal Value
dBsm	Decibels relative to a square meter

DE	Double Exponential
DoF	Degrees of Freedom
EFIE	Electric Field Integral Equation
EM	ElectroMagnetic
EMFIE	Electric-Magnetic Field Integral Equation
EPFL	École Polytechnique Fédérale de Lausanne
erfc	Complementary error function
EuCAP	European Conference on Antennas and Propagation
FFT	Fast Fourier Transform
GB	GigaByte
GF	Green's Function
GL	Gauss-Legendre
GLaguerre	Gauss-Laguerre
GMRES	Generalized Minimal RESidual Method
ICEAA	International Conference on Electromagnetics in Advanced Applications
IE	Integral Equations
IEEE	Institute of Electrical and Electronics Engineers
IES3	Integral Equation Solver for 3-D extraction
ILU	Incomplete Lower Upper factorization
INTELECT	INtegral Techniques for ELECTromagnetics
KUL	Katholieke Universiteit Leuven
LEMA	Laboratory of Electromagnetics and Acoustics
LL	Linear-Linear Basis Functions
LN/LT	Linear Normal / Linear Tangential
MDA	Matrix Decomposition Algorithm
MFIE	Magnetic Field Integral Equation
MLACA	MultiLevel Adaptive Cross Approximation
MLFIA	Multilevel Field Interpolator Algorithm

MLFMA	MultiLevel Fast Multipole Method
MLMDA	MultiLevel Matrix Decomposition Algorithm
MoM	Method of Moments
MSCBD	MultiScale Compressed Block Decomposition
MTT-S	Microwave Symposium Digest
ODDM	Overlapped Domain-Decomposition Method
PC	Personal Computer
PEC	Perfectly Electric Conductor
RAM	Random Access Memory
RCS	Radar Cross Section
RWG	Rao-Wilton-Glisson basis functions
SIAM	Society for Industrial and Applied Mathematics
SVD	Singular Value Decomposition
SWR	Standing Wave Ratio
TEM	Transversal ElectroMagnetic
TFQMR	Transpose Free Quasi Minimal Residual
TSC	Teoria del Senyal i Comunicacions
UAB	Universitat Autònoma de Barcelona
UNEX	UNiversidad de EXtremadura
UPC	Universitat Politècnica de Catalunya
UPM	Universidad Politécnica de Madrid
UPV	Universidad Politécnica de Valencia
VIE	Volume Integral Equation

CHAPTER 1

Introduction

THIS chapter puts all the work developed during the realization of this thesis in context. It establishes what was done before in literature showing weak points or gaps to be filled. Many of them have been attempted by the author at some time in the last four years. The proposed solutions are introduced as objectives of the dissertation. Finally, a description of the distribution of the document is given.

1.1 State of the art

1.1.1 Method of Moments generalities

Surface integral equation formulations have proved to be one of the most powerful methods for the solution of various electromagnetic (EM) antenna and scattering problems [1, 2, 3]. Galerkin variants of the method of moments (MoM) [4] are most often utilized for the numerical solution of these electromagnetic surface integral equations.

Normally, three different formulations are encountered, the Electric Field Integral Equation (EFIE), the Magnetic Field Integral Equation (MFIE) and the Combined Field Integral Equation (CFIE), which is a linear combination of the other two. However, each one of the above-mentioned integral operators involve a handful of complications, which include, among others, the so-called low-frequency breakdown, the bad conditioning of the MoM matrix, especially for the EFIE, and inaccuracies in the solution for objects of certain shape or electrical size.

In front of all these difficulties, there are a number of possible strategies, all of them valid and usually complementary. The first one is to develop a new formulation which ideally avoids all the previously encountered problems. Some attempts in this direction can be found in literature (Augmented-EFIE [5, 6], Current and Charge Integral Equation (CCIE) [7]), but they are still in a preliminary state. Another strategy to face these problems is trying to mitigate them, for instance with a preconditioning matrix [8, 9] or a clever choice of basis and testing functions. There is also the possibility of reducing the possible sources of error, for example improving the accuracy on the computation of the impedance matrix elements, to have a better understanding of the root of the issue.

1.1.2 Basis and testing functions

The divergence-conforming basis functions ensure normal continuity of the current across the edges arising from the discretization. They are well-suited for the MoM discretization of the EFIE involving perfectly electric conducting (PEC) objects. The Rao-Wilton-Glisson (RWG) [10] set exhibits constant normal/linear tangential (CN/LT) variation along the edge. The variation of the Linear-linear (LL) set [11, 12, 13] is linear normal/linear tangential (LN/LT), which doubles the number of unknowns. Even though the LL set carries out a higher-order expansion of the current, both RWG and LL sets produce a piecewise constant expansion of the charge density, whereby they are zero-order examples of divergence-conforming sets.

A MoM-discretization of the EFIE for perfectly conducting objects (PeC-EFIE) with the traditional divergence-conforming basis functions suffers from the so-called low-frequency breakdown. At very low frequency regime, the contribution of the vector potential to the impedance matrix becomes negligible, according to the finite machine precision, compared with the scalar potential. This makes the discretization of the PeC-EFIE ill-conditioned and the solution inaccurate.

The rearrangement of the divergence-conforming basis functions as the combination of solenoidal and nonsolenoidal, quasi-irrotational, sets of functions leads to MoM-discretizations of the PeC-EFIE also valid at very low frequencies. The solenoidal subspace captures the null space of the scalar potential and thus allows a stable and accurate discretization at very low frequencies.

The Loop-Tree [14] or the Loop-Star [14, 15, 16] basis functions develop a rearrangement of the RWG current space. The Loop basis functions, expanding the RWG-solenoidal space, are defined around vertices and have zero divergence. They gather all the closed paths arising from the RWG discretization of closed or open and simply connected surfaces. The Tree or the Star basis functions capture the remaining nonsolenoidal space, whereby they have non-zero divergence.

Furthermore, there exist generalized Loop-Tree decomposition schemes for higher-order MoM-discretizations of the EFIE for perfectly conducting objects [17]. Such schemes, though, are not required for the RWG and LL basis functions, which are complete to zero divergence-order. To our knowledge, there is no such decomposition for the LL basis functions available.

1.1.3 Accurate computation of the matrix elements

Weakly singular integrals appear in the EFIE formulations, when divergence-conforming basis and testing functions are incorporated in the Galerkin MoM. Due to the non smooth behavior of the weakly singular integral kernels, classical numerical quadrature rules fail to meet the requirements for high precision results, and more sophisticated techniques are needed to tackle this problem. Generally, various regularization methods for the computation of 4-D weakly singular integrals have appeared in the literature over the last decades and interested readers could consult [18] for a more detailed history of relevant previous work.

On the other hand, 4-D hyper-singular integrals (following the convention used in EM community) arise in the numerical solution of MFIE and CFIE formulations, since the latter are combination of EFIE and MFIE formulations. Over the last years, numerous techniques have been presented for the accurate and efficient evaluation of those multidimensional hyper-singular integrals, which can be roughly categorized into two main families: singularity cancellation methods [19, 20, 21, 22, 23] and singularity subtraction methods [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. The logical expectation is that both methods should lead to superior accuracy compared with direct implementation of quadrature rules. This is not necessarily the case, especially when the quest for machine precision is combined with the need of improved efficiency.

1.1.4 Green's functions

We could decide to include part of the problem to solve into the Green's function (GF) itself rather than discretizing everything. The reason is the reduction in the number of unknowns that we can achieve. However, the GF is hard to find and calculate in a general case. Nonetheless, this is not the case of a multi-layered media, with infinite planar layers or with symmetry properties, or a cavity problem, where we avoid the discretization of the cavity walls.

In the case of rectangular cavities, the Green's function can be expressed as a triple infinite series of cavity modes or images. Both series have their advantages and disadvantages. While the modal series satisfies the boundary conditions and converges fast for far-interactions, it is unable to catch the singular behavior of the GF when source and observer are close to each other. On the other hand, the image series works best in the latter case, but converges slowly if source and observer are far away. Furthermore, it satisfies the boundary conditions only in the limit.

Due to their slow convergence these triple infinite series must be accelerated. For this, various techniques can be found in literature [35, 36, 37, 38, 39, 40], that are often general approaches not taking into account the physics behind the problem.

Another approach is Ewald's technique [41, 42, 43, 44], which is one of the most popular ones in the framework of Cartesian coordinates due to its high efficiency and precision. It is based on a hybridization of both, modal and images series developments combining the "best" of them. Unfortunately, the terms in the images part inside Ewald's method requires the evaluation of the complementary error function (erfc), which is computationally expensive, mainly for the complex arguments needed in frequency dependent problems. To mitigate this problem, the GF could be split into a static and dynamic part using Kummer's transform [45]. But still, the static part contains the evaluation of the erfc with real-valued arguments.

1.1.5 Fast iterative solvers

The main drawback of the MoM is the costly construction, storage, and solution considering the unavoidable dense linear system. As a consequence, a wide range of fast methods [46] have been developed for accelerating the iterative solution of the electromagnetic integral equations [47] discretized by Method of Moments (MoM) [4]. Most of them are based on multilevel subdomain decomposition and require a computational cost per iteration of order $N \log N$ or $N \log^2 N$. One of these methods is the Multilevel Fast Multipole Algorithm (MLFMA) [48]. The MLFMA has been widely used in the last years to solve very large electromagnetic problems [49, 50] due to its excellent computational efficiency.

The main drawback of the MLFMA is the dependence of its formulation on the problem Green's function. Notwithstanding, other general purpose methods have been developed. For instance, the Multilevel Matrix Decomposition Algorithm (MLMDA) [51, 52, 53], exploits the compressibility of MoM sub-matrices corresponding to well separated sub-scatterers by using equivalent surface basis/testing functions cleverly distributed that radiate and receive the same fields as the original basis/testing functions. Another fast solver, the Adaptive Integral Method (AIM) [54] replaces the actual basis and testing functions with a regular volumetrical grid, that again radiates and receives the same fields as the original discretization, in order to efficiently compute the integral equations convolution using the FFT algorithm.

However, all the aforementioned methods rely on the appropriate selection of elements with an equivalent physical behavior, either multipoles or equivalent surface or volume basis/testing functions. Hence the interest of purely algebraic methods, whose formulation is still independent of the problem Green's function and operate solely with some of the impedance matrix elements, such as the IE-QR Algorithm [55], the IES3 [56, 57] and the Adaptive Cross Approximation (ACA) [58]. Unfortunately, these algebraic methods present an asymptotic computational time and memory requirement not as good as that of the above-mentioned methods.

1.1.6 Fast direct solvers

These new iterative algorithms have enormously extended the range of problems that MoM can manage. The maximum affordable number of unknowns used to be limited to a few thousands. Now problems with hundreds of thousands or even millions of unknowns are within reach, depending on the available computational resources.

Based on the previous, one might be tempted to conclude that direct solution of the MoM matrix equation, through Gaussian elimination or LU decomposition, has become obsolete. This is not entirely the case, for the following number of reasons:

- First of all, the fast algorithms are very efficient for electrically large structures, while the direct solution is faster for small and medium size problems, depending on the computational resources and on the specifics of the problem. The turning point may be of the order of a few thousand unknowns.

- Furthermore, iterative solution methods for matrix equations yield the solution to the linear system for only one independent vector (excitation vector) at a time. Consequently, the computational effort is proportional to the number of independent vectors. By contrast, the bulk of the effort in LU decomposition, which is the generation of the L and U factors of the impedance matrix, needs to be done only once for as many independent vectors as needed.

- Finally, the convergence rate of iterative methods can vary in an unpredictable way. It is related to the matrix condition number, which is notoriously bad for the EFIE in large problems. The only remedy is the use of a good preconditioner with a relatively large number of non-zero elements, but the construction of such a preconditioner becomes the bottleneck of the computation [59, 60, 61].

Therefore, the introduction of an accelerated direct solver would be of much interest.

1.1.7 Post-processing

A lot of resources can be saved in antenna design with the proper use of fast and accurate simulation tools. Before its existence, one was forced to build an expensive prototype for each new modification. Therefore, they call not only for the solution of the MoM linear system but for the obtention of useful information which could be actually measured and contrasted at any time.

The S-parameters or the input impedance are very useful parameters for industry, giving information on the behavior of the antenna or multi-port system. The computation of these parameters requires the definition of a port in the MoM discretization and some post-processing with the resulting surface equivalent currents. Traditionally, the RWG basis functions must have the edges coinciding with the port feeding line. This is a drawback for the final user, who needs either to force this on the original mesh or to re-mesh near the port. Moreover, there are some situations in which the mesh is given by an external peer and it cannot be changed.

1.2 Objectives and hypothesis

In the previous section, several deficiencies found in literature related to the Method of Moments have been highlighted. In the following, a list with the topics which the author has attempted during the realization of this thesis is presented. Moreover, a brief description of how the author has contributed to fill these gaps is included for each item. They have been subdivided in two different lists, depending on whether the item has been placed in the present manuscript or not. In any case, when they are related to one or several publications, either as journal or conference papers, it will be appropriately referenced. These references can be found at the end of the document at the "Author Publications and Awards" Section. The nomenclature APJ-i and APC-i is used meaning Author Publication i in Journal and Conference paper, respectively.

1.2.1 Included in this document

- Fast Iterative solver - MLACA [APJ-1, APC-10-14]:

  We develop the MLACA as a MoM fast iterative solver. It is a multilevel version of the ACA, maintaining the condition of being purely algebraic and GF independent, but reducing the asymptotic computational time and memory requirement.

- Singular integrals - direct evaluation method [APJ-2, APC-2,4,7]:

  The direct evaluation method for the computation of singular integrals is extended to the hyper-singular integrals appearing in the matrix elements of the MFIE and CFIE formulations. It is proved to reach unmatched efficiency and accuracy.

- Near-singular integrals:

  The near-singular case in the hyper-singular integrals appearing in the MFIE and CFIE formulation is analyzed and a possible treatment is given.

- Multi-port systems:

  A generalization of port definition is given so that the port definition is independent of the actual mesh, not needing any re-meshing. It gives freedom to the final user.

1.2.2 Not included in this document

- Green's functions - Rectangular cavity [APJ-3]:

  A novel accelerated technique for the computation of the static part of a rectangular cavity GF is developed. Similarly to Ewald's method, it combines the "best" of modal and images series. Unlike Ewald's method is has a clear physical interpretation and it does not require the evaluation of the erfc function.

- Basis functions

- Self-loop [APJ-5, APC-12,13]:

  A Loop-Star decomposition for the LL set is obtained with the inclusion of the Self-loop basis functions. The Self-loop basis function are solenoidal (zero divergence). The LL basis functions are decomposed into solenoidal (RWG loops + Self-loops) and non-solenoidal (RWG stars) parts to make the EFIE stable at very low frequencies, when a LL discretization is utilized.
- Orthogonal basis functions [APJ-4, APC-4,9]:

  A set of orthogonal basis functions as an alternative to the RWG is introduced to improve the accuracy of MFIE for sharp-edged objects in the low frequency regime.
- Macrobasis functions [APC-16]:

  Some facts on the use of macrobasis functions (functions defined over relatively large domains) were highlighted and analyzed.

- Integral equation formulation - Electric-Magnetic Field Integral Equation (EMFIE) [APJ-4, APC-1]:

  The EMFIE is a formulation which forces at the same time electric and magnetic boundary conditions. It is expected to solve the problems arising in other formulations with the proper selection of basis and testing functions.

- Accelerated direct solver

  - CBD[APJ-7,9,10, APC-26,27]:

    Contribution to the Compressed Block Decomposition algorithm for the direct solution of moderately large electromagnetic problems.
  - MDA-CBI [APC-20,23]:

    Introduction of the MDA algorithm inside the CBD as an alternative to the ACA.
  - MSCBD [APJ-6, APC-5,6,8,19]:

    Multiscale version of the CBD algorithm, which reduces the computational cost and memory requirement, allowing for the solution of problems electrically larger.

- Fast iterative solvers

  - MDA-SVD [APJ-8, APC-8,11,20,28]:

    Introduction of a Singular Value Decomposition (SVD) re-compression to the MDA algorithm which improves performance.
  - MLFIA [APC-18,25]:

    The Multilevel Field Interpolator Algorithm is a fast iterative method based on the introduction of equivalent functions, phase subtraction, interpolation from one level to the next and phase restoring.

1.3 Methodology

The utilized methodology throughout the thesis has been in general, in this order: detection of a problem present or stated in literature; deep analysis of the problem to localize its causes, if any, usually based on numerical experiments; development of a theory, sometimes based on previous existing works, combined with the simultaneous development of an algorithm related to the theory; simulation of relevant numerical examples which prove the validity of the developed theory and algorithm and its consequences.

1.4 Organization of the document

The present document has been globally subdivided in five parts: introduction, fast solvers, singular integrals, multi-port systems and conclusions. As a general remark, every chapter starts with a short paragraph in an abstract-like manner, usually followed by an introduction with a specific state of the art to place the treated problem. Afterwards, the body of the chapter comes, to be followed by numerical results and final conclusions.

The first part comprises the present introductory chapter and Chapter 2, which gives a brief review of the Method of Moments (MoM), this being the common point to all the work-lines presented in this dissertation. No novel material is found in this part.

The second part, Fast Solvers, comprises Chapters 3 and 4, which detail the theory, implementation and testing of the Multilevel Adaptive Cross Approximation (MLACA) algorithm. In particular, Chapter 3 states the method in its original form and Chapter 4 sets a theoretically justified modification in the tree decomposition. As a consequence of this modification, the results are much improved making the method worth even for moderate electrically large electromagnetic problems.

The third part, Singular Integrals, contains Chapters 5 and 6. The first one describes the direct evaluation method adapted to the hyper-singular integrals appearing in the MFIE formulation. The numerical results section proves the superiority in efficiency and accuracy in comparison with the existing alternatives in literature to tackle these kind of integrals when the triangle domains share a common edge or vertex. Chapter 6 continues with the treatment of this kind of integrals, though forcing to use exclusively numerical integrations. Some problems arise and a number of possible solutions are presented, including the treatment of the near-singular case, where the triangle domains are very close to each other but not touching.

The forth part, Multi-port Systems, coincides with Chapter 7 which deals with the obtention of measurable parameters, like the input impedance of a system, from the current density solution obtained from the MoM. Furthermore, a new theory to deal with ports, gives freedom to a final user of an electromagnetic software to define the mesh independently of the position of the port.

Finally, some conclusions and further discussion from the present dissertation are outlined in the last part.

Chapter 2

Method of Moments (MoM)

This chapter provides a brief overview, in an introductory manner, of the so-called Method of Moments. However no novel material is included in the chapter, the MoM represents the pilar around which the rest of the chapters find a connecting point. Therefore, it is interesting, if not mandatory, to supply the basics of the formulation, particularly for the Galerkin surface integral equations, which are the most used ones during the manuscript.

2.1 Introduction

The MoM became popular in the middle 60s thanks to the work of Mei-Van Bladel [62], Richmond [63] and Harrington [64, 4]. Since then, its use has grown in parallel with the modern computer resources.

2.2 General algorithm formulation

In the process of solving integral equations we can represent these functional equations in the form:

$$\mathcal{F}X = Y \tag{2.1}$$

where $\mathcal{F}$ is a linear operator, X is the unknown function (equivalent currents) and the function Y is the independent term (incident field). To solve (2.1) in a computer, it is necessary to discretize functions and operators converting the functional equation into a matrix equation or system.

Firstly, the unknown function X is approximated by a finite linear combination of basis functions x_n

$$X_N = \sum_{n=1}^{N} c_n x_n \approx X \tag{2.2}$$

where the N coefficients c_n are the samples of the discretization of X and constitute the unknowns of the numerical problem to solve. In order that X_N is a good approximation of X, it is necessary that the basis functions x_n belong to the domain of the operator $\mathcal{F}$, i.e., that they respect the boundary and differentiability conditions of the operator.

Substituting (2.2) into the functional equation (2.1) and using the linearity of the operator $\mathcal{F}$ we obtain

$$Y_N = \sum_{n=1}^{N} c_n \mathcal{F} x_n \approx Y \tag{2.3}$$

which is a functional equation as well, though with N numerical unknowns c_n instead. So that an appropriate solution exists, it is necessary that the function Y can be adequately approximated by a linear combination of the functions $\mathcal{F}x_n$. To accomplish this, the basis functions x_n must be chosen so that the functions $\mathcal{F}x_n$ are in the range space of the operator $\mathcal{F}$ and that the set $\{\mathcal{F}x_n\}$ is a complete base of the range space of $\mathcal{F}$ when the order of the discretization N tends to infinite. It is equivalent to say that the field must be representable in the basis of functions $\{\mathcal{F}x_n\}$.

The error in the boundary conditions or residual is

$$R = Y - Y_N = Y - \sum_{n=1}^{N} c_n \mathcal{F} x_n. \tag{2.4}$$

To convert the functional equation (2.1) into a matrix system of M equations with N unknowns, we set to zero the residual adjusted with M weighting functions w_m

$$\langle w_m, R \rangle = 0 \qquad\qquad m = 1 \ldots M \tag{2.5}$$

where the scalar product is defined as the Hilbert inner product

$$\langle g, f \rangle = \int_{D(S)} g^*(\mathbf{r}') f(\mathbf{r}') ds' \tag{2.6}$$

resulting finally in the system of M equations and N unknowns

$$\langle w_m, Y \rangle = \sum_{n=1}^{N} c_n \langle w_m, \mathcal{F} x_n \rangle \qquad\qquad m = 1...M \tag{2.7}$$

which may be written in matrix form as

$$[\mathbf{A}]\mathbf{c} = \mathbf{b} \qquad A_{mn} = \langle w_m, \mathcal{F} x_n \rangle \qquad b_m = \langle w_m, Y \rangle\,. \tag{2.8}$$

To solve this system, if $M = N$ the matrix $[\mathbf{A}]$ is square and the system easily solved as

$$\mathbf{c} = [\mathbf{A}]^{-1}\mathbf{b} \tag{2.9}$$

otherwise ($M > N$), we need to solve the system in a minimum least squares way utilizing the pseudo-inverse or generalized inverse

$$\mathbf{c} = ([\mathbf{A}]^H [\mathbf{A}])^{-1} [\mathbf{A}]^H \mathbf{b} \tag{2.10}$$

where $[\mathbf{A}]^H$ is the adjoint matrix of $[\mathbf{A}]$, i.e., the conjugate transpose of $[\mathbf{A}]$.

When the coefficients c_n are calculated utilizing either (2.9) or (2.10), the numerical solution of the problem X_N is obtained from (2.2). In general, it is an approximated solution because the imposed conditions (2.5) do not necessarily imply a zero residual and because the functional space in which the function X is defined has an infinite dimension, while it is approximated by a function X_N with a finite number N of coefficients c_n.

2.3 Integral operators in electromagnetism

The boundary conditions on the surface S of a perfectly electric conductor (PEC) state that the total tangential electric and magnetic fields ($\mathbf{E}^t(\mathbf{r})$, $\mathbf{H}^t(\mathbf{r})$) must accomplish

$$\begin{aligned} \hat{\mathbf{n}} \times \mathbf{E}^t(\mathbf{r})|_S &= \hat{\mathbf{n}} \times (\mathbf{E}^i(\mathbf{r}) + \mathbf{E}^s(\mathbf{r}))|_S = 0 \\ \hat{\mathbf{n}} \times \mathbf{H}^t(\mathbf{r})|_S &= \hat{\mathbf{n}} \times (\mathbf{H}^i(\mathbf{r}) + \mathbf{H}^s(\mathbf{r}))|_S = \mathbf{J}(\mathbf{r}) \end{aligned} \tag{2.11}$$

where $\hat{\mathbf{n}}$ is the outward normal vector at the observation point. Moreover, we have decomposed the total fields into incident ($\mathbf{E}^i(\mathbf{r})$, $\mathbf{H}^i(\mathbf{r})$) and scattered ($\mathbf{E}^s(\mathbf{r})$, $\mathbf{H}^s(\mathbf{r})$) fields. The formers are those which would exist in the absence of the object, whereas the latter are the ones generated by the induced current on the surface of the object ($\mathbf{J}(\mathbf{r})$).

Utilizing the mixed potentials theory and considering that the induced magnetic currents $\mathbf{M}(\mathbf{r})$ are zero (only PEC are considered), the scattered electric and magnetic fields can be expressed in terms of the electric vector potential $\mathbf{A}(\mathbf{r})$ and the electric scalar potential $\Phi(\mathbf{r})$ as

$$\begin{aligned} \mathbf{E}^s(\mathbf{r}) &= -j\omega\mathbf{A}(\mathbf{r}) - \nabla\Phi(\mathbf{r}) \\ \mathbf{H}^s(\mathbf{r}) &= \frac{1}{\mu}\nabla \times \mathbf{A}(\mathbf{r}). \end{aligned} \tag{2.12}$$

The vector and scalar potentials can be calculated using its respective dyadic Green's function which include the information of the medium where the electric currents are immersed [65]:

$$\begin{aligned} \mathbf{A}(\mathbf{r}) &= \mu \int_S G(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\, ds' \\ \Phi(\mathbf{r}) &= -\frac{1}{j\omega\epsilon} \int_S G(\mathbf{r},\mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')\, ds'. \end{aligned} \tag{2.13}$$

Unless oppositely said, we mainly consider the homogeneous space Green's function, although many things can be extrapolated to other existing Green's functions:

$$G(\mathbf{r},\mathbf{r}') = \frac{e^{-jkR}}{4\pi R} = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}. \tag{2.14}$$

In some cases and without loss of generality, the 4π constant is not considered to simplify the notation, in which case it is always stated.

Substituting (2.13) into (2.12) and developing, we obtain the following expressions of the fields scattered by the electric currents

$$\begin{aligned} \mathbf{E}^s(\mathbf{r}) &= -j\omega\mu \int_S G(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\, ds' - \frac{1}{j\omega\epsilon}\nabla \int_S G(\mathbf{r},\mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')\, ds' \\ \mathbf{H}^s(\mathbf{r}) &= \int_S \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{J}(\mathbf{r}')\, ds' \end{aligned} \tag{2.15}$$

which can be represented in terms of the linear operators

$$\begin{aligned}\mathcal{L}(\mathbf{J}) &= -jk\eta \int_S G(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\, ds' + j\frac{\eta}{k}\nabla \int_S G(\mathbf{r},\mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')\, ds' \\ \mathcal{K}(\mathbf{J}) &= \int_S \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{J}(\mathbf{r}')\, ds'.\end{aligned} \tag{2.16}$$

Finally, introducing (2.16) into (2.11) we obtain the expressions of the so-called electric field integral equation (EFIE) and magnetic field integral equation (MFIE)

$$\begin{aligned}\hat{\mathbf{n}} \times \mathcal{L}(\mathbf{J})|_S &= -\hat{\mathbf{n}} \times \mathbf{E}^i(\mathbf{r})|_S \qquad \text{EFIE} \\ \frac{1}{2}\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}} \times \mathcal{K}(\mathbf{J})|_S &= -\hat{\mathbf{n}} \times \mathbf{H}^i(\mathbf{r})|_S \qquad \text{MFIE}\end{aligned} \tag{2.17}$$

where the integral inside $\mathcal{K}(\mathbf{J})$ must be understood in the Cauchy Principal Value (CPV) sense, as the other part is in the $\mathbf{J}(\mathbf{r})/2$ term.

When including these operators in the framework of the Method of Moments, with the inclusion of convenient basis and testing functions, it claims for the evaluation of the integrals at points $\mathbf{r} \in S$, which are infinitesimally close to the source point $\mathbf{r}'$, and therefore with a singular integrand. Consequently, to compute the matrix elements, we need to treat singular integrals, weakly-singular in the EFIE or hyper-singular in the MFIE, which claim for sophisticated techniques or quadrature rules. These sort of singular integrals are treated in Chapters 5 and 6.

Both EFIE and MFIE apply for closed surfaces, whereas only EFIE can be applied for open surfaces. Moreover, when the problem is solved at a frequency equal to the resonance frequency of the object, the uniqueness theorem fails and infinite solutions might appear accomplishing the boundary conditions (they accomplish the homogeneous integral equations). This is translated into a very ill-conditioned matrix system, which becomes hardly solvable.

To avoid the resonance problem, having into account that the resonance frequencies of EFIE and MFIE differ form each other, the combined field integral equation (CFIE) appeared as a linear combination of the EFIE and MFIE [3]. The main problems of the CFIE are that we need to compute both EFIE and MFIE integrals, making it less efficient and that, at least in its first approach, it is only applicable to closed surfaces.

As a last remark, the EFIE is a first kind integral operator while the MFIE is a second kind operator. This property has direct consequences in the conditioning of the posterior matrix as well as in the precision of the solution in terms of the number of unknowns.

2.4 Basis and testing functions

Having a deep look at (2.16) one observe that an appropriate choice of basis functions must be able to correctly represent the current $\mathbf{J}(\mathbf{r})$ and its divergence. Presently, three schemes are mainly being used to represent three-dimensional objects: *rooftops* [66] over rectangular patches, Rao-Wilton-Glisson (RWG) basis functions over triangular patches [10] and the wire model when a part of the object can be approximated by a thin wire. The main advantage of the triangular mesh against the rectangular one is the larger flexibility to better model arbitrary surfaces with complex shapes.

Although this is the general case, some effort is still being done in different groups both to find better behaved local basis functions, especially for the MFIE, and to use macrobasis functions defined over large domains instead of very localized [67, 68, 69, 70, 71, 72]. The author has contributed with some publications in this sense during the realization of the thesis, although it has not been included in the present document (refer to Author Publications Section [APJ-5, APC-1,4,9,12,13,16]).

We will now focus on triangular meshes with RWG basis functions and the wire model as they have been our choice for the examples along the whole dissertation. As for the testing or weighting functions, the same RWG or wire functions will be used. When basis and testing functions coincide it is known as Galerkin method and has been shown to be more precise than *point-matching* method, which uses Dirac deltas as testing functions. Moreover, with Galerkin method, some simplifications in the formulation [10] can be done accelerating the computation of the elements of the impedance matrix corresponding to the interaction between two far triangles, supposing that the incident field and the potentials are constant inside these elements.

2.4.1 RWG functions

Once the object has been meshed into triangular elements, we define the RWG basis function $\mathbf{f}_n(\mathbf{r})$ associated to each edge l_n, common to the triangles T_n^+ and T_n^- (Fig. 2.1) as

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{l_n}{2A_n^+}\boldsymbol{\rho}_n^+(\mathbf{r}) & \text{if } \mathbf{r} \in T_n^+ \\ \frac{l_n}{2A_n^-}\boldsymbol{\rho}_n^-(\mathbf{r}) & \text{if } \mathbf{r} \in T_n^- \end{cases} \tag{2.18}$$

where $\boldsymbol{\rho}_n^+(\mathbf{r})$ is the vector from the vertex of T_n^+ opposite to l_n to $\mathbf{r}$, $\boldsymbol{\rho}_n^-(\mathbf{r})$ from $\mathbf{r}$ to the vertex of T_n^- opposite to l_n and $A_n^\pm$ is the area of the triangle $T_n^\pm$.

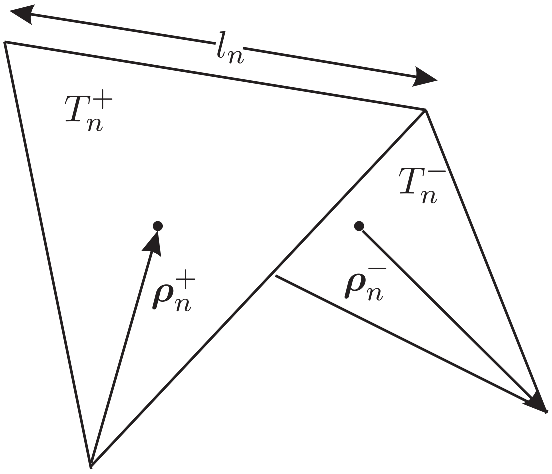


Figure 2.1: Triangular elements pair with a common edge above which the RWG basis functions are defined.

This div-conforming basis function have the following properties:

- The function $\mathbf{f}_n(\mathbf{r})$ has no normal components crossing the external edges of the triangles T_n^+ and T_n^-. The normal component across the internal edge l_n is equal to 1, constant and continuous along the edge. Then, in the approximation of the total current

$$\mathbf{J}(\mathbf{r}) \approx \mathbf{J}_N(\mathbf{r}) = \sum_{n=1}^{N} c_n \mathbf{f}_n(\mathbf{r}) \tag{2.19}$$

where N is the number of edges common to two triangles, the coefficients c_n are the value of the current density which crosses the edge l_n in the normal direction. A positive value of c_n indicates a current flux flowing from T_n^+ to T_n^-. The value of $\mathbf{J}(\mathbf{r})$ inside each triangle is a linear vectorial interpolation of the three values c_n of the components of $\mathbf{J}(\mathbf{r})$ normal to the three edges.

- The surface charge density is

$$-j\omega\rho_s = \nabla_s \cdot \mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{l_n}{A_n^+} & \text{if } \mathbf{r} \in T_n^+ \\ -\frac{l_n}{A_n^-} & \text{if } \mathbf{r} \in T_n^- \end{cases} \tag{2.20}$$

  so that the total charge in the pair of elements is null.

- In the case of open surfaces, there are no normal components of $\mathbf{J}_N(\mathbf{r})$ in the external boundary, which corresponds with the physical reality.

2.4.2 Wire model functions

We have considered a very simple model which assumes a constant value of the current in the revolution of the cylinder, i.e., it is constant along each circular section. We have a basis function per each vertex common to a pair of wire segments, and its value decreases linearly to zero when going to the outer edges of these wires. In particular, the expression of the basis function $\mathbf{f}_n(\mathbf{r})$ is

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{1}{(2\pi a)l_n^+}(\mathbf{r}_c - \mathbf{r}_1) & \text{if } \mathbf{r} \in W_n^+ \\ \frac{1}{(2\pi a)l_n^-}(\mathbf{r}_2 - \mathbf{r}_c) & \text{if } \mathbf{r} \in W_n^- \end{cases} \tag{2.21}$$

where a is the radius of the cylinders, $l_n^\pm$ is the length of the cylinder $W_n^\pm$, $\mathbf{r}_c$ is the projection of $\mathbf{r}$ into the axis of the cylinder and $\mathbf{r}_1$ and $\mathbf{r}_2$ are the opposite vertices of the cylinders W_n^+ and W_n^-, respectively.

2.5 Solvers

Commonly, the size of the discretized elements is chosen proportionally to the wavelength λ (around $\lambda/10$ gives normally quite accurate results) considered that a larger variation in the surface current is expected for larger frequencies. It is translated into an increasing number of unknowns N with the growing of the electrical size of the object. Moreover, the MoM impedance matrix is a full matrix, so the matrix storage scales with $O(N^2)$, limiting the size of the objects that we can analyze if no improvement or compression is applied. To have an idea, with a PC with 64GB of RAM memory, problems no larger than 65000 unknowns can be solved, while applying the fast solvers existing in literature or in particular the ones developed in this thesis, we are able to solve problems in this same machine with more than 1 million of unknowns, which would need more than 16000GB of memory.

These solvers can be categorized into direct and iterative solvers, whose treatment is now in order.

2.5.1 Direct methods

A direct method requires the computation of the inverse of the matrix or an equivalent factorization, and the independent vectors are applied afterwards to this inverse matrix. The direct methods are very well-suited for problems which require the solution of a system for many different independent vectors, as is the case of the mono-static RCS for different incident angles. Furthermore, they are less affected than iterative methods by the bad-conditioning of the matrix, as a solution is always reached. However, traditional general direct methods scale with $O(N^3)$ what makes it unfeasible for objects with moderate electrical size. Notwithstanding, compression and inversion techniques based on the physics of the problem, can reduce this asymptotical behaviors, making affordable, in a reasonable time, the solution of much larger objects. When the solution of the system is only needed for a reduced number of incident fields, iterative methods are still the appropriate choice. The author has contributed in the development of these accelerated direct methods ([APJ-6,9,10, APC-5,6,8,19,20,23,26,27]).

2.5.2 Iterative methods

Iterative methods are the most used nowadays for the solution of very large MoM problems. They start from an initial guess of the solution, usually not very important and set to a null vector, and performing one or more matrix-vector products per iteration, depending on the actual method, they converge to the solution reducing the norm of the residual r_i at each iteration i:

$$\begin{aligned} \text{System to solve:} \quad & Ax = b \\ & r_i = \frac{\|Ax_i - b\|_2}{\|b\|_2}. \end{aligned} \tag{2.22}$$

There are several iterative methods present in literature, each of them with different properties which determine per each problem the number of iterations necessary to converge up to a certain residual. Without entering into the details, we just mention some of them: conjugate gradient (CG), bi-conjugate gradient (BiCG), conjugate gradient squared (CGS), bi-conjugate gradient stabilized (BiCGSTAB), transpose free quasi minimal residual (TFQMR) and generalized minimal residual method (GMRES). Unless stated otherwise, we will use the GMRES for all the simulations.

The computational time of an iterative method is proportional to the product of the number of iterations k and the cost of a matrix-vector product. If no compression is applied on the original matrix it becomes $O(kN^2)$ which will represent an unreasonable time for very large problems. Notwithstanding, several fast solvers have appeared in literature during the last decades which are able to reduce the computational order of a matrix-vector product as well as of the computation of the compressed form of the matrix. Without being exhaustive, a list of these techniques include: MLFMA, MDA-SVD [APJ-7,8, APC-8,11,18,20,25,28], ACA, MLACA [APJ-1, APC-10,14] (Chapters 3 and 4).

2.5.2.1 Preconditioning

Another important parameter which determines the total time to obtain a solution with an iterative solver is the number of iterations. This number is dependent on the actual iterative method we are using, but most worryingly, it can grow, even to order N, when the condition number of the matrix to solve grows. This is the case of the EFIE formulation in the vast majority of cases. To avoid this bad-conditioning effect, and therefore to reduce the number of iterations, we can introduce a preconditioning matrix. The idea behind the preconditioning is the following: modifying our system so that the solution is the same and the new resulting matrix has a better condition number, i.e.

$$M^{-1}Ax = M^{-1}b \tag{2.23}$$

where if M is as equal to A as possible, the product $M^{-1}A$ will be very close to the identity matrix. Evidently, if $M = A$, one iteration suffices, but the cost of the preconditioning M^{-1} is the same as directly solving the original system. Therefore, there is a trade-off between the cost and size of the preconditioner and the number of iterations of the system.

There are several preconditioning techniques in literature, being the most used an incomplete LU decomposition (ILU) [8] and the recently developed Calderon preconditioner [9], based on the Buffa-Christiansen basis functions.

Fast Solvers

Chapter 3

Multilevel Adaptive Cross Approximation (MLACA)

The Multilevel Adaptive Cross Approximation (MLACA) is proposed as a fast method to accelerate the matrix-vector products in the iterative solution of the linear system that results from the discretization of electromagnetic Integral Equations (IE) with the Method of Moments (MoM). The single level ACA, already described in literature, is extended with a multilevel recursive algorithm in order to improve the asymptotical complexity of both the computational cost and the memory requirements. The main qualities of ACA are maintained: it is purely algebraic and independent of the integral equation kernel Green's function as long as it produces pseudo-rank-deficient matrix blocks. Examples are presented to show the potentiality of the method, although we need to wait to the next Chapter 4, where a slight modification is introduced, to have really relevant examples for the EM community. The algorithm is presented in such a way that it can be easily implemented on top of an existing MoM code with most commonly used Green's functions.

3.1 Introduction

In recent years, a wide range of fast methods [46] have been developed for accelerating the iterative solution of the electromagnetic integral equations [47] discretized by Method of Moments (MoM) [4]. Most of them are based on multilevel subdomain decomposition and require a computational cost per iteration of order $N \log N$ or $N \log^2 N$. One of these methods is the Multilevel Fast Multipole Algorithm (MLFMA) [48]. The MLFMA has been widely used in the last years to solve very large electromagnetic problems [49, 50] due to its excellent computational efficiency.

The main drawback of the MLFMA is the dependence of its formulation on the problem Green's function. Notwithstanding, other general purpose methods have been developed. For instance, the Multilevel Matrix Decomposition Algorithm (MLMDA) [51, 52, 53] or its renewed version, the Matrix Decomposition Algorithm - Singular Value Decomposition (MDA-SVD) [73] exploit the compressibility of MoM sub-matrices corresponding to well separated sub-scatterers by using equivalent surface basis/testing functions cleverly distributed that radiate and receive the same fields as the original basis/testing functions. Another fast solver, the Adaptive Integral Method (AIM) [54] replaces the actual basis and testing functions with a regular volumetrical grid, that again radiates and receives the same fields as the original discretization, in order to efficiently compute the integral equations convolution using the FFT algorithm.

However, all the aforementioned methods rely on the appropriate selection of elements with an equivalent physical behavior, either multipoles or equivalent surface or volume basis/testing functions. Hence the interest of purely algebraic methods, whose formulation is independent of the problem Green's function and operate solely with some of the impedance matrix elements, such as the IE-QR Algorithm [55], the IES3 [56, 57] and the Adaptive Cross Approximation (ACA) [58]. Unfortunately, these algebraic methods present an asymptotic computational time and memory requirement not as good as that of the above-mentioned methods.

The Adaptive Cross Approximation method (ACA), which is the basis of the multilevel algorithm proposed here, was developed in 2000 by Bebendorf [58] and since then has been widely used to solve large magnetostatic problems [74]. Recently, the ACA has been applied also to antenna radiation and RCS computation [75, 76]. For very large problems, the compression of well separated sub-scatterers matrix blocks with ACA has an asymptotical computational complexity of $O(\bar{N}^3)$ for matrix compression and $O(\bar{N}^2)$ for storage, as proved in [77], where $\bar{N}$ is the number of unknowns of each sub-scatterer. In practice, these asymptotical behaviors are not reached unless $\bar{N}$ is of the order of several millions, and therefore for moderately large values of $\bar{N}$ one usually observes a much lower computational complexity.

The aim of this chapter is to present a multilevel version of the ACA method, using physical concepts similar to those presented in [51] for the MLMDA. Maintaining the purely algebraic formulation and in comparison with the single level ACA, the computational cost and memory requirements are reduced to $O(\bar{N}^2)$ and $O(\bar{N}\log\bar{N})$, respectively. The memory, and therefore the matrix-vector product time, is comparable to that of the MLFMA.

3.2 Description of the Algorithm

3.2.1 Binary Tree Domain Decomposition

The idea of the binary domain decomposition is to start with a discretized object and split it recursively so that at each new level, every subset of the previous level is subdivided into two new subsets. The criterion for the partition of each subset can be set depending on the application but is usually related with the distribution of the discrete samples in space. Fig. 3.1 displays a representation of such a decomposition.

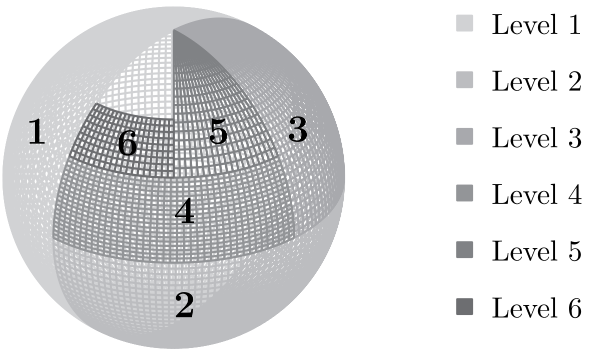


Figure 3.1: Multilevel decomposition of a sphere resulting from a binary tree subdivision. Only one block for each level is represented. This decomposition is used at two different stages in the algorithm: the global object multilevel subdivision tree and for simple MLACA boxes.

A formalization of this idea can be found in appendix A. Just note that the splitting criterion in our algorithm is adjusting a cuboid to each subdomain and cutting it in half across its largest dimension.

Similarly to many existing methods, this decomposition is first used to break the global object under study, and consequently the associated impedance matrix, into near and far interactions. Furthermore, and most importantly for us, when a far-interaction submatrix is to be computed, both the source and the observation boxes are split using this binary decomposition in order to apply the MLACA algorithm.

This was the original criterion we chose, and the one presented in some conferences. However, as will be shown in the next Chapter, another tree subdivision based on the solid angle seen between each pair of far-boxes will produce an important improvement on the behavior of the algorithm.

In the matrix representations throughout the chapter, the basis and testing functions will be reordered following the tree decomposition described, in order to obtain a block-like representation. It means that functions in the same box of the tree are represented by contiguous rows/columns of the matrix.

3.2.2 Adaptive Cross Approximation - Singular Value Decomposition (ACA-SVD)

The Adaptive Cross Approximation (ACA) [58] is a pure algebraic technique for matrix compression. It accurately decomposes a low τ-rank matrix arising from an asymptotically smooth function into the product of two new matrices in a compressed form:

$$M \approx UV \tag{3.1}$$

being M a $m \times n$ matrix, U a $m \times k$ matrix and V a $k \times n$ matrix with $k \ll m, n$, where k is the τ-rank or number of degrees of freedom (DoF). The error in this approximation is controlled by a threshold τ which determines when to stop looking for more columns and rows of U and V, respectively.

Most of the open Green functions in electromagnetism lead to an integral equation function (EFIE, MFIE, CFIE, ...) which, when dealing with the interaction between two well-separated blocks in space, lead in turn to an interaction matrix with a deficient pseudo-rank that can be efficiently compressed with the ACA technique.

The ACA has a computational cost or number of operations scaling with $k^2(m+n)$ whereas the final amount of necessary memory scales with $k(m+n)$. The last quantity coincides also with the number of matrix elements evaluations in M to decompose $M \approx UV$ allowing us not to compute the whole matrix. Although this number of evaluations is asymptotically smaller than the computational cost of the ACA, for medium size problems it can dominate the total computational time as Green function evaluations can be highly time consuming.

We are going to apply a singular value decomposition (SVD) recompression (see algorithm **ACA-SVD-R** below) to the obtained ACA compression for two different reasons. First of all, to have a better compression as the last or smallest singular values coming from the ACA decomposition are usually not very accurate and we can remove them without losing accuracy. And secondly, it will allow us to make the right-hand side outer matrix orthonormal, which is a key point in the multilevel ACA (MLACA), because all the information of the matrix (singular values) is kept in the left-hand side.

Table 3.1: Asymptotic Computational Order Cost of the Different Used Sub-methods

	dims(M)	DoF(M)	Comp. Cost
QR(M)	$m \times k$	-	mk^2
ACA(M, τ)	$m \times n$	k	$k^2(m+n)$
SVD(M, τ)	$k \times k$	$k' < k$	k^3
ACA-SVD-R(M, τ)	$m \times n$	k	$k^2(m+n)$

Function ACA-SVD-R(M, τ)

```
$U, V \leftarrow \mathbf{ACA}(M, \tau/10)$
$Q_u, R_u \leftarrow \mathbf{QR}(U)$
$Q_v, R_v \leftarrow \mathbf{QR}(V^H)$
$U, S, V \leftarrow \mathbf{SVD}(R_u R_v^H, \tau)$
return $Q_u US, VQ_v^H$
```

where the subfunctions accomplish:

$$Q, R = \mathbf{QR}(M) \Longrightarrow M = QR \tag{3.2}$$

$$U, V = \mathbf{ACA}(M, \tau) \Longrightarrow M \approx UV \tag{3.3}$$

$$U, S, V = \mathbf{SVD}(M, \tau) \Longrightarrow M \approx USV \tag{3.4}$$

being **QR** and **SVD** the well known matrix algebra algorithms [78].

It can be easily proved that $M \approx (Q_u US)(VQ_v^H)$ and that the returning right-hand side outer matrix VQ_v^H is orthonormal taking into account the orthogonality properties of the **QR** and **SVD** decompositions. The computational cost and required storage scale the same as the single ACA (see Table 3.1).

In the algorithm implementation there will be two different functions **ACA-SVD-R** and **ACA-SVD-MR**. The only difference between them is that in the former we have the input matrix M already computed whereas in the latter the elements are computed on demand with the corresponding Green function.

3.2.3 Degrees of Freedom

Let us consider the behavior of the number of degrees of freedom (DoF) of the matrix representing the interaction between two well-separated boxes of samples, source and observation boxes, respectively.

For a given frequency and distance, when the size of the source box grows by a certain factor and the size of the observation box decreases by the same factor, the number of DoF remains asymptotically constant for large boxes (a proof of this can be found in [51]). This property allows us to maintain the same number of DoF from one step to the other in the MLACA algorithm.

When the frequency grows, the sizes of the boxes and the distance between them become electrically larger if they are physically the same. Now, if the physical size of one of the boxes, the source box for instance, is reduced, so that its electrical size is constant, then the number of DoF is also constant. This can be understood considering that the growth in electrical distance compensates the growth in electrical size of the observation box. This means that if only the frequency is changed, the only necessary thing to do to maintain the same number of DoF per sub-block is to fix the electrical size of the finest level, which is what the defined binary tree does. This property allows us to maintain the same number of DoF, more boxes, but with constant DoF, when the frequency changes in the MLACA algorithm.

3.2.4 Multilevel Adaptive Cross Approximation (MLACA)

Hereupon and up to Section 3.5 we address the interaction between two boxes or sets of samples contained within two spheres that do not intersect each other. These boxes come from a domain decomposition of the object that will be introduced in Section 3.6. We shall call the two separated boxes, source and observation boxes, respectively. Applying the Method of Moments (MoM) it is possible to express the interaction between them with an impedance matrix Z_{mn} with dimensions $m \times n$, where m and n are the number of basis functions in which the observation and source boxes have been discretized, respectively. In the first place, we must subdivide each box recursively into smaller domains in a binary tree manner, so that in each subdivision approximately half of the basis functions goes to each side. If L is the number of levels, at the finest level we have in each subset or subdomain approximately $M = \bar{N}/2^L$ elements, considering $\bar{N} = n = m$. For asymptotical analysis, L is chosen to yield a fixed minimum box electrical size.

For the sake of clarity, the algorithm will be explained below in the easiest way to understand. The next Section will address the actual implementation in which a reordering of operations improves performance.

In step 0 (Fig. 3.2(a)) Z_{mn} is transformed into the product of two new matrices $A_0^{(1)}$ and $B_0^{(1)}$. The procedure to obtain them is to split Z_{mn} into strips, each corresponding to the interaction of one subset of basis functions at the finest level in the source box with the whole observation box (see first row of pictures in Fig. 3.3). Each of those strips only has k degrees of freedom. They can be compressed with the ACA-SVD algorithm and regrouped (see Fig. 3.2(a)) as follows. The left-hand side matrices resulting from each ACA-SVD are horizontally concatenated forming the new matrix $A_0^{(1)}$, whereas the right-hand side matrices are diagonally concatenated forming the new block-diagonal matrix $B_0^{(1)}$. In that way, the product of the two new matrices gives us an approximation of the original Z_{mn} matrix. Then only the 'B' matrices need to be permanently stored and the 'A' matrices will be subdivided and re-compressed in the next steps of the algorithm.

At step 1 ($i = 1$), the matrix $A_0^{(1)}$ from level 0 is transformed into four new matrices $A_0^{(2)}$, $A_1^{(2)}$, $B_0^{(2)}$, $B_1^{(2)}$ (Fig. 3.2(b)) as follows. Due to the proper re-ordering of columns according to the tree decomposition of the source box, consecutive couples of strips in matrix $A_0^{(1)}$ correspond to the two children of each box in the next decomposition level. Similarly, due to the proper re-ordering of rows according to the tree decomposition of the observation box, the matrix $A_0^{(1)}$ can be horizontally split obtaining that each subblock corresponds to each one of the two halves of the observation object. The resulting upper and lower matrices have new strips with again k degrees of freedom because the size of the involved boxes is halved and doubled in the observation and source objects, respectively. As the actual number of columns of the strips is $2k$, they can be re-compressed with the ACA-SVD and re-group for each matrix in the same manner as in step 0. The obtained new

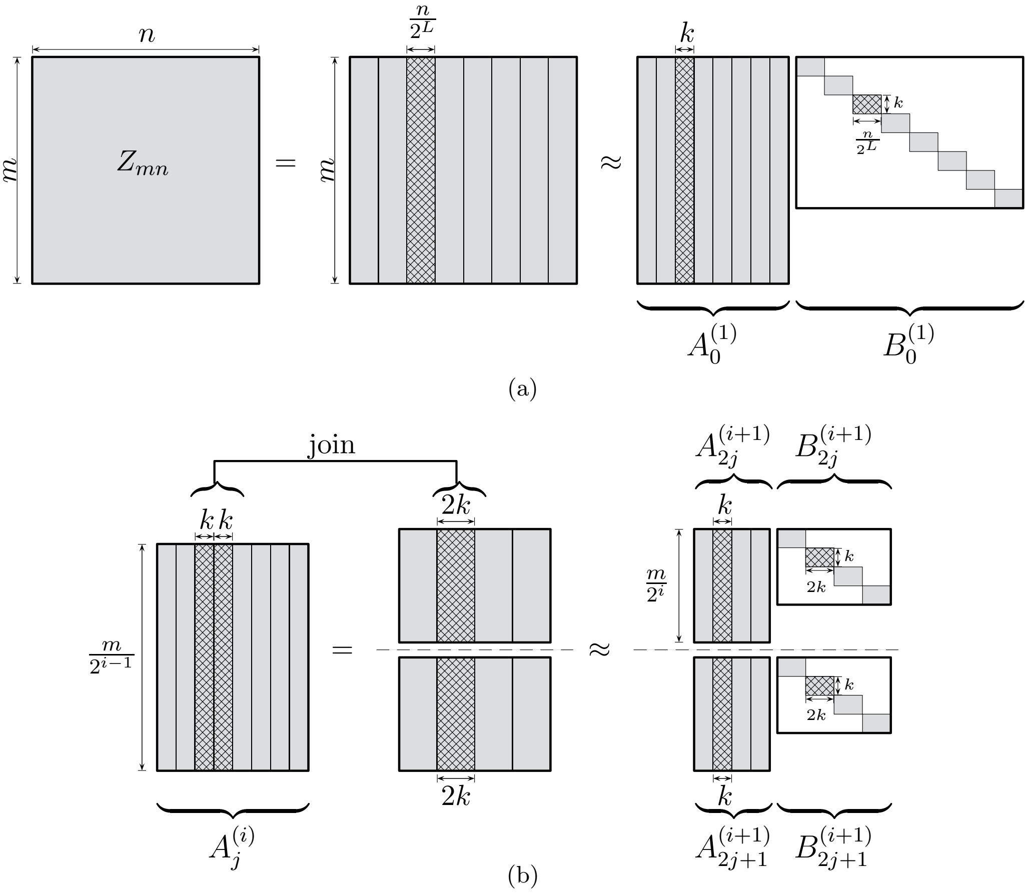


Figure 3.2: Graphical representation of the matrix transformations in the MLACA algorithm described in Section 3.2. First step or step 0 (a) and step i (b) with $i \geq 1$.

matrices $A_0^{(2)}$ and $A_1^{(2)}$ are sent to the next step in order to equally proceed recursively.

Generalizing this procedure in a recursive manner, at step i, the set of 'A' matrices from step $i-1$, $A_j^{(i)}$ $j = 0, \ldots, 2^{i-1}-1$, are transformed into four new matrices $A_{2j}^{(i+1)}$, $A_{2j+1}^{(i+1)}$, $B_{2j}^{(i+1)}$, $B_{2j+1}^{(i+1)}$ (Fig. 3.2(b)). The initial matrix $A_j^{(i)}$ is split into two sets of rows corresponding to the two children subdomains in the observation box tree. On the other hand, the strips are grouped in pairs corresponding to the upper level, or parent box, in the source box tree. This procedure is represented from the object point of view in Fig. 3.3 where each row corresponds to a step of the algorithm. As mentioned in Section 3.2.1, the submatrix grouping and subdivision corresponds to the object domain decomposition (Fig. 3.1 and Fig. 3.3). As the source domain is doubled in size and the observation domain is divided by two, the number of degrees of freedom of the new substrips is again k [51], and therefore can be re-compressed using the ACA-SVD algorithm to obtain the new 'A' and 'B' matrices. After the step $i = L$ we have the whole set of matrices which correctly combined represent a compressed version of the matrix Z_{mn}. Only the 'B' matrices and the '$A^{(L+1)}$' from the last step need to be permanently stored.

In this procedure, all the matrices we extract to the right are orthonormal. According to appendix B, each time we apply **ACA-SVD-R** to a strip, the resulting left-hand side matrix will have exactly the same singular values as the original strip. If a SVD was not applied after the ACA compression in function **ACA-SVD-R**, the singular values would not be the same.

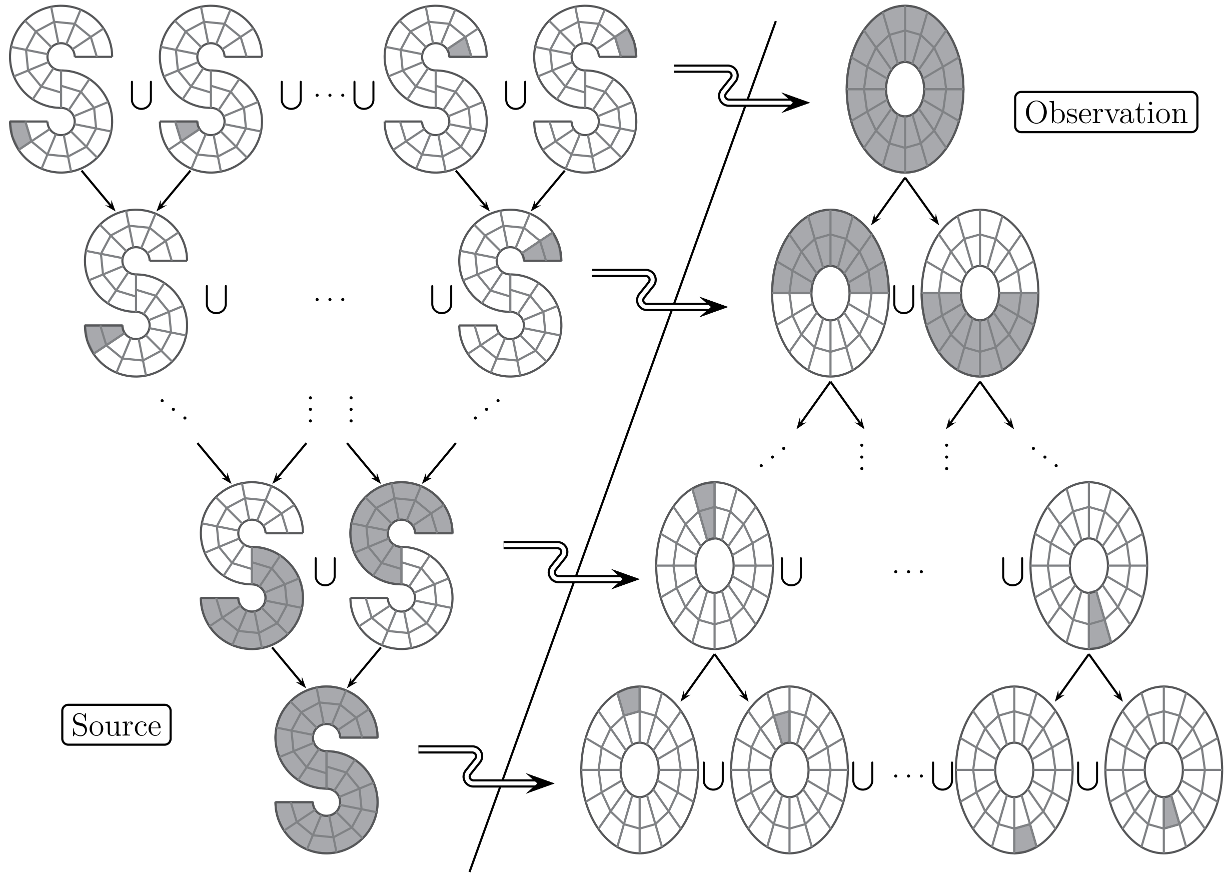


Figure 3.3: Schematic representation of the described multilevel procedure. Each row of figures represents each step on the algorithm. Gray blocks are the ones considered at each step. It starts with the interaction between all the finest level boxes of the source box "S" and the total observation box "O". In the next steps, the source blocks are sequentially grouped whilst the observation boxes are split.

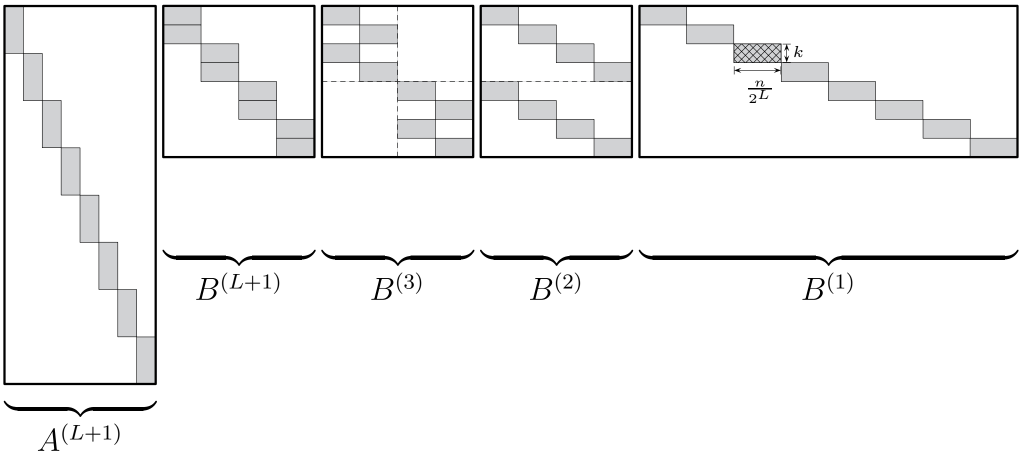


Figure 3.4: Multilevel decomposition matrices for $L = 3$ levels. White parts are zero elements whereas gray blocks are filled.

In order to have our original matrix Z_{mn} decomposed as a product of matrices (see Fig. 3.4 and 3.5) we need to regroup the obtained matrices as follows:

$$B^{(1)} = B_0^{(1)}. \tag{3.5}$$

For each $i = 2, \ldots, L+1$:

$$B^{(i)} = \begin{bmatrix} B_0^{(i)} & & & & & & \\ B_1^{(i)} & & & & & & \\ & \ddots & & & & & \\ & & B_{2j}^{(i)} & & & & \\ & & B_{2j+1}^{(i)} & & & & \\ & & & \ddots & & & \\ & & & & B_{2^{i-1}-2}^{(i)} & \\ & & & & B_{2^{i-1}-1}^{(i)} \end{bmatrix}; \qquad A^{(L+1)} = \begin{bmatrix} A_0^{(L+1)} & & \\ & \ddots & \\ & & A_{2^L-1}^{(L+1)} \end{bmatrix}. \tag{3.6}$$

Note that in this representation we have matrices with a high sparsity distributed by blocks, but actually we do not need to form those sparse matrices because we already know how to use the subblocks.

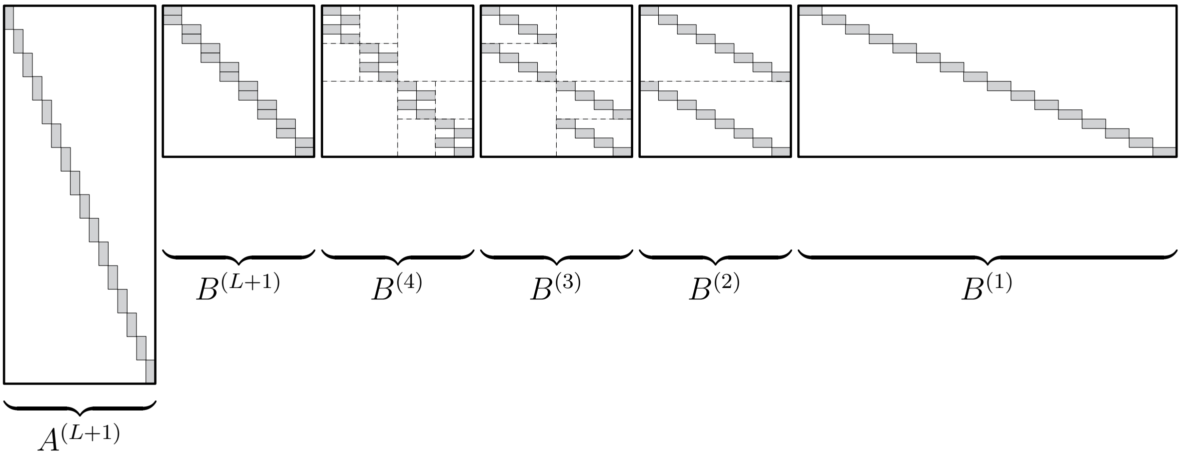


Figure 3.5: Multilevel decomposition matrices for $L = 4$ levels. White parts are zero elements whereas gray blocks are filled.

3.3 Algorithm Implementation

The previous procedure has a computational cost of $O(\bar{N}^2)$ and a final storage of $O(\bar{N} \log \bar{N})$. However, during the algorithm the amount of necessary memory at some steps can be temporarily of

$O(\bar{N}^2)$ (for instance, at step 0, the matrix $A_0^{(1)}$ has $O(\bar{N}^2)$ elements). Nevertheless, this temporary storage can be reduced by simply reordering the operations. It means that we do exactly the same operations and the result is exactly the same but the order of the operations is different.

A representation of that reordering is shown in Fig. 3.6 and 3.7 for respectively 2 and 3 levels. Each row in the figure represents all the matrix blocks involved at each step of the algorithm. The numbers above each matrix determine the order in which the operations are computed.

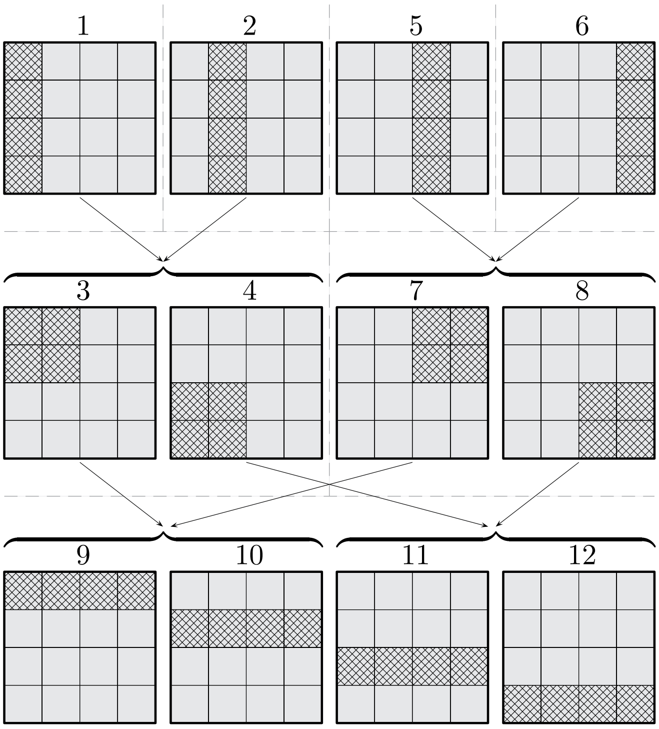


Figure 3.6: Representation of the order to follow (numbers) showing schematically which are the elements of the matrix involved. The arrows display which steps are necessary from the previous row in order to compute the matrices below. In this case there are $L = 2$ levels.

Inside the algorithms, the notation of $A_i^{(l)}[\nu]$ and $B_i^{(l)}[\nu]$ is consistent with the matrices in Fig. 3.2 and in the previous section. Here the ν inside the square brackets means that it corresponds to the ν-th vertical strip of the corresponding matrix.

Function computeCompressedMatrixMlaca(L, τ)

```
$A_0^{(L+1)}[1], \dots, A_{2^L-1}^{(L+1)}[1]$ = **mlacaRec**($L+1, 1, \tau$)
**store** $A_0^{(L+1)}[1], \dots, A_{2^L-1}^{(L+1)}[1]$
```

Function $A_0^{(l)}[\nu], \dots, A_{2^{l-1}-1}^{(l)}[\nu]$ = mlacaRec(l, ν, τ)

```
**if** $l = 1$ **then**
    $A_0^{(1)}[\nu], B_0^{(1)}[\nu] \leftarrow$ **ACA-SVD-MR**($Z_{mn_\nu}, \tau$)
    **store** $B_0^{(1)}[\nu]$
```

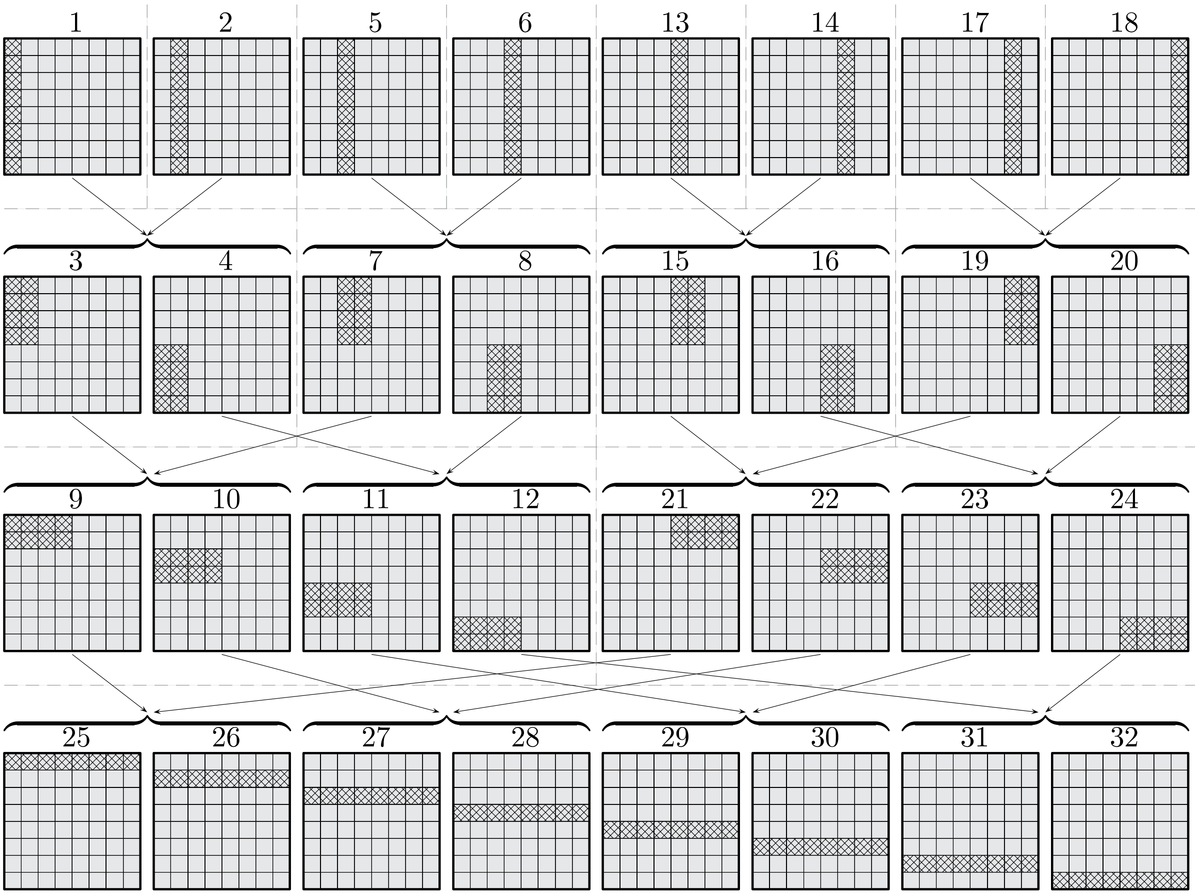


Figure 3.7: Representation of the order to follow (numbers) showing schematically which are the elements of the matrix involved. The arrows display which steps are necessary from the previous row in order to compute the matrices below. In this case there are $L = 3$ levels.

```
    return $A_0^{(1)}[\nu]$
else
    $A_0^{(l-1)}[2\nu-1], \dots, A_{2^{l-2}-1}^{(l-1)}[2\nu-1] \leftarrow$
            mlacaRec$(l-1, 2\nu-1, \tau)$
    $A_0^{(l-1)}[2\nu], \dots, A_{2^{l-2}-1}^{(l-1)}[2\nu] \leftarrow$
            mlacaRec$(l-1, 2\nu, \tau)$
    for $j = 0$ to $2^{l-2}-1$ do
        $aux \leftarrow \left[ A_j^{(l-1)}[2\nu-1] \quad A_j^{(l-1)}[2\nu] \right]$
        $auxUpper$, $auxLower \leftarrow$ splitMatrix$(aux)$
        $A_{2j}^{(l)}[\nu], B_{2j}^{(l)}[\nu] \leftarrow$ ACA-SVD-R$(auxUpper,\tau)$
        $A_{2j+1}^{(l)}[\nu], B_{2j+1}^{(l)}[\nu] \leftarrow$ ACA-SVD-R$(auxLower,\tau)$
        store $B_{2j}^{(l)}[\nu], B_{2j+1}^{(l)}[\nu]$
    end for
    return $A_0^{(l)}[\nu], \ldots, A_{2^{l-1}-1}^{(l)}[\nu]$
end if
```

In order to compute the compressed interaction matrix we only need to call once the function **computeCompressedMatrixMlaca** described above which in turn calls the recursive function **mlacaRec**. The important work is done by the function **mlacaRec** and the former function just initialize the recursion and stores the final resulting matrices.

Function y **= matrixVector(**A, B, x**)**

```
for $l = 1$ to $L + 1$ do
    $x \leftarrow$ matrixVectorLev$(l, B, x)$
end for
for $i = 1$ to $2^{L-1}$ do
    $y_{m_i} \leftarrow A_i^{(L+1)}[1]x\{i\}$
end for
return $y$
```

Function y **= matrixVectorLev(**l, B, x**)**

```
if $l = 1$ then
    for $i = 1$ to $2^L$ do
        $y\{i\} \leftarrow B_0^{(1)}[i]x\{i\}$
    end for
else
    for $i = 1$ to $2^L$ do
        $x_{aux}\{i\} \leftarrow \begin{bmatrix} x\{2i-1\} \\ x\{2i\} \end{bmatrix}$
    end for
    for $i = 1$ to $2^L$ do
        $j \leftarrow \left\lfloor \frac{i-1}{2^{L-l+1}} \right\rfloor$
        $\nu \leftarrow i - j2^{L-l+1}$
        $s \leftarrow \nu + \left\lfloor \frac{i-1}{2^{L-l+2}} \right\rfloor 2^{L-l+1}$
        $y\{i\} \leftarrow B_j^{(l)}[\nu]x_{aux}\{s\}$
    end for
```

```
15: end if
16: return y
```

Once the compressed matrix is computed, to make a matrix-vector product with a vector x we only need to call the function **matrixVector**. This function does the product of each matrix in the multilevel decomposition with the corresponding sub-vector from right to left. Each matrix-vector product with each matrix $B^{(l)}$ is performed by the subfunction **matrixVectorLev**.

There are some parts of the algorithms described above which need a further explanation. There are four issues implicitly related with the binary decomposition defined in Section 3.2.1 and Appendix A. In the **mlacaRec** function line 2, the subindex n_ν is $n_\nu = T_\nu^{(L)}$ and in function **matrixVectorLev** line 3, $x\{i\}$ is $x\{i\} = x(T_i^{(L)})$, T being the tree decomposition of the source box under study. In function **mlacaRec** line 10, the function **splitMatrix**(aux) takes into account that the matrix aux contains the rows related to a certain box in a certain level of the tree decomposition of the observation box and it simply splits the matrix in an upper and lower side in terms of the subdivision of that box at the next level. In function **matrixVector** line 5 y_{m_i} is $y_{m_i} = y(T_i^{(L)})$ being T the tree decomposition of the observation box.

3.4 Computational Cost and Storage

We remind that this section deals with time and storage of the MLACA algorithm applied to the interaction between two separated sets of samples and not to the global algorithm applied to the entire problem under study.

Here k is the constant number of degrees of freedom of each submatrix interaction inside the multilevel procedure defined in Section 3.2.4.

A first initial point is the knowledge of the dimensions of the different matrices involved in the algorithm and from that, it will be easy to evaluate the complexity. For each level $l = 1, \ldots, L+1$, each matrix inside that level $j = 0, \ldots, 2^{l-1} - 1$ and each strip of the matrix $\nu = 1, \ldots, 2^{L-l+1}$, the matrix $A_j^{(l)}[\nu]$ is $\frac{m}{2^{l-1}} \times k$. Similarly, for $l = 2, \ldots, L+1$, $j = 0, \ldots, 2^{l-1} - 1$ and $\nu = 1, \ldots, 2^{L-l+1}$, $B_j^{(l)}[\nu]$ is a matrix $k \times 2k$. And for $\nu = 1, \ldots, 2^L$, $B_0^{(1)}[\nu]$ is $k \times \frac{n}{2^L}$.

The final storage, and therefore the computational cost of a matrix-vector product, corresponds with the whole set of B matrices and the $A^{(L+1)}$ matrices. The total size of the B matrices S_B can be computed as follows:

$$\begin{aligned} S_B &= \sum_{l=1}^{L+1} \sum_{j=0}^{2^{l-1}-1} \sum_{\nu=1}^{2^{L-l+1}} \text{size}(B_j^{(l)}[\nu]) & (3.7)\\ &= \sum_{\nu=1}^{2^L} \text{size}(B_0^{(1)}[\nu]) + \sum_{l=2}^{L+1} \sum_{j=0}^{2^{l-1}-1} \sum_{\nu=1}^{2^{L-l+1}} \text{size}(B_j^{(l)}[\nu]) & (3.8)\\ &= \sum_{\nu=1}^{2^L} k\frac{n}{2^L} + \sum_{l=2}^{L+1} \sum_{j=0}^{2^{l-1}-1} \sum_{\nu=1}^{2^{L-l+1}} 2k^2 & (3.9)\\ &= kn + \sum_{l=2}^{L+1} 2^{l-1}2^{L-l+1}2k^2 = kn + \sum_{l=2}^{L+1} 2k^2 2^L & (3.10)\\ &= kn + 2k^2 L 2^L & (3.11) \end{aligned}$$

and the total of the $A^{(L+1)}$ matrices:

$$S_{A^{(L+1)}} = \sum_{j=0}^{2^L-1} \text{size}(A_j^{(L+1)}[1]) = \sum_{j=0}^{2^L-1} k\frac{m}{2^L} = 2^L k \frac{m}{2^L} = km. \tag{3.12}$$

Therefore, the final storage is:

$$S_{END} = S_B + S_{A^{(L+1)}} = kn + 2k^2 L 2^L + km = 2k\bar{N} + 2k^2 \frac{\bar{N}}{M} \log \frac{\bar{N}}{M} = O(\bar{N}\log\bar{N}) \tag{3.13}$$

where it has been used that $\bar{N} = m = n$ and $L = \log\frac{\bar{N}}{M}$ and that k and M are independent of $\bar{N}$. So finally, the total necessary amount of memory to store the matrix and the computational cost of a matrix-vector product is $O(\bar{N}\log\bar{N})$.

Now let us consider which is the necessary memory at the worst part of the algorithm, including temporary variables. We will do it without considering the B matrices and we will sum them up in the end. Observing the function **mlacaRec** we can see that for $l = 1$ the worst case is when we are at line 4 where we need to store $A_0^{(1)}[\nu]$ and therefore:

$$S^{(1)} = mk. \tag{3.14}$$

When $l > 1$ there are two possible places where the amount of memory can be greater, so we must consider the maximum of both of them. The first point would be inside line 7 computing **mlacaRec** with a smaller level. In that case we have previously stored the set of matrices $A_0^{(l-1)}[2\nu-1], \ldots, A_{2^{l-2}-1}^{(l-1)}[2\nu-1]$ and the amount of memory which needs the same function for the last level:

$$S_1^{(l)} = S^{(l-1)} + \sum_{j=0}^{2^{l-2}-1} \text{size}(A_j^{(l-1)}[2\nu-1]) = S^{(l-1)} + \sum_{j=0}^{2^{l-2}-1} \text{size}\frac{m}{2^{l-2}}k = S^{(l-1)} + km. \tag{3.15}$$

The other possible point is when we are at line 15 where we have stored $A_0^{(l-1)}[2\nu-1], \ldots, A_{2^{l-2}-1}^{(l-1)}[2\nu-1]$ and $A_0^{(l-1)}[2\nu]$,..., $A_{2^{l-2}-1}^{(l-1)}[2\nu]$ and $A_0^{(l)}[\nu], \ldots, A_{2^{l-1}-1}^{(l)}[\nu]$. Making a similar count it can be proved that:

$$S_2^{(l)} = 3km. \tag{3.16}$$

Summarizing:

$$\begin{cases} S^{(1)} = mk \\ S^{(l)} = \max\left\{S_1^{(l)}, S_2^{(l)}\right\}, \quad l > 1. \end{cases} \tag{3.17}$$

For $l > 2$, $S_1^{(l)}$ is always greater than $S_2^{(l)}$. Therefore:

$$\begin{cases} S^{(2)} = 3mk \\ S^{(l)} = S_1^{(l)} = S^{(l-1)} + km, \quad l > 2 \end{cases} \tag{3.18}$$

and as it grows with l, the worst case is $S^{(L+1)}$ and:

$$S^{(L+1)} = S^{(2)} + (L-1)km = 3km + (L-1)km = k\bar{N}(2 + \log\frac{\bar{N}}{M}). \tag{3.19}$$

And finally, adding the amount of necessary memory for the B matrices we obtain:

$$S_{TOTAL} = kn + 2k^2L2^L + 3km + (L-1)km = 3k\bar{N} + (k + 2\frac{k^2}{M})\bar{N}\log\frac{\bar{N}}{M} = O(\bar{N}\log\bar{N}). \tag{3.20}$$

So it has been proved that at the worst point, the temporary storage of the algorithm is still $O(\bar{N}\log\bar{N})$.

Only remains to obtain the computational cost of the construction of the compressed matrix. To do so we will compute the cost of the function **mlacaRec** for a certain level l and we will obtain then a recursion which finally will give us the total computational cost. For $l = 1$ we only need to do an ACA (line 2) of the matrix Z_{mn_ν} with size $m \times \frac{n}{2^L}$ and k degrees of freedom. Therefore:

$$CC^{(1)} = \left(k^2 + \alpha k\right)\left(m + \frac{n}{2^L}\right) \tag{3.21}$$

where α is the factor which tells us how much a single evaluation of the Green function costs with respect to a single complex number operation.

To compute the cost at level l, $CC^{(l)}$, with $l > 1$ we go line by line. In line 6 and line 7 we call the function **mlacaRec** with level $l-1$ and they cost $CC^{(l-1)}$ each one. Then, the rest of the time is spent in lines 11 and 12 which are done 2^{l-2} times because they are inside a loop. Each one computes an ACA of a matrix $\frac{m}{2^{l-2}} \times 2k$ with k degrees of freedom. So each line costs $k^2(\frac{m}{2^{l-2}} + 2k)$. Putting it together:

$$CC^{(l)} = 2CC^{(l-1)} + 2^{l-2}2k^2\left(\frac{m}{2^{l-2}} + 2k\right) = 2CC^{(l-1)} + 2k^2m + k^32^l \tag{3.22}$$

obtaining finally the recursion:

$$\begin{cases} CC^{(1)} = \left(k^2 + \alpha k\right)\left(m + \frac{n}{2^L}\right) \\ CC^{(l)} = 2CC^{(l-1)} + 2k^2m + k^32^l, \quad l > 1. \end{cases} \tag{3.23}$$

Solving the recursion we can obtain an implicit formula:

$$CC^{(l)} = 2^{l-1}CC^{(1)} + 2k^2m\left(2^{l-1} - 1\right) + k^3\left(l - 1\right)2^l. \tag{3.24}$$

And the total computational cost of function **computeCompressedMatrixMlaca** is obtained by substituting l by $L+1$ in the last equation:

$$\begin{aligned} CC_{TOTAL} &= CC^{(L+1)} && (3.25)\\ &= 2^L CC^{(1)} + 2k^2m\left(2^L - 1\right) + k^3 L 2^{L+1} && (3.26)\\ &= \frac{3k^2+\alpha k}{M}\bar{N}^2 + 2k^3\frac{\bar{N}}{M}\log\frac{\bar{N}}{M} + \left(\alpha k - k^2\right)\bar{N} && (3.27)\\ &= O(\bar{N}^2). && (3.28) \end{aligned}$$

So finally, the computational cost of computing the matrix is $O(\bar{N}^2)$, and the computational cost of a matrix-vector product as well as the storage is $O(\bar{N}\log\bar{N})$.

3.5 SVD Threshold

Similarly to what happens with real numbers, if we have a product of matrices and we want to assure a certain relative error in the resulting matrix, we have to consider the worst case where all the relative errors of each matrix are summed. So, to have a better error than with the direct application of the ACA-SVD (equivalent to the MLACA with $L = 0$), we need to use a threshold $\tau/(L+2)$ instead of τ because the matrix is decomposed into the product of $L+2$ matrices. So actually we run **computeCompressedMatrixMlaca**$(L, \tau/(L+2))$. The influence of that change of threshold in the computational costs and storage is analyzed below.

The only thing which is affected by the threshold is the number of DoF k at each level that now depends on L and therefore on $\bar{N}$. Let us consider that the singular values decay exponentially up to some point:

$$\sigma_t = s_0 10^{-\beta t}, \qquad \beta > 0, t \geq t_0. \tag{3.29}$$

Then we must accomplish

$$\frac{\sigma_{k\left(\frac{\tau}{L+2}\right)}}{\sigma_0} \leq \frac{\tau}{L+2} \tag{3.30}$$

and substituting equation (3.29) for $t = k\left(\frac{\tau}{L+2}\right)$ and isolating k we finally obtain:

$$k\left(\frac{\tau}{L+2}\right) = \left\lceil \frac{1}{\beta}\left[\log_{10}(L+2) + \log_{10}\left(\frac{s_0}{\tau\sigma_0}\right)\right]\right\rceil = O\left(\log_{10}\left(\log\frac{\bar{N}}{M}\right)\right). \tag{3.31}$$

As the coefficients of the order in the last section were proportional to k^2, what we need to add is a factor $\log^2(\log(\bar{N}))$ to the $O(\bar{N}^2)$ computational cost and $O(\bar{N}\log\bar{N})$ storage. In practice the new factor is negligible against the others.

3.6 Global algorithm (Putting it together)

In this section we are going to consider a complete radiation or scattering problem. It is necessary to subdivide the whole set of interactions in near and far interactions, so that we can apply the multilevel MLACA algorithm described above to the far parts. Commonly in the literature, a hierarchical octal tree is used to split the object or the related matrix and this approach can also be applied in our case if desired. Notwithstanding, we propose the utilization of a binary tree also for this hierarchical subdivision. The differences in the results should not be too high in the vast majority of the cases. The only gain comes when the object discretization is very non-uniformly distributed or the object is much larger in one dimension than the rest, because in the proposed decomposition the boxes are adapted to the object at each subdivision.

Once the matrix is divided in far and near interactions, the near parts are directly computed whereas the far parts are computed with the presented MLACA algorithm. Then we have a routine to perform a matrix-vector product which can be introduced in any existing iterative method, like a GMRES for instance. This procedure adds a $\log N$ factor to the aforementioned asymptotical costs, obtaining finally a computational cost growing with $O(N^2 \log N)$ and a storage and matrix-vector product with $O(N \log^2 N)$.

3.7 Numerical Results

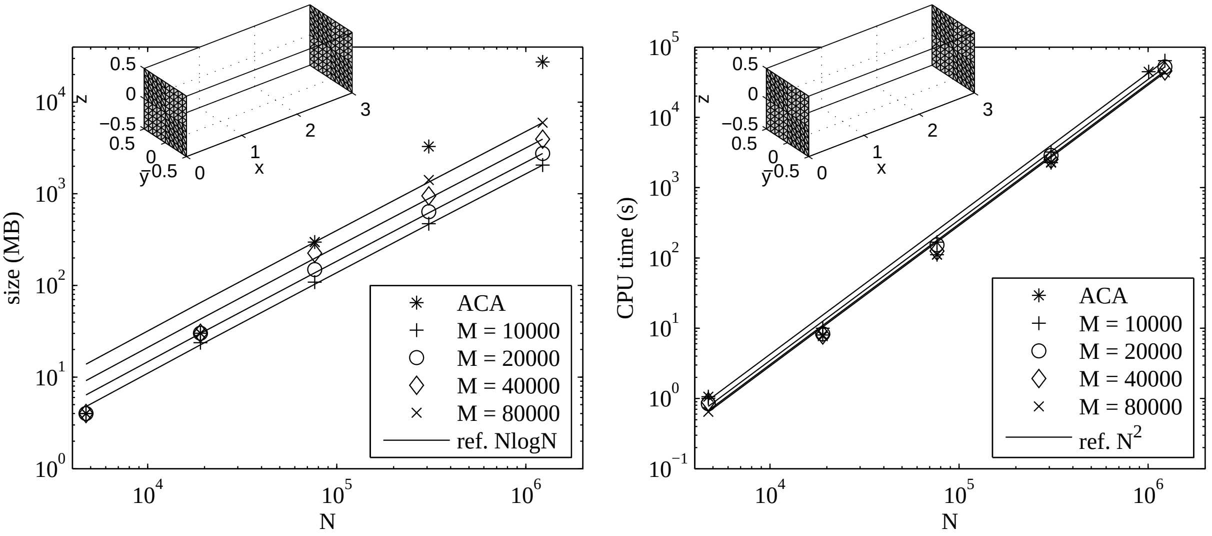
All the numerical experiments reported in this section have been performed on a PC with 64 GB of RAM and a Dual Intel Xeon X5460 processor at 3.16 GHz (8 cores). The computations were done in MATLAB© 7.8.0, always using only one CPU.

We present three different experiments where we consider only the interaction between two sets of samples separated in space. They are the most important in order to evaluate the computational cost and the necessary memory of the algorithm. They could well be a part of a major object or electromagnetic problem. The evaluation for a complete object is left to the next chapter.

3.7.1 Space Separated Blocks

3.7.1.1 Opposite Square Plates

In this experiment a couple of 1m×1m square plates are placed face to face, separated 3m. Varying the frequency, and accordingly the number of unknowns $\bar{N}$ of each square plate to have a fixed discretization size of $\lambda/10$, we perform the computation of the interaction MoM-EFIE matrix between the two plates with the presented MLACA. Fig. 3.8 shows the required memory and the computational time of this simulation when a threshold $\tau = 10^{-3}$ is used. Several computations have been performed varying the size of the finest level in the multilevel decomposition. In the legend, the size of this finest level is shown with a value M which is the approximate number of unknowns at the finest level box. As the discretization size is electrically maintained, it is equivalent to changing the box size. It has also been compared with the ACA method, which can be seen as the MLACA with $L = 0$ levels. Reference lines with the expected asymptotical behavior have been included as well. We can observe how the MLACA perfectly fits the theoretical asymptotic lines when the number of unknowns grows. The compression rate, in comparison with the single level ACA, is much higher, in particular when the number of unknowns grows, as the storage in ACA is asymptotically $\bar{N}^2$ instead of $\bar{N}\log\bar{N}$. The compression is better when the size of the finest level is smaller, i.e. when a larger number of levels is used. However, the computation becomes slightly slower, so the choice of the minimum box size is transformed in a trade-off between storage and computational time. For very large problems, the CPU time of the single level ACA starts to grow faster than $O(\bar{N}^2)$, theoretically $O(\bar{N}^3)$ [77], whereas MLACA is maintained at $O(\bar{N}^2)$. This ACA $\bar{N}^3$ behavior is hardly seen in the figure because this asymptotical behavior is just starting at the greatest represented problem sizes.



(a) Storage

(b) CPU time

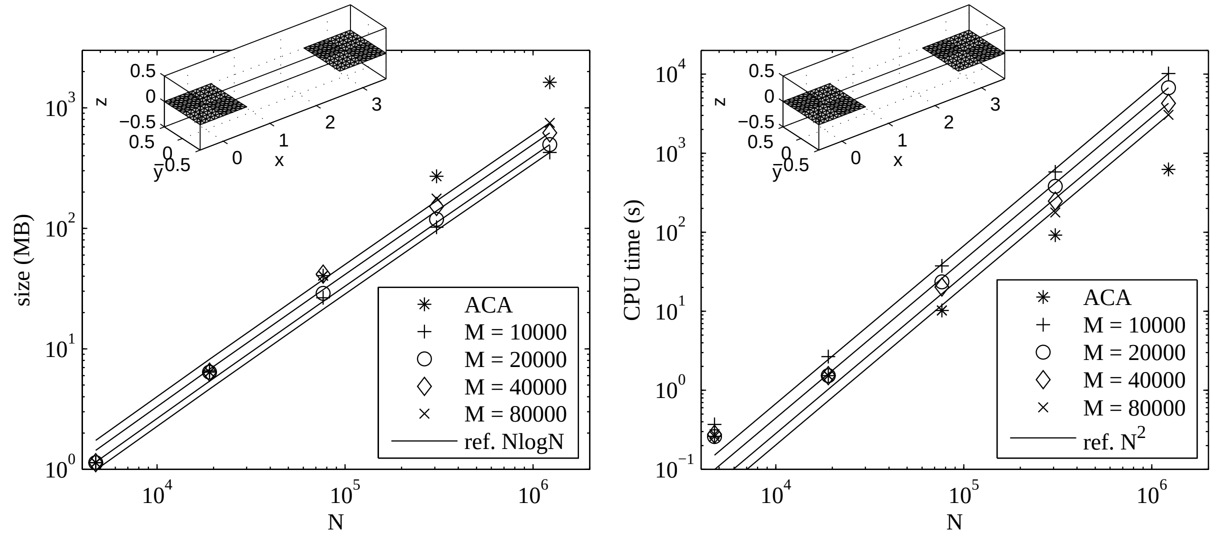
Figure 3.8: Storage requirement and CPU time versus the number of unknowns, with a fixed electrical discretization size, in the interaction between two separated opposite square plates for different finest level sizes in the MLACA. The threshold τ equals 10^{-3}.

3.7.1.2 Parallel Square Plates

The same experiment, except for the position of the plates, is now in order. The two square plates are placed parallel instead of opposite with the centers separated 3m. Fig. 3.9 displays the same kind of results as in the last experiment with the new configuration. Again, the theoretical asymptotic behaviors are perfectly followed and the compression rate is better with the multilevel version than with the single level ACA. In this case, however, ACA is quicker than its multilevel version because the number of DoF, and consequently the ACA complexity, depend on the solid angle between the two groups, which in this case is quasi-1D. Nevertheless, even in this biased case, the expected complexity for very large blocks of ACA is $O(\bar{N}^2)$ as in the MLACA, but with a poorer storage complexity $O(\bar{N}^{3/2})$.

3.7.1.3 Spheres

In this case, the objects are replaced by spheres with radius 0.5m whose centers are separated 4m. This is a case in between the two studied ones, as it definitely has 2D against 2D as in the opposite plates but it also has depth, similarly to the parallel plates. Fig. 3.10 shows the results for

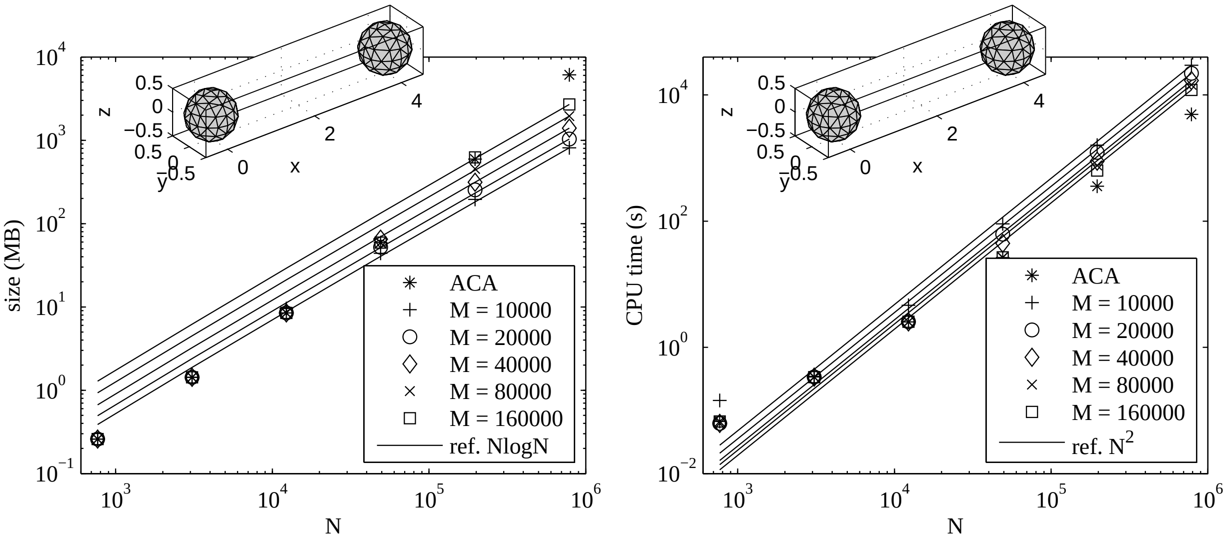


(a) Storage

(b) CPU time

Figure 3.9: Storage requirement and CPU time versus the number of unknowns, with a fixed electrical discretization size, in the interaction between two separated parallel square plates for different finest level sizes in the MLACA. The threshold τ equals 10^{-3}.

this experiment. We can conclude the same as in the parallel plates although in this case the ACA is already behaving $O(\bar{N}^2)$ and it is expected to behave $O(\bar{N}^3)$ for larger objects. The MLACA keeps following the expected asymptotical behaviors.



(a) Storage

(b) CPU time

Figure 3.10: Storage requirement and CPU time versus the number of unknowns, with a fixed electrical discretization size, in the interaction between two separated spheres for different finest level sizes in the MLACA. The threshold τ equals 10^{-3}.

3.8 Conclusion

The MLACA has been presented as a novel multilevel version of the well-known Adaptive Cross Approximation (ACA) algorithm, for the compression of matrices appearing in the Method of Moments discretization of radiation and scattering electromagnetic problems.

Equally to ACA, MLACA applies when the source and field domains corresponding to the impedance matrix block being compressed are well-separated in space, which needs a previous hierarchical domain decomposition of the object under study and a different treatment of compressible far field blocks from non-compressible near field blocks.

It has been proven, both theoretically and numerically, that

1. The storage and the matrix-vector product (or iteration) time scale asymptotically as $O(N \log^2 N)$ and the matrix compression time scales as $O(N^2 \log N)$.

2. That the accuracy is easily controllable varying a threshold τ.

In the presented results, the gain with respect to ACA is not very impressive. However, in the next chapter they will further improve thanks to a modification on the tree decomposition of far-interaction boxes.

Chapter 4

Modified Multilevel Adaptive Cross Approximation

Although we could well have included this chapter inside the previous one, we have preferred to separate it as a modification of the MLACA in its original version. The main reason is twofold: to first focus the attention of the reader on the kernel of the method itself, making it clearer; and to highlight the importance of the present modification to have a real improvement in efficiency and compression with respect to the ACA method already for moderately large objects and not only asymptotically for huge objects. The idea is to decompose each two far boxes with a tree based on the solid angle of one block against the other rather than on the size of the sub-boxes at each level.

4.1 Tree subdivision based on the solid angle

The number of Degrees of Freedom (DoF) of the interaction between two well-separated blocks of samples depends essentially on the solid angle that one block occupies seen from the center of the other, rather than only on the electrical size of the blocks. In a step of the MLACA, one of the blocks is subdivided so that the interaction of the smaller sub-blocks with the other block has a reduced number of DoF in comparison with the parent box. In other words, the DoF are split as best as possible between the new sub-blocks.

Consequently, a block subdivision particularized for every block interaction, depending on both source and observation boxes together, based on the solid angle seen from one box to the other, should give much better numerical results, as will be shown in Section 4.2. Fig. 4.1 shows a pair of example tree subdivisions, a coplanar case and a perpendicular case. A formalization of this procedure is now in order.

Let us introduce the boxes of samples "Box$_1$" and "Box$_2$" with centers $\mathbf{c}_1$ and $\mathbf{c}_2$ and with samples at points $\mathbf{v}_1^i$ and $\mathbf{v}_2^i$, respectively. We want to deal with the tree subdivision of "Box$_2$" when interacting with "Box$_1$". The reverse case is completely analogous. Overall, we need to place spherical coordinates (see Fig. 4.2(a)) centered at the center of "Box$_1$", $\mathbf{c}_1$. The orientation must be so that the other block, Box$_2$ is centered at the x axis, or equivalently at $\phi = 0$ and $\theta = 0$ (see Fig. 4.2(b)). This is to avoid deformations of the object and an unbalance between ϕ and θ which would appear close to the poles, when seen in spherical coordinates. To do this, the following space transformations are necessary.

First a translation with vector $\mathbf{c}_1$ to place the center of coordinates at the center of Box$_1$:

$$\mathbf{r}_{\mathrm{t}} = \mathbf{r} - \mathbf{c}_1. \tag{4.1}$$

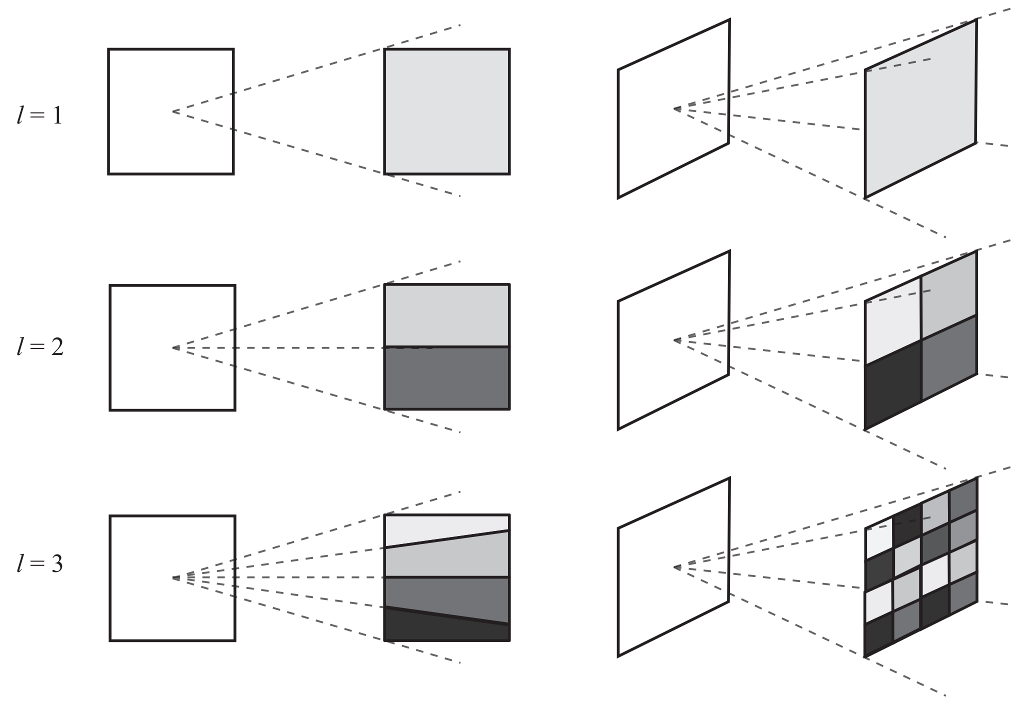
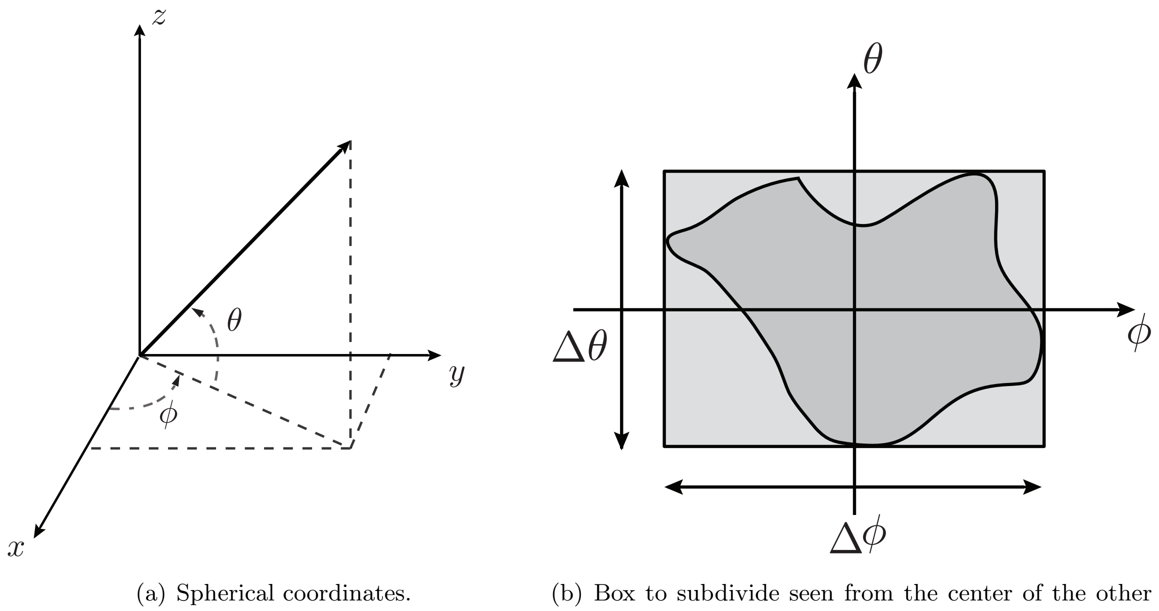


Figure 4.1: Representation of the subdivision of the blocks of samples based on the solid angle seen from the other block. Coplanar on the left and perpendicular on the right.



(a) Spherical coordinates.

(b) Box to subdivide seen from the center of the other box.

Figure 4.2: Representation of the spherical coordinates on the left and what we see of one block of samples from the center of the other in spherical coordinates centered at the center of the box.

Afterwards a rotation with axis z and angle θ_1 to put Box$_2$ centered in the plane $y = 0$:

$$\theta_1 = \tan^{-1}\left(\frac{((\mathbf{c}_2)_\mathrm{t})_y}{((\mathbf{c}_2)_\mathrm{t})_x}\right) \tag{4.2}$$

$$\mathbf{r}_{\text{rot1}} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{r}_{\text{t}}. \tag{4.3}$$

Finally, another rotation with axis y and angle θ_2 is performed to definitively center Box$_2$ at the x axis:

$$\theta_2 = \tan^{-1}\left(\frac{((\mathbf{c}_2)_{\text{rot1}})_z}{((\mathbf{c}_2)_{\text{rot1}})_x \cos(\theta_1) + ((\mathbf{c}_2)_{\text{rot1}})_y \sin(\theta_1)}\right) \tag{4.4}$$

$$\mathbf{r}_{\text{rot2}} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_1) \\ 0 & 1 & 0 \\ -\sin(\theta_1) & 0 & \cos(\theta_1) \end{bmatrix} \cdot \mathbf{r}_{\text{rot1}}. \tag{4.5}$$

Once the transformation have been performed, it only remains to express the obtained Box$_2$ points $(\mathbf{v}_2^i)_{\text{rot2}}$ into the angular spherical coordinates ϕ_i and θ_i

$$\begin{aligned} \phi_i &= \tan^{-1}\left(\frac{((\mathbf{v}_2^i)_{\text{rot2}})_y}{((\mathbf{v}_2^i)_{\text{rot2}})_x}\right) \\ \theta_i &= \tan^{-1}\left(\frac{\sqrt{((\mathbf{v}_2^i)_{\text{rot2}})_x^2 + ((\mathbf{v}_2^i)_{\text{rot2}})_y^2}}{((\mathbf{v}_2^i)_{\text{rot2}})_z}\right). \end{aligned} \tag{4.6}$$

The block Box$_2$ is then enclosed in a rectangular box with sides $\Delta\phi$ and $\Delta\theta$ defined as

$$\begin{aligned} \Delta\phi &= \max_i(\phi_i) - \min_i(\phi_i) \\ \Delta\theta &= \max_i(\theta_i) - \min_i(\theta_i). \end{aligned} \tag{4.7}$$

At each level, we split the block in pieces which have half the solid angle of the parent at the previous level. The empirical criterion we have chosen for the definition of solid angle (which is not the regular one in the mathematic community) is

$$\text{Solid Angle} = \max(\Delta\phi, \Delta\theta) \cdot \left[\min\left(\frac{((\mathbf{v}_2^i)_{\text{rot2}})_x}{\lambda}\right)\right]^{0.6}. \tag{4.8}$$

4.2 Numerical results

All the numerical experiments reported in this section have been performed on a PC with 64 GB of RAM and a Dual Intel Xeon X5460 processor at 3.16 GHz (8 cores). The computations were done in MATLAB© 7.8.0, always using only one CPU.

We present two different experiments where we consider only the interaction between two sets of samples separated in space (Section 4.2.1). They are the most important in order to evaluate the computational cost and the necessary memory of the algorithm. They could well be a part of a major object or electromagnetic problem. Then, three complete object examples are presented (Section 4.2.2).

In all the presented numerical experiments, when reference is made to a relative error in a

Table 4.1: Storage requirement and CPU time in the interaction between two separated parallel square plates with $\bar{N} = 2428200$ samples per plate for different number of levels in the Modified MLACA. The threshold τ equals 10^{-3}.

	$L = 0$ (ACA)	$L = 1$	$L = 2$	L=3
Time	3h 24m	3h 8m	3h 16m	3h 42m
Size (GB)	16.47	10.27	6.79	4.9

computed (vector) parameter X, this error has been calculated according to

$$\frac{\|X - X_{ref}\|}{\|X_{ref}\|}, \tag{4.9}$$

where $\|\cdot\|$ is the 2-norm and X_{ref} is an independently created reference solution. The proper reference solution is usually hard to obtain when dealing with extremely large-scale problems due to the limitations in the available resources. Therefore, the obtained error is usually an estimated error which has only into account the error introduced by the accelerated solver rather than the actual error, which would include the effect of the finite discretization among others.

4.2.1 Space Separated Blocks

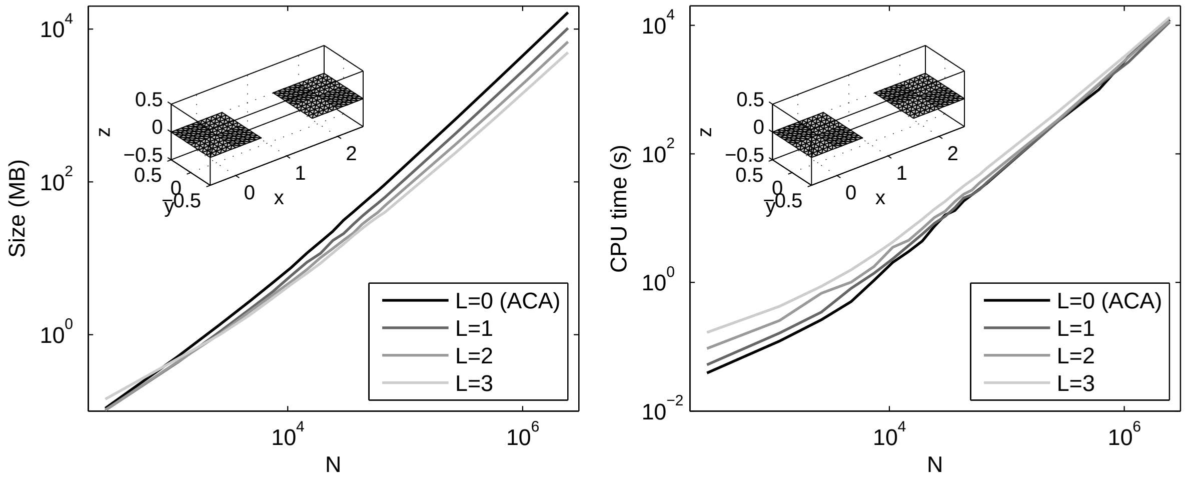
4.2.1.1 Parallel Square Plates

In this experiment, a couple of 1m×1m square plates are placed in the same plane, separated 2m. Varying the frequency, and accordingly the number of unknowns $\bar{N}$ of each square plate to have a fixed discretization of $\lambda/10$, we perform the computation of the interaction MoM-EFIE matrix between the two plates with the presented modified MLACA, for different number of used levels.

Fig. 4.3 shows the required memory and the computational time of this simulation when a threshold $\tau = 10^{-3}$ is used. Clearly, the memory requirement decreases significantly with the number of levels, mainly when the electrical size of the plates grows.

With respect to the CPU time, the optimum for small electrical sizes is the single level ACA ($L = 0$). Nonetheless, there is a certain point, approximately for $\bar{N} = 29800$ or a plate side of 10λ, where the time for $L = 1$ levels starts to be equal or even better when the plates size increase. The same happens for $L = 2$ at a further point, approximately for $\bar{N} = 270000$ or a plate side of 30λ. Exactly the same will happen for $L = 3$, not represented in the figure as it is beyond the maximum computed value, and so on and so forth. Therefore, an appropriate utilization to have a good trade-off between necessary memory and CPU time is the following: using the maximum number of levels but assuring that the CPU time is at least equal to the CPU time for $L = 0$. The number of levels to use is then chosen depending on the solid angles between the two blocks of samples. For larger solid angles, larger number of levels.

To stress the gain that we are actually obtaining, we have extracted the results for the largest computed parallel square plates ($\bar{N} = 2428200$ or square side 90λ) and included them in Table 4.1. A good trade-off in this case would be to use $L = 2$, whose computation takes 8 minutes less than with the single level ACA with 41% of the memory (from 16.47GB to 6.79GB).



(a) Storage

(b) CPU time

Figure 4.3: Storage requirement and CPU time versus the number of unknowns, with a fixed electrical discretization size, in the interaction between two separated parallel square plates for different number of levels in the Modified MLACA. The threshold τ equals 10^{-3}.

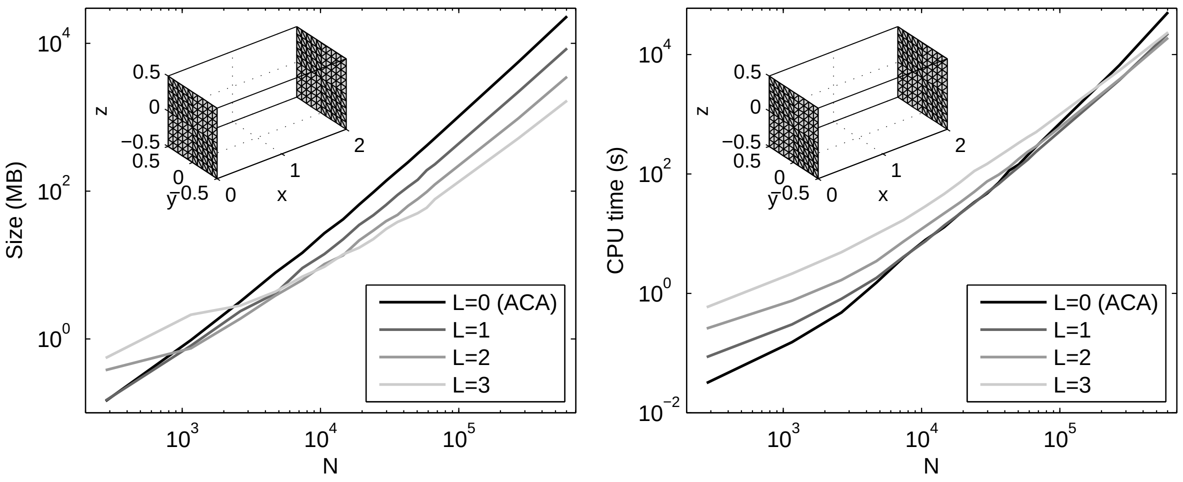
4.2.1.2 Opposite Square Plates

As will be shown in the present Section, the gain is much greater when the two square plates are placed face to face instead of in the same plane. The two 1m×1m square plates are oppositely placed and separated 2m. Again, the computation of the MoM-EFIE compressed interaction matrix is computed for different frequencies and number of MLACA levels. The used threshold is $\tau = 10^{-3}$ and the discretized elements have an average length of $\lambda/10$.

Fig. 4.4 shows the required memory and the computational time of this simulation. For electrically large blocks, the memory requirement is much smaller when the number of levels grows. For small blocks, the size with more levels is larger because the threshold is reduced proportionally to the number of levels (see Chapter 3).

As for the CPU time, the single level ACA ($L = 0$) is faster until blocks with about $\bar{N} = 7400$ samples or a side size of 5λ. From this point, $L = 1$ becomes the reference until blocks with $\bar{N} = 67200$ or a side size of 15λ. Afterwards, $L = 2$ is the fastest and it follows successively for larger electrical sizes and number of MLACA levels. In this case, the better behavior of the multilevel, even in terms of CPU time is much more evident.

Similarly to the parallel square plates case, we have included the results for the electrically largest simulated plates into Table 4.2. It corresponds to square plates with $\bar{N} = 606600$ samples or a side size of 45λ. If we use $L = 2$ levels, it only takes a 37% of the time with respect to the ACA and it only needs a 15% of the memory. Furthermore, with $L = 3$, although it takes a 46% of the ACA time which is a bit slower than with $L = 2$, it only needs a 7.2% of the memory. Therefore, the number of MLACA levels to use will depend on our resources. If we are more restricted by memory, we can use a larger number of levels. If CPU time is our main constraint there is always an optimum number of levels in terms of time, with still an excellent compression.



(a) Storage

(b) CPU time

Figure 4.4: Storage requirement and CPU time versus the number of unknowns, with a fixed electrical discretization size, in the interaction between two separated opposite square plates for different number of levels in the Modified MLACA. The threshold τ equals 10^{-3}.

Table 4.2: Storage requirement and CPU time in the interaction between two separated opposite square plates with $\bar{N} = 606600$ samples per plate for different number of levels in the Modified MLACA. The threshold τ equals 10^{-3}.

	$L = 0$ (ACA)	$L = 1$	$L = 2$	$L = 3$
Time	14h 15m	6h 16m	5h 19m	6h 33m
Size (GB)	23.31	8.57	3.54	1.68

4.2.2 Complete Problem Solutions

4.2.2.1 Large PEC Square Plate

The first object under study is a PEC square plate with a length side of 59.7λ. The threshold τ for the SVD re-compressions is 10^{-3} and the size of the finest level in the subdivision of the object to split in far and near interactions is 0.5λ. The square plate has $N = 1385840$ RWG unknowns with an average edge length of the discretized elements of $\lambda/10$. In this case we have used the EFIE formulation and as iterative method we have used a GMRES with a residual threshold to stop of 10^{-5}. Note that in this case all the far interactions are of the kind where there are square plates in the same plane, which as proved before, is the case where the MLACA needs electrically larger blocks to be worth using.

The reference current vector to compute the error has been obtained solving the problem with the MLACA with a threshold $\tau = 10^{-4}$. We tried to obtain it with the ACA to be fairer but it was then beyond the memory limit with the correspondent slowing down of the simulation when the swapping to disk started, making it unfeasible in a reasonable time.

Table 4.3 shows the performance of this simulation with three different methods, the single level ACA, the MLACA with the tree subdivision introduced in Chapter 3 and the MLACA with the modified tree subdivision introduce in the present Chapter. With the modified MLACA method

Table 4.3: MLACA performance on currents computation of 59.7λ side PEC square plate utilizing EFIE formulation. $N = 1385840$ unknowns and threshold $\tau = 10^{-3}$.

	ACA	MLACA (Old)	Modified MLACA
Build time	5h 57m	6h 27m	6h 11m
Size Near (GB)	9.30	9.30	9.30
Size Far (GB)	38.22	34.71	32.75
GMRES threshold	10^{-5}	10^{-5}	10^{-5}
# iterations	141	142	142
Iterations time	4h 44m	4h 35m	4h 30m
Total Time	10h 41m	11h 2m	10h 41m
Total Size (GB)	47.52	44.01	42.05
Error in $\mathbf{J}$(%)	0.29	0.18	0.19

we obtain a further compression of $5.5GB$ with respect to the ACA with the same total solution time. It represents a reduction of an 11% of the total necessary memory. The estimated error is also lower for the MLACA than for the ACA, and in particular is of the same order as the selected threshold τ. With respect to the old MLACA, the gain in memory is lower and the CPU time is higher, justifying the importance of the tree subdivision in terms of the solid angle presented in this Chapter.

Something worth mentioning, which will appear in all the simulation from here on, is that the iterations time is not improved in the same amount as the used memory. This is because the GMRES has an additional cost apart from a matrix-vector product which increases with each iteration.

Fig. 4.5 shows the bi-static RCS of this simulation for the different methods. As expected from the errors in the surface currents, an excellent agreement can be observed for the RCS.

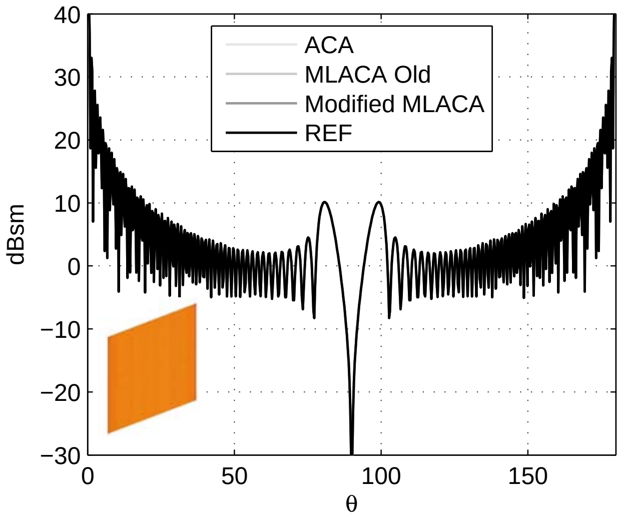


Figure 4.5: Bistatic RCS of a 59.7λ side PEC square plate with $N = 1385840$ unknowns.

Note that we only use multiple levels for the far interaction of electrically large blocks. Therefore,

Table 4.4: MLACA performance on currents computation of 26.4λ diameter PEC sphere utilizing CFIE formulation. $N = 786432$ unknowns and threshold $\tau = 10^{-1}$.

	ACA	MLACA (Old)	Modified MLACA
Build time	4h 17m	5h 18m	4h 40m
Size Near (GB)	7.19	7.19	7.19
Size Far (GB)	29.12	25.72	22.35
GMRES threshold	10^{-5}	10^{-5}	10^{-5}
# iterations	77	77	77
Iterations time	1h 19m	1h 18m	1h 17m
Total Time	5h 36m	6h 36m	5h 57m
Total Size (GB)	36.31	32.91	29.54
Error in $\mathbf{J}$(%)	7.97	7.63	7.80

in this case only a few interactions are improved, which do not represent a considerable percentage of the total matrix size. However, this compression will increase considerably when the object is larger. This behavior will be better analyzed for the PEC missile in Section 4.2.2.3.

4.2.2.2 Large PEC Sphere

In order to avoid conditioning problems, which are not the issue of this thesis, and having a good convergence, in the next two examples we will only analyze closed geometries using the CFIE formulation. A drawback of this approach is that the matrix is not symmetric. Therefore, all the interactions need to be computed.

The next object under study is a PEC sphere with a diameter of 26.4λ with $N = 786432$ unknowns. The threshold τ for the SVD re-compressions is 10^{-1}, the size of the finest level in the subdivision of the object to split in far and near interactions is again 0.5λ and the threshold for the GMRES residual to stop is 10^{-5}. The discretization has an average length of the elements of $\lambda/10$.

The reference current vector to compute the error has been obtained solving the problem with the ACA with a threshold $\tau = 10^{-3}$. Therefore, in any case it should benefit the ACA simulation, but as expected, the error will be slightly smaller for the MLACA.

Table 4.4 shows the results of this simulation. In this case, the MLACA has only been used at the first level (largest blocks) of the object subdivision where there are far interactions, even though it should not be applied yet if we want to optimize the CPU time. The memory requirement is 6.77GB lower for the modified MLACA than for the ACA with CPU time 21 minutes higher. It represents a gain in memory of a 18.6% with an increasing of 6.6% of CPU time. When the object becomes electrically larger, the gain in memory will be much more important together with a time improvement. The estimated relative error is slightly smaller for the MLACA and of the same order as the used threshold τ. With respect to the old version of the MLACA, it doubles the compression while reducing also the CPU time in 39 minutes.

Fig. 4.6 shows the bi-static RCS of the sphere computed from the surface currents obtained with the different methods. We have also included the reference available for the sphere from the Mie series development. There are slight differences which are normal considering the errors in the surface currents. These errors can be easily reduced by decreasing the threshold τ, which has been set very high for this case ($\tau = 10^{-1}$).

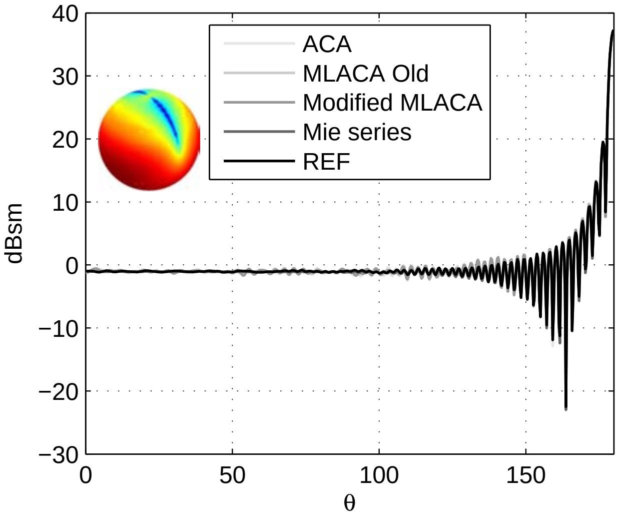


Figure 4.6: Bistatic RCS of a PEC sphere with diameter 26.4λ and $N = 786432$ unknowns.

4.2.2.3 PEC Missile

Finally, we analyze a PEC missile presented in [79] which has a length of 75λ. The shape and dimensions of the missile can be extracted from the referenced paper. The missile has been discretized with $N = 1222419$ unknowns and an average length of the elements of $\lambda/10$ as in the previous cases. As in the sphere, we use the CFIE formulation with a threshold $\tau = 10^{-1}$ and the size of the finest level in the subdivision of the object to split in far and near interactions of 0.5λ. However, the residual tolerance to stop the GMRES algorithm is set to 10^{-3} due to the worse convergence rate.

We were not able to obtain a good reference solution as we are very close to the RAM memory constraint of our computer. However, a comparison of the bi-static RCS with our results (see Fig. 4.7) and the one included in [79], calculated with a very different approach, shows a good agreement.

Table 4.5 shows the results for this simulation. In this case, there are two columns for the modified MLACA. We use modified MLACA 1 when multiple levels are only applied to the block interactions at the subdivision levels 2 and 3 of the object which correspond with the largest far-interaction blocks. In the modified MLACA 2, also the forth level far-interactions are calculated with multiple levels. With the inclusion of the forth level, the build time is a bit higher but the gain in compression is considerable. In this case, the number of iterations is higher for the ACA than the others. Often the one which takes longer to converge is the one with a higher relative error in the matrix elements but this cannot be concluded in the absence of a good reference. In any case, the total time of the modified MLACA is smaller than the ACA and even the build time, without considering the iterations is very close. The gain in memory is $10.4GB$ with respect to the ACA which corresponds with a 18% of the total.

To evaluate the contribution of the MLACA to the different far-interaction blocks, we have

Table 4.5: MLACA performance on currents computation of 75λ PEC missile utilizing CFIE formulation. $N = 1222419$ unknowns and threshold $\tau = 10^{-1}$.

	ACA	MLACA (Old)	Modified MLACA 1	Modified MLACA 2
Build time	9h 45m	10h 11m	9h 52m	10h 9m
Size Near (GB)	8.55	8.55	8.55	8.55
Size Far (GB)	49.59	40.12	43.78	39.18
GMRES threshold	10^{-3}	10^{-3}	10^{-3}	10^{-3}
# iterations	235	195	197	195
Iterations time	9h 40m	7h 25m	7h 28m	7h 23m
Total Time	19h 25m	17h 35m	17h 20m	17h 32m
Total Size (GB)	58.14	48.67	52.33	47.73

Table 4.6: MLACA performance on the matrix computation of 75λ PEC missile utilizing CFIE formulation. The contribution to sub-blocks of the original object of different sizes is contemplated. $N = 1222419$ unknowns and threshold $\tau = 10^{-1}$.

	level 2		level 3		level 4	
	ACA	MLACA	ACA	MLACA	ACA	MLACA
# interactions	6	6	48	48	6892	6892
Size (GB)	5.50	1.36	3.05	1.38	22.11	17.50
Time	53m	56m	20m	24m	2h 1m	2h 18m

included Table 4.6. When the MLACA is applied to the relatively electrically small blocks at level 4 of the object subdivision we save a 21% of the memory but with an increase of a 14% of CPU time. In the next larger blocks level 3, the gain in memory is already a 55% with only 4 minutes more of time. And finally, the gain at the largest level 2, with only 6 far-interactions, is a 75% with just 3 extra minutes. Therefore, if the object is electrically larger, the compression of the larger block interactions will be much better, and also the contribution of the largest blocks interactions will represent a good proportion of the total memory.

Fig. 4.7 shows the bi-static RCS computed with the different methods. There are some small differences which are normal considering the high threshold we are using. The main difference is in the single level ACA.

4.3 Conclusion

A modified tree decomposition based on the solid angle has been presented as an improvement to the Multilevel Adaptive Cross Approximation method. Several numerical experiments have corroborated the gain in terms of memory requirement and CPU time with respect to the single level version ACA, already for object of around one million of unknowns.

It has been proved that the MLACA will be much superior for problems that are one or two orders of magnitude larger. Therefore, the author expects that the MLACA will find its place in the framework of supercomputers and parallelization, which are becoming a reality nowadays even at very low prices.

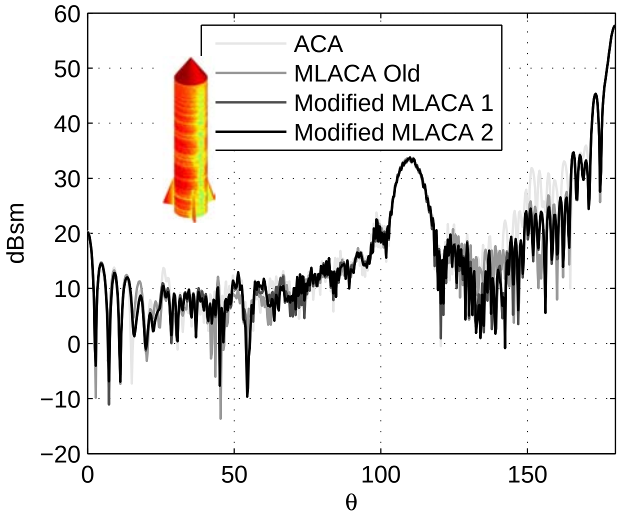


Figure 4.7: Bistatic RCS of a 75λ length PEC missile with $N = 1222419$ unknowns.

Singular Integrals

Chapter 5

Fast and accurate computation of hyper-singular integrals in Galerkin surface integral equation formulations via the direct evaluation method

In this chapter, hyper-singular 4-D integrals, arising in the Galerkin discretization of surface integral equation formulations, are computed by means of the direct evaluation method. The proposed scheme extends the basic idea of the singularity cancellation methods, usually employed for the regularization of the singular integral kernel, by utilizing a series of coordinate transformations combined with a re-ordering of the integrations. The overall algebraic manipulation results in smooth 2-D integrals – in the edge adjacent case – or smooth 3-D integrals – in the vertex adjacent case – that can be easily evaluated via standard quadrature rules. Finally, the reduction of the dimensionality of the original integrals together with the smooth behavior of the associated integrands lead to unmatched accuracy and efficiency.

5.1 Introduction

Surface integral equation formulations have proved to be one of the most powerful methods for the solution of various electromagnetic (EM) antenna and scattering problems [1, 2, 3]. Galerkin variants of the method of moments (MoM) [4] are most often utilized for the numerical solution of these electromagnetic surface integral equations, calling for the calculation of four-dimensional (or double) integrals with singular kernels. Typically, the singular integrals arising in surface field integral equations can be categorized according to the behavior of the kernel into weakly singular ($1/R$), strongly singular ($1/R^2$) and hyper-singular ($1/R^3$). In EM community, though, the definition might be quite different, i.e., kernels with $1/R^2$ and $1/R^3$ behaviors are called hyper-singular and super hyper-singular, respectively, as discussed in detail by Tong and Chew [80, 81].

To be more specific, weakly singular integrals appear in the electric field integral equation (EFIE) formulations, when divergence-conforming basis and testing functions, like the popular Rao-Wilton-Glisson (RWG) [10], are incorporated in the Galerkin MoM. Due to the non smooth behavior of the weakly singular integral kernels, classical numerical quadrature rules fail to meet the requirements for high precision results, and more sophisticated techniques are needed to tackle this problem. Generally, various regularization methods for the computation of 4-D weakly singular integrals have appeared in the literature over the last decades and interested readers could consult [18] for a more detailed history of relevant previous work.

On the other hand, 4-D hyper-singular integrals (following here and for the rest of the chapter the convention used in EM community) arise in the numerical solution of magnetic field integral equation (MFIE) and combined field integral equation (CFIE) formulations, since the latter are combination of EFIE and MFIE formulations. Over the last years, numerous techniques have been presented for the accurate and efficient evaluation of those multidimensional hyper-singular integrals, which can be roughly categorized into two main families: singularity cancellation methods [19, 20, 21, 22, 23] and singularity subtraction methods [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. The logical expectation is that both methods should lead to superior accuracy compared with direct implementation of quadrature rules. This is not necessarily the case, as demonstrated in this chapter, especially when the quest for machine precision is combined with the need of improved efficiency.

In this chapter, the direct evaluation method tailored for the hyper-singular integrals arising in Galerkin surface integral equation formulations is developed. The proposed method utilizes a series of coordinate transformations together with a re-ordering of the integrations in order to reduce the dimensionality of the 4-D hyper-singular integrals into 2-D smooth integrals – in the edge adjacent case – or smooth 3-D integrals – in the vertex adjacent case – that can be easily computed via generalized Cartesian product rules based on standard Gauss quadratures, readily available in the literature. The direct evaluation method was originally introduced by Gray et al. [82], [83] for the evaluation of super hyper-singular Galerkin surface integrals arising in static problems and recently it was extended by Polimeridis et al. for the case of the weakly singular integrals in Galerkin MoM problems over coincident [18] and over edge adjacent and vertex adjacent triangle elements [84]. The extension to the dynamic hyper-singular integrals presented in this chapter was developed during a short stay in the Laboratory of Electromagnetics and Acoustics (LEMA) department at the École Polytechnique Fédérale de Lausanne (EPFL) in collaboration with Polimeridis et al.

In the following Section, the singular integrals arising in Galerkin surface integral equation formulations are reported. In Sections 5.3 and 5.4, the direct evaluation method for the hyper-singular integrals over edge adjacent and vertex adjacent triangle elements, respectively, is presented in detail. The final formulas in the last part of Sections 5.3 and 5.4 together with the functions provided in Appendix D form the complete setup for a convenient software implementation. Finally, in Section 5.5, the proposed method is applied in some typical test case problems and the results are compared with some of the most powerful methods available in the literature. Furthermore, the sensitivity of the method performance against a tolerable deformation in the triangle elements is presented.

5.2 Singular integrals

The evaluation of the Galerkin MoM matrix elements in EFIE, MFIE or CFIE formulations, using RWG basis functions $\mathbf{f}$ and either RWG or $\hat{\mathbf{n}}\times$RWG testing functions $\mathbf{g}$, calls for the calculation of the following singular integrals [29]:

$$I_1 := \int_{S_m} \mathbf{g}(\mathbf{r}) \cdot \nabla \int_{S_n} G(\mathbf{r},\mathbf{r}') \nabla_s' \cdot \mathbf{f}(\mathbf{r}') \, dS' dS \tag{5.1}$$

$$I_2 := \int_{S_m} \mathbf{g}(\mathbf{r}) \cdot \int_{S_n} G(\mathbf{r},\mathbf{r}') \, \mathbf{f}(\mathbf{r}') \, dS' dS \tag{5.2}$$

$$I_3 := \int_{S_m} \mathbf{g}(\mathbf{r}) \cdot \int_{S_n} \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{f}(\mathbf{r}') \, dS' dS \tag{5.3}$$

which incorporate the free-space Green's function

$$G(\mathbf{r},\mathbf{r}') = G(R) = \frac{e^{-jkR}}{R} \tag{5.4}$$

with $R = |\mathbf{r} - \mathbf{r}'|$ being the distance function and $k = \omega\sqrt{\varepsilon\mu}$ the wavenumber. Here, S_m and S_n correspond to the supports of the associated testing and basis functions.

The integrands of the aforementioned integrals become singular when the integration domains S_m and S_n have common points. Consequently, regular numerical quadrature rules can not converge with a reasonable number of integration points and the need appears for more sophisticated techniques to compute them. The integral I_1 can be transformed, for the basis and testing functions under consideration, to a weakly singular integral [29]. Therefore, both I_1 and I_2 can be included in the cases which were already treated in previous works within the framework of the direct evaluation method [18],[84].

Although the basis and testing functions used herein are defined over triangle pairs, the above integrals can be reduced to integrals over the forming triangles, following a "triangle by triangle" assembly procedure to fill the MoM impedance matrix. Hence, we consider the integrals over the triangles E_P and E_Q instead of the domains S_m and S_n. In this chapter, we focus particularly on the solution of the hyper-singular integral

$$I := \int_{E_P} \mathbf{g}(\mathbf{r}) \cdot \int_{E_Q} \nabla G(\mathbf{r},\mathbf{r}') \times \mathbf{f}(\mathbf{r}')\, dS' dS \tag{5.5}$$

when the triangles E_P and E_Q lie on different planes while either sharing a common edge or only a common vertex. The co-planar case is directly zero because the integrand equals zero.

5.3 Direct evaluation method - Edge Adjacent

5.3.1 Equilateral triangle parameter space

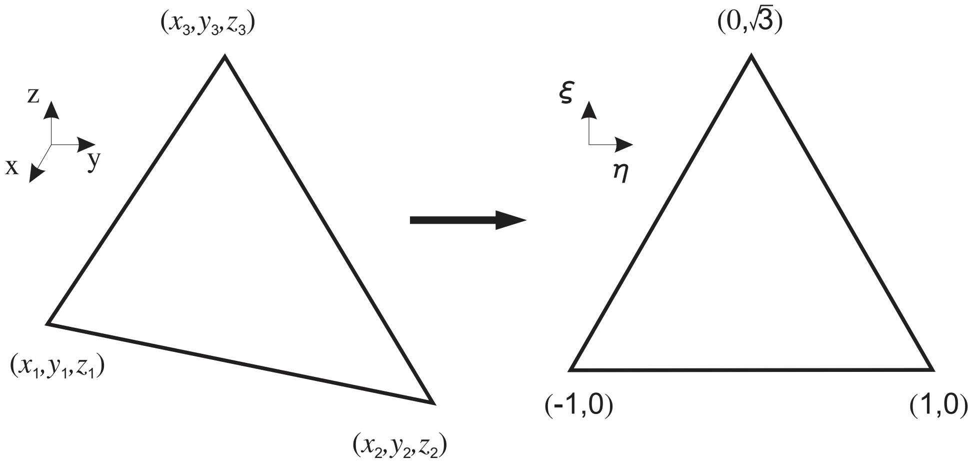


Figure 5.1: Equilateral parameter space transformation (original triangle → master triangle).

The first step of the direct evaluation method is to introduce an appropriate parameter space.

In this manuscript, an equilateral parameter space $\{\eta, \xi\}$, where $-1 \leq \eta \leq 1$, $0 \leq \xi \leq \sqrt{3}(1 - |\eta|)$ (see Fig. 5.1) is employed for each one of the triangles, following the original work by Gray et al. [82], [83]. This specific choice of parameter space is rather convenient for executing the edge adjacent and the vertex adjacent integrations, in which we are interested here, due to its symmetry. The governing transformation matrices are given by

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_2-x_1}{2} & \frac{2x_3-x_1-x_2}{2\sqrt{3}} \\ \frac{y_2-y_1}{2} & \frac{2y_3-y_1-y_2}{2\sqrt{3}} \\ \frac{z_2-z_1}{2} & \frac{2z_3-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = [\mathbf{A}] + [\mathbf{Q}] \begin{bmatrix} \eta \\ \xi \end{bmatrix} \tag{5.6}$$

with the Jacobian being a constant, $J = A/\sqrt{3}$, where A is the area of the original triangle. The hyper-singular integral (5.5) in the new parametric space can be evaluated as

$$I = (J_p J_q) \int_{-1}^{1} d\eta \int_{0}^{\xi(\eta)} d\xi \int_{-1}^{1} d\eta' \int_{0}^{\xi(\eta')} \mathbf{g}(\eta, \xi) \cdot \left(\nabla G(R) \times \mathbf{f}(\eta', \xi')\right) d\xi' \tag{5.7}$$

where $\xi(\eta) = \sqrt{3}(1 - |\eta|)$ and J_p and J_q are the associated Jacobians of the transformation from the original triangles to the equilateral parameter space master triangles. Moreover, R is a function of all six nodal coordinates (for the two original triangles) as well as the four variables (two for the inner integral E_Q and two for the outer integral E_P) of the equilateral triangle parameter space.

Next, we need to orient the elements so that the shared edge is defined by $\xi = 0$ for E_P, and $\xi' = 0$ for E_Q, hence, the singularity is characterized by $\eta + \eta' = \xi = \xi' = 0$ (see Fig. 5.2). The position vectors in the new parametric space are given according to (5.6), taking into account the notation in Fig. 5.2, where $E_P : (\mathbf{r}_{1p}, \mathbf{r}_{2p}, \mathbf{r}_{3p}) \equiv (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ and $E_Q : (\mathbf{r}_{1q}, \mathbf{r}_{2q}, \mathbf{r}_{3q}) \equiv (\mathbf{r}_2, \mathbf{r}_1, \mathbf{r}_4)$,

$$\mathbf{r} = \mathbf{r_P} = \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_2-x_1}{2} & \frac{2x_3-x_1-x_2}{2\sqrt{3}} \\ \frac{y_2-y_1}{2} & \frac{2y_3-y_1-y_2}{2\sqrt{3}} \\ \frac{z_2-z_1}{2} & \frac{2z_3-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = \frac{\mathbf{r}_1 + \mathbf{r}_2}{2} + \boldsymbol{\alpha}_{e1}\eta + \boldsymbol{\alpha}_{e2}\xi \tag{5.8}$$

$$\mathbf{r}' = \mathbf{r_Q} = \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_1-x_2}{2} & \frac{2x_4-x_1-x_2}{2\sqrt{3}} \\ \frac{y_1-y_2}{2} & \frac{2y_4-y_1-y_2}{2\sqrt{3}} \\ \frac{z_1-z_2}{2} & \frac{2z_4-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta' \\ \xi' \end{bmatrix} = \frac{\mathbf{r}_1 + \mathbf{r}_2}{2} - \boldsymbol{\alpha}_{e1}\eta' - \boldsymbol{\alpha}_{e3}\xi' \tag{5.9}$$

where

$$\begin{aligned} \boldsymbol{\alpha}_{e_1} &= \frac{\mathbf{r}_2 - \mathbf{r}_1}{2} \\ \boldsymbol{\alpha}_{e_2} &= \frac{2\mathbf{r}_3 - \mathbf{r}_1 - \mathbf{r}_2}{2\sqrt{3}} \\ \boldsymbol{\alpha}_{e_3} &= -\frac{2\mathbf{r}_4 - \mathbf{r}_1 - \mathbf{r}_2}{2\sqrt{3}} \end{aligned} \tag{5.10}$$

which only depend on the vertices of the triangles. The distance function vector $\mathbf{R}$ can be derived as follows:

$$\mathbf{R} = \mathbf{r_P} - \mathbf{r_Q} = \boldsymbol{\alpha}_{e_1}(\eta + \eta') + \boldsymbol{\alpha}_{e_2}\xi + \boldsymbol{\alpha}_{e_3}\xi'. \tag{5.11}$$

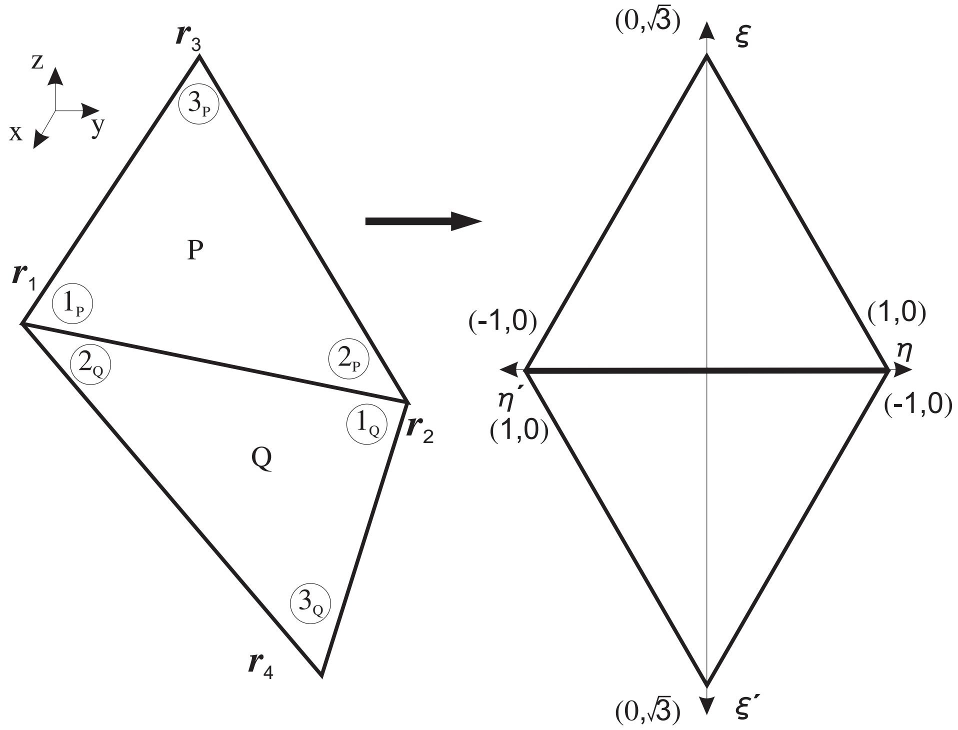


Figure 5.2: Orientation of the triangular elements both in the original and the equilateral triangle parameter space in the edge adjacent case.

5.3.2 First analytical integration - Polar coordinates changes

Thereafter, a polar coordinate transformation within the inner integration (triangle Q) is introduced to cancel the line of singularity defined by $\xi = \xi' = 0$ and $\eta = -\eta'$ (Fig. 5.3(a)),

$$\begin{aligned} \eta' &= \rho \cos(\theta) - \eta \\ \xi' &= \rho \sin(\theta). \end{aligned} \tag{5.12}$$

The Jacobian of the transformation is ρ, which accumulated to the previous variable changes yields

$$J = \frac{A_p A_q}{3} \rho. \tag{5.13}$$

The distance function vector takes the form

$$\mathbf{R} = \boldsymbol{\alpha}_{e_1} \rho \cos(\theta) + \boldsymbol{\alpha}_{e_2} \xi + \boldsymbol{\alpha}_{e_3} \rho \sin(\theta). \tag{5.14}$$

The integration with respect to θ should be split into two pieces, as shown in Fig. 5.3(a),

$$\begin{aligned} I =&\frac{A_pA_q}{3}\int_{-1}^{1} d\eta \int_{0}^{\xi(\eta)} d\xi \\ &\left[\int_{0}^{\Theta_1(\eta)} d\theta \int_{0}^{L(\eta,\theta)} \mathbf{g}(\eta,\xi)\cdot\left(\nabla G\times\mathbf{f}(\eta,\theta,\rho)\right)\rho d\rho\right. \\ &\left.+\int_{\Theta_1(\eta)}^{\pi} d\theta \int_{0}^{L(\eta,\theta)} \mathbf{g}(\eta,\xi)\cdot\left(\nabla G\times\mathbf{f}(\eta,\theta,\rho)\right)\rho d\rho\right] \end{aligned} \tag{5.15}$$

where

$$\begin{aligned} \Theta_1(\eta) &= \frac{\pi}{2}-\tan^{-1}\left(\frac{\eta}{\sqrt{3}}\right) \\ L(\eta,\theta) &= \begin{cases} \dfrac{\sqrt{3}(1+\eta)}{\sin(\theta)+\sqrt{3}\cos(\theta)}, & 0<\theta<\Theta_1 \\ \dfrac{\sqrt{3}(1-\eta)}{\sin(\theta)-\sqrt{3}\cos(\theta)}, & \Theta_1<\theta<\pi. \end{cases} \end{aligned} \tag{5.16}$$

Bearing in mind that the break-point in θ is only a function of η, the integration can be directly re-arranged,

$$\begin{aligned} I =&\frac{A_pA_q}{3}\int_{-1}^{1} d\eta \int_{0}^{\Theta_1(\eta)} d\theta \int_{0}^{\xi(\eta)} d\xi \int_{0}^{L(\eta,\theta)} \mathbf{g}(\eta,\xi)\cdot\left(\nabla G(R)\times\mathbf{f}(\eta,\theta,\rho)\right)\rho d\rho \\ &+\frac{A_pA_q}{3}\int_{-1}^{1} d\eta \int_{\Theta_1(\eta)}^{\pi} d\theta \int_{0}^{\xi(\eta)} d\xi \int_{0}^{L(\eta,\theta)} \mathbf{g}(\eta,\xi)\cdot\left(\nabla G(R)\times\mathbf{f}(\eta,\theta,\rho)\right)\rho d\rho. \end{aligned} \tag{5.17}$$

As the singularity now occurs when $\rho=\xi=0$, we proceed in accordance with [82] by introducing a second polar coordinate transformation,

$$\begin{aligned} \rho &= \Lambda\cos(\Psi) \\ \xi &= \Lambda\sin(\Psi) \end{aligned} \tag{5.18}$$

as shown in Fig. 5.3(b). The distance function vector in the new parametric system takes the form

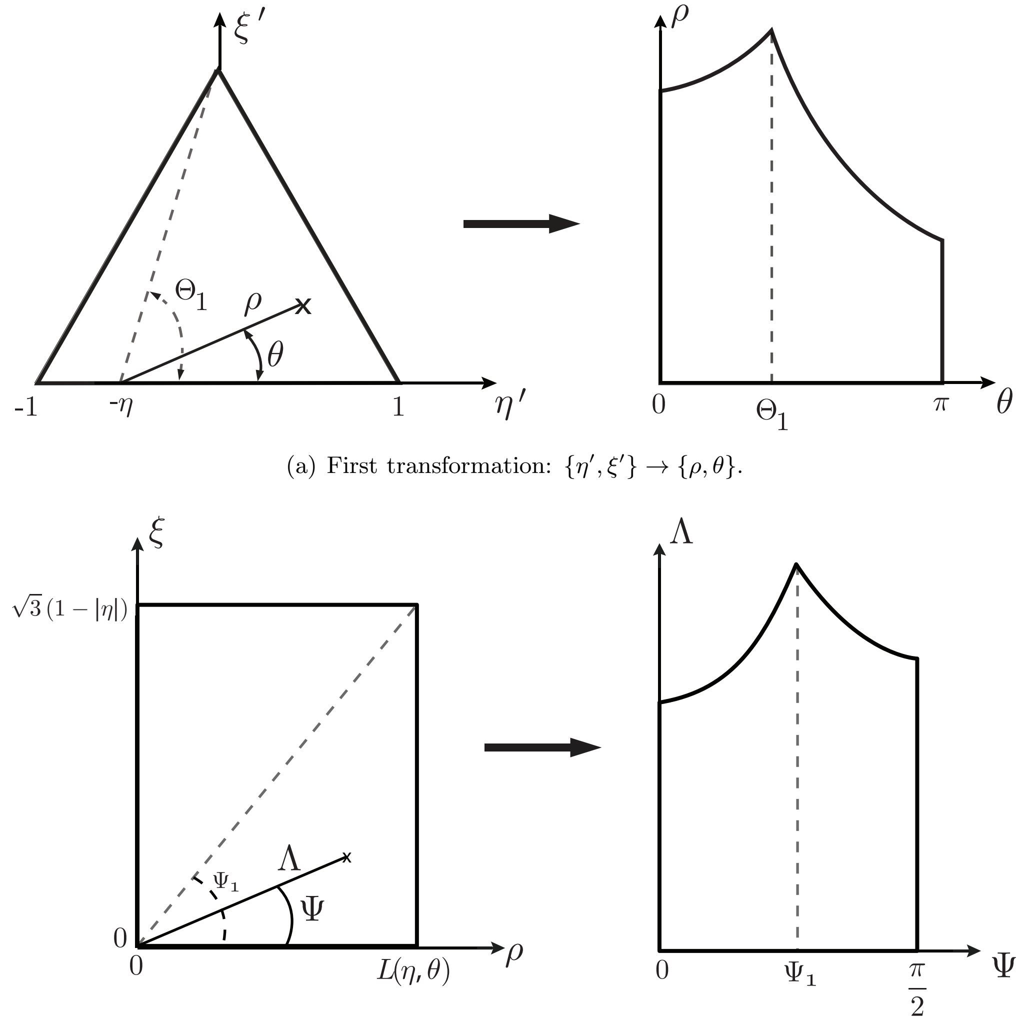
$$\mathbf{R} = \Lambda\left[\boldsymbol{\alpha}_{e_1}\cos(\Psi)\cos(\theta)+\boldsymbol{\alpha}_{e_2}\sin(\Psi)+\boldsymbol{\alpha}_{e_3}\cos(\Psi)\sin(\theta)\right] = \Lambda\mathbf{B}(\theta,\Psi) \tag{5.19}$$

and the original integral can be written as a sum of 3-D integrals (whose integration limits are defined later on),

$$I = \sum_{l=0}^{1}\sum_{m=0}^{1}\int_{-1}^{1} d\eta \int_{\Theta_l}^{\Theta_{l+1}} d\theta \int_{\Psi_m}^{\Psi_{m+1}} A(\eta,\theta,\Psi,\Lambda_L(\eta,\theta,\Psi))d\Psi \tag{5.20}$$

since the integral over Λ,

$$A(\eta,\theta,\Psi,\Lambda_L) = \frac{A_pA_q}{3}\cos(\Psi)\int_{0}^{\Lambda_L} \mathbf{g}(\eta,\Lambda,\Psi)\cdot\left(\nabla G(R)\times\mathbf{f}(\eta,\theta,\Lambda,\Psi)\right)\Lambda^2 d\Lambda \tag{5.21}$$



(a) First transformation: $\{\eta', \xi'\} \rightarrow \{\rho, \theta\}$.

(b) Second transformation: $\{\rho, \xi\} \rightarrow \{\Lambda, \Psi\}$.

Figure 5.3: Polar coordinate transformations employed in the first analytical integration.

can be evaluated analytically, as will be shown below. In the last expression one can easily identify the final Jacobian after the whole set of parametric transformations,

$$J = \frac{A_p A_q}{3} \Lambda^2 \cos(\Psi). \tag{5.22}$$

The main kernel of the hyper-singular integral can be extended as

$$\nabla G(R) = \nabla \left(\frac{e^{-jkR}}{R} \right) = -\frac{e^{-jkR}}{R^3} \left(1 + jkR\right) \mathbf{R} \tag{5.23}$$

where both the distance vector $\mathbf{R} = \mathbf{r_P} - \mathbf{r_Q}$ and its modulus R are proportional to Λ,

$$\begin{aligned}
&\mathbf{R} = \Lambda \mathbf{B}(\theta,\Psi) \\
&\mathbf{B}(\theta,\Psi) = \boldsymbol{\alpha}_{e_2}\sin(\Psi) + \boldsymbol{\alpha}_{e_1}\cos(\Psi)\cos(\theta) + \boldsymbol{\alpha}_{e_3}\cos(\Psi)\sin(\theta) \\
&R = \Lambda B(\theta,\Psi) \\
&B(\theta,\Psi) = \sqrt{\beta_{e_1}\sin^2(\Psi) + \beta_{e_2}\sin(\Psi)\cos(\Psi) + \beta_{e_3}\cos^2(\Psi)}
\end{aligned} \tag{5.24}$$

where

$$\begin{aligned}
\beta_{e_1} &= |\boldsymbol{\alpha}_{e_2}|^2 \\
\beta_{e_2} &= 2\left[(\boldsymbol{\alpha}_{e_1}\cdot\boldsymbol{\alpha}_{e_2})\cos(\theta) + (\boldsymbol{\alpha}_{e_2}\cdot\boldsymbol{\alpha}_{e_3})\sin(\theta)\right] \\
\beta_{e_3} &= |\boldsymbol{\alpha}_{e_1}|^2\cos^2(\theta) + 2(\boldsymbol{\alpha}_{e_1}\cdot\boldsymbol{\alpha}_{e_3})\cos(\theta)\sin(\theta) \\
&\quad + |\boldsymbol{\alpha}_{e_3}|^2\sin^2(\theta).
\end{aligned} \tag{5.25}$$

Substituting (5.24) in (5.23) yields

$$\nabla G(R) = -\frac{e^{-jk\Lambda B(\theta,\Psi)}}{\Lambda^2 B(\theta,\Psi)^3}\left(1 + jk\Lambda B(\theta,\Psi)\right)\mathbf{B}(\theta,\Psi). \tag{5.26}$$

One of the key points of the direct evaluation method is that the singularity $1/\Lambda^2$ is canceled out with the Jacobian (5.22), obtaining finally:

$$\begin{aligned}
A(\eta,\theta,\Psi,\Lambda_L) = \frac{A_p A_q}{3}\frac{\cos(\Psi)}{B(\theta,\Psi)^3}\int_0^{\Lambda_L} & e^{-jk\Lambda B(\theta,\Psi)}\left(1 + jk\Lambda B(\theta,\Psi)\right) \\
& \mathbf{g}(\eta,\Lambda,\Psi)\cdot\left(\mathbf{f}(\eta,\theta,\Lambda,\Psi)\times\mathbf{B}(\theta,\Psi)\right)d\Lambda.
\end{aligned} \tag{5.27}$$

Inasmuch as the testing and basis functions $\mathbf{g}(\eta,\Lambda,\Psi)$ and $\mathbf{f}(\eta,\theta,\Lambda,\Psi)$ are either RWG or $\hat{\mathbf{n}}\times$RWG functions, it is easy to prove (see Appendix C) that after applying the parametric transformations they have the following pattern:

$$\mathbf{v}_1(\theta,\Psi) + \mathbf{v}_2(\theta,\Psi)\Lambda + \mathbf{v}_3(\theta,\Psi)\eta \tag{5.28}$$

where $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are vectors independent of Λ and η, and are determined by the actual functions we use. Consequently, in (5.27), only terms proportional (considering the variable Λ) to any of the following:

$$\begin{aligned}
&e^{-jk\Lambda B(\theta,\Psi)}; \quad \Lambda e^{-jk\Lambda B(\theta,\Psi)} \\
&\Lambda^2 e^{-jk\Lambda B(\theta,\Psi)}; \quad \Lambda^3 e^{-jk\Lambda B(\theta,\Psi)}
\end{aligned} \tag{5.29}$$

appear. Of course, all of them can be analytically integrated with respect to Λ, therefore, justifying the argument below (5.21).

After the first analytical integration, which can be done with the help of a symbolic mathematical software like *Maple* for the sake of a systematic approach, the function A takes the form

$$A(\eta,\theta,\Psi,\Lambda_L) = \sum_{i=1}^{12} c_i(\theta,\Psi)M_i(\eta,\Lambda_L,B(\theta,\Psi)) \tag{5.30}$$

where $M_i(\eta, \Lambda, B)$ are fixed functions defined in Table 5.1 and the coefficients $c_i(\theta, \Psi)$ depend on the actual functions $\mathbf{g}$ and $\mathbf{f}$ under consideration.

Table 5.1: Functions $M_i(\eta, \Lambda, B)$.

$M_1(\eta, \Lambda, B) = 1$	$M_7(\eta, \Lambda, B) = \eta^2$
$M_2(\eta, \Lambda, B) = e^{-jkB\Lambda}$	$M_8(\eta, \Lambda, B) = \eta e^{-jkB\Lambda}$
$M_3(\eta, \Lambda, B) = \Lambda e^{-jkB\Lambda}$	$M_9(\eta, \Lambda, B) = \eta^2 e^{-jkB\Lambda}$
$M_4(\eta, \Lambda, B) = \Lambda^2 e^{-jkB\Lambda}$	$M_{10}(\eta, \Lambda, B) = \eta\Lambda e^{-jkB\Lambda}$
$M_5(\eta, \Lambda, B) = \Lambda^3 e^{-jkB\Lambda}$	$M_{11}(\eta, \Lambda, B) = \eta^2\Lambda e^{-jkB\Lambda}$
$M_6(\eta, \Lambda, B) = \eta$	$M_{12}(\eta, \Lambda, B) = \eta\Lambda^2 e^{-jkB\Lambda}$

The integration limits in (5.20) are given by

$$\Theta_0 = 0; \quad \Theta_1 = \frac{\pi}{2} - \tan^{-1}\left(\frac{\eta}{\sqrt{3}}\right); \quad \Theta_2 = \pi \tag{5.31}$$

and

$$\Psi_0 = 0; \quad \Psi_1 = \tan^{-1}\left(\frac{\sqrt{3}(1-|\eta|)}{L(\eta,\theta)}\right); \quad \Psi_2 = \frac{\pi}{2} \tag{5.32}$$

while the integration limits with respect to Λ take the following form:

$$\Lambda_L(\eta,\theta,\Psi) = \begin{cases} \dfrac{L(\eta,\theta)}{\cos(\Psi)}, & \Psi_0 < \Psi < \Psi_1 \\ \dfrac{\sqrt{3}(1-|\eta|)}{\sin(\Psi)}, & \Psi_1 < \Psi < \Psi_2. \end{cases} \tag{5.33}$$

Since the angle integrals cannot be evaluated analytically, it will be necessary to re-arrange the aforementioned integrals by placing the η integral in the innermost position, taking into account that the functions $M_i(\eta, \Lambda, B)$ can be integrated analytically in terms of η. As a last remark, note that for some combinations of testing and basis functions $\mathbf{g}$ and $\mathbf{f}$, only some of the terms in (5.30) appear, reducing the actual computational cost for those cases.

5.3.3 Second analytical integration - Reordering

In the next step, the integration is split for positive and negative values of η. Exploiting the symmetry of the equilateral triangle parameter space, though, it suffices to proceed only to the analytical integration over region $\eta \geq 0$ [83]. The remaining part can be easily computed by flipping the elements around, as presented in subsection 5.3.4.

Before starting with the second analytical integration, we introduce the three possible values of the function Λ_L, for the case $\eta \geq 0$, depending on the zone of the $\{\theta, \Psi\}$ plane:

$$\begin{aligned}
\Lambda_{L_1} &= \frac{\sqrt{3}(1-\eta)}{\cos(\Psi)(\sin(\theta) - \sqrt{3}\cos(\theta))} \\
\Lambda_{L_2} &= \frac{\xi(\eta)}{\sin(\Psi)} = \frac{\sqrt{3}(1-\eta)}{\sin(\Psi)} \\
\Lambda_{L_3} &= \frac{\sqrt{3}(1+\eta)}{\cos(\Psi)(\sin(\theta) + \sqrt{3}\cos(\theta))}.
\end{aligned} \tag{5.34}$$

5.3.3.1 Integration over region $\Theta_1 \leq \theta \leq \pi$

In this case, the splitting of Ψ integrals is independent of η, thus, leading, without complication, to the following two integrals:

$$I^{\theta+} = \int_0^1 d\eta \int_{\theta_\eta}^{\pi} d\theta \int_0^{\Psi_1^+} A(\eta, \theta, \Psi, \Lambda_{L_1}) d\Psi + \int_0^1 d\eta \int_{\theta_\eta}^{\pi} d\theta \int_{\Psi_1^+}^{\pi/2} A(\eta, \theta, \Psi, \Lambda_{L_2}) d\Psi \tag{5.35}$$

where the upper limit of the integration with respect to Λ (manifested as a parameter in the M functions) is different in the two terms,

$$\Lambda_L = \begin{cases} \Lambda_{L_1}, & 0 < \Psi < \Psi_1 \\ \Lambda_{L_2}, & \Psi_1 < \Psi < \pi/2. \end{cases} \tag{5.36}$$

Moreover,

$$\theta_\eta \equiv \Theta_1 = \frac{\pi}{2} - \tan^{-1}\left(\frac{\eta}{\sqrt{3}}\right) \tag{5.37}$$

and the Ψ integral is split at

$$\Psi_A \triangleq \Psi_1^+ = \tan^{-1}\left(\sin(\theta) - \sqrt{3}\cos(\theta)\right). \tag{5.38}$$

Since Ψ_1 is not a function of η, the integral with respect to η can be moved in the innermost integration, once it is interchanged with the θ integral according to the geometry shown in Fig. 5.4 and the inversion formula

$$\eta_\theta = \frac{\sqrt{3}}{\tan(\theta)}. \tag{5.39}$$

Finally, switching the integrals results in

$$\begin{aligned}
I^{\theta+} = &\int_{\pi/2}^{\pi} d\theta \int_0^{\Psi_A} X^a(\theta, \Psi) d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} X^b(\theta, \Psi) d\Psi \\
&+ \int_{\pi/2}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} X^c(\theta, \Psi) d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\pi/2} X^d(\theta, \Psi) d\Psi
\end{aligned} \tag{5.40}$$

where $X^\alpha(\theta, \Psi)$, $\alpha = a, b, c, d$, are the following analytically integrated with respect to η functions:

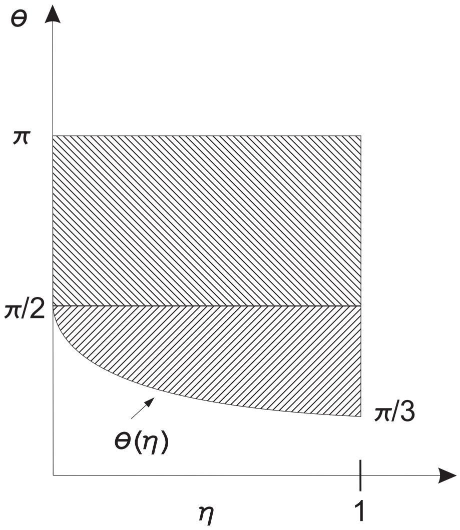


Figure 5.4: The $\{\eta, \theta\}$ domain for the first shift of the integral, $\Theta_1 \leq \theta \leq \pi$.

$$\begin{aligned}
X^a(\theta, \Psi) &= \int_0^1 A(\eta, \theta, \Psi, \Lambda_{L_1}) d\eta \\
X^b(\theta, \Psi) &= \int_{\eta_\theta}^1 A(\eta, \theta, \Psi, \Lambda_{L_1}) d\eta \\
X^c(\theta, \Psi) &= \int_0^1 A(\eta, \theta, \Psi, \Lambda_{L_2}) d\eta \\
X^d(\theta, \Psi) &= \int_{\eta_\theta}^1 A(\eta, \theta, \Psi, \Lambda_{L_2}) d\eta.
\end{aligned} \tag{5.41}$$

5.3.3.2 Integration over region $0 < \theta \leq \Theta_1$

In this case, the two starting integrals are given by

$$I^{\theta-} = \int_0^1 d\eta \int_0^{\theta_\eta} d\theta \int_0^{\Psi_1^-} A(\eta, \theta, \Psi, \Lambda_{L_3}) d\Psi + \int_0^1 d\eta \int_0^{\theta_\eta} d\theta \int_{\Psi_1^-}^{\pi/2} A(\eta, \theta, \Psi, \Lambda_{L_2}) d\Psi \tag{5.42}$$

where, again,

$$\Lambda_L = \begin{cases} \Lambda_{L_3}, & 0 < \Psi < \Psi_1 \\ \Lambda_{L_2}, & \Psi_1 < \Psi < \pi/2 \end{cases} \tag{5.43}$$

and

$$\theta_\eta \equiv \Theta_1 = \frac{\pi}{2} - \tan^{-1}\left(\frac{\eta}{\sqrt{3}}\right) \tag{5.44}$$

only now the breakpoint in Ψ is a function of η,

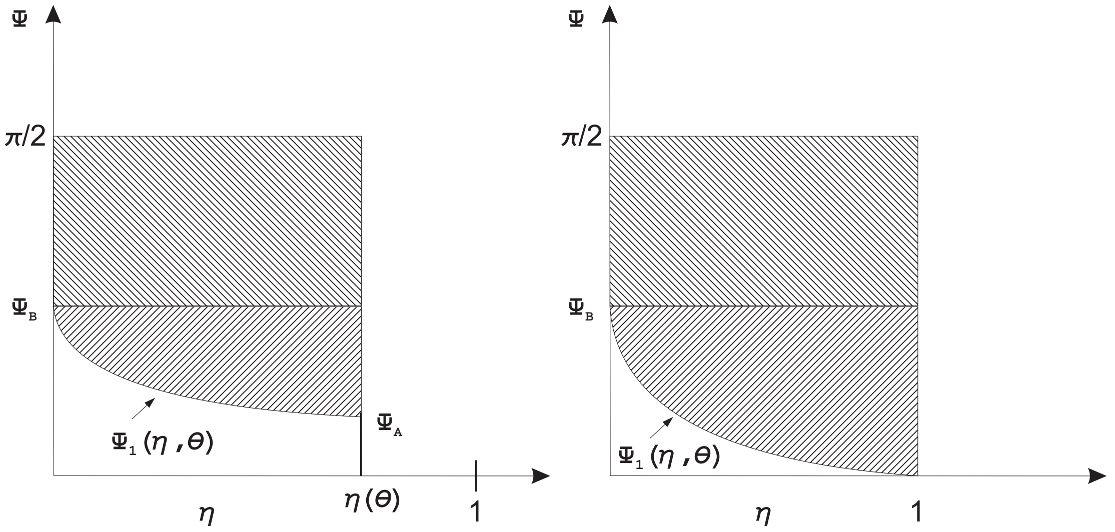
$$\Psi_1^- = \tan^{-1}\left(\frac{\xi(\eta)}{L(\eta, \theta)}\right) = \tan^{-1}\left(\frac{1-\eta}{1+\eta}\left[\sin(\theta) + \sqrt{3}\cos(\theta)\right]\right). \tag{5.45}$$

As before, the θ and η integrals are easily interchanged, the domain being the region below the

$\theta(\eta)$ curve in Fig. 5.4. This results in the four integrals

$$\begin{aligned} I^{\theta_-} = &\int_0^{\pi/3} d\theta \int_0^1 d\eta \int_0^{\Psi_1^-} A(\eta,\theta,\Psi,\Lambda_{L_3})d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_0^{\eta_\theta} d\eta \int_0^{\Psi_1^-} A(\eta,\theta,\Psi,\Lambda_{L_3})d\Psi \\ &+ \int_0^{\pi/3} d\theta \int_0^1 d\eta \int_{\Psi_1^-}^{\pi/2} A(\eta,\theta,\Psi,\Lambda_{L_2})d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_0^{\eta_\theta} d\eta \int_{\Psi_1^-}^{\pi/2} A(\eta,\theta,\Psi,\Lambda_{L_2})d\Psi \end{aligned} \tag{5.46}$$

where η_θ is given in (5.39). Using the fact that $\Psi_1^-(\eta = 1, \theta) = 0$, the geometry for interchanging η and Ψ is shown in Figs. 5.5(a) and 5.5(b).



(a) Case $\eta(\theta) < 1$.

(b) Case $\eta(\theta) = 1$.

Figure 5.5: The domain for interchanging the integrals $\{\eta, \Psi\}$, for a fixed value of θ.

Integration over region $0 < \Psi \le \Psi_1^-$:

Moving the η integral to the front in the first two integrals in (5.46) ($0 < \Psi < \Psi_1^-$) and using that $\Psi_1^-(\eta = \eta_\theta) \equiv \Psi_A$, results in

$$\begin{aligned} I^{\theta_-,\Psi_-} = &\int_0^{\pi/3} d\theta \int_0^{\Psi_B} X^e(\theta,\Psi)d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} X^f(\theta,\Psi)d\Psi \\ &+ \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} X^g(\theta,\Psi)d\Psi \end{aligned} \tag{5.47}$$

where

$$\Psi_B \triangleq \Psi_1^-(\eta = 0) = \tan^{-1}\left(\sin(\theta) + \sqrt{3}\cos(\theta)\right) \tag{5.48}$$

and

$$\begin{aligned} X^e(\theta,\Psi) = X^g(\theta,\Psi) &= \int_0^{\Psi_\eta} A(\eta,\theta,\Psi,\Lambda_{L_3})d\eta \\ X^f(\theta,\Psi) &= \int_0^{\eta_\theta} A(\eta,\theta,\Psi,\Lambda_{L_3})d\eta. \end{aligned} \tag{5.49}$$

Again, the functions $X^\alpha(\theta,\Psi)$, $\alpha = e,f,g$, are integrated analytically and the integration limit Ψ_η is given by

$$\Psi_\eta = \frac{1 - \frac{\tan(\Psi)}{\sin(\theta)+\sqrt{3}\cos(\theta)}}{1 + \frac{\tan(\Psi)}{\sin(\theta)+\sqrt{3}\cos(\theta)}} = \frac{1-\gamma}{1+\gamma}. \tag{5.50}$$

Integration over region $\Psi_1^- < \Psi \leq \pi/2$:

Similarly, the second two integrals in (5.46) ($\Psi_1^- < \Psi < \pi/2$) become, after the analytical integration with respect to η,

$$\begin{aligned} I^{\theta_-,\Psi_+} = &\int_0^{\pi/3} d\theta \int_0^{\Psi_B} X^h(\theta,\Psi)d\Psi + \int_0^{\pi/3} d\theta \int_{\Psi_B}^{\pi/2} X^i(\theta,\Psi)d\Psi \\ &+ \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} X^k(\theta,\Psi)d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_B}^{\pi/2} X^l(\theta,\Psi)d\Psi \end{aligned} \tag{5.51}$$

where

$$\begin{aligned} X^h(\theta,\Psi) &= \int_{\Psi_\eta}^1 A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta \\ X^i(\theta,\Psi) &= \int_0^1 A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta \\ X^k(\theta,\Psi) &= \int_{\Psi_\eta}^{\eta_\theta} A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta \\ X^l(\theta,\Psi) &= \int_0^{\eta_\theta} A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta. \end{aligned} \tag{5.52}$$

Finally, by adding all the components from (5.40), (5.47) and (5.51), we end up to the following formula for the integration over positive values of η:

$$I^{\eta_+} = I^{\theta_-} + I^{\theta_+} = I^{\theta_-,\Psi_-} + I^{\theta_-,\Psi_+} + I^{\theta_+}. \tag{5.53}$$

More specifically, the hyper-singular integral for $\eta > 0$ has been reduced, after re-grouping the integrals over disjoint domains in the space $\{\theta,\Psi\}$ (see Fig. 5.6), to the following 2-D smooth integral:

$$
\begin{aligned}
I^{\eta+} = &\int_0^{\pi/3} d\theta \int_0^{\Psi_B} \chi^{a+}(\theta,\Psi)d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} \chi^{a+}(\theta,\Psi)d\Psi \\
&+ \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} \chi^{b+}(\theta,\Psi)d\Psi + \int_{\pi/2}^{\pi} d\theta \int_0^{\Psi_A} \chi^{c+}(\theta,\Psi)d\Psi \\
&+ \int_0^{\pi/2} d\theta \int_{\Psi_B}^{\pi/2} \chi^{d+}(\theta,\Psi)d\Psi + \int_{\pi/2}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} \chi^{d+}(\theta,\Psi)d\Psi
\end{aligned}
\tag{5.54}
$$

which depends only on the four different functions

$$
\begin{aligned}
&\chi^{a+}(\theta,\Psi) = X^e + X^h = X^d + X^g + X^k = \int_0^{\Psi_\eta} A(\eta,\theta,\Psi,\Lambda_{L_3})d\eta + \int_{\Psi_\eta}^1 A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta \\
&\chi^{b+}(\theta,\Psi) = X^b + X^f = \int_0^{\eta_\theta} A(\eta,\theta,\Psi,\Lambda_{L_3})d\eta + \int_{\eta_\theta}^1 A(\eta,\theta,\Psi,\Lambda_{L_1})d\eta \\
&\chi^{c+}(\theta,\Psi) = X^a = \int_0^1 A(\eta,\theta,\Psi,\Lambda_{L_1})d\eta \\
&\chi^{d+}(\theta,\Psi) = X^i = X^d + X^l + X^c = \int_0^1 A(\eta,\theta,\Psi,\Lambda_{L_2})d\eta.
\end{aligned}
\tag{5.55}
$$

The function $\chi^{a+}(\theta,\Psi)$ is defined over the integration domains Ω_1 and Ω_2, $\chi^{b+}(\theta,\Psi)$ over Ω_3, $\chi^{c+}(\theta,\Psi)$ over Ω_4 and $\chi^{d+}(\theta,\Psi)$ over Ω_5 and Ω_6, as shown in Fig. 5.6. Considering the expressions of A in (5.30) and the linearity of the integrals, the abovementioned functions equal

$$
\chi^{\alpha+}(\theta,\Psi) = \sum_{i=1}^{12} c_i(\theta,\Psi) N_i^{\alpha}(B(\theta,\Psi),\theta,\Psi)
\tag{5.56}
$$

where $\alpha \in \{a,b,c,d\}$ and the new functions $N_i^{\alpha}(B,\theta,\Psi)$ are defined using $M_i(\eta,\Lambda,B)$ as:

$$
\begin{aligned}
&N_i^a(B,\theta,\Psi) = \int_0^{\Psi_\eta} M_i(\eta,\Lambda_{L_3},B)d\eta + \int_{\Psi_\eta}^1 M_i(\eta,\Lambda_{L_2},B)d\eta \\
&N_i^b(B,\theta,\Psi) = \int_0^{\eta_\theta} M_i(\eta,\Lambda_{L_3},B)d\eta + \int_{\eta_\theta}^1 M_i(\eta,\Lambda_{L_1},B)d\eta \\
&N_i^c(B,\theta,\Psi) = \int_0^1 M_i(\eta,\Lambda_{L_1},B)d\eta \\
&N_i^d(B,\theta,\Psi) = \int_0^1 M_i(\eta,\Lambda_{L_2},B)d\eta.
\end{aligned}
\tag{5.57}
$$

5.3.4 Equilateral triangle parameter space's symmetry effects

The remaining case (integration over negative values of η) can be handled by simply flipping the master triangles (triangles in the equilateral triangle parametric space) and employing the formulas

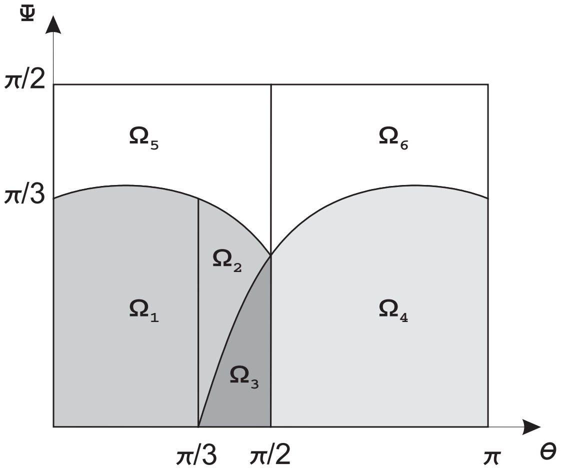


Figure 5.6: Integration domains in the $\{\theta, \Psi\}$ space.

for the $\eta > 0$ case (Fig. 5.7). More specifically, the flipping transformation is stated as

$$\begin{bmatrix}\eta\\ \xi\end{bmatrix} = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{H}\\ \Xi\end{bmatrix} \tag{5.58}$$

$$\begin{bmatrix}\eta'\\ \xi'\end{bmatrix} = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{H}'\\ \Xi'\end{bmatrix} \tag{5.59}$$

with the overall additional multiplicative Jacobian being equal to 1. After applying the two polar coordinate transformations, analogously to (5.12) and (5.18) but with the extra aforementioned transformation, the distance function R takes the form

$$R = \Lambda B(\pi - \theta, \Psi). \tag{5.60}$$

From the formulation point of view, the integral over $\eta < 0$, starting from eq. (5.7) can be expressed as:

$$I^{\eta^-} = (J_p J_q) \int_{-1}^{0} d\eta \int_{0}^{\xi(\eta)} d\xi \int_{-1}^{1} d\eta' \int_{0}^{\xi(\eta')} \mathbf{g}(\eta,\xi) \cdot \big(\nabla G(R) \times \mathbf{f}(\eta',\xi')\big)\, d\xi' \tag{5.61}$$

and after applying the flipping changes

$$I^{\eta^-} = (J_p J_q) \int_{0}^{1} dH \int_{0}^{\xi(-H)} d\xi \int_{-1}^{1} dH' \int_{0}^{\xi(-H')} \mathbf{g}(-H,\xi) \cdot \big(\nabla G(R) \times \mathbf{f}(-H',\xi')\big)\, d\xi'. \tag{5.62}$$

Considering that the integration limit function $\xi(x) = \sqrt{(}3)(1 - |x|)$ only depends on the absolute value of its arguments, the aforementioned integral can be expressed equivalently as

$$I^{\eta^-} = (J_p J_q) \int_{0}^{1} d\eta \int_{0}^{\xi(\eta)} d\xi \int_{-1}^{1} d\eta' \int_{0}^{\xi(\eta')} \mathbf{g}(-\eta,\xi) \cdot \big(\nabla G(R) \times \mathbf{f}(-\eta',\xi')\big)\, d\xi'. \tag{5.63}$$

which is equivalent to the integration over the positive η but evaluating the integrand in $-\eta$ and

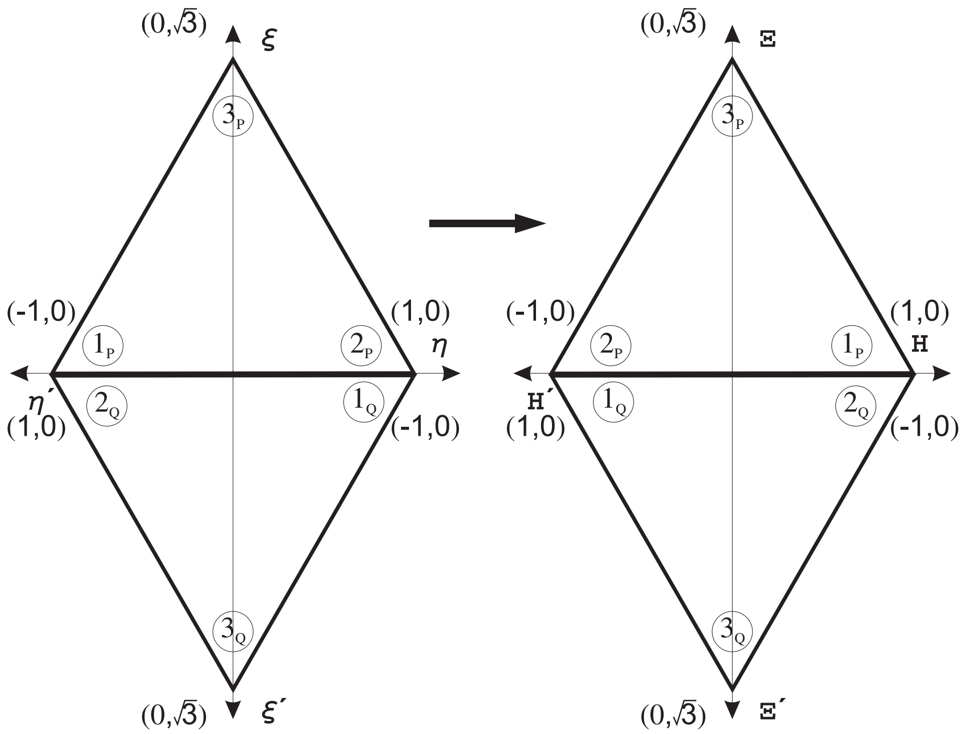


Figure 5.7: Flipping of the master triangles.

$-\eta'$. The new polar coordinate change

$$\begin{aligned} -\eta' &= \rho\cos{(\theta)} + \eta \\ \xi' &= \rho\sin{(\theta)} \end{aligned} \tag{5.64}$$

equals the original one just using $\pi-\theta$ instead of θ and nothing affects the second polar coordinates change.

5.3.5 Final formulas

Summarizing, the objective hyper-singular integral (5.5) can be computed by simply evaluating numerically the 2-D smooth integral

$$\begin{aligned} I = &\int_0^{\pi/3} d\theta \int_0^{\Psi_B} \chi^a(\theta,\Psi)d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} \chi^a(\theta,\Psi)d\Psi + \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} \chi^b(\theta,\Psi)d\Psi \\ &+ \int_{\pi/2}^{\pi} d\theta \int_0^{\Psi_A} \chi^c(\theta,\Psi)d\Psi + \int_0^{\pi/2} d\theta \int_{\Psi_B}^{\pi/2} \chi^d(\theta,\Psi)d\Psi + \int_{\pi/2}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} \chi^d(\theta,\Psi)d\Psi \end{aligned} \tag{5.65}$$

where the functions $\chi^\alpha(\theta,\Psi)$ with $\alpha \in \{a,b,c,d\}$ are given by

$$\begin{aligned} \chi^\alpha(\theta,\Psi) = \sum_{i=1}^{12} &c_i^{\eta^+}(\theta,\Psi)N_i^\alpha(B^+,\theta,\Psi) \\ &+ c_i^{\eta^-}(\theta,\Psi)N_i^\alpha(B^-,\theta,\Psi) \end{aligned} \tag{5.66}$$

with

$$\begin{aligned} B^{+} &= \sqrt{\beta_{e1}\sin^{2}(\Psi) + \beta_{e2}\sin(\Psi)\cos(\Psi) + \beta_{e3}\cos^{2}(\Psi)} \\ B^{-}(\theta,\Psi) &= B^{+}(\pi-\theta,\Psi) \\ c_i^{\eta^{-}}(\theta,\Psi) &= c_i^{\eta^{+}}(\pi-\theta,\Psi)S_i \end{aligned} \tag{5.67}$$

and the case-dependent sign S_i:

$$S_i = \begin{cases} -1, & i = 6,8,10,12 \\ +1, & \text{otherwise.} \end{cases} \tag{5.68}$$

Also, $c_i^{\eta^{+}}(\theta,\Psi)$ are the coefficients of the functions $M_i(\eta,\Lambda_L,B^{+})$ in (5.30) after the first analytical integration and the functions $N_i^{\alpha}(B,\theta,\Psi)$ are explicitly derived in Appendix D. The sign S_i comes from those elements M_i which have an odd exponent for η, which carries a minus sign when changed to $-\eta$.

5.4 Direct evaluation method - Vertex Adjacent

5.4.1 Equilateral triangle parameter space

Equivalently to the edge adjacent case, we first transform each of the two original triangles to an equilateral triangle parameter space (see Fig. 5.8). However, in this case, it must be set so that the common vertex $\mathbf{r}_3$ goes to the point $(\eta,\xi) = (0,\sqrt{3})$ in the first triangle $E_P : (\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3)$, and to $(\eta',\xi') = (0,\sqrt{3})$ in the second triangle $E_Q : (\mathbf{r}_3,\mathbf{r}_4,\mathbf{r}_5)$. The singularity is then placed at $\eta = \eta' = 0$ and $\xi = \xi' = \sqrt{3}$ in the new parameter space. Keeping the notation used in the edge adjacent case, the transformation can be expressed as

$$\mathbf{r} = \mathbf{r_P} = \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_2-x_1}{2} & \frac{2x_3-x_1-x_2}{2\sqrt{3}} \\ \frac{y_2-y_1}{2} & \frac{2y_3-y_1-y_2}{2\sqrt{3}} \\ \frac{z_2-z_1}{2} & \frac{2z_3-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = \frac{\mathbf{r}_1+\mathbf{r}_2}{2} + \boldsymbol{\alpha}_{e1}\eta + \boldsymbol{\alpha}_{e2}\xi \tag{5.69}$$

$$\mathbf{r}' = \mathbf{r_Q} = \begin{bmatrix} \frac{x_4+x_5}{2} \\ \frac{y_4+y_5}{2} \\ \frac{z_4+z_5}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_5-x_4}{2} & \frac{2x_3-x_4-x_5}{2\sqrt{3}} \\ \frac{y_5-y_4}{2} & \frac{2y_3-y_4-y_5}{2\sqrt{3}} \\ \frac{z_5-z_4}{2} & \frac{2z_3-z_4-z_5}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta' \\ \xi' \end{bmatrix} = \frac{\mathbf{r}_4+\mathbf{r}_5}{2} + \boldsymbol{\alpha}_{e4}\eta' + \boldsymbol{\alpha}_{e5}\xi' \tag{5.70}$$

where the vectors $\boldsymbol{\alpha}_i$ depend only in the vertex coordinates of the triangles and are explicitly defined as

$$\begin{aligned} \boldsymbol{\alpha}_{e1} &= \frac{\mathbf{r}_2-\mathbf{r}_1}{2} \\ \boldsymbol{\alpha}_{e2} &= \frac{2\mathbf{r}_3-\mathbf{r}_1-\mathbf{r}_2}{2\sqrt{3}} \\ \boldsymbol{\alpha}_{e4} &= \frac{\mathbf{r}_5-\mathbf{r}_4}{2} \\ \boldsymbol{\alpha}_{e5} &= \frac{2\mathbf{r}_3-\mathbf{r}_4-\mathbf{r}_5}{2\sqrt{3}}. \end{aligned} \tag{5.71}$$

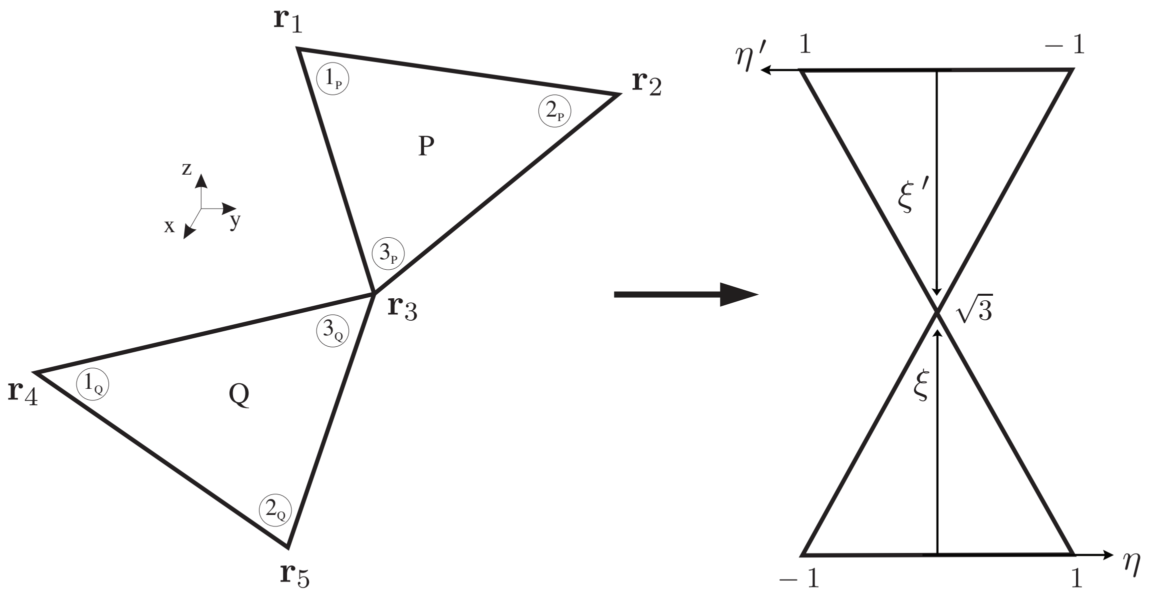


Figure 5.8: Orientation of the triangular elements both in the original and the equilateral triangle parameter space in the vertex adjacent case.

The Jacobian of this transformation is

$$J = \frac{A_p A_q}{3} \tag{5.72}$$

where A_p and A_q are the areas of the respective triangles. The new integral to evaluate becomes then:

$$I = \frac{A_p A_q}{3} \int_{-1}^{1} d\eta \int_{0}^{\xi(\eta)} d\xi \int_{-1}^{1} d\eta' \int_{0}^{\xi(\eta')} \mathbf{g}(\eta,\xi) \cdot \left(\nabla G(R) \times \mathbf{f}(\eta',\xi')\right) d\xi' \tag{5.73}$$

which includes again the integration limit function $\xi(\eta) = \sqrt{3}(1-|\eta|)$. Differently from the previous section, the symmetries with respect to η and to η' can be exploited, not just one of them. In this section we will carry the whole symmetry development from the very beginning, so that the reader can have two slightly different points of view of the same thing. Therefore, the last integral can be expressed, considering the integration of positive η and η' and benefiting from the symmetry of the function $\xi(\eta)$, as

$$I = \frac{A_p A_q}{3} \int_{0}^{1} d\eta \int_{0}^{\xi(\eta)} d\xi \int_{0}^{1} d\eta' \int_{0}^{\xi(\eta')} F(\eta,\xi,\eta',\xi') d\xi' \tag{5.74}$$

where the integrand function equals

$$\begin{aligned} F(\eta,\xi,\eta',\xi') =& F^{\eta^+}(\eta,\xi,\eta',\xi') + F^{\eta^+}(-\eta,\xi,\eta',\xi') \\ &+ F^{\eta^+}(\eta,\xi,-\eta',\xi') + F^{\eta^+}(-\eta,\xi,-\eta',\xi') \end{aligned} \tag{5.75}$$

whereas the generator or basic function F^{η^+} for the four symmetries is

$$F^{\eta^+}(\eta,\xi,\eta',\xi') = \mathbf{g}(\eta,\xi)\cdot\left(\nabla G(R(\eta,\xi,\eta',\xi'))\times\mathbf{f}(\eta',\xi')\right). \tag{5.76}$$

5.4.2 Analytical integration - Polar coordinates changes

In order to cancel out the singularity, we perform a first polar coordinate transformation per triangle with center at the common vertex (see Fig. 5.9)

$$\begin{aligned} \xi &= \sqrt{3}-\rho_p\cos(\theta_p), \quad & \eta &= \rho_p\sin(\theta_p) \\ \xi' &= \sqrt{3}-\rho_q\cos(\theta_q), \quad & \eta' &= \rho_q\sin(\theta_q). \end{aligned} \tag{5.77}$$

In the new parametric space $\{\rho_p,\theta_p,\rho_q,\theta_q\}$, the singularity is placed at $\rho_p=\rho_q=0$, a fact which is evidenced in the new expression of the distance function vector

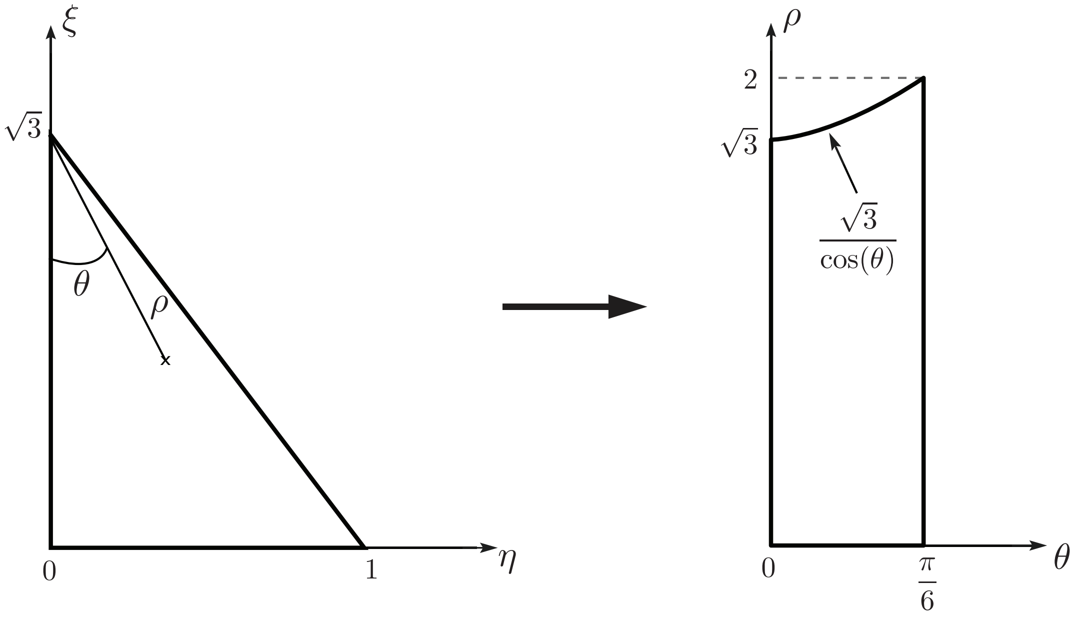


Figure 5.9: First polar coordinate transformation which applies to both triangle elements ($\{\eta,\xi\}\rightarrow\{\theta_p,\rho_p\}$ and $\{\eta',\xi'\}\rightarrow\{\theta_q,\rho_q\}$).

$$\mathbf{R} = \mathbf{r}_P-\mathbf{r}_Q = \boldsymbol{\alpha}_{e1}\rho_p\sin(\theta_p)-\boldsymbol{\alpha}_{e2}\rho_p\cos(\theta_p)-\boldsymbol{\alpha}_{e4}\rho_q\sin(\theta_q)+\boldsymbol{\alpha}_{e5}\rho_q\cos(\theta_q). \tag{5.78}$$

The accumulated overall Jacobian, considering that the two previous polar changes have a Jacobian ρ_p and ρ_q, respectively, is

$$J = \frac{A_pA_q}{3}\rho_p\rho_q. \tag{5.79}$$

The integral under study is then transformed in the new parametric space to

$$I = \frac{A_pA_q}{3}\int_0^{\pi/6}d\theta_p\int_0^{\pi/6}d\theta_q\int_0^{\sqrt{3}/\cos(\theta_p)}d\rho_p\int_0^{\sqrt{3}/\cos(\theta_q)}\rho_p\rho_qF(\theta_p,\theta_q,\rho_p,\rho_q)d\rho_q \tag{5.80}$$

where the integrals have been easily re-ordered as the integration limits of θ_q and ρ_p are independent. In the new parametric space, when the integrand function depends on $-\eta$, this corresponds to a $-\theta_p$ dependence and the same happens with $-\eta'$ and $-\theta_q$. Therefore, the integrand function can be expressed as

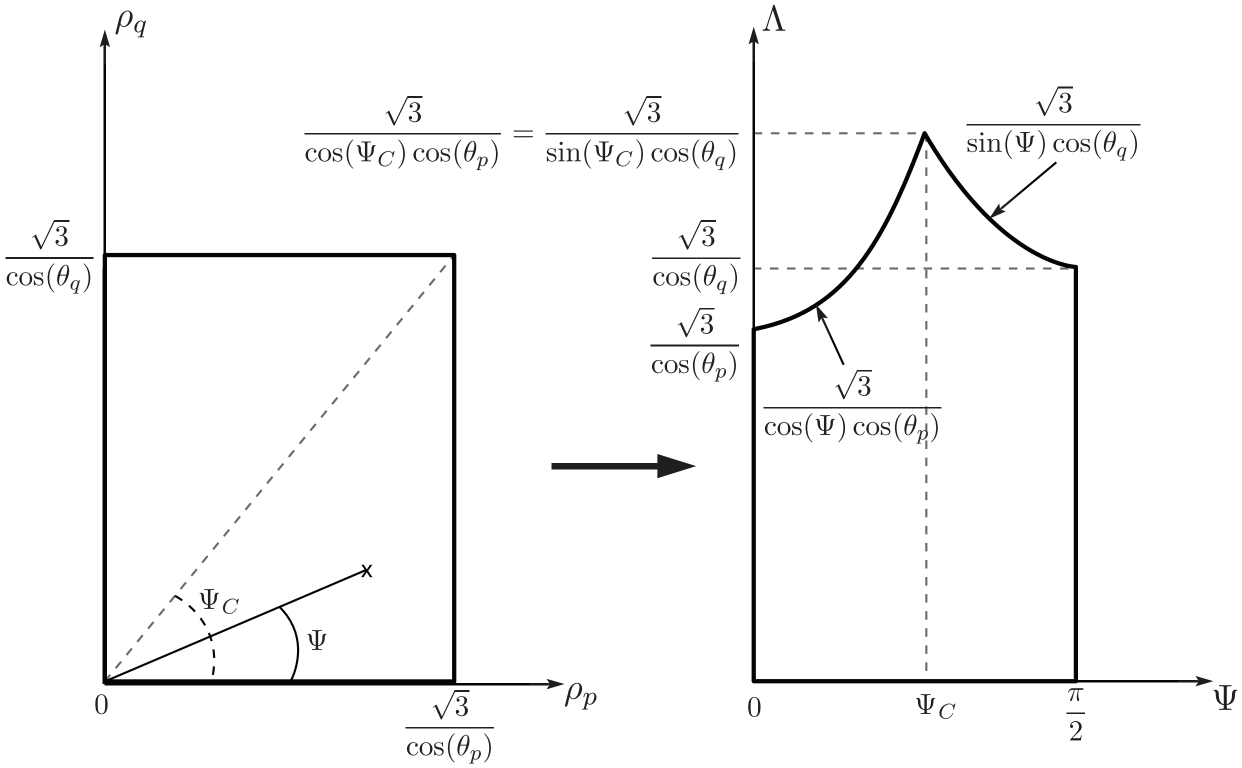
$$\begin{aligned} F(\theta_p,\theta_q,\rho_p,\rho_q) =& F^{\eta^+}(\theta_p,\theta_q,\rho_p,\rho_q) + F^{\eta^+}(-\theta_p,\theta_q,\rho_p,\rho_q) \\ &+ F^{\eta^+}(\theta_p,-\theta_q,\rho_p,\rho_q) + F^{\eta^+}(-\theta_p,-\theta_q,\rho_p,\rho_q) \end{aligned} \tag{5.81}$$

where the generator equals

$$F^{\eta^+}(\theta_p,\theta_q,\rho_p,\rho_q) = \mathbf{g}(\theta_p,\rho_p)\cdot\left(\nabla G(R(\theta_p,\theta_q,\rho_p,\rho_q)) \times \mathbf{f}(\theta_q,\rho_q)\right). \tag{5.82}$$

Another polar coordinates transformation to remove the singularity which still remains at $\rho_p = \rho_q = 0$ is now in order (see Fig. 5.10):

$$\begin{aligned} \rho_p &= \Lambda\cos(\Psi) \\ \rho_q &= \Lambda\sin(\Psi). \end{aligned} \tag{5.83}$$



ρ_q
$\frac{\sqrt{3}}{\cos(\Psi_C)\cos(\theta_p)} = \frac{\sqrt{3}}{\sin(\Psi_C)\cos(\theta_q)}$
$\frac{\sqrt{3}}{\cos(\theta_q)}$
Ψ_C
Ψ
0
$\frac{\sqrt{3}}{\cos(\theta_p)}$
ρ_p
Λ
$\frac{\sqrt{3}}{\sin(\Psi)\cos(\theta_q)}$
$\frac{\sqrt{3}}{\cos(\theta_q)}$
$\frac{\sqrt{3}}{\cos(\theta_p)}$
$\frac{\sqrt{3}}{\cos(\Psi)\cos(\theta_p)}$
0
Ψ_C
$\frac{\pi}{2}$
Ψ

Figure 5.10: Second polar coordinate transformation employed in the analytical integration in the vertex adjacent case.

The singularity is displaced to $\Lambda = 0$, with the peculiarity that the distance vector function is proportional to Λ:

$$\mathbf{R} = \Lambda\mathbf{\Gamma}(\theta_p, \theta_q, \Psi) \tag{5.84}$$

being the proportionality vector Γ:

$$\mathbf{\Gamma}(\theta_p, \theta_q, \Psi) = (\boldsymbol{\alpha}_{e_1} \sin(\theta_p) - \boldsymbol{\alpha}_{e_2} \cos(\theta_p)) \cos(\Psi) - (\boldsymbol{\alpha}_{e_4} \sin(\theta_q) + \boldsymbol{\alpha}_{e_5} \cos(\theta_q)) \sin(\Psi). \tag{5.85}$$

The new overall Jacobian is

$$J = \frac{A_p A_q}{3} \Lambda^3 \sin(\Psi) \cos(\Psi) \tag{5.86}$$

which, similarly to the edge adjacent case, will remove the singularity $1/R^2$. Actually, it would be able to remove a singularity up to $1/R^3$. Furthermore, the resulting integral is analytically integrable with respect to Λ, as will be seen in a while.

After applying the last coordinate change the integral takes the form:

$$\begin{aligned} I = \frac{A_p A_q}{3} \int_0^{\pi/6} d\theta_p \int_0^{\pi/6} d\theta_q \Bigg[\int_0^{\Psi_C} d\Psi \int_0^{\Lambda_{L_4}} \Lambda^3 \cos(\Psi) \sin(\Psi) F(\theta_p, \theta_q, \Psi, \Lambda) d\Lambda + \\ \int_{\Psi_C}^{\pi/2} d\Psi \int_0^{\Lambda_{L_5}} \Lambda^3 \cos(\Psi) \sin(\Psi) F(\theta_p, \theta_q, \Psi, \Lambda) d\Lambda \Bigg] \end{aligned} \tag{5.87}$$

where the integration limits are defined as:

$$\begin{aligned} \Psi_C &= \tan^{-1}\left(\frac{\cos(\theta_p)}{\cos(\theta_q)}\right) \\ \Lambda_{L_4} &= \frac{\sqrt{3}}{\cos(\Psi)\cos(\theta_p)} \\ \Lambda_{L_5} &= \frac{\sqrt{3}}{\sin(\Psi)\cos(\theta_q)}. \end{aligned} \tag{5.88}$$

Regrouping the integrand and the integral with respect to Λ in a new function Ω it can be rewritten as:

$$I = \int_0^{\pi/6} d\theta_p \int_0^{\pi/6} d\theta_q \left[\int_0^{\Psi_C} \Omega(\theta_p, \theta_q, \Psi, \Lambda_{L_4}) d\Psi + \int_{\Psi_C}^{\pi/2} \Omega(\theta_p, \theta_q, \Psi, \Lambda_{L_5}) d\Psi \right]. \tag{5.89}$$

Substituting the new expression of $\mathbf{R}$ in eq. (5.23) the main kernel of the hyper-singular integral can be extended as

$$\nabla G(R) = -\frac{e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)}}{\Lambda^2 \Gamma(\theta_p, \theta_q, \Psi)^3} (1 + jk\Lambda\Gamma(\theta_p, \theta_q, \Psi)) \mathbf{\Gamma}(\theta_p, \theta_q, \Psi) \tag{5.90}$$

which after being included in the expression of Ω yields

$$\begin{aligned} \Omega(\theta_p, \theta_q, \Psi, \Lambda_L) =& \Omega^{\eta^+}(\theta_p, \theta_q, \Psi, \Lambda_L) + \Omega^{\eta^+}(-\theta_p, \theta_q, \Psi, \Lambda_L) \\ &+ \Omega^{\eta^+}(\theta_p, -\theta_q, \Psi, \Lambda_L) + \Omega^{\eta^+}(-\theta_p, -\theta_q, \Psi, \Lambda_L) \end{aligned} \tag{5.91}$$

where

$$\begin{aligned}\Omega^{\eta^+}(\theta_p,\theta_q,\Psi,\Lambda_L) &= \frac{A_pA_q}{3}\cos(\Psi)\sin(\Psi)\int_0^{\Lambda_L}\Lambda^3F^{\eta^+}(\theta_p,\theta_q,\Psi,\Lambda)d\Lambda \\ &= \frac{A_pA_q}{3}\frac{\cos(\Psi)\sin(\Psi)}{\Gamma(\theta_p,\theta_q,\Psi)^3}\int_0^{\Lambda_L}\Lambda e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)}\left(1+jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)\right) \\ &\qquad \mathbf{g}(\theta_p,\Psi,\Lambda)\cdot\left(\mathbf{f}(\theta_q,\Psi,\Lambda)\times\mathbf{\Gamma}(\theta_p,\theta_q,\Psi)\right)d\Lambda.\end{aligned} \tag{5.92}$$

Similarly to the edge adjacent case, as the testing and basis functions $\mathbf{g}(\theta_p,\Psi,\Lambda)$ and $\mathbf{f}(\theta_q,\Psi,\Lambda)$ are either RWG or $\hat{\mathbf{n}}\times$RWG functions, it is easy to prove (see Appendix C) that after applying the parametric transformations they have the following pattern:

$$\mathbf{v}_1(\theta_p,\theta_q,\Psi)+\mathbf{v}_2(\theta_p,\theta_q,\Psi)\Lambda \tag{5.93}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are vectors independent of Λ, and are determined by the actual functions we use. Consequently, in the last expression of the integral, only terms proportional (considering the variable Λ) to any of the following:

$$\begin{aligned}&\Lambda e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)}; \quad \Lambda^2e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)} \\ &\Lambda^3e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)}; \quad \Lambda^4e^{-jk\Lambda\Gamma(\theta_p,\theta_q,\Psi)}\end{aligned} \tag{5.94}$$

appear. Of course, all of them can be analytically integrated with respect to Λ, justifying the aforementioned argument.

5.4.3 Final formulas

Summarizing, to compute the original hyper-singular integral 5.5 in the vertex-adjacent case, we only need to perform the two 3-D numerical integrals in (5.89), where the integrand function Ω has been previously computed via a symbolic software like *Maple* utilizing the expressions in equations (5.91) and (5.92).

5.5 Numerical results

In this section, we present some numerical results for the assessment of the proposed method in terms of accuracy and efficiency, particularly for the edge-adjacent case, which is the most challenging. More specifically, it would be sufficient to analyze a single, albeit representative, case among the various combinations of basis and testing functions where all the terms in (5.29) are present before the first analytical integration. The other possible combinations do not have any particularity which could change the behavior of the results.

To start with, following the notation in Fig. 5.2, we consider two edge adjacent triangles (T1:$\{\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3\}$ and T2:$\{\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_4\}$) with the following vertices (see Fig. 5.11):

$$\begin{aligned}\mathbf{r}_1 &= [0,0,0]; \quad \mathbf{r}_2 = [0,0.1\lambda,0] \\ \mathbf{r}_3 &= [0,0,0.1\lambda]; \quad \mathbf{r}_4 = [0.1\lambda,0,0]\end{aligned} \tag{5.95}$$

where $\lambda = 1$ [m] corresponds to the wavelength. Note that $\mathbf{r}_1$ and $\mathbf{r}_2$ are the two common vertices.

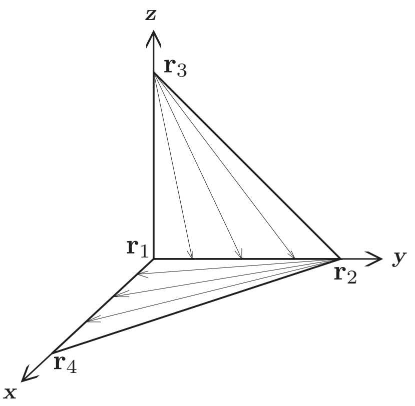


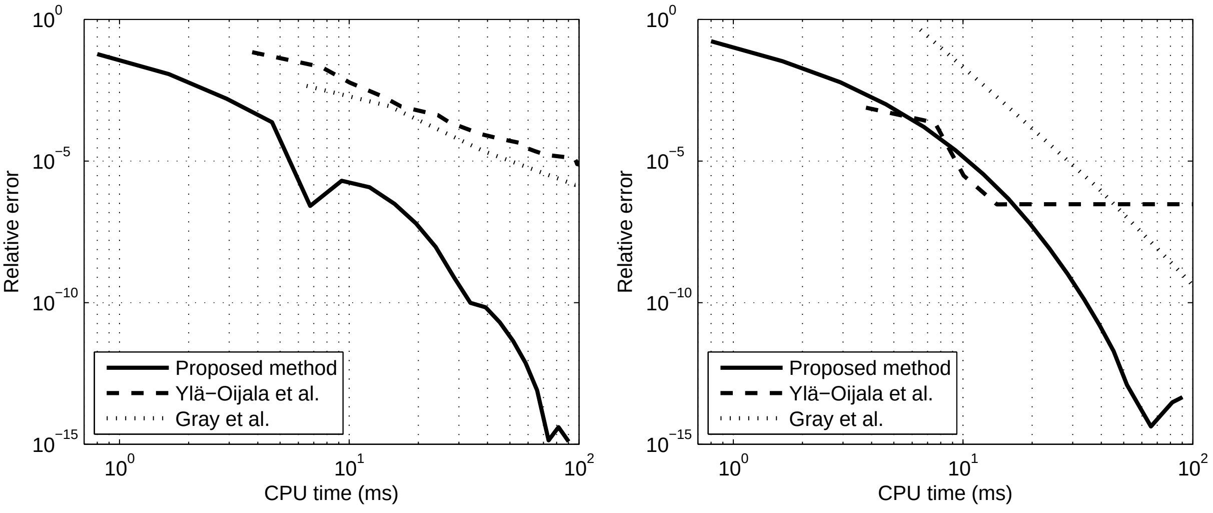
Figure 5.11: First case under study and the most representative one.

The actual computed integral is

$$I = \int_{E_P} \mathbf{f}_3(\mathbf{r}) \cdot \int_{E_Q} \nabla G(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_1'(\mathbf{r}')\, dS' dS \tag{5.96}$$

where $\mathbf{f}_3$ and $\mathbf{f}_1^{'}$ are the RWG functions

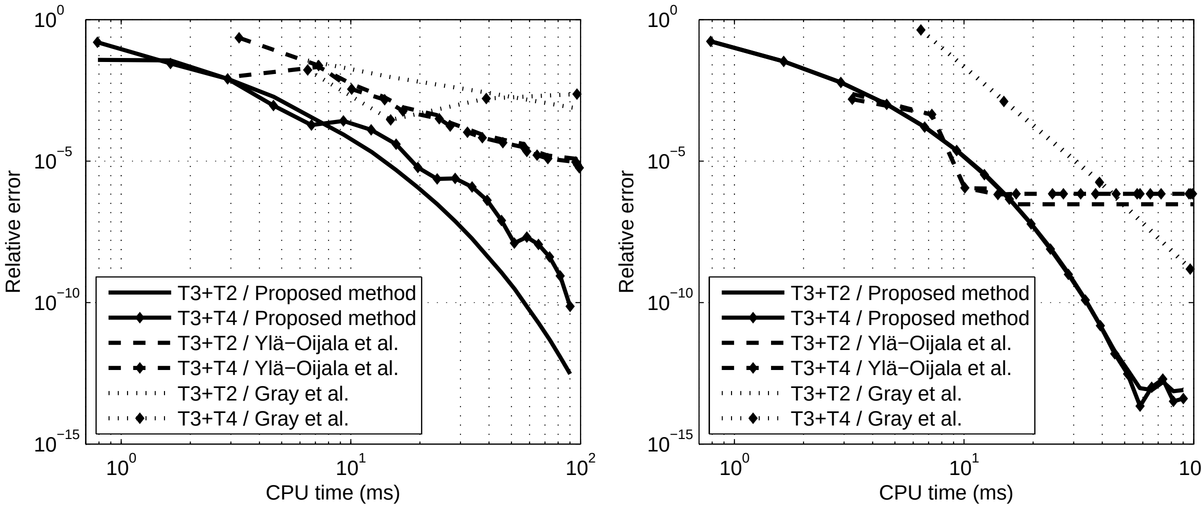
$$\begin{aligned} \mathbf{f}_3(\mathbf{r}) &= \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{2A_p}(\mathbf{r} - \mathbf{r}_3) \\ \mathbf{f}_1^{'}(\mathbf{r}') &= \frac{|\mathbf{r}_4 - \mathbf{r}_2|}{2A_q}(\mathbf{r}' - \mathbf{r}_2). \end{aligned} \tag{5.97}$$



(a) Real part.

(b) Imaginary part.

Figure 5.12: Comparison of the relative error as a function of the CPU time for the triangles T1 and T2.



(a) Real part.

(b) Imaginary part.

Figure 5.13: Comparison of the relative error as a function of the CPU time for the distorted triangles.

In Figs. 5.12a and 5.12b, the relative error

$$\varepsilon = \frac{|I - I_{\text{ref}}|}{|I_{\text{ref}}|} \tag{5.98}$$

in calculating the real part (hyper-singular portion) and the imaginary part (weakly singular portion) of the hyper-singular integral (5.96) in terms of the CPU time is presented. For the sake of a fair comparison, we selected two of the most proven methods available in the literature, i.e., the singularity subtraction method presented by Ylä-Oijala and Taskinen in [29] and the direct evaluation method applied over the static integral, again in a singularity subtraction fashion, presented by Gray et al. [83]. Without entering into much detail, the solution from the aforementioned singularity subtraction method is obtained with the subtraction of the first two terms of the Taylor expansion of the free-space Green's function G

$$\nabla G = \nabla \left(G - \frac{1}{R} + \frac{k^2}{2} R \right) + \nabla \frac{1}{R} - \frac{k^2}{2} \nabla R. \tag{5.99}$$

The integrals related to the last two terms of the expression above can be computed semi - analytically, where the remaining numerical integrals apply only to smooth functions. The first term has a continuous derivative and, therefore, it can be integrated numerically as well. The choice of removing two terms is somehow optimum as there is no real improvement after removing more terms. The number of integration points for the singularity subtraction method is fixed for the inner integral and a Gaussian quadrature tailored for triangles [85] is used for the outer integral.

Moreover, the reference solution has been obtained by means of the proposed method in combination with the symbolic software *Maple*, using high precision arithmetic (300 digits) in all the computations together with a high number of integration points until it converges smoothly up to any desired accuracy. The exact value up to 32 significant digits equals

$$\begin{aligned} I = {} & 3.4928883683897266018383577695620 \cdot 10^{-3} \\ & - j2.2540732129690316163209769145458 \cdot 10^{-5}. \end{aligned} \tag{5.100}$$

According to the results, as depicted in Fig. 5.12, the proposed method converges up to machine precision, even for the hyper-singular portion, much faster than the rest. To the best of our knowledge, there is no method readily available in the literature that could provide better accuracy in reasonable computational times. The behavior of the proposed method clearly reveals one of its key features, i.e., the remaining integrands after analytical integrations and re-ordering are sufficiently smooth functions. Hence, simple interpolatory quadratures, like the standard Gauss-Legendre, are sufficient even for highly accurate results.

This is not the case, though, for the other competing methods, where the remaining integrands after the cancellation or the subtraction of the singularity are non sufficiently smooth functions of the outer (observation) triangle's arguments. The accurate integration of such functions calls for sophisticated 2-D cubatures, like the recently developed double-exponential based generalized Cartesian product rules [86], still with considerable computational overhead. This problem would therefore arise in any of the traditional singularity cancellation or singularity subtraction methods mentioned in the introduction, considering that all of them deal separately with inner and outer integrals.

As for the imaginary part, it seems that the singularity subtraction method stagnates at a relative error of 10^{-7}. A fair explanation of this behavior is based on the fact that the fixed number of integration points used for the inner integral is not sufficient for high precision and it should be increased, deteriorating even more its overall efficiency.

Although the example presented above stands as a representative case in the vast majority of the state-of-the-art triangular meshing schemes, two additional extreme cases have been selected in order to highlight the robustness of the proposed method against deformed triangles with bad aspect ratio or quality. One commonly used quality measure for such "distorted" triangles is the ratio between the radius of the largest inscribed circle (times two) and the smallest circumscribed circle:

$$q = 2\frac{r_{\text{in}}}{r_{\text{out}}}. \tag{5.101}$$

According to [87], a triangle with quality factor q lower than 0.7 or a minimum angle lower than 30 degrees is unacceptable in a mesh, and there are techniques to convert these bad meshes into others, fulfilling those requirements [88, 89, 90]. Therefore, we have selected two triangles T3:$\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_5\}$ and T4:$\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_6\}$ which are below this limit (quality factor of 0.46 and minimum angle of 30 degrees), i.e.,

$$\mathbf{r}_5 = [-0.05\lambda, 0, 0.087\lambda]; \quad \mathbf{r}_6 = [-0.05\lambda, 0.087\lambda, 0]. \tag{5.102}$$

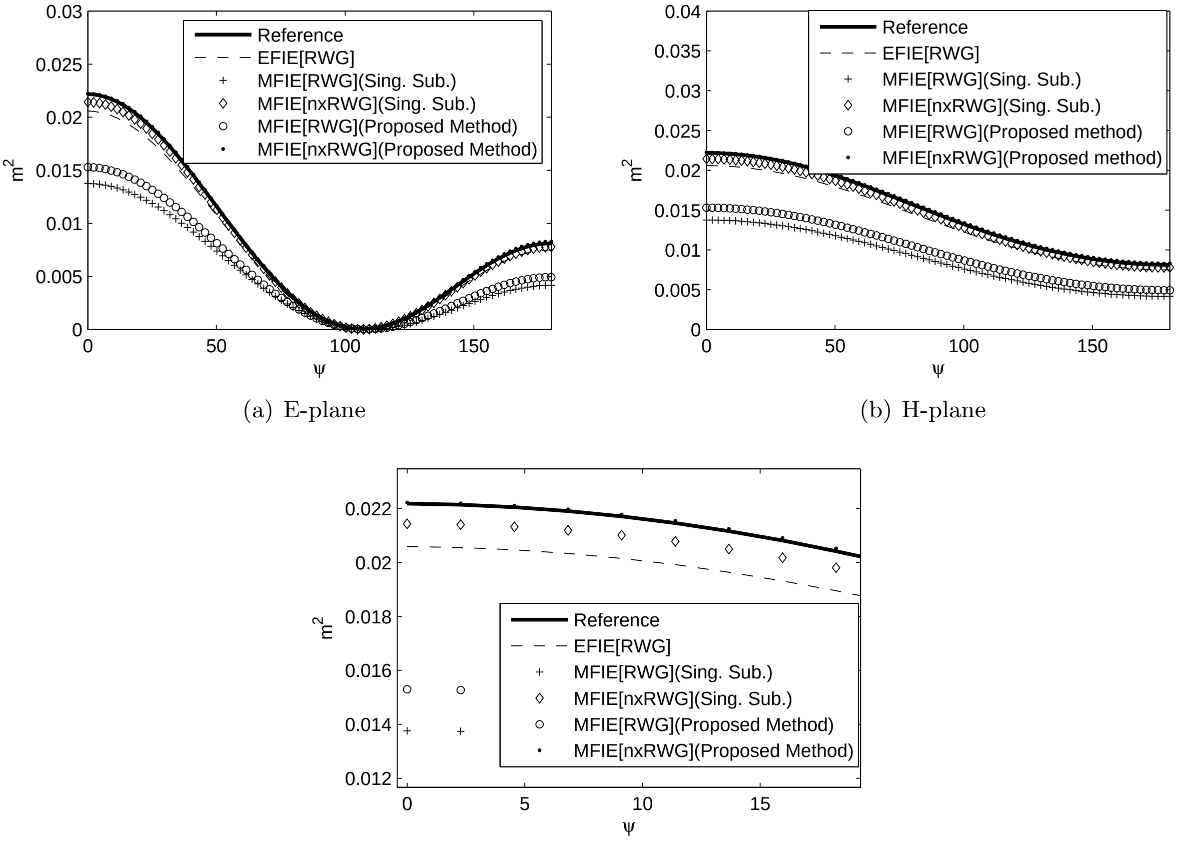
The results for the combinations of triangles T3+T2 and T3+T4 are depicted in Fig. 5.13. The reference solutions have been obtained in the same way as the previous case and they equal, respectively,

$$I = 3.1419955732525062504931041862416 \cdot 10^{-3} \\ - j1.9600239487556817889309129166796 \cdot 10^{-5} \tag{5.103}$$

$$I = 3.5226217019446727628139765841737 \cdot 10^{-3} \\ - j1.7135151374120059440368032702308 \cdot 10^{-5}. \tag{5.104}$$

Although convergence slightly deteriorates for the real part, the proposed method is still able to practically reach machine precision in less than 0.1s and the same conclusions as in the previous, more representative case can be drawn.

Evidently, the regularization of the hyper-singular kernel together with the reduction of the dimensionality of the original integral from 4-D to 2-D via the direct evaluation method has resulted in formulas which provide *numerically exact* results (more than 13 significant digits for the relative error) with significantly reduced computational effort. In particular, using codes implemented in MATLAB® and run in a computer with an Intel® Core 2 Quad, 2.83 GHz (no parallelization has been done), Linux 2.6.28 Ubuntu and MATLAB® 7.7.0.471, the proposed scheme reaches a relative error smaller than 10^{-5} in 10ms and almost machine precision in about 100ms.



(a) E-plane

(b) H-plane

(c) E-plane zoomed in

Figure 5.14: Bi-static Radar Cross Section (RCS) of a 0.25λ regular tetrahedron .

Finally, a relevant example is presented which shows the importance of leading an accurate integration of the Method of Moments matrix elements. More specifically, an electrically small regular tetrahedron with side 0.25λ has been chosen and modeled with a very coarse regular mesh of $N = 150$ edges or basis functions, where all the triangles are equilateral. The reference value has been obtained with a very fine mesh ($N = 23436$ edges), accumulating many triangular elements close to the tetrahedron edges and solving with EFIE formulation. The following five simulations have been performed for the coarse mesh: 1) Standard EFIE, 2), 3) MFIE with RWG and $\hat{\mathbf{n}}\times$RWG basis functions, respectively, together with standard singularity subtraction approach, 4), 5) MFIE with RWG and $\hat{\mathbf{n}}\times$RWG basis functions, respectively, with the edge adjacent interactions computed with the proposed method. Note, that using $\hat{\mathbf{n}}\times$RWG basis functions imply the same singular integrals [91, 92]. For the singularity subtraction, four integration points per triangle were used. Fig. 5.14 shows the bi-static RCS for all the cases. Clearly, MFIE with RWG basis functions has the worst performance, although there is a substantial gain when the edge-adjacent integration is done accurately. In this case there is an important error inherent to the formulation itself. However, when using MFIE with $\hat{\mathbf{n}}\times$RWG basis functions, the performance is even better than with EFIE for the same mesh and an excellent result, without computational overhead, is obtained if the integration is accurately done with the proposed method, as highlighted when zooming in (see Fig. 5.14(c)). Moreover, the fast and accurate evaluation of the matrix elements, allows the proper study of the other (of different nature) well-known problems appearing in MFIE formulations.

5.6 Conclusion

In this chapter, the direct evaluation method is developed for the computation of the hypersingular integrals arising in Galerkin surface integral equation formulations. The key feature of the proposed scheme lies in the appropriate regularization of the singular integrand together with the reduction of the dimensionality of the original integral from 4-D to 2-D. The final formulas presented herein succeed in providing numerical results of unmatched accuracy (close to the machine precision) and efficiency, thus improving substantially the accuracy of the impedance matrix elements in field integral equation formulations as well as reducing the overall filling time. In addition, we hope that with the detailed analysis and the ready-to-use form of the final formulas, the direct evaluation method will find its place in standard mathematical subroutine libraries widely used in the computational electromagnetics community.

Another important remark after the whole study can be extracted:

- Coincident triangles: A singularity $1/R$ can be canceled out and three out of four integrals can be analytically integrated.

- Edge Adjacent: A singularity $1/R^2$ can be canceled out and two out of four integrals can be analytically integrated.

- Vertex adjacent: A singularity $1/R^3$ can be canceled out and one out of four integrals can be analytically integrated.

Fortunately, the integration over coincident triangles in the MFIE formulation is directly zero because its integrand is zero and it is unnecessary to tackle the $1/R^2$ singularity in this case.

However, it sets a challenge for the near-singular MFIE case, i.e., when the two triangles have a part of their surface very close to each other but not touching.

Chapter 6

On the numerical integration of singular integrals

This chapter includes a series of aspects which arise when dealing with singular integrals, and in particular the hyper-singular integrals coming from Galerkin surface integral equation formulations treated in the previous chapter. Specifically, the direct evaluation method is tackled when no analytical integrations are done and the effect of the re-ordering in the numerical integration is investigated. Furthermore, the singular behavior of the field, i.e., the integrand of the outer integral, even when the $1/R^2$ singularity has been canceled out, is highlighted, including the development of a set of solutions.

6.1 Analysis of numerical integration in the direct evaluation method

In this section, some aspects of the direct evaluation method for the computation of hyper-singular 4-D integrals, arising in the Galerkin discretization of surface integral formulations, are analyzed. In particular the edge adjacent case will be our test case. In the previous chapter, utilizing a series of coordinate transformations combined with a re-ordering of the integrations, the integrand not only was regularized but also it was casted in a convenient form in order to perform two of the integrals analytically. Here, we are going to test and compare the behavior when only one or none of the dimensions is done analytically. Furthermore, the importance of the re-ordering will be outlined, even when no analytical integration is performed.

For the sake of clarity and completeness in the chapter, some points regarding hyper-singular integrals and the direct evaluation method will be briefly reviewed in the next two subsections.

6.1.1 Hyper-singular integrals

The hyper-singular integrals appearing in MFIE, and therefore in CFIE formulations using RWG basis functions, can be reduced to integrals of the form

$$I := \int_{E_P} \mathbf{g}(\mathbf{r}) \cdot \int_{E_Q} \nabla G(\mathbf{r}, \mathbf{r}') \times \mathbf{f}(\mathbf{r}') \, dS' dS \tag{6.1}$$

where the function $\mathbf{f}(\mathbf{r})$ is a RWG basis function and $\mathbf{g}(\mathbf{r})$ can be either a RWG or a $\hat{\mathbf{n}}\times$RWG testing function defined over the triangles E_Q and E_P, respectively. $G(\mathbf{r}, \mathbf{r}')$ corresponds to the free space Green's function $G(R) = e^{-jkR}/R$, depending on the distance $R = |\mathbf{r} - \mathbf{r}'|$.

In particular, the case where the two triangles have a common edge (see Fig. 6.1 on the left), which is actually the most challenging one, is in order.

6.1.2 Direct evaluation method

The main idea of the direct evaluation method is to cancel out the singularity $1/R^2$ of the integrand, performing a set of parameter transformations. Following the procedure described in the last chapter and references therein, we start by moving to equilateral parameter spaces $\{\eta, \xi\}$ and $\{\eta', \xi'\}$ in each triangle (see Fig. 6.1):

$$\begin{aligned}
\mathbf{r} &= \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_2-x_1}{2} & \frac{2x_3-x_1-x_2}{2\sqrt{3}} \\ \frac{y_2-y_1}{2} & \frac{2y_3-y_1-y_2}{2\sqrt{3}} \\ \frac{z_2-z_1}{2} & \frac{2z_3-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix} \\
\mathbf{r}' &= \begin{bmatrix} \frac{x_2+x_1}{2} \\ \frac{y_2+y_1}{2} \\ \frac{z_2+z_1}{2} \end{bmatrix} + \begin{bmatrix} \frac{x_1-x_2}{2} & \frac{2x_4-x_1-x_2}{2\sqrt{3}} \\ \frac{y_1-y_2}{2} & \frac{2y_4-y_1-y_2}{2\sqrt{3}} \\ \frac{z_1-z_2}{2} & \frac{2z_4-z_1-z_2}{2\sqrt{3}} \end{bmatrix} \begin{bmatrix} \eta' \\ \xi' \end{bmatrix}
\end{aligned} \tag{6.2}$$

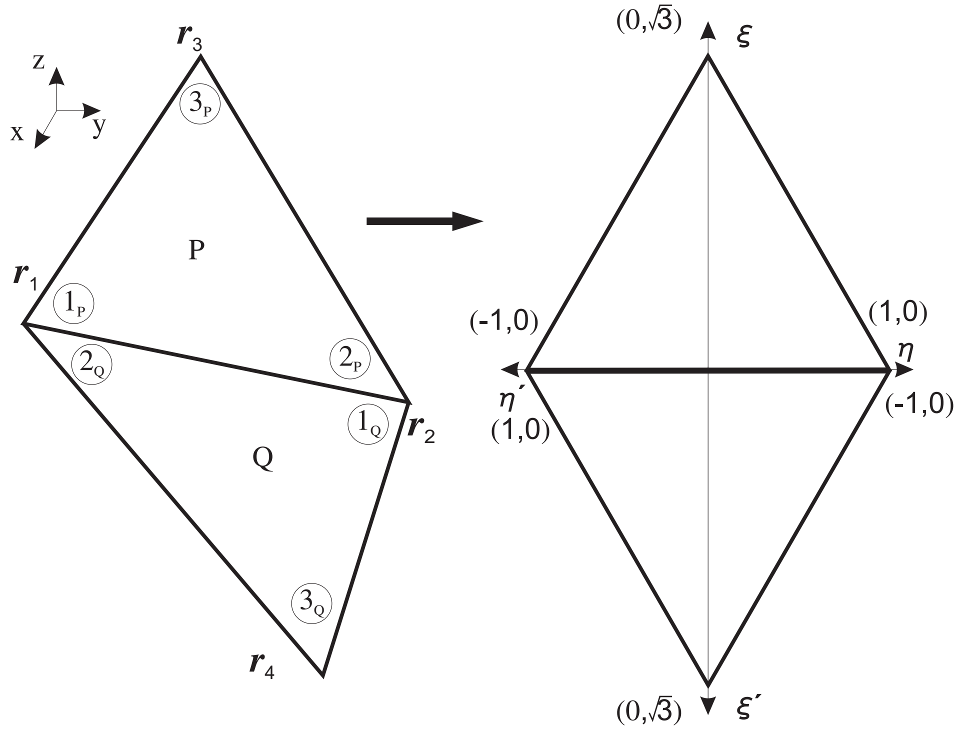


Figure 6.1: Geometry of the triangular elements both in the original and the equilateral triangle parameter space.

Subsequently, two further consecutive polar coordinates changes are carried out:

$$\begin{aligned} \eta' &= \rho\cos(\theta) - \eta \\ \xi' &= \rho\sin(\theta) \end{aligned} \quad \text{and} \quad \begin{aligned} \rho &= \Lambda\cos(\Psi) \\ \xi &= \Lambda\sin(\Psi). \end{aligned} \tag{6.3}$$

Considering that the distance function R becomes proportional to Λ, the singularity, now placed at $\Lambda = 0$, is removed thanks to the global Jacobian

$$J = \frac{A_p A_q}{3}\Lambda^2 \cos(\Psi) \tag{6.4}$$

yielding after some manipulations:

$$\begin{aligned} I = &\int_0^1 d\eta \int_0^{\Theta_1} d\theta \int_0^{\Psi_C} d\Psi \int_0^{\Lambda_{L_3}} F(\Lambda, \eta, \Psi, \theta)\, d\Lambda \\ &+ \int_0^1 d\eta \int_0^{\Theta_1} d\theta \int_{\Psi_C}^{\pi/2} d\Psi \int_0^{\Lambda_{L_2}} F(\Lambda, \eta, \Psi, \theta)\, d\Lambda \\ &+ \int_0^1 d\eta \int_{\Theta_1}^{\pi} d\theta \int_0^{\Psi_A} d\Psi \int_0^{\Lambda_{L_1}} F(\Lambda, \eta, \Psi, \theta)\, d\Lambda \\ &+ \int_0^1 d\eta \int_{\Theta_1}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} d\Psi \int_0^{\Lambda_{L_2}} F(\Lambda, \eta, \Psi, \theta)\, d\Lambda \end{aligned} \tag{6.5}$$

where the transformed integrand F is smooth, with no singularities, and can be computed, in view of a symmetry at $\eta = 0$ in the equilateral parameter space, as

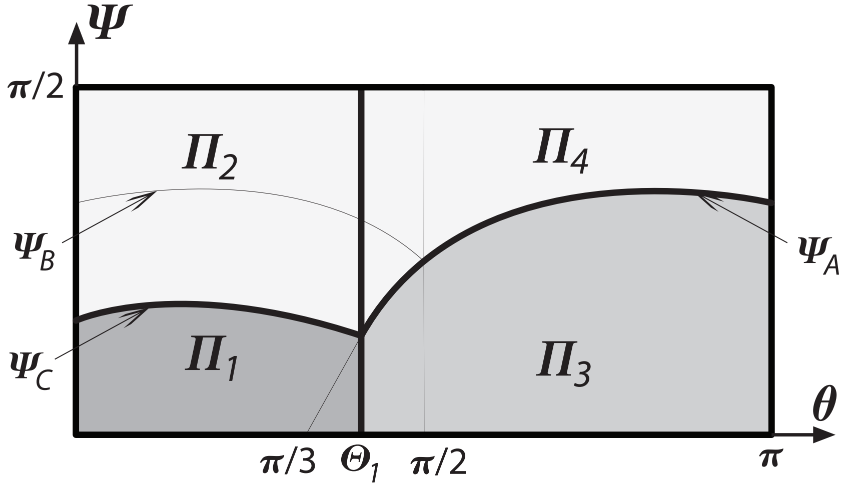


Figure 6.2: Integration domains in eq. (6.5).

$$\begin{aligned} F(\Lambda, \eta, \Psi, \theta) =& F^+(\Lambda, \eta, \Psi, \theta) + F^+(\Lambda, -\eta, \Psi, \pi - \theta) \\ F^+(\Lambda, \eta, \Psi, \theta) =& \frac{A_p A_q}{3} \cos(\Psi) \left(\frac{e^{-jk\Lambda B(\theta,\Psi)}}{B(\theta, \Psi)^3} (1 + jk\Lambda B(\theta, \Psi)) \right) \\ & \mathbf{g}(\Lambda, \eta, \theta, \Psi) \cdot (\mathbf{f}(\Lambda, \eta, \theta, \Psi) \times \mathbf{B}(\theta, \Psi)) \end{aligned} \tag{6.6}$$

being $\mathbf{B}$ the proportionality constant, in terms of Λ, of the distance vector function $\mathbf{R} = \Lambda \mathbf{B}(\theta, \Psi)$:

$$\mathbf{B} = \boldsymbol{\alpha}_{e_2} \sin(\Psi) + \boldsymbol{\alpha}_{e_1} \cos(\Psi) \cos(\theta) + \boldsymbol{\alpha}_{e_3} \cos(\Psi) \sin(\theta). \tag{6.7}$$

In the last expression, $\boldsymbol{\alpha}_{e_i}$ are constant vectors which only depend on the triangles' vertices:

$$\begin{aligned}
\boldsymbol{\alpha}_{e_1} &= \frac{\mathbf{r}_2 - \mathbf{r}_1}{2} \\
\boldsymbol{\alpha}_{e_2} &= \frac{2\mathbf{r}_3 - \mathbf{r}_1 - \mathbf{r}_2}{2\sqrt{3}} \\
\boldsymbol{\alpha}_{e_3} &= -\frac{2\mathbf{r}_4 - \mathbf{r}_1 - \mathbf{r}_2}{2\sqrt{3}}.
\end{aligned} \tag{6.8}$$

The set of integration limits in expression (6.5) equal

$$\begin{aligned}
&\Theta_1(\eta) = \frac{\pi}{2} - \tan^{-1}\left(\frac{\eta}{\sqrt{3}}\right) \\
&\Psi_A = \tan^{-1}\left(\sin(\theta) - \sqrt{3}\cos(\theta)\right) \\
&\Psi_B = \tan^{-1}\left(\sin(\theta) + \sqrt{3}\cos(\theta)\right) \\
&\Psi_C = \tan^{-1}\left(\frac{1-\eta}{1+\eta}\left(\sin(\theta) + \sqrt{3}\cos(\theta)\right)\right) \\
&\Lambda_{L_1} = \frac{\sqrt{3}(1-\eta)}{\cos(\Psi)(\sin(\theta) - \sqrt{3}\cos(\theta))} \\
&\Lambda_{L_2} = \frac{\xi(\eta)}{\sin(\Psi)} = \frac{\sqrt{3}(1-\eta)}{\sin(\Psi)} \\
&\Lambda_{L_3} = \frac{\sqrt{3}(1+\eta)}{\cos(\Psi)(\sin(\theta) + \sqrt{3}\cos(\theta))}.
\end{aligned} \tag{6.9}$$

To simplify the notation we introduce a new function including the integration with respect to Λ:

$$A(\Lambda, \eta, \Psi, \theta) = \int_0^{\Lambda} F(\Lambda', \eta, \Psi, \theta) d\Lambda'. \tag{6.10}$$

This integral can be performed either analytically or numerically. Indeed, numerical experiments show that six integration points suffice to obtain machine precision.

6.1.3 Analysis of the integration domains

The original integral has been subdivided, using (6.5), into four integrals over the domains Π_i presented in Fig. 6.2. Though it is a 2-D cut representation, it should be understood as a four-dimensional domain, varying the shape of the domains when η sweeps from zero to one, due to the η-dependence of the curves Ψ_C and Θ_1. The fourth dimension is evidenced with Λ going from zero to Λ_{L_i}, where i is determined by the gray tone in the figure, being Λ_{L_2} the lightest, Λ_{L_1} the one in the middle and Λ_{L_3} the darkest.

Notwithstanding, these domains can be split to have more regular domains, so that integration re-ordering can be immediately applied. The new set of integration domains, nine to be more specific, is represented in Fig. 6.3 (2-D cut) and Fig. 6.4 (3-D cut, including η-dependence). In this exploded figure, the adjacent 3D blocks are depicted separated in space to better show their shape. Clearly, the union of all the domains Ω_i covers exactly the cuboid $(\theta, \Psi, \eta) \in [0, \pi] \times [0, \pi/2] \times [0, 1]$.

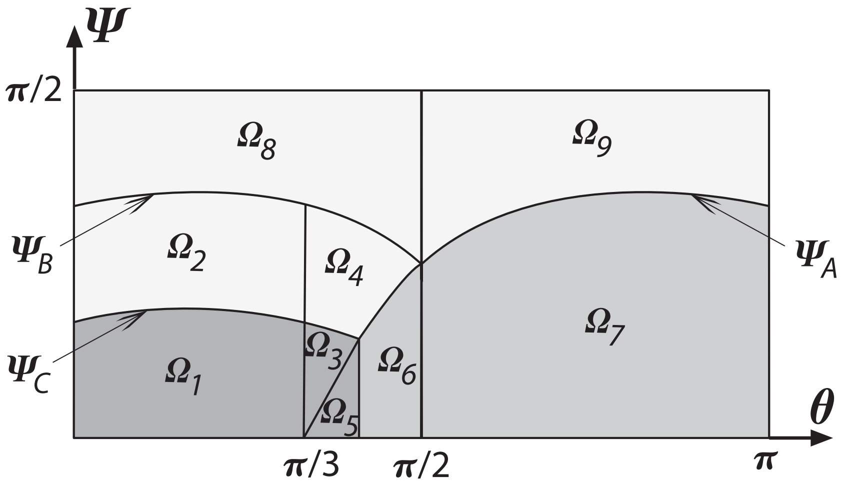


Figure 6.3: Integration domains after splitting in nine integrals, for some fixed $0 < \eta < 1$.

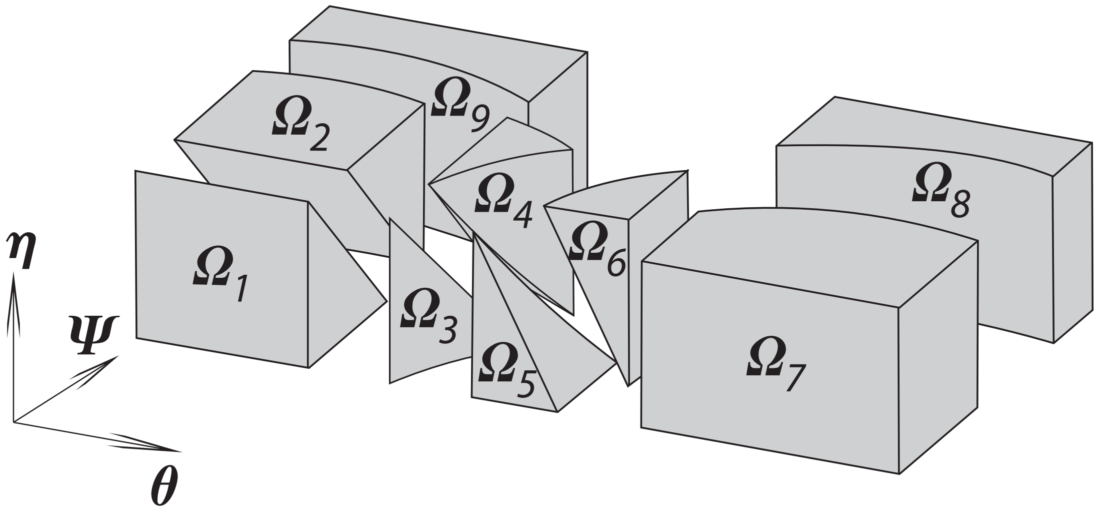


Figure 6.4: Integration domains after splitting in nine integrals.

The integral (6.5) is then computed as

$$I = \sum_{i=1}^{9} I^{(i)} \tag{6.11}$$

where the constituent integrals equal

$$\begin{aligned} I^{(1)} &= \int_0^1 d\eta \int_0^{\pi/3} d\theta \int_0^{\Psi_C} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\Psi \\ &= \int_0^{\pi/3} d\theta \int_0^{\Psi_B} d\Psi \int_0^{\Psi_\eta} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.12}$$

$$\begin{aligned} I^{(2)} &= \int_0^1 d\eta \int_0^{\pi/3} d\theta \int_{\Psi_C}^{\Psi_B} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &= \int_0^{\pi/3} d\theta \int_0^{\Psi_B} d\Psi \int_{\Psi_\eta}^{1} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.13}$$

$$\begin{aligned} I^{(3)} &= \int_0^1 d\eta \int_{\pi/3}^{\Theta_1} d\theta \int_{\Psi_A}^{\Psi_C} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} d\Psi \int_0^{\Psi_\eta} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.14}$$

$$\begin{aligned} I^{(4)} &= \int_0^1 d\eta \int_{\pi/3}^{\Theta_1} d\theta \int_{\Psi_C}^{\Psi_B} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &\quad + \int_0^1 d\eta \int_{\Theta_1}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/3}^{\pi/2} d\theta \int_{\Psi_A}^{\Psi_B} d\Psi \int_{\Psi_\eta}^{1} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.15}$$

$$\begin{aligned} I^{(5)} &= \int_0^1 d\eta \int_{\pi/3}^{\Theta_1} d\theta \int_0^{\Psi_A} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} d\Psi \int_0^{\eta_\theta} A(\Lambda_{L_3}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.16}$$

$$\begin{aligned} I^{(6)} &= \int_0^1 d\eta \int_{\Theta_1}^{\pi/2} d\theta \int_0^{\Psi_A} A(\Lambda_{L_1}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/3}^{\pi/2} d\theta \int_0^{\Psi_A} d\Psi \int_{\eta_\theta}^{1} A(\Lambda_{L_1}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.17}$$

$$\begin{aligned} I^{(7)} &= \int_0^1 d\eta \int_{\pi/2}^{\pi} d\theta \int_0^{\Psi_A} A(\Lambda_{L_1}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/2}^{\pi} d\theta \int_0^{\Psi_A} d\Psi \int_0^1 A(\Lambda_{L_1}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.18}$$

$$\begin{aligned} I^{(8)} &= \int_0^1 d\eta \int_0^{\pi/2} d\theta \int_{\Psi_B}^{\pi/2} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &= \int_0^{\pi/2} d\theta \int_{\Psi_B}^{\pi/2} d\Psi \int_0^1 A(\Lambda_{L_2}, \eta, \Psi, \theta) d\eta \end{aligned} \tag{6.19}$$

$$\begin{aligned} I^{(9)} &= \int_0^1 d\eta \int_{\pi/2}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &= \int_{\pi/2}^{\pi} d\theta \int_{\Psi_A}^{\pi/2} d\Psi \int_0^1 A(\Lambda_{L_2}, \eta, \Psi, \theta) d\eta. \end{aligned} \tag{6.20}$$

Each integral $I^{(i)}$ in (6.12)-(6.20) is displayed with two different integration orders to cover the domain Ω_i, the original one $(\eta, \theta, \Psi, \Lambda)$ and the one bringing η integration right after Λ's one, $(\theta, \Psi, \eta, \Lambda)$. The latter was tackled in the previous chapter because the integrand function $A(\Lambda, \eta, \Psi, \theta)$ is analytically integrable with respect to η, allowing us to perform two analytical integrations and only two dimensions numerically, θ and Ψ. However, we are going to analyze the effect of the integration order when the four dimensions are integrated numerically using generalized Cartesian product rules based on standard Gauss quadratures. There are two main reasons to attempt to do everything numerically: the theoretical knowledge we obtain from this and the increased simplicity of the implementation. Of course, the best option would be to obtain an optimal set of weights and integration points adapted to these particular 3-D subdomains, but this still remains as a task for future investigation. In Section 6.1.4, we will see that the re-ordered one is close to this optimal even performing classical 1-D Gauss integrations, whereas with the original order it fails to converge in a reasonable time in certain cases.

When the domains have a shape close to a cuboid, as is the case of Ω_7, Ω_8 and Ω_9, the re-ordering has almost no effect. Contrarily, when the shape changes to something close to a pyramid, the integration points are distributed over the domain differently depending on the order. In our case, it is numerically proved that it is important to make η-integration after Λ-integration. It assures that for each taken angles θ and Ψ, we have N_{1D} integration points in η, although they are accumulated in a corner. From a theoretical point of view, an accurate inspection of the expression of the integrand shows a smoother behavior for θ and Ψ parameters than for η, which is polynomial apart from the complex exponential factors. Furthermore, the non-smooth η-dependence of the integration limits for θ and Ψ in the non re-ordered case provokes a loss in the smoothness of the integrand function with respect to η as showed in Section 6.1.4.

6.1.4 Numerical Results

6.1.4.1 Case under study

Without loss of generality, we are going to consider the representative integral case:

$$I = \int_{E_P} \mathbf{f}_3(\mathbf{r}) \cdot \int_{E_Q} \nabla G(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_1'(\mathbf{r}')\, dS' dS \tag{6.21}$$

where $\mathbf{f}_3$ and $\mathbf{f}_1^{'}$ are, following the notation in Fig. 6.1, the RWG functions

$$\mathbf{f}_3(\mathbf{r}) = \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{2A_p}(\mathbf{r} - \mathbf{r}_3); \quad \mathbf{f}_1^{'}(\mathbf{r}') = \frac{|\mathbf{r}_4 - \mathbf{r}_2|}{2A_q}(\mathbf{r}' - \mathbf{r}_2) \tag{6.22}$$

which after applying the coordinate changes described in Section 6.1.2 can be expressed as

$$\begin{aligned} \mathbf{f}_3(\Lambda, \eta, \Psi, \theta) &= \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{2A_p}\left(-\sqrt{3}\alpha_{e_2} + \alpha_{e_1}\eta + \alpha_{e_2}\sin(\Psi)\Lambda\right) \\ \mathbf{f}_1'(\Lambda, \eta, \Psi, \theta) &= \frac{|\mathbf{r}_4 - \mathbf{r}_1|}{2A_q}\left(-\alpha_{e_1} + \alpha_{e_1}\eta - (\alpha_{e_1}\cos(\theta) + \alpha_{e_3}\sin(\theta))\cos(\Psi)\Lambda\right). \end{aligned} \tag{6.23}$$

Substituting (6.23) into (6.6), the integrand function is

$$\begin{aligned} F_{31}(\Lambda, \eta, \Psi, \theta) = {}& \frac{|\mathbf{r}_2 - \mathbf{r}_1||\mathbf{r}_4 - \mathbf{r}_1|}{12}(\alpha_{e_2} \cdot (\alpha_{e_1} \times \alpha_{e_3}))\cos^2(\Psi)\sin(\theta) \\ & \left[\frac{1}{(B^+)^3}(\sqrt{3}(1-\eta) - \sin(\Psi)\Lambda)(1 + jk\Lambda B^+)e^{-jk\Lambda B^+}\right. \\ & \left. + \frac{1}{(B^-)^3}(\sqrt{3}(1+\eta) - \sin(\Psi)\Lambda)(1 + jk\Lambda B^-)e^{-jk\Lambda B^-}\right] \end{aligned} \tag{6.24}$$

and applying (6.10) yields

$$
\begin{aligned}
A_{31}(\Lambda,\eta,\Psi,\theta) = \frac{|\mathbf{r}_2-\mathbf{r}_1||\mathbf{r}_4-\mathbf{r}_1|}{12}&(\alpha_{e_2}\cdot(\alpha_{e_1}\times\alpha_{e_3}))\cos^2(\Psi)\sin(\theta) \\
&\Bigg[\frac{1}{(B^+)^3}\left(\sqrt{3}(1-\eta)S_2^+ + (-\sin(\Psi)+\sqrt{3}(1-\eta)jkB^+)S_3^+ \;\; -\sin(\Psi)jkB^+S_4^+\right) \\
&+\frac{1}{(B^-)^3}\left(\sqrt{3}(1+\eta)S_2^- + (-\sin(\Psi)+\sqrt{3}(1+\eta)jkB^-)S_3^- \;\; -\sin(\Psi)jkB^-S_4^-\right)\Bigg]
\end{aligned} \tag{6.25}
$$

where

$$
\begin{aligned}
S_2 &:= \int_0^{\Lambda} e^{-jk\Lambda' B}d\Lambda' = \frac{1-e^{-jk\Lambda B}}{jkB} \\
S_3 &:= \int_0^{\Lambda} \Lambda' e^{-jk\Lambda' B}d\Lambda' = \frac{S_2-\Lambda e^{-jk\Lambda B}}{jkB} \\
S_4 &:= \int_0^{\Lambda} (\Lambda')^2 e^{-jk\Lambda' B}d\Lambda' = \frac{2S_3-\Lambda^2 e^{-jk\Lambda B}}{jkB}.
\end{aligned} \tag{6.26}
$$

We consider the triangles defined by the vertices:

$$
\begin{aligned}
\mathbf{r}_1 &= [0,0,0]; \quad \mathbf{r}_2 = [0,0.1\lambda,0] \\
\mathbf{r}_3 &= [0,0,0.1\lambda]; \quad \mathbf{r}_4 = [0.1\lambda,0,0].
\end{aligned} \tag{6.27}
$$

6.1.4.2 Results

We start isolating the error sources due to each one of the integration parameters. The justified solution presented in the previous chapter is taken as a reference. Using the re-ordered version, which we know works properly, we set the number of 1-D integration points of each variable at a sufficient value to have machine precision. Then, only one of the parameters varies its number of integration points to see the influence of this integration alone. Fig. 6.5 shows the results of this experiment. We can see how for the two inner integrals (Λ, η), just six integration points per dimension suffice to obtain machine precision. This is the part which can be analytically integrated. The other two parameters converge with a slightly lower rate.

Fig. 6.6 shows the required CPU times to get the different relative errors. The three first curves in the plot correspond to the re-ordered case doing two, three and four numerical integrations, respectively. In view of the result of the last paragraph, in the second and the third curves, the number of integration points for η and Λ has been set to six when integrated numerically. We can observe that the optimum case is when two dimensions are carried out analytically, as expected. The fourth curve corresponds to integrating four dimensions numerically but without applying the re-ordering, indicated as w-r. In this case, to assure convergence, also the number of integration points in η is gradually increased. The convergence without re-ordering is clearly worse than the re-ordered one.

As a representative case of the sub-integrals with a deteriorated behavior when no re-ordering

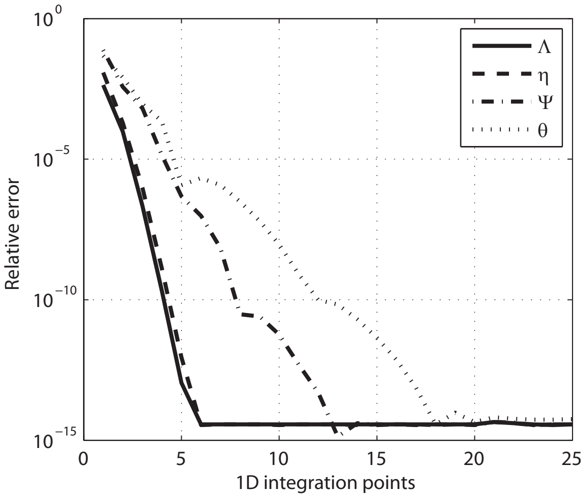


Figure 6.5: Relative error due to the number of integration points in each dimension avoiding the other error sources.

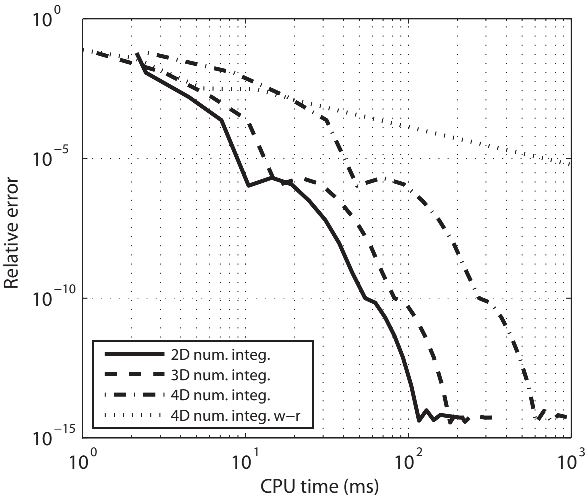


Figure 6.6: Relative error and CPU time relation for different approaches of the direct evaluation method.

is applied, the sub-integral $I^{(2)}$ is further analyzed:

$$\begin{aligned} I^{(2)} &= \int_0^1 d\eta \int_0^{\pi/3} d\theta \int_{\Psi_C}^{\Psi_B} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi \\ &= \int_0^{\pi/3} d\theta \int_0^{\Psi_B} d\Psi \int_{\Psi_\eta}^{1} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\eta. \end{aligned} \tag{6.28}$$

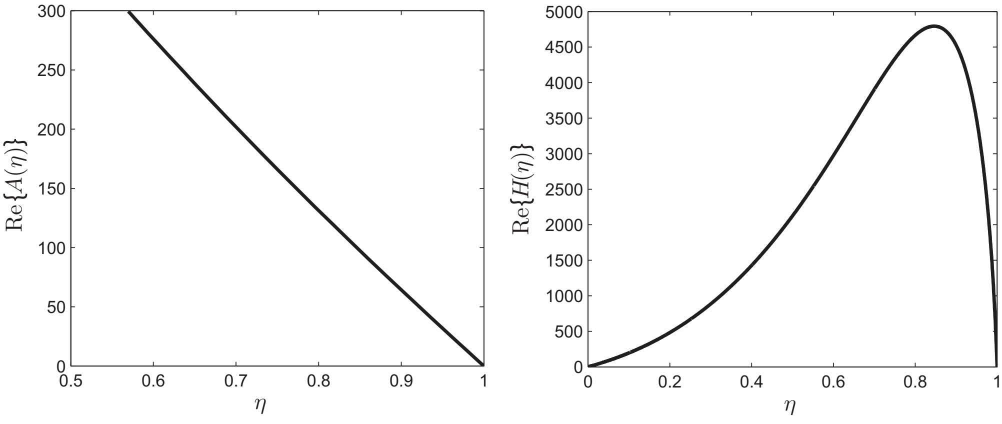
In the non re-ordered one, the integral can be grouped in an integration with respect to η as

$$I^{(2)} = \int_0^1 H(\eta) d\eta \tag{6.29}$$

where the integrand function $H(\eta)$ is:

$$H(\eta) = \int_0^{\pi/3} d\theta \int_{\Psi_C}^{\Psi_B} A(\Lambda_{L_2}, \eta, \Psi, \theta) d\Psi. \tag{6.30}$$

Fig. 6.7 displays the integrand function with respect to η before and after integrating with respect to θ and Ψ. In the figure on the left, the function can be approximated with a polynomial of very low degree considering its smoothness. This fact explains the excellent convergence of the integral in terms of the number of integration points when a Gauss quadrature rule is used in the re-ordered case. However, when the integrations with respect to θ and Ψ are held before η (figure on the right), the integrand becomes far from a low order polynomial, explaining the degradation of the convergence rate when no re-ordering is performed.



(a) With re-ordering for a fixed θ and Ψ.

(b) Without re-ordering.

Figure 6.7: Real part of the integrand with respect to η for the integral $I^{(2)}$.

6.1.5 Conclusions

In this section, an alternative implementation of the direct evaluation method for the computation of the hyper-singular 4-D integrals is introduced. Although two of the four integrals can be performed analytically, we have considered here the possibility of using standard numerical quadratures for three and even for the four dimensions. This has the advantage of resulting in easier and more compact implementations, especially in the case of a full 4-D numerical integration. For this last situation, the relevance of re-ordering the individual integrals has been clearly shown. Obviously, re-ordering is essential for bringing the analytical integrations to the front. But even when no analytical integration is attempted, re-ordering is a crucial step in the success of full 4-D numerical evaluations. As clearly demonstrated, a full 4-D numerical implementation of the direct evaluation method can achieve the same level of accuracy as hybrid analytical-numerical methods, with similar efficiency, if a proper re-ordering is applied.

As a connection to the next section, although the integrand function has been properly regularized, canceling out any singularity, the last integration can have a singular-like convergence due to the irregularity introduced by the previous integrations, inherent in their integration limits.

6.2 Singular behavior of the field when the observation point approaches the source domain

This section presents, based on relevant numerical examples, the relatively well-known fact that the magnetic field is singular when approaching the source triangle domain. In other words, even though the kernel might be regularized, the function after the inner integration has a logarithmic singular behavior. This observation leads to some interesting conclusions and problems which are treated in the next section.

In particular, the following function is considered

$$H(\mathbf{r}) := \int_{E_Q} \frac{e^{-jkR}}{R^2}\, dS' \tag{6.31}$$

which contains the most singular part of the ∇G kernel. Without loss of generality, the integration source triangle E_Q for the rest of the chapter will be $E_Q : \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_4\}$, included in the XY plane, where:

$$\begin{aligned} \mathbf{r}_1 &= [0,0,0]; \quad \mathbf{r}_2 = [0, 0.1\lambda, 0] \\ \mathbf{r}_3 &= [0,0,0.1\lambda]; \quad \mathbf{r}_4 = [0.1\lambda, 0, 0]. \end{aligned} \tag{6.32}$$

The function $H(\mathbf{r})$ has been accurately evaluated using the R^2 radial-angular transformation, a singularity cancellation method described in [22], although other techniques such as singularity subtraction would be possible as well.

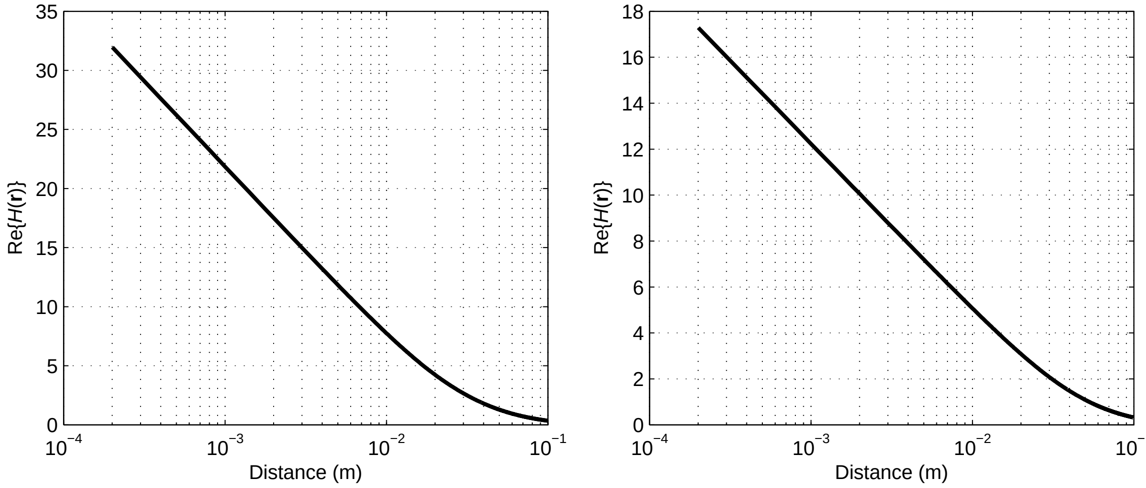
The first experiment consists in approaching the triangle along lines perpendicular to the triangle in two different points, the center of the triangle and the center of its edge in the y axis. The choice of the second point comes from our quest for the computation of the edge-adjacent or near-edge-adjacent cases. Fig. 6.8 shows the results for the real part of $H(\mathbf{r})$ in the two experiments. The imaginary part has been omitted due to its non-singular behavior. Clearly, the real part is proportional to the logarithm of the distance when $\mathbf{r}$ tends to the triangle

$$\mathrm{Re}\{H(\mathbf{r})\} = -k\log(d) \quad , d \to 0. \tag{6.33}$$

The proportionality constant k is a bit larger when approaching the triangle center than when approaching the edge.

The second experiment analyzes the behavior when passing the triangle parallelly. For this, the function $H(\mathbf{r})$ is computed in a large triangle parallel to E_Q but separated in the z-direction a distance 10^{-16}m. Fig. 6.9 displays this result, showing the logarithmic singularity which appears in the real part at the edges of the inner triangle. Therefore, some sophisticated techniques are necessary to numerically perform the outer integral when there are parts very close to the source triangle.

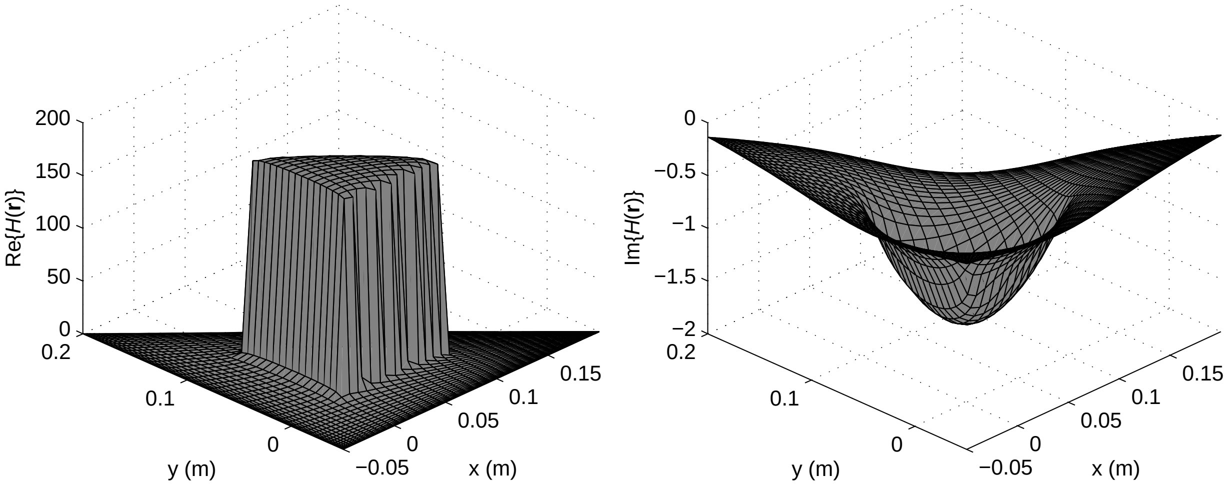
To further analyze how close to the triangle this sort of discontinuity appears another experiment is in order. $H(\mathbf{r})$ is evaluated along a line parallel to the x axis but regressively displaced in the z-direction. Fig. 6.10 and 6.11 show the real and imaginary parts, respectively, in film-like frames for the different z distances to the triangle. The imaginary part stabilizes and has a smooth shape close to the triangle edges. The real part, however, becomes sharper and sharper when the distance



(a) Center of the triangle.

(b) Center of the edge of the triangle.

Figure 6.8: Real part of $H(\mathbf{r})$ along a line perpendicular to the integration triangle at two different points inside the triangle.



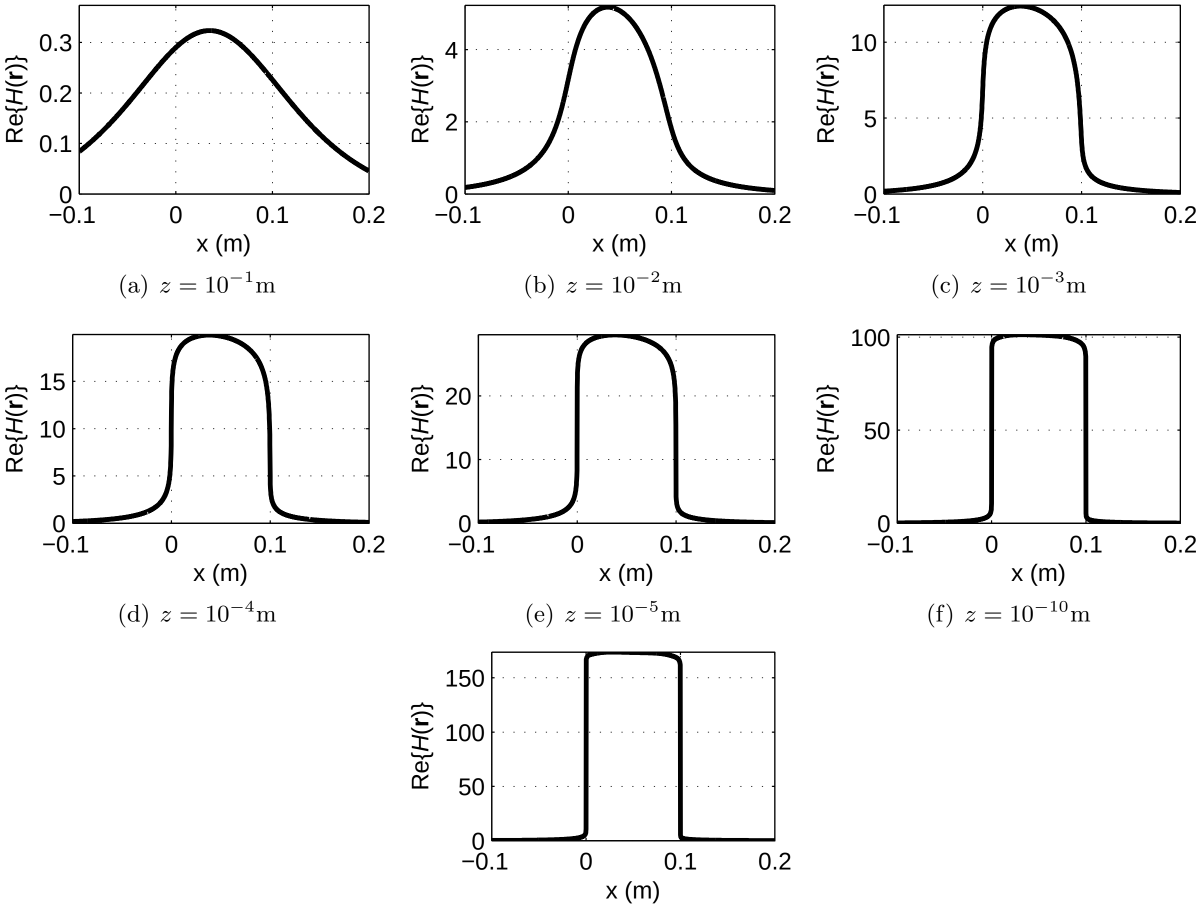
(a) Real Part.

(b) Imaginary Part.

Figure 6.9: Value of $H(\mathbf{r})$ sampling at a height very close ($z = 10^{-16}$m) to the integration triangle. The domain is larger than the original triangle to better see its effect when approaching the triangle edges.

to the triangle edge decreases. These real part results are summarized in Fig. 6.12.

Two important conclusions can be extracted from the previous results: If we were to compute the self-term, i.e., the outer integral over an observation triangle equal to the source one, or a triangle very close to it, some sophisticated numerical quadrature rules would be necessary to properly handle the singularity close to the edges; For the edge-adjacent case, the singularity appears not only in the direction which approaches the triangle perpendicularly but also in the direction parallel to the inner triangle when it is very close to the common edge. Notwithstanding, all the existing singularity cancellation and singularity subtraction methods in literature must handle this problem independently of the regularization of the kernel, which is valid for the inner integral alone.



(a) $z = 10^{-1}$m

(b) $z = 10^{-2}$m

(c) $z = 10^{-3}$m

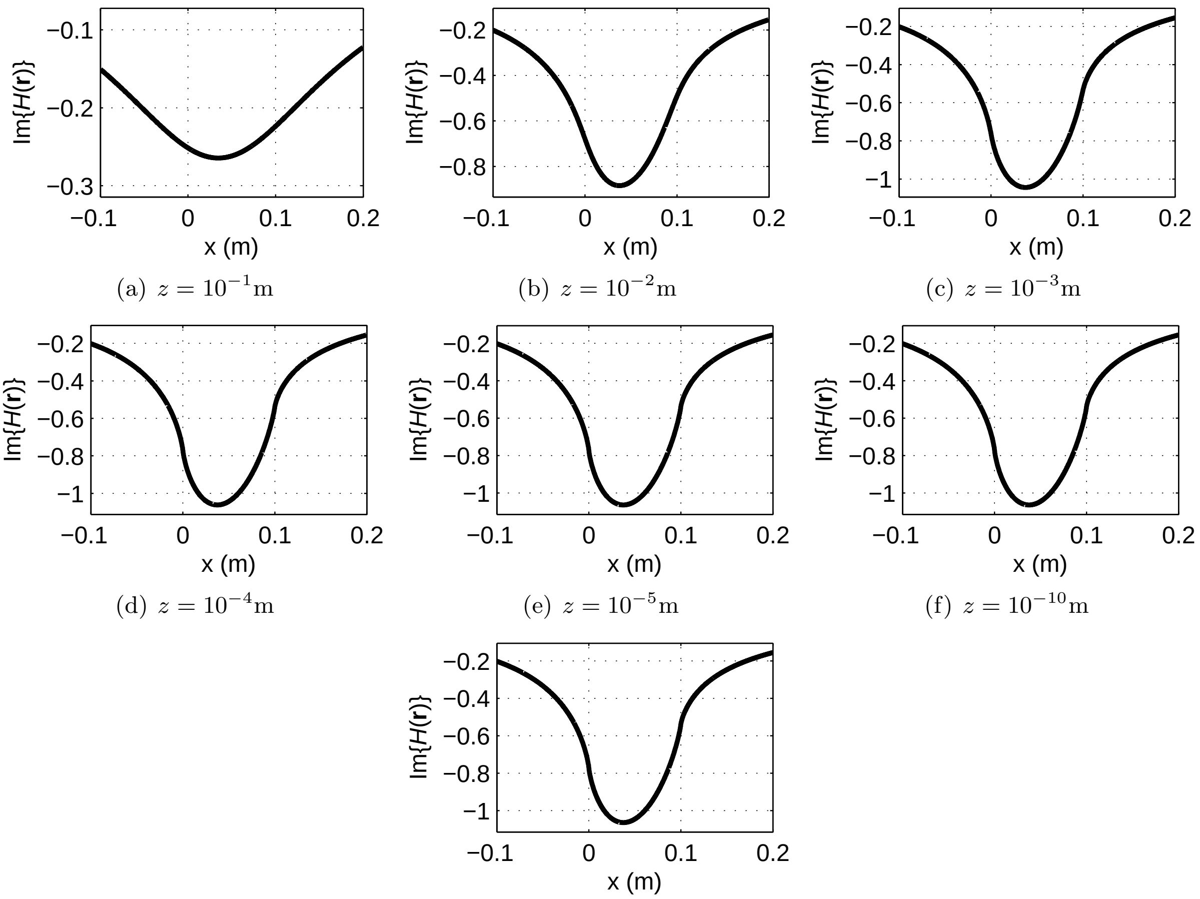
(d) $z = 10^{-4}$m

(e) $z = 10^{-5}$m

(f) $z = 10^{-10}$m

(g) $z = 10^{-15}$m

Figure 6.10: Real part of $H(\mathbf{r})$ when sampling along a line parallel to the x axis, where there is an edge of the integration triangle, for different heights or distances to the triangle.



(a) $z = 10^{-1}$m

(b) $z = 10^{-2}$m

(c) $z = 10^{-3}$m

(d) $z = 10^{-4}$m

(e) $z = 10^{-5}$m

(f) $z = 10^{-10}$m

(g) $z = 10^{-15}$m

Figure 6.11: Imaginary part of $H(\mathbf{r})$ when sampling along a line parallel to the x axis, where there is an edge of the integration triangle, for different heights or distances to the triangle.

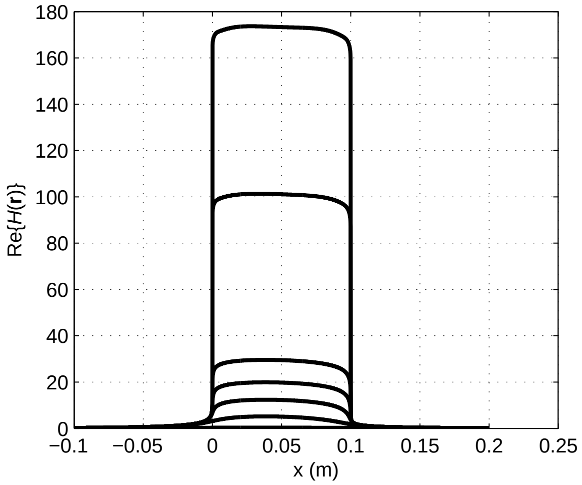


Figure 6.12: Real part of $H(\mathbf{r})$ when sampling along a line parallel to the x axis, where there is an edge of the integration triangle, for different heights or distances to the triangle.

6.3 Possible solutions and open issues

Before starting with the real discussion, let us introduce, avoiding the fine details, a couple of 1-D integration quadrature rules. The first one is the regular Gauss-Legendre quadrature rule, which exactly integrates polynomials up to order $2n-1$ where n is the number of integration points. This quadrature is very well suited for smooth functions, which are accurately approximated by low order polynomials. The other is the Double Exponential quadrature rule [93], also known as tanh-sinh quadrature, which unlike Gaussian quadrature is very well suited for functions with singularities or infinite derivatives at one or both endpoints of the integration interval. Both quadrature rules allow arbitrary-precision integration by increasing the number of integration points. However, the convergence rate can be appreciably different depending on the shape of the function. After applying a variable change, the integration points in the Double Exponential are taken arbitrarily close to the endpoints of the integration interval, allowing to extract all the information present at the singular point with a reduced number of integration points.

With this in mind we will focus on different possibilities to numerically deal with the outer integration

$$I = \int_{E_P} H(\mathbf{r})\, dS \tag{6.34}$$

for the different relative positions between inner E_Q and outer E_P triangles. Of course, if the two triangles are very separated, the best option is using a 2-D Gauss-Legendre cubature rule, considered the smoothness of the field function $H(\mathbf{r})$ at large distances. A completely different story appears when they are closely placed, as was shown in the last section. The edge-adjacent and near-edge-adjacent cases are accurately treated in the next paragraphs and some hints are given for the rest of the situations.

We select the representative edge-adjacent example E_P:$\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ and E_Q:$\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_4\}$ with the following vertices:

$$\begin{aligned} \mathbf{r}_1 &= [0,0,0]; \quad \mathbf{r}_2 = [0, 0.1\lambda, 0] \\ \mathbf{r}_3 &= [0,0,0.1\lambda]; \quad \mathbf{r}_4 = [0.1\lambda, 0, 0]. \end{aligned} \tag{6.35}$$

Firstly, we express the outer integral in the equilateral parameter space introduced in previous sections:

$$I = \frac{A_p}{\sqrt{3}} \int_0^{\sqrt{3}} d\xi \int_{\xi/(\sqrt{3}-1)}^{(1-\xi)/\sqrt{3}} H(\mathbf{r}(\eta,\xi))\, d\eta. \tag{6.36}$$

The integrand function $H(\mathbf{r})$, which corresponds to the inner integration, is again accurately evaluated using a singularity cancellation method. As was analyzed in the previous section, $H(\mathbf{r})$ has a logarithmic singularity when ξ tends to zero and when η tends to its extremes 1 and -1 for very small values of ξ. This implies that a Gauss-Legendre quadrature rule applied to each of the variables η and ξ fails to converge properly in an efficient manner.

A possible solution for this slow convergence was presented in [94, 86], which stated that utilizing a Double Exponential rule at each variable solved the problem. Effectively, this approach achieves close to machine precision in a reasonable time, but it is still further improvable. Note that we need to apply Double Exponential to both variables and not only one because the singularity is present in both directions.

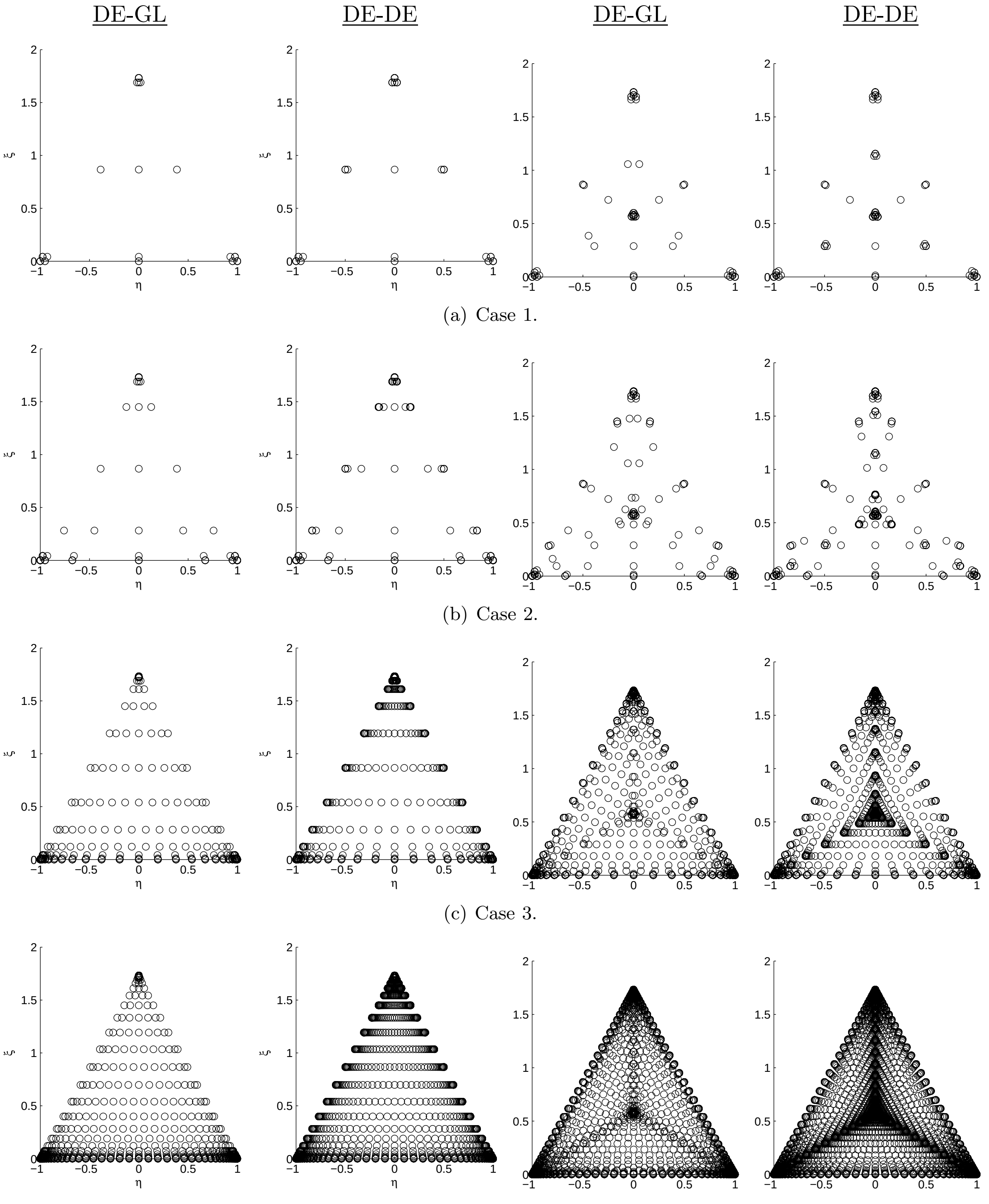
The last technique, from here on called DE-DE (not to be confused with Direct Evaluation), has three main causes of inefficiency: it accumulates many integration points close to the upper vertex or $\xi = \sqrt{3}$, where the function is smoothest, because the Double Exponential considers singularity at both limits of ξ; it considers that the integrand with respect to η is singular at its integration extremes independently of the fixed ξ value, however, this singularity is only present for extremely small values of ξ; the integrand function is smoother as a η function and with a smaller domain when ξ grows and this is not considered either.

The solution proposed here is based on the Double Exponential one though removing unnecessary integration points in a clever manner. A threshold of 0.1 is established for ξ, so that for values of ξ smaller than the threshold, Double Exponential is used in the η integration, as in the DE-DE technique. However, for values of ξ larger than the threshold, we use a Gauss-Legendre quadrature rule utilizing the smoothness of the integrand in terms of η for large values of ξ. Furthermore, the number of integration points utilized at each one of these GL integrations is decreased proportionally as ξ grows. A minimum of three integration points per 1D integration has been set to avoid further problems. For the ξ integration we use Double Exponential exactly as in the DE-DE case.

The first and second columns of subfigures in Fig. 6.13 show the integration points inside the equilateral triangle parameter space in four equivalent cases for the combined DE-GL technique presented here and the DE-DE technique, respectively. The number of integration points increases at each case and the relative error of the whole integral is approximately reduced proportionally for both cases. Clearly, the number of integration points with the new technique is smaller than in the DE-DE method. Before showing some experimental results, note that the case where the triangles are coincident or very close to each other, which has a logarithmic singularity along the three edges, as proved in the previous section, can be dealt with subdividing the equilateral triangle by its barycenter and applying the aforementioned techniques to each subtriangle. The distribution of integration points for this case is shown in the third and forth columns of Fig. 6.13 for the DE-GL and DE-DE techniques, respectively. The accumulation of integration points in the center of the triangle, where the function is smooth, is clearly larger in the DE-DE case.

Fig. 6.14 shows the relative error of the real part of the integral both in terms of the number of integration points for the outer triangle and in terms of the required CPU time in the edge-adjacent case under study. The reference solution is taken from the last chapter. The two methods presented here, the existing DE-DE and the proposed DE-GL converge relatively fast close to machine precision. However, as expected, when the regular Gauss-Legendre quadrature rule is used, the convergence is much slower. The improvement obtained with the combined method with respect to the DE-DE method translates in half of the integration points or half the CPU time to have the same relative error.

These results, however impressive, are still far from the Direct Evaluation method performance, presented in the previous chapter. Nonetheless, the advantage of the present numerical technique is that it can be applied in a number of different situations and not only the vertex or edge adjacent. For instance, we next show its behavior when the edges of the triangles are slightly separated in the z-direction. In particular, the new outer triangle is displaced as E_P:$\{\mathbf{r}_1 + z\hat{\mathbf{z}}, \mathbf{r}_2 + z\hat{\mathbf{z}}, \mathbf{r}_3 + z\hat{\mathbf{z}}\}$. Fig. 6.15 and 6.16 display the results for two different displacements $z = 10^{-2}$m and $z = 10^{-5}$m, respectively. The reference for these cases are not accurately justified as in the previous case. However, the convergence of all the methods up to nine significant digits should be enough to consider the results reliable to this extent. In view of the results, when the triangles are extremely



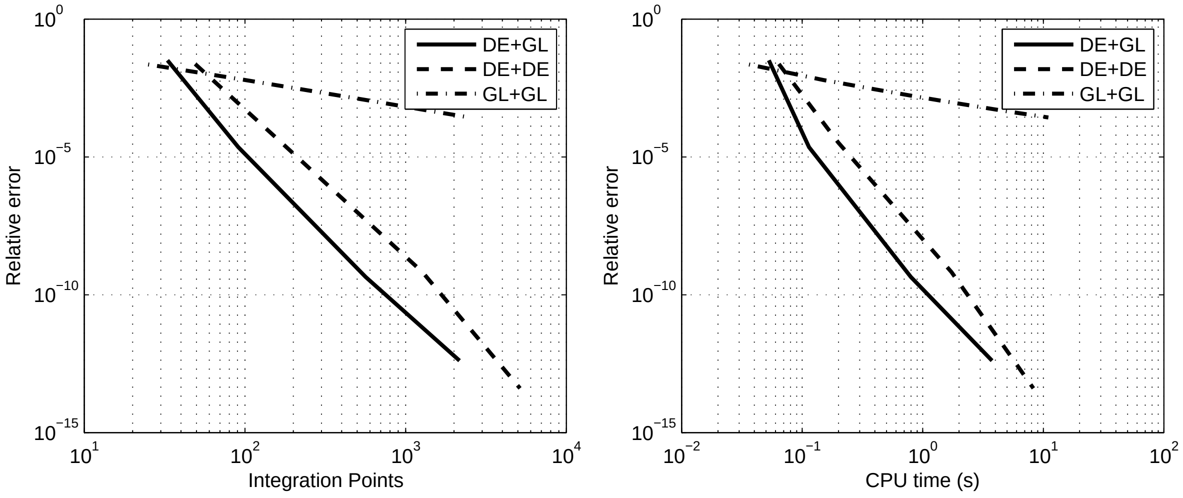
(a) Case 1.

(b) Case 2.

(c) Case 3.

(d) Case 4.

Figure 6.13: Distribution of the integration points, increasing from (a) to (d), in the equilateral triangle space. The first and second column correspond to the case when the singularity is placed along an edge using DE+GL and DE+DE methods to sample η and ξ, respectively. The third and forth columns are the same when the singularity is at all the edges.

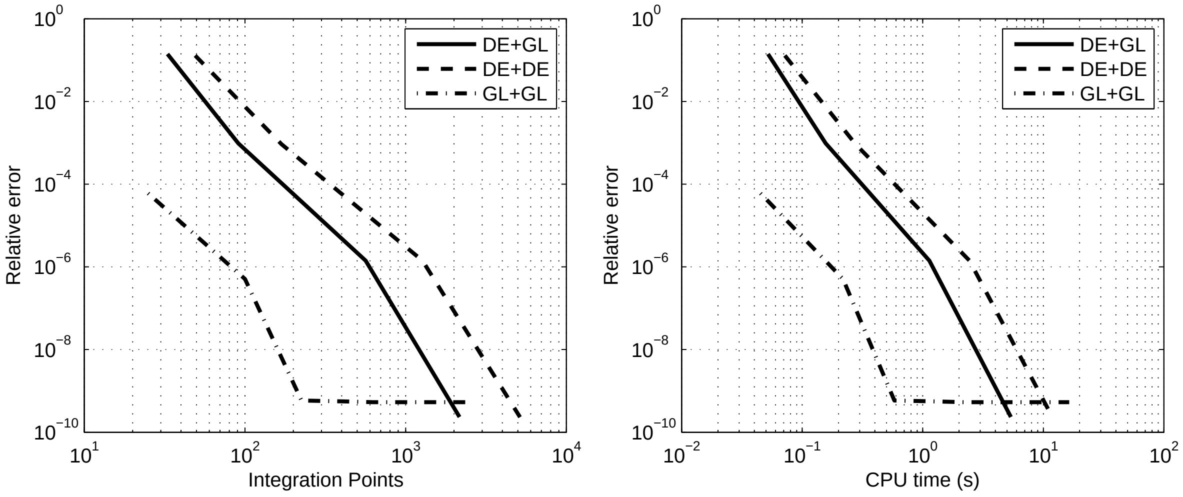


(a) Integration Points.

(b) CPU time.

Figure 6.14: Convergence of the different presented methods for the outer integral when the inner and outer triangle share a common edge.

close, $z = 10^{-5}$m, it is worth to use the presented techniques. Contrarily, when the distance starts to grow, about $z = 10^{-2}$m, Gauss-Legendre quadrature rule becomes again the best choice.

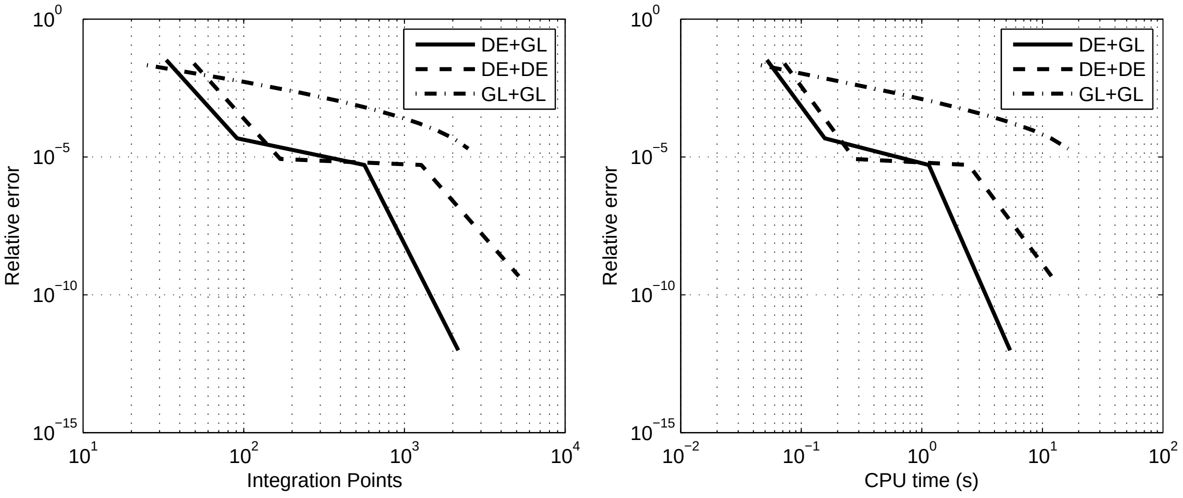


(a) Integration Points.

(b) CPU time.

Figure 6.15: Convergence of the different presented methods for the outer integral when the inner and outer triangle have an edge separated a distance $z = 10^{-2}$m.

Considering that the singularity of the field function is logarithmic when ξ tends to 0, a Gauss-Laguerre quadrature rule can be used as an alternative to the DE for the ξ integration after some manipulations, with the advantage that it treats singularity only in one of the integration extremes,



(a) Integration Points.

(b) CPU time.

Figure 6.16: Convergence of the different presented methods for the outer integral when the inner and outer triangle have an edge separated a distance $z = 10^{-5}$m.

which is our case. The integrand for ξ integration is

$$M(\xi) = \int_{\xi/(\sqrt{3}-1)}^{(1-\xi)/\sqrt{3}} H(\mathbf{r}(\eta, \xi))\, d\eta \tag{6.37}$$

and a variable transformation $\xi = \sqrt{3}e^{-t}$ yields

$$I = \frac{A_p}{\sqrt{3}} \int_0^{\sqrt{3}} M(\xi)\, d\xi = A_p \int_0^{\infty} M(\sqrt{3}e^{-t})e^{-t}\, dt = A_p \int_0^{\infty} S(t)e^{-t}\, dt \tag{6.38}$$

which has exactly the shape of the Gauss-Laguerre integration, where the integrand is

$$S(t) = M(\sqrt{3}e^{-t})) \tag{6.39}$$

and the weighting function is the exponential e^{-t}. As the integrand $M(\xi)$ behaves logarithmically when ξ tends to zero, it is translated in an integrand function $S(t)$ very smooth, and therefore integrable with a reduced number of integration points using Gauss-Laguerre rule:

$$M(\xi) \approx -k \log \xi \quad \Longrightarrow \quad S(t) \approx -k(-t + \log \sqrt{3}). \tag{6.40}$$

Due to the smoothness of the integrand function $S(t)$, the summation in the Gauss-Laguerre quadrature rule has been truncated when the weights where smaller than the machine precision constant 10^{-16}.

Fig. 6.17 shows the results comparing Double Exponential in the ξ integration with respect to the utilization of Gauss-Laguerre, whilst the rest of parameters are set to have machine precision. We can infer a very similar behavior between the two methods. The main disadvantage of the Gauss-Laguerre one is that it could fail with different kernels, losing then the general purpose objective.

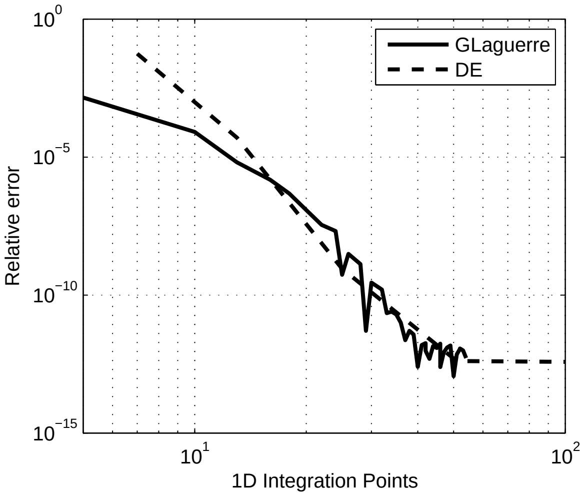


Figure 6.17: Convergence when using double exponential or Gauss Laguerre integration rules for the ξ integration when the inner and outer triangle share a common edge.

Finally, in the presented cases it is very clear where the singularity is placed inside the outer triangle. However, in a more general case, the relative position between inner and outer triangles is a priori unknown. Therefore, there is a first important step in which the singularity position and the kind -point-like, line-like or surface-like- must be determined. After this, the triangle should be split accordingly in pieces which are equivalent to the previously treated cases. This issue becomes even harder when a higher order mesh is considered, as there should be a search for the closest points in the non-planar triangles. This work is left as a matter of future research.

6.4 Conclusions

After the whole development and analysis of the previous and the present chapters, our choice for an accurate and efficient evaluation of the 4-D integrals appearing in Galerkin surface integral formulations is the following:

- If the inner and outer triangles are well separated (no points closer than 10^{-2}m), 2-D Gauss-Legendre integration rule adapted to a triangle domain per triangle.

- If the triangles share a common vertex, a common edge or are coincident, the best choice is using the Direct Evaluation method and performing the maximum of integrals analytically. For an easier to implement formulation, all the integrals can be done numerically as well, though keeping the variable changes and integration re-ordering of the Direct Evaluation method.

- If the triangles have a very near interaction (closer than 10^{-2}m) at some point, the first thing is to determine where this near interaction is placed and separate the integration triangles in domains which are in one of the treated cases. It is necessary to determine if the near interaction is point-like, line-like or surface-like because the singularity is then differently distributed. This point is still an open issue, specially regarding efficiency.

- If the triangles have an edge-like near interaction, we use singularity cancellation or singularity subtraction for the inner integral and the technique presented in this chapter, which combines Double Exponential and Gauss-Legendre integration points, for the outer integral. Although the results with this technique are already as accurate as desired and more efficient than other existing ones, there is still some space for optimization.

- If the triangles have a surface-like near interaction, the same as in the previous item applies, although this time the outer integral must deal with the singularity placed at all the edges, splitting the triangle by its barycenter in three new sub-triangles and applying the edge-like behavior at each sub-triangle.

Multi-port Systems

Chapter 7

Multi-port Systems

This chapter tries to establish a connection between two, sometimes distant though highly related, worlds. On one hand we encounter the numerical or computational universe, particularly the Method of Moments, to which we have dedicated most of this thesis. On the other, the more practical electromagnetic system or antenna design area. The latter calls for the computation of certain parameters such as S-parameters or input impedance among others which give a descriptive and important idea of the behavior of the final system, and which might be actually measured in a future prototype.

7.1 Introduction

Before starting, let us define what we understand by a multi-port system. Imagine an electromagnetic device which can be included in a bounded domain whose communication with the outside world is reduced to N ports (see Fig. 7.1). One way of analyzing or characterizing this kind of systems, commonly used in microwave theory, is observing the output of a certain port when the rest of the ports are either loaded with a controlled impedance or with a voltage or current source.

Consequently, the first necessary thing is to properly model a port in our discretized space arising from the MoM formulations. When a port is excited with a source, it is modeled with an incident field which would produce the same, or at least similar, effect as the real source, in the absence of this one. It determines in turn the values of the independent vector in the MoM linear system when it is projected on the testing functions. If more than one port is excited, we can easily apply superposition due to the linearity of the system.

The main contribution of the author to this chapter is the δ-gap source generalization to ports crossing the RWG or the wire model basis functions arbitrarily, not necessarily at a triangle edge or wire vertex as was done before to our knowledge. It provides freedom to define the position of the ports, not having to care anymore about the discretization along the port. Although some authors claim that the δ-gap source gives not very accurate results in some cases, it is still interesting because the integral appearing in the δ-gap excitation coincides with the integral used to compute the current related to the port as will be shown in Section 7.3. This current is used to compute the Z-parameters, which are in turn necessary to compute the S-parameters first and the auto-impedance in the end.

The chapter starts with the description of the two most used port excitation models in the Method of Moments, the δ-gap and the magnetic frill (Sec. 7.2). Secondly, considering that it is easy to simulate a short-circuit in the MoM, we explain how to obtain the Z-parameters from the solution currents from the MoM (Sec. 7.3). Thirdly, we formulate the computation of the S-

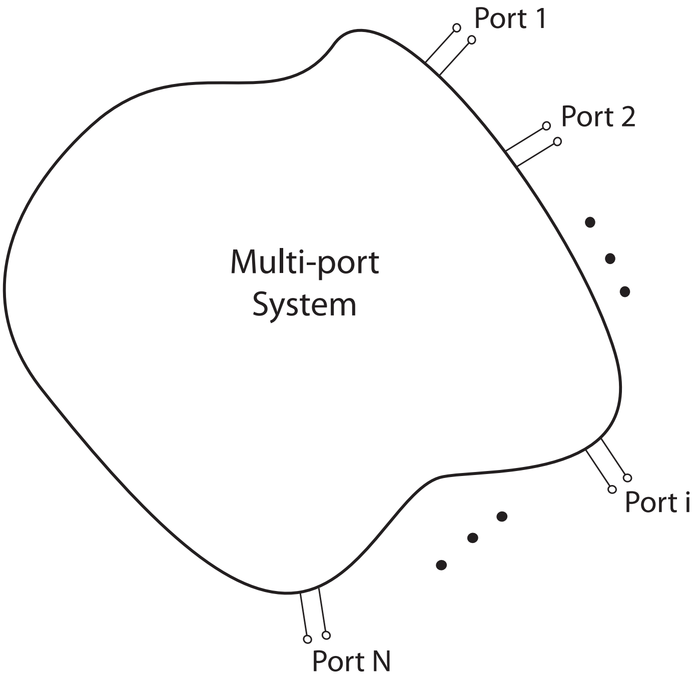


Figure 7.1: Multi-port system scheme.

parameters from the Z-parameters with the inclusion of a reference load Z_0 in the system (Sec. 7.4). Subsequently, we get the desired input impedances of the multi-port system from the S-parameters and the loads applied to each port. Finally, some numerical results with commonly used test cases are shown to corroborate the explained formulation (Sec. 7.6).

7.2 Port feeding

A convenient way of defining a port in our discretized electromagnetic problem is from the intersection of the discretization elements (triangles or wires) with a pre-defined plane near a determined point. The intersection will be a linear path, usually a close path, where we consider that the port is situated. We need to imagine that the conductor is infinitesimally separated at this path and the source is placed between these two separated paths. When no forced feeding conditions are imposed in the port under study, it behaves as though it was short-circuited, taking into account that the current can freely pass across the path.

7.2.1 δ-gap source

The main idea behind the δ-gap source model is forcing a 1V voltage gap following the conductor when we cross the feeding plane and a constant voltage in each part of the conductor everywhere else. As we will see, to formally obtain this voltage we need an incident electric field which behaves as a Dirac delta distribution, leading to the model name. This incident electric field is actually not Maxwellian and therefore not physical. Nonetheless, the simplified version presented here is still very often used due to its simplicity, taking into account that sufficiently good results are commonly obtained in the vast majority of problems. Furthermore, as shown in Sec. 7.3, the δ-gap EFIE independent vector coincides with some necessary coefficients to compute the current crossing the port from the resulting surface current vector.

We start with the analysis when the port is defined over RWG basis functions and the same ideas, with slight modifications, will do the work for wire model basis functions. As the RWG functions are defined over triangles, it suffices to consider separately each triangle and the total will be the addition of all of them. Our port is defined by a plane Π with normal vector $\hat{\mathbf{n}}$ characterized by the points

$$\Pi : \{\mathbf{r}|\hat{\mathbf{n}} \cdot \mathbf{r} = D\}. \tag{7.1}$$

We only care about the triangles which cross the plane, because in the rest we want to have constant voltage, easily obtained with an incident electric field equal to zero. For the triangles crossing the plane we have the situation shown in Fig. 7.2, where the vertex $\mathbf{r}_2$ of the triangle is the one alone at one side of the plane. Fixed a normal vector to the triangle $\hat{\mathbf{n}}_T$ we can uniquely and conveniently define a sorted orthonormal $\mathbb{R}^3$ vector basis $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{n}}_T)$, where $\hat{\mathbf{u}}$ has the direction of the intersection of the plane with the triangle and $\hat{\mathbf{v}}$ is orthogonal to $\hat{\mathbf{u}}$ following the other direction of the triangle and oriented with the normal of the plane $\hat{\mathbf{n}}$. The sense of $\hat{\mathbf{u}}$ is then defined to accomplish $\hat{\mathbf{u}} \times \hat{\mathbf{v}} = \hat{\mathbf{n}}_T$. The actual computation of this parameters will be shown further on.

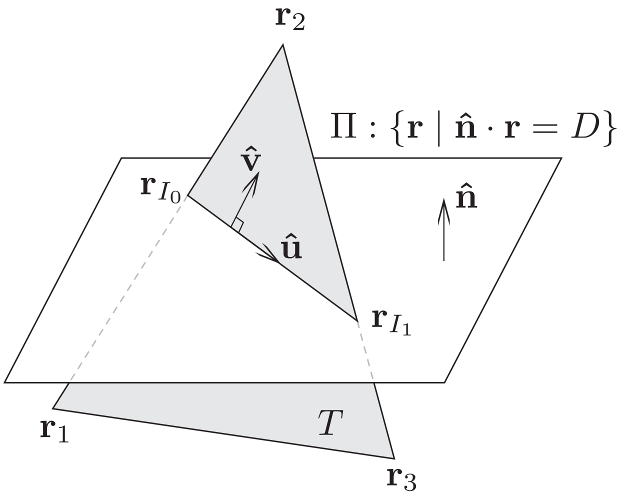


Figure 7.2: Triangle from the discretization crossing the feeding plane Π and variables involved.

Any point in the space can then be expressed in the new basis in the new rectangular coordinates (u, v, n) as:

$$\mathbf{r} = \mathbf{r}_{I_0} + u\hat{\mathbf{u}} + v\hat{\mathbf{v}} + n\hat{\mathbf{n}}_T \tag{7.2}$$

where $\mathbf{r}_{I_0}$ is the intersection of the edge defined by the points $\mathbf{r}_1$ and $\mathbf{r}_2$ of the triangle with the plane Π. The triangle is then included in the points with coordinate n equaling zero. We define the so-called delta-gap incident electric field over the triangle, considering only the tangential component, as

$$\mathbf{E}^i(\mathbf{r})\Big|_T = \hat{\mathbf{v}}\delta(v). \tag{7.3}$$

To corroborate that with this incident field we have a 1V gap when crossing the plane, we integrate the incident field following a path over the triangle which crosses the intersection line perpendicularly, or equivalently in the $\hat{\mathbf{v}}$ direction, as follows:

$$\begin{aligned} V &= \int_{-\infty}^{\infty} \mathbf{E}^i(\mathbf{r}) \cdot d\mathbf{l} = \int_{-\infty}^{\infty} \hat{\mathbf{v}}\delta(v) \cdot d\hat{\mathbf{v}} \\ &= \int_{-\infty}^{\infty} \delta(v) dv = 1\text{V}. \end{aligned} \tag{7.4}$$

To compute the n-th element of the independent vector of the EFIE formulation with this incident field we simply apply the following integration over the triangle corresponding to the RWG basis function $\mathbf{f}_n$:

$$E_n = \left\langle \mathbf{f}_n(\mathbf{r}), \mathbf{E}^i(\mathbf{r}) \right\rangle = \int_T \mathbf{f}_n(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}) \, dS = \int_T \mathbf{f}_n(\mathbf{r}) \cdot \hat{\mathbf{v}}\delta(g(\mathbf{r})) \, dS \tag{7.5}$$

where $g(\mathbf{r}) = v = (\mathbf{r} - \mathbf{r}_{I_0}) \cdot \hat{\mathbf{v}}$. Applying the general mathematical property

$$\int_{\mathbb{R}^n} f(\mathbf{x})\delta(g(\mathbf{x}))d\mathbf{x} = \int_{g^{-1}(0)} \frac{f(\mathbf{x})}{|\nabla g|} d\sigma(\mathbf{x}) \tag{7.6}$$

the bi-dimensional integral in (7.5) can be computed, after removing the δ, as a 1-dimensional integral over the intersection line:

$$E_n = \int_0^l \mathbf{f}_n(\mathbf{r}_{I_0} \pm u\hat{\mathbf{u}}) \cdot \hat{\mathbf{v}} \, du \tag{7.7}$$

where it has been used that $\nabla g = \hat{\mathbf{v}}$ and l is the length of the intersection line. The RWG basis function can be expressed as:

$$\mathbf{f}_n(\mathbf{r}) \Big|_T = \pm \frac{l_n}{2A_T} (\mathbf{r} - \mathbf{r}_i) \tag{7.8}$$

where $\mathbf{r}_i$ is the opposite vertex of the edge under consideration in the triangle T. Substituting into (7.7) finally yields

$$\begin{aligned} E_n &= \pm \frac{l_n}{2A_T} \int_0^l (\mathbf{r}_{I_0} \pm u\hat{\mathbf{u}} - \mathbf{r}_i) \cdot \hat{\mathbf{v}} \, du \\ &= \pm \frac{l_n}{2A_T} (\mathbf{r}_{I_0} - \mathbf{r}_i) \cdot \hat{\mathbf{v}} \int_0^l \, du \\ &= \pm l \frac{l_n}{2A_T} (\mathbf{r}_{I_0} - \mathbf{r}_i) \cdot \hat{\mathbf{v}}. \end{aligned} \tag{7.9}$$

It still remains to actually compute the vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$. To do so, we first compute the two intersection points

$$\begin{aligned} \mathbf{r}_{I_0} &= \mathbf{r}_1 + \frac{D - \hat{\mathbf{n}} \cdot \mathbf{r}_1}{\hat{\mathbf{n}} \cdot (\mathbf{r}_2 - \mathbf{r}_1)} (\mathbf{r}_2 - \mathbf{r}_1) \\ \mathbf{r}_{I_1} &= \mathbf{r}_3 + \frac{D - \hat{\mathbf{n}} \cdot \mathbf{r}_3}{\hat{\mathbf{n}} \cdot (\mathbf{r}_2 - \mathbf{r}_3)} (\mathbf{r}_2 - \mathbf{r}_3). \end{aligned} \tag{7.10}$$

We can then introduce the vector defined by these two points

$$\mathbf{u}_1 = \mathbf{r}_{I_1} - \mathbf{r}_{I_0} \tag{7.11}$$

whose length and unitary counterpart are

$$\begin{aligned} l &= |\mathbf{u}_1| \\ \hat{\mathbf{u}}_1 &= \frac{\mathbf{u}_1}{l}. \end{aligned} \tag{7.12}$$

The vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ are finally:

$$\begin{aligned} \hat{\mathbf{u}} &= S_i \hat{\mathbf{u}}_1 \\ \hat{\mathbf{v}} &= \hat{\mathbf{n}}_T \times \hat{\mathbf{u}} \end{aligned} \tag{7.13}$$

where S_i is just a case-dependent sign computed as:

$$S_i = \begin{cases} +1, & \text{if } \hat{\mathbf{n}} \cdot (\hat{\mathbf{n}}_T \times \hat{\mathbf{u}}_1) > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{7.14}$$

We have considered so far that the triangle completely crosses the plane, but it could happen that the intersection line exactly coincides with one of the edges of the triangle, as shown in Fig. 7.3. In that case we could either consider that the plane is actually placed slightly upwards and then apply the same as before for the upper triangle or that it is downwards and use the lower triangle. The result is obviously independent of this choice as we show below.

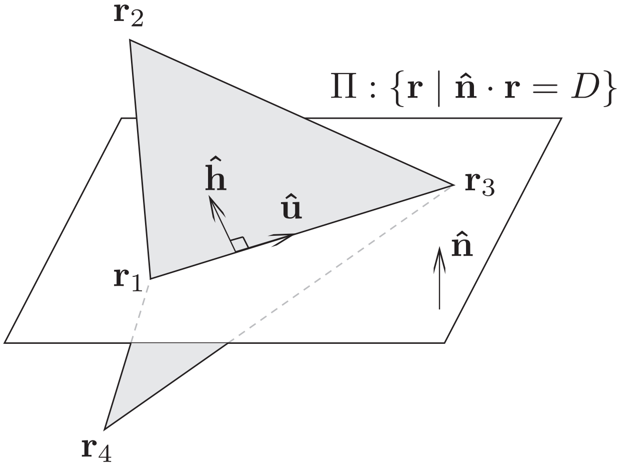


Figure 7.3: Situation when an edge of the discretized triangle element exactly coincides with the feeding plane Π.

The vector $\hat{\mathbf{v}}$ in that case equals the height vector of the triangle $\hat{\mathbf{h}}$. When $\mathbf{r}_i$ is either $\mathbf{r}_1$ or $\mathbf{r}_3$ then $E_n = 0$ because $\hat{\mathbf{v}}$ is orthogonal to $\mathbf{r}_{I_0} - \mathbf{r}_i$. When the edge associated to the RWG is the intersection line then the opposite vertex $\mathbf{r}_i$ is $\mathbf{r}_2$ or $\mathbf{r}_4$ depending on the triangle. When we apply

(7.9), for instance to the upper triangle, we get:

$$E_n = \pm l_n \frac{l_n}{2A_T} (\mathbf{r}_1 - \mathbf{r}_2) \cdot \hat{\mathbf{h}} = \pm l_n \frac{l_n h}{2A_T} = \pm l_n. \tag{7.15}$$

As we can see, it only depends on the length of the edge, which is common in both triangles, and therefore independent of the choice as we wanted to show.

This concludes the analysis for RWG basis functions, and now it is time to do almost the same when the fed basis functions are wire segments. The new situation is represented in Fig. 7.4 where $\mathbf{r}_{I_c}$ is the intersection of the axis of the cylindrical segment with the plane. The intersection of the cylinder surface with the plane is an ellipse. Due to the fact that the wire model assumes that the current distribution is constant along the angle coordinate of the cylinder, it is better to consider that the actual feeding is in a plane passing through the same point $\mathbf{r}_{I_c}$ but with normal vector equal to the direction of the cylinder (see Fig. 7.5). The intersection is then a circle, much easier to deal with.

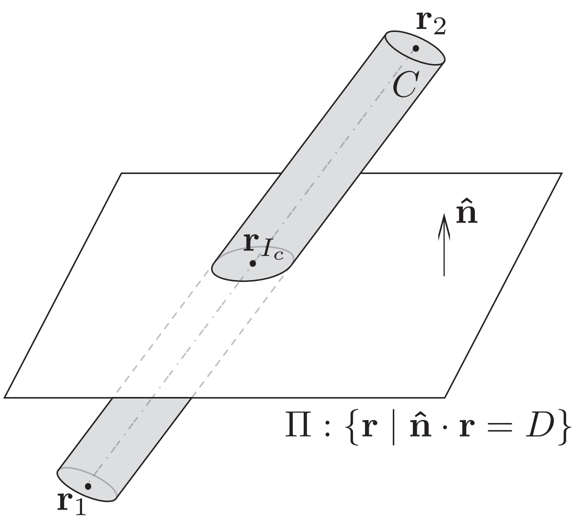


Figure 7.4: Original situation for the discretized cylindrical wire element intersected by the feeding plane Π.

Similarly to what was done with the triangles, we can introduce a new ordered vector basis $(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{v}})$ as shown in Fig. 7.6 where $\hat{\mathbf{v}}$ is the unitary vector following the direction of the wire and oriented according to the normal vector $\hat{\mathbf{n}}_2$. $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ are chosen arbitrarily so that they are in the plane Π_2 and $\hat{\mathbf{u}}_1 \times \hat{\mathbf{u}}_2 = \hat{\mathbf{v}}$. Then, any point in the space in cylindrical coordinates (ρ, ϕ, v) can be expressed as:

$$\mathbf{r} = \mathbf{r}_{I_c} + \rho \cos(\phi) \hat{\mathbf{u}}_1 + \rho \sin(\phi) \hat{\mathbf{u}}_2 + v \hat{\mathbf{v}}. \tag{7.16}$$

The points of the cylinder are included in the points with coordinate ρ equal to the radius of the cylinder a. The tangential incident electric field is again defined as:

$$\mathbf{E}^i(\mathbf{r}) \Big|_C = \hat{\mathbf{v}} \delta(v). \tag{7.17}$$

This incident field assures a 1V gap among the two cylinder parts defined by the intersection

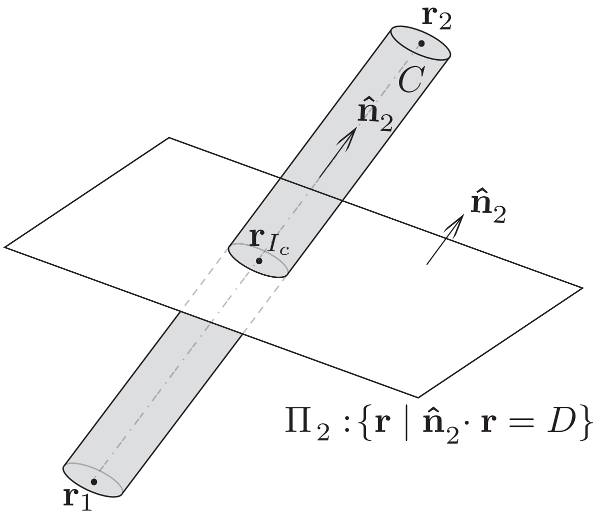


Figure 7.5: Situation for the discretized cylindrical wire element intersected by the feeding plane Π_2 after orienting the feeding plane with the axis of the cylinder.

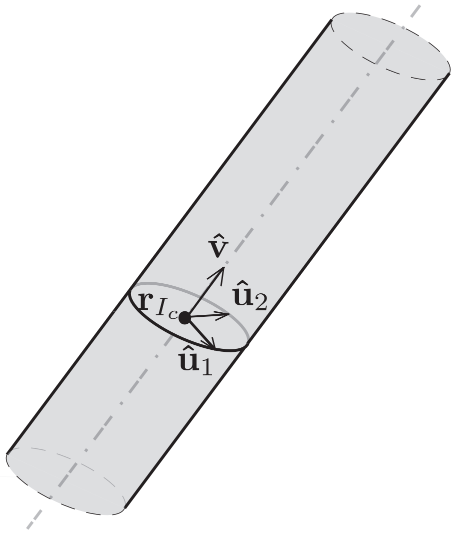


Figure 7.6: New reference vector basis at the intersection of the cylinder with the feeding plane.

circumference. To compute the independent vector element we proceed as before, now considering that the basis function is:

$$\mathbf{f}_n(\mathbf{r}) = \pm \frac{1}{(2\pi a) l_n} (\mathbf{r}_c - \mathbf{r}_i) \tag{7.18}$$

where l_n is the length of the cylinder, $\mathbf{r}_c$ is the projection of $\mathbf{r}$ into the axis of the cylinder and $\mathbf{r}_i$ is either $\mathbf{r}_1$ or $\mathbf{r}_2$. Therefore,

$$E_n = \langle \mathbf{f}_n(\mathbf{r}), \mathbf{E}^i(\mathbf{r}) \rangle = \int_C \mathbf{f}_n(\mathbf{r}) \cdot \mathbf{E}^i(\mathbf{r}) \, dS = \int_C \mathbf{f}_n(\mathbf{r}) \cdot \hat{\mathbf{v}} \delta(g(\mathbf{r})) \, dS = a \int_0^{2\pi} \mathbf{f}_n(\mathbf{r}) \cdot \hat{\mathbf{v}} \, d\phi \tag{7.19}$$

and substituting the basis function:

$$E_n = \pm \frac{1}{2\pi l_n} \int_0^{2\pi} (\mathbf{r}_{I_c} - \mathbf{r}_i) \cdot \hat{\mathbf{v}} \, d\phi = \pm \frac{(\mathbf{r}_{I_c} - \mathbf{r}_i) \cdot \hat{\mathbf{v}}}{l_n}. \tag{7.20}$$

The intersection point is easily computed as:

$$\mathbf{r}_{I_c} = \mathbf{r}_1 + \frac{D - \hat{\mathbf{n}} \cdot \mathbf{r}_1}{\hat{\mathbf{n}} \cdot (\mathbf{r}_2 - \mathbf{r}_1)} (\mathbf{r}_2 - \mathbf{r}_1). \tag{7.21}$$

And the vector $\hat{\mathbf{v}}$ is:

$$\hat{\mathbf{v}} = S_i \hat{\mathbf{v}}_1 \tag{7.22}$$

with

$$\hat{\mathbf{v}}_1 = \frac{\mathbf{r}_2 - \mathbf{r}_1}{l_n} \tag{7.23}$$

$$S_i = \begin{cases} +1, & \text{if } \hat{\mathbf{n}} \cdot \hat{\mathbf{v}}_1 > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{7.24}$$

Similarly to the triangles, when the intersection point coincides with one of the extremes of the cylinder, then we can choose to use the upper or the lower cylinder and the result will be the same. In this case the result is zero, or ±1 depending on the basis function we are considering.

7.2.2 Magnetic frill source

The magnetic frill generator was introduced to calculate the near as well as the far zone fields from coaxial apertures [95]. To use this model the feed gap is replaced with a circumferentially directed magnetic current density that exists over an annular aperture with inner radius a, which is usually chosen to fit as well as possible the elements at the port, and an outer radius b determined by the characteristic impedance of the supposed transmission line feeding the port.

Over the annular aperture of the magnetic frill generator the electric field is represented by the TEM mode field distribution of a coaxial transmission line given by

$$\mathbf{E}_f = \hat{\mathbf{a}}_\rho \frac{V_i}{2\rho' \ln(b/a)}. \tag{7.25}$$

Therefore the corresponding equivalent magnetic current density M_f for the magnetic frill generator used to represent the aperture is equal to

$$M_f = -2\hat{\mathbf{n}} \times E_f = -2\hat{\mathbf{a}}_z \times \hat{\mathbf{a}}_\rho E_\rho = -\hat{\mathbf{a}}_\phi \frac{V_i}{\rho' \ln(b/a)}. \tag{7.26}$$

The fields generated by the magnetic frill generator of (7.26) can be numerically calculated using the electric vector potential $\mathbf{F}$ as stated in [95], where some approximations are done.

7.3 Z-parameters

Given a N ports system network, the Z-parameters relate the voltage present at a certain port knowing the currents crossing all the ports. More specifically, if $\mathbf{V}^T = [V_1, \ldots, V_N]$ is a vector containing the voltages at each port and $\mathbf{I}^T = [I_1, \ldots, I_N]$ is a vector with the currents crossing each port, then the Z-parameters are the elements of a $N \times N$ matrix $[\mathbf{Z}]$ which accomplishes:

$$\mathbf{V} = [\mathbf{Z}]\mathbf{I}. \tag{7.27}$$

Each equation, reads

$$V_i = \sum_{j=1}^{N} Z_{ij} I_j \tag{7.28}$$

and therefore the element Z_{ij} can be computed leaving all the ports but the j-th open-circuited and reading the voltage at port i:

$$Z_{ij} = \left.\frac{V_i}{I_j}\right|_{I_k=0 \quad \forall k \neq j}. \tag{7.29}$$

Unfortunately, it can be hard to model an open-circuit in the MoM. However, modeling a short-circuit is immediate, as it is equivalent to leave the basis functions at the port as they are, allowing the current to flow freely. Hence, it is better to first compute the impedance parameters matrix $[\mathbf{Y}]$ and compute $[\mathbf{Z}]$ as the inverse of this matrix. The equation is now,

$$\mathbf{I} = [\mathbf{Y}]\mathbf{V} \tag{7.30}$$

or row by row

$$I_i = \sum_{j=1}^{N} Y_{ij} V_j \tag{7.31}$$

and the Y_{ij} elements can be easily computed short-circuiting all the ports but the j-th and reading the current crossing port i:

$$Y_{ij} = \left.\frac{I_i}{V_j}\right|_{V_k=0 \quad \forall k \neq j}. \tag{7.32}$$

The port j is fed with one of the sources previously described in last section. Afterwards it suffices to perform

$$[\mathbf{Z}] = [\mathbf{Y}]^{-1}. \tag{7.33}$$

In order to compute the voltage V_j in terms of the surface currents obtained after solving the MoM linear system, we can go through the supplied power, considering always the situation described above, i.e., feeding at port j and short-circuit everywhere else. On one hand, the supplied power P_s corresponds with the product of voltage and current at port j:

$$P_s = V_j I_j. \tag{7.34}$$

On the other hand, the supplied power can also be computed from the field and surface currents enclosed in a volume:

$$P_s = -\int_V (\mathbf{H} \cdot \mathbf{M} + \mathbf{E} \cdot \mathbf{J}) dV. \tag{7.35}$$

In our case of PEC objects, the magnetic currents equal zero, the electric field is the considered incident electric field and the electric currents are the solution of the MoM system. Substituting the development of the currents in terms of the basis functions in (7.35), we can write:

$$P_s = -\sum_n J_n \int_{T_n^+ \bigcup T_n^-} \mathbf{E}^i(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}) dS = -\sum_n J_n \left\langle \mathbf{f}_n, \mathbf{E}^i \right\rangle = -\sum_n J_n E_n \tag{7.36}$$

From (7.34) and (7.36) and isolating V_j we get the relation:

$$V_j = -\frac{\sum_n J_n E_n}{I_j} \tag{7.37}$$

and substituting (7.37) into (7.32):

$$Y_{ij} = -\left.\frac{I_i I_j}{\sum_n J_n E_n}\right|_{V_k=0 \quad \forall k \neq j} . \tag{7.38}$$

It still remains the computation of the current at each port I_k. We are going to show that it can be computed from the δ-gap incident field associated to the corresponding port k, independently of the actual used source:

$$I_k = \sum_n J_n E_n^{\delta\text{-gap (port } k)}. \tag{7.39}$$

This is the reason why it is so important to have a good and versatile representation of the δ-gap source. The surface equivalent current is

$$\mathbf{J}_s(\mathbf{r}) = \sum_n J_n \mathbf{f}_n(\mathbf{r}) \quad [\text{A/m}]. \tag{7.40}$$

However, it considers that the current is defined over an infinitely thin layer and to have the real current density we need to multiply by a δ distribution in the direction normal to the conductor. In particular, for RWG basis functions and using the same notation as at the starting of the chapter:

$$\mathbf{J}(\mathbf{r}) = \sum_n J_n \mathbf{f}_n(\mathbf{r}) \delta(n) \quad [\text{A/m}^2]. \tag{7.41}$$

Then, the current crossing the port is computed integrating the current density over a surface with directions $\hat{\mathbf{n}}_T$ and $\hat{\mathbf{u}}$ close to each feeding triangle:

$$I_k = \int_{S_k} \mathbf{J} \cdot d\mathbf{s} = \sum_n J_n \int_0^l \mathbf{f}_n(\mathbf{r}) \cdot \hat{\mathbf{v}} \, du = \sum_n J_n E_n^{\delta\text{-gap (port } k)} \tag{7.42}$$

as we wanted to prove. The same can also be obtained for wires, considering that the current density is now:

$$\mathbf{J}(\mathbf{r}) = \sum_n J_n \mathbf{f}_n(\mathbf{r}) \delta(\rho - a) \quad [\text{A/m}^2]. \tag{7.43}$$

Summarizing,

$$Y_{ij} = -\frac{\left(\sum_n J_n E_n^{\delta\text{-gap (port } i)}\right)\left(\sum_n J_n E_n^{\delta\text{-gap (port } j)}\right)}{\sum_n J_n E_n} \tag{7.44}$$

where E_n is the tested incident field when only the port j is excited, J_n are the coefficients solution of the MoM for said incident field and $E_n^{\delta\text{-gap (port } i)}$ and $E_n^{\delta\text{-gap (port } j)}$ are the tested fields when a

δ-gap incident field is considered at ports i and j, respectively. Finally, $[\mathbf{Z}]$ is obtained inverting the matrix $[\mathbf{Y}]$.

7.4 S-parameters

Voltages and currents are a priori difficult to measure directly at microwave frequencies. The Z-parameters matrix requires ideal open-circuits, which are hard to obtain due to parasitic capacitances and radiation. Likewise, the Y matrix requires short-circuits, which are impossible to obtain at high frequencies due to the finite inductance. Furthermore, many active devices could oscillate under the open or short terminations. On the contrary, the measurement of S-parameters only involves measurement of relative quantities, such as the SWR or the location of the first minima relative to the load, which can be done for any frequency.

The S-parameters matrix $[\mathbf{S}]$ is defined as

$$\mathbf{b} = [\mathbf{S}]\mathbf{a} \tag{7.45}$$

where the vectors $\mathbf{a}$ and $\mathbf{b}$ are the so-called incident and reflected power waves, respectively, and are in turn defined as

$$\begin{aligned} \mathbf{a} &= \frac{1}{2\sqrt{Z_0}}(V + Z_0 I) \\ \mathbf{b} &= \frac{1}{2\sqrt{Z_0}}(V - Z_0 I) \end{aligned} \tag{7.46}$$

being Z_0 the reference load with which all the ports are loaded.

The matrix $[\mathbf{S}]$ can be easily obtained from the Z-parameters matrix with the expression [96]

$$[\mathbf{S}] = ([\mathbf{Z}] + Z_0[\mathbf{Id}])^{-1}([\mathbf{Z}] - Z_0[\mathbf{Id}]) \tag{7.47}$$

where $[\mathbf{Id}]$ is the $N \times N$ identity matrix.

7.5 Auto-impedance or input impedance

In this case we are interested in the impedance that an external observer to the multi-port system is seeing at port i, $Z_{\text{in}}^{(i)}$, when the rest of the ports $j \neq i$ are loaded with a predefined load Z_{l_j}, utilizing the S-parameters matrix $[\mathbf{S}]$. This impedance can be computed in terms of the reflexion coefficient $\Gamma_{\text{in}}^{(i)}$ as

$$Z_{\text{in}}^{(i)} = Z_0 \frac{1 + \Gamma_{\text{in}}^{(i)}}{1 - \Gamma_{\text{in}}^{(i)}}. \tag{7.48}$$

The reflexion coefficient is in turn calculated using the S parameters with the following expression:

$$\Gamma_{\text{in}}^{(i)} = S_{i,i} + \mathbf{S}_{(i)}^{H}[\mathbf{\Gamma}]_{(i)}([\mathbf{Id}] - [\mathbf{S}]_{(i)(i)}[\mathbf{\Gamma}]_{(i)})^{-1}\mathbf{S}_{(i)}^{V} \tag{7.49}$$

where $[\mathbf{S}]_{(i)(i)}$ is the $(N-1) \times (N-1)$ sub-matrix of $[\mathbf{S}]$ in which row and column i have been removed, $\mathbf{S}_{(i)}^{H}$ is the i-th row of $[\mathbf{S}]$ removing the i-th element, $\mathbf{S}_{(i)}^{V}$ is the i-th column of $[\mathbf{S}]$ removing the i-th element and $[\mathbf{\Gamma}]_{(i)}$ is a diagonal matrix in which the diagonal equals the outer reflexion coefficients Γ_{l_j} at each port except the port i:

$$[\mathbf{S}] = \begin{bmatrix} S_{1,1} & \dots & S_{1,N} \\ \vdots & \ddots & \vdots \\ S_{N,1} & \dots & S_{N,N} \end{bmatrix} \tag{7.50}$$

$$[\mathbf{S}]_{(i)(i)} = \begin{bmatrix} S_{1,1} & \dots & S_{1,i-1} & S_{1,i+1} & \dots & S_{1,N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{i-1,1} & \dots & S_{i-1,i-1} & S_{i-1,i+1} & \dots & S_{i-1,N} \\ S_{i+1,1} & \dots & S_{i+1,i+1} & S_{i+1,i+1} & \dots & S_{i+1,N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{N,1} & \dots & S_{N,i+1} & S_{N,i+1} & \dots & S_{N,N} \end{bmatrix} \tag{7.51}$$

$$\mathbf{S}^{H}_{(i)} = \begin{bmatrix} S_{i,1} & \dots & S_{i,i-1} & S_{i,i+1} & \dots & S_{i,N} \end{bmatrix} \tag{7.52}$$

$$\mathbf{S}^{V}_{(i)} = \begin{bmatrix} S_{1,i} \\ \vdots \\ S_{i-1,i} \\ S_{i+1,i} \\ \vdots \\ S_{N,i} \end{bmatrix} \tag{7.53}$$

$$[\mathbf{\Gamma}]_{(i)} = \begin{bmatrix} \Gamma_{l_1} & & & & & \\ & \ddots & & & & \\ & & \Gamma_{l_{i-1}} & & & \\ & & & \Gamma_{l_{i+1}} & & \\ & & & & \ddots & \\ & & & & & \Gamma_{l_N} \end{bmatrix} \tag{7.54}$$

$$\Gamma_{l_j} = \frac{Z_{l_j} - Z_0}{Z_{l_j} + Z_0} \tag{7.55}$$

There are a couple of particular cases which are worth a special mention. The first one is the bi-port system in which the expression for $\Gamma^{(1)}_{\text{in}}$ is reduced to

$$\Gamma^{(1)}_{\text{in}} = S_{1,1} + \frac{S_{1,2}S_{2,1}\Gamma_{l_1}}{1 - S_{2,2}\Gamma_{l_1}}. \tag{7.56}$$

The second one is when all the ports are loaded with the reference impedance Z_0 (matched ports), in which the reflexion coefficient $\Gamma^{(1)}_{\text{in}}$ coincides with the S-parameter $S_{i,i}$.

7.6 Numerical results

To assess the validity of the presented scheme to treat ports defined arbitrarily over a discretization, independently of the position of the discretized elements, we have selected a 0.5λ dipole with

Table 7.1: Input impedance and S-parameter of a 0.5λ height dipole with radius 0.001 with different meshes and feeding models.

	Z_{in}	S ($Z_0 = 50$)
δ-gap Wires Mesh1	85.85+45.69j	0.3387+0.2224j
δ-gap Wires Mesh2	84.79+46.50j	0.3370+0.2287j
Mag. Frill Wires Mesh1	85.78+45.74j	0.3386+0.2228j
Mag. Frill Wires Mesh2	84.79+46.50j	0.3370+0.2287j
δ-gap RWG Mesh1	86.21+47.56j	0.3456+0.2285j
δ-gap RWG Mesh2	85.49+48.03j	0.3444+0.2324j
Mag. Frill RWG Mesh1	85.74+47.85j	0.3447+0.2310j
Mag. Frill RWG Mesh2	85.45+48.06j	0.3443+0.2327j

a radius of 0.001m being the wavelength $\lambda = 1$m. For the wires model, two different meshes have been selected, mesh1 and mesh2, dividing the dipole in 50 and 51 segments, respectively. Mesh1 has a vertex segment exactly at the central point of the dipole where the feeding port is placed. Therefore, it can be considered as the usual case found in literature. Oppositely, mesh2 has no vertex at the feeding point, and the central segment crosses the feeding in the middle point. For the RWG model, the dipole has been considered with a square section and again two different meshes, mesh1 and mesh2, with the same properties as the in the wire model, have been selected.

The resulting input impedance and S parameter for the different aforementioned models and discretizations of the dipole are shown in Table 7.1. We can observe a very good agreement between the different simulations, considering that the dipole is slightly different in the wires model (cylindrical) and the RWG model (square section) and the effect of the different discretizations. Therefore, the developed method allows a freedom to the user to define the port position and orientation without having to care about the mesh or any mesh refinement.

7.7 Conclusion

It has been highlighted the importance of having an accurate and versatile port definition and treatment in which no meshing restrictions are necessary to follow by a hypothetical user. The approach presented here is completely general and avoids any mesh refinement, making it easier to integrate in the framework of independent routines developed by different people. The obtention of the most common parameters used in antenna design from the method of moments system solution has been reviewed. A relevant dipole example has been analyzed to show the correctness of the proposed scheme.

Thesis Conclusions

Chapter 8

Conclusions and discussion

This chapter summarizes to which extent the author has contributed to the electromagnetic (EM) community. In particular, to the Method of Moments (MoM), as one of the main subjects inside numerical simulation techniques for the study of EM problems. Then, a number of hints for further development or future work is given.

8.1 Conclusions

As stated in Chapter 1, there are several open issues regarding the solution of electromagnetic problems with the MoM. Many of them have been addressed and solved with success in the present document. In the following, the most important concluding remarks are stated:

1. We wanted a fast iterative solver for the MoM electromagnetic problems which was purely algebraic, independent of the Green's function and with improved computational cost and memory requirements. The Multilevel Adaptive Cross Approximation (MLACA) has been developed:

    - The MLACA is a purely algebraic algorithm as it only uses several ACA and SVD decompositions which are purely algebraic algorithms themselves.
    - It is independent of the GF of the problem as long as the MoM matrix is block-wise compressible, which is the most common case in electromagnetics and other physical fields. It is basically a consequence of the decreasing of the GF with the distance.
    - It has been proved to have a memory requirement growing with $O(N \log^2 N)$, in contrast with its single-level counterpart (ACA) which grows with $O(N^2 \log N)$.
    - A matrix-vector product, and therefore the CPU time per iteration, scales with $O(N \log^2 N)$ against the $O(N^2 \log N)$ of the ACA.
    - The computational cost for the creation of the matrix scales with $O(N^2 \log N)$, whereas it was $O(N^3 \log N)$ for the ACA.
    - The accuracy due to the solving method can be reduced as much as desired, with the correspondent increase in computational cost.
    - Numerical examples have corroborated the advantages of utilizing the MLACA for electromagnetic problems with around one million of unknowns.
    - Based on the theory and on the numerical examples with large separated groups of samples, a much larger gain is expected for larger problems.

2. We wanted to tackle the fast and accurate computation of the hyper-singular integrals arising in the MoM matrix elements of the MFIE and CFIE formulations. The direct evaluation method has been extended to this sort of integrals:

    - The direct evaluation method has been proved to reach machine precision much faster than any previously existing technique.
    - It has been presented so that it can be implemented and used by EM community without a great effort. Even a fully numerical and very simple implementation is possible with a small reduction in the efficiency.
    - It is robust against reasonably distorted meshes.
    - It improves total accuracy and CPU time of a MoM electromagnetic problem.

3. The non-singular but near-hyper-singular integrals arising in the MoM matrix elements of the MFIE and CFIE formulations for closely-placed non-touching elements needed a detailed analysis and treatment. It has been properly analyzed and solved when necessary.

    - Numerical examples have localized the source of the problem in the outer 2-D integration although the singularity has been previously canceled out.
    - The Double Exponential-Gauss Legendre (DE-GL) improves the performance of DE-DE approximately by half the time.
    - The Gauss-Laguerre integration technique is presented as a good alternative to the DE technique, and better for a very small number of integration points.
    - Only extremely close elements need a special treatment.

4. An extra freedom to the user was desired when defining a mesh in which a port will be defined to apply the MoM. A generalization of the δ-gap feeding together with a justification of its use in the determination of certain system parameters of practical importance have been presented.

8.2 Summary of Contributions

This section comprises a list with the main author contributions during the thesis realization period. The most important remarks of several contributions which are only referenced in the present document, but not developed, are also included.

1. Developed a fast and accurate algorithm, the MLACA, for the iterative solution of large-scale MoM problems.

2. Demonstrated the improved performance of MLACA with respect to ACA, keeping its best characteristics.

3. Developed a fast and accurate technique for the calculation of hyper-singular integrals of dynamic MoM problems with MFIE and CFIE formulations. Obtained up to machine precision, which is the best one could do, in a more than reasonable time.

4. Demonstrated a much better performance in the integration than any other existing technique.

5. Developed a possible strategy to deal with all the integrals arising in MFIE and CFIE formulations.
6. Proved the correctness of a new generalized port definition independent of the mesh in its vicinity.
7. Developed a new accelerated technique for the computation of the static part of a rectangular cavity GF, with an efficiency comparable to Ewald method, but with a physical interpretation.
8. Developed a new set of basis functions, the Self-Loop, to overcome the low frequency breakdown of the EFIE when Linear-Linear basis functions are used.
9. Introduction of the orthogonal basis functions for the improvement of the accuracy of MFIE for sharp-edged objects.
10. A new integral equation formulation, the EMFIE, is developed.
11. A useful direct method, the MSCBD, for the direct solution of large-scale MoM problems is developed.
12. Improved the MDA iterative solver performance with the introduction of a SVD re-compression.

8.3 Future Research

- A systematical approach to obtain the optimum value for the parameters in the MLACA algorithm is desirable. As its behavior basically depends on the computer and system themselves, a determination with the previous simulations of some relevant test cases is presented as the best option. This overload should be performed only once per computer so its contribution to the final computational cost is irrelevant.
- The MLACA is somehow optimum in terms of asymptotical compression. However, the computational cost could be further improved, if a technique less costly than the ACA was used. Perhaps a fusion of MLMDA and MLACA could improve the efficiency while maintaining compression and robustness.
- The efficiency of the computation of near-singular integrals arising in MFIE and CFIE formulations for closely spaced triangle elements might be further improved with a more sophisticated technique.
- A generalization of the direct evaluation method to higher order basis functions remains open. Based on the theory developed during this document it should be practically direct. The only difference is the expression of the basis functions in the new parameters spaces, but the kernel and therefore the singularity cancellation remains equal.
- The direct evaluation method can be extended to polygonal domain meshes other than triangles. It should be doable with slight changes in the formulation.
- Finally, a more challenging issue is the generalization of the direct evaluation method to non-planar meshes. In this case, it should be studied how the non-linear behavior affects to the formulation.

Appendix A

Binary Tree Domain Decomposition

Let us formalize the procedure in section 3.2.1 for our particular case. The object is discretized in a set of N basis (or testing) functions which are defined in a reduced portion of the space (local subdomain functions):

$$F = \{\mathbf{f}_1(\mathbf{r}), \ldots, \mathbf{f}_N(\mathbf{r})\} \,. \tag{A.1}$$

Note that it suffices to deal with the indexes because each index is uniquely related to a basis function. A center point is assigned to each basis function which represents the position of the geometrical domain of the function. Its definition slightly depends on the basis functions we use (RWG on a triangular mesh, rooftops, ...), but the idea is clear. This point in the space will determine the position of the function when the decomposition is done. The subsets of indexes of the decomposition can be computed with the following simple recursion:

$$\begin{cases} T_0^{(0)} = \{1, \ldots, N\} \\ T_{2i}^{(l)}, T_{2i+1}^{(l)} = \textbf{splitBox}(T_i^{(l-1)}) \quad l = (1, \ldots, L), \forall i \end{cases} \tag{A.2}$$

where $T_i^{(l)}$ is the box i at level l and L is the total number of levels of the decomposition. These sets need to fulfil the following properties:

$$T_{2i}^{(l)} \cup T_{2i+1}^{(l)} = T_i^{(l-1)} \text{ and } T_{2i}^{(l)} \cap T_{2i+1}^{(l)} = \emptyset \tag{A.3}$$

The breaking function **splitBox** works as follows: a minimum size cuboid (oriented with the axes) containing all the center points of its box argument is determined. Then, it is split in half along its longest dimension, getting the two new sets of indexes. The reason to cut the box with that criterion is that we want the largest box at a given level to be as small as possible. The total number of levels L is determined by applying the recursion until the sizes of the boxes at that level are smaller than a certain quantity. That quantity is a first parameter to the algorithm.

Appendix B

Orthonormal Matrices Properties

There are a good number of properties which orthonormal matrices accomplish. However, we are going to show just three important and easy to proof properties which are applied in the MLACA algorithm. Before starting, let us define an orthonormal matrix. A general complex matrix U is orthonormal if and only if $U^H U = I$, being U^H the U matrix transposed and conjugated, and I the identity matrix with the appropriate dimensions.

The first property is that the product of orthonormal matrices is in turn an orthonormal matrix.

Proof. Let be U_1 and U_2 two orthonormal matrices. We want to proof that the matrix product $U = U_1 U_2$ is orthonormal as well. To do so, it is only necessary to check that $U^H U = I$:

$$U^H U = (U_1 U_2)^H (U_1 U_2) = U_2^H U_1^H U_1 U_2 = \tag{B.1}$$

$$= U_2^H I U_2 = U_2^H U_2 = I \tag{B.2}$$

□

The second property is that a block-diagonal matrix U

$$U = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_N \end{bmatrix} \tag{B.3}$$

where the blocks in the diagonal U_i are orthonormal, is also orthonormal.

Proof.

$$U^H U = \begin{bmatrix} U_1^H & & \\ & \ddots & \\ & & U_N^H \end{bmatrix} \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_N \end{bmatrix} \tag{B.4}$$

$$= \begin{bmatrix} U_1^H U_1 & & \\ & \ddots & \\ & & U_N^H U_N \end{bmatrix} = I \tag{B.5}$$

□

And the last, but not the least is that the matrices A and $A\widetilde{V}^H$ being $\widetilde{V}$ orthonormal, have exactly the same singular values and therefore, it is equivalent to do a compression with a certain threshold τ in one or the other.

Proof. Let be $A = USV^H$ the singular value decomposition of the matrix A. Then

$$A\widetilde{V}^H = USV^H\widetilde{V}^H = US(\widetilde{V}V)^H \tag{B.6}$$

which is the unique singular value decomposition of the matrix $A\widetilde{V}^H$, being S the singular values matrix and U and $\widetilde{V}V$ the outer orthonormal matrices. So S is the singular values matrix in both cases A and $A\widetilde{V}^H$ as we wanted to prove. □

Appendix C

Basis functions definitions

C.1 Edge adjacent changes

In equation (5.5) we accept different possibilities for the functions $\mathbf{g}$ and $\mathbf{f}$. $\mathbf{f}$ could be any of the RWG basis functions $\mathbf{f}'_1(\mathbf{r}')$, $\mathbf{f}'_2(\mathbf{r}')$ or $\mathbf{f}'_3(\mathbf{r}')$ below and $\mathbf{g}$ could be any of the RWG weighting functions $\mathbf{f}_1(\mathbf{r})$, $\mathbf{f}_2(\mathbf{r})$ or $\mathbf{f}_3(\mathbf{r})$ or their $\hat{\mathbf{n}}$ cross versions $\mathbf{g}_1(\mathbf{r})$, $\mathbf{g}_2(\mathbf{r})$ or $\mathbf{g}_3(\mathbf{r})$ being $\hat{\mathbf{n}}$ the unitary normal vector of the triangle P corresponding to the outer integral.

$$\begin{aligned}
\mathbf{f}'_1(\mathbf{r}') &= \frac{|\mathbf{r}_4 - \mathbf{r}_1|}{2A_q}(\mathbf{r}' - \mathbf{r}_2) \\
\mathbf{f}'_2(\mathbf{r}') &= \frac{|\mathbf{r}_4 - \mathbf{r}_2|}{2A_q}(\mathbf{r}' - \mathbf{r}_1) \\
\mathbf{f}'_3(\mathbf{r}') &= \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{2A_q}(\mathbf{r}' - \mathbf{r}_4) \\
\mathbf{f}_1(\mathbf{r}) &= \frac{|\mathbf{r}_3 - \mathbf{r}_2|}{2A_p}(\mathbf{r} - \mathbf{r}_1) \\
\mathbf{f}_2(\mathbf{r}) &= \frac{|\mathbf{r}_3 - \mathbf{r}_1|}{2A_p}(\mathbf{r} - \mathbf{r}_2) \\
\mathbf{f}_3(\mathbf{r}) &= \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{2A_p}(\mathbf{r} - \mathbf{r}_3) \\
\mathbf{g}_1(\mathbf{r}) &= \hat{\mathbf{n}} \times \mathbf{f}_1(\mathbf{r}) \\
\mathbf{g}_2(\mathbf{r}) &= \hat{\mathbf{n}} \times \mathbf{f}_2(\mathbf{r}) \\
\mathbf{g}_3(\mathbf{r}) &= \hat{\mathbf{n}} \times \mathbf{f}_3(\mathbf{r})
\end{aligned} \tag{C.1}$$

After applying all the parametric transformations defined in section 5.3 and removing the constants $l_n/(2A)$ we finally obtain the following expressions for the possible basis and weighting functions:

$$\begin{aligned}
\mathbf{f}'_1(\eta,\theta,\Lambda,\Psi) &\propto -\boldsymbol{\alpha}_{e_1} + \boldsymbol{\alpha}_{e_1}\eta - (\boldsymbol{\alpha}_{e_1}\cos(\theta) + \boldsymbol{\alpha}_{e_3}\sin(\theta))\cos(\Psi)\Lambda \\
\mathbf{f}'_2(\eta,\theta,\Lambda,\Psi) &\propto \boldsymbol{\alpha}_{e_1} + \boldsymbol{\alpha}_{e_1}\eta - (\boldsymbol{\alpha}_{e_1}\cos(\theta) + \boldsymbol{\alpha}_{e_3}\sin(\theta))\cos(\Psi)\Lambda \\
\mathbf{f}'_3(\eta,\theta,\Lambda,\Psi) &\propto \sqrt{3}\boldsymbol{\alpha}_{e_3} + \boldsymbol{\alpha}_{e_1}\eta - (\boldsymbol{\alpha}_{e_1}\cos(\theta) + \boldsymbol{\alpha}_{e_3}\sin(\theta))\cos(\Psi)\Lambda \\
\mathbf{f}_1(\eta,\theta,\Lambda,\Psi) &\propto \boldsymbol{\alpha}_{e_1} + \boldsymbol{\alpha}_{e_1}\eta + \boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda \\
\mathbf{f}_2(\eta,\theta,\Lambda,\Psi) &\propto -\boldsymbol{\alpha}_{e_1} + \boldsymbol{\alpha}_{e_1}\eta + \boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda \\
\mathbf{f}_3(\eta,\theta,\Lambda,\Psi) &\propto -\sqrt{3}\boldsymbol{\alpha}_{e_2} + \boldsymbol{\alpha}_{e_1}\eta + \boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda
\end{aligned} \tag{C.2}$$

$$\begin{aligned}
\mathbf{g}_1(\eta,\theta,\Lambda,\Psi) &\propto \hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\eta+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda\\
\mathbf{g}_2(\eta,\theta,\Lambda,\Psi) &\propto -\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\eta+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda\\
\mathbf{g}_3(\eta,\theta,\Lambda,\Psi) &\propto -\sqrt{3}\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\eta+\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\sin(\Psi)\Lambda
\end{aligned} \tag{C.3}$$

As a curiosity, note that the integral (5.5) is always zero when the basis and weighting functions are $\mathbf{f}_1(\mathbf{r})$ and $\mathbf{f}'_2(\mathbf{r}')$ or $\mathbf{f}_2(\mathbf{r})$ and $\mathbf{f}'_1(\mathbf{r}')$, respectively, independently of the triangles vertices. It can be seen with the following development:

$$\mathbf{f}_1\cdot(\mathbf{R}\times\mathbf{f}'_2)=\mathbf{f}_1\cdot\left(\left(\frac{2A_p}{|\mathbf{r}_3-\mathbf{r}_2|}\mathbf{f}_1-\frac{2A_q}{|\mathbf{r}_4-\mathbf{r}_2|}\mathbf{f}'_2\right)\times\mathbf{f}'_2\right)=\frac{2A_p}{|\mathbf{r}_3-\mathbf{r}_2|}\mathbf{f}_1\cdot\left(\mathbf{f}_1\times\mathbf{f}'_2\right)=0 \tag{C.4}$$

$$\mathbf{f}_2\cdot(\mathbf{R}\times\mathbf{f}'_1)=\mathbf{f}_2\cdot\left(\left(\frac{2A_p}{|\mathbf{r}_3-\mathbf{r}_1|}\mathbf{f}_2-\frac{2A_q}{|\mathbf{r}_4-\mathbf{r}_1|}\mathbf{f}'_1\right)\times\mathbf{f}'_1\right)=\frac{2A_p}{|\mathbf{r}_3-\mathbf{r}_1|}\mathbf{f}_2\cdot\left(\mathbf{f}_2\times\mathbf{f}'_1\right)=0 \tag{C.5}$$

where it has been used that the cross product of two parallel vectors and the dot product of orthogonal vectors are zero.

C.2 Vertex adjacent changes

Similarly to the edge adjacent case, in equation (5.5) we accept different possibilities for the functions $\mathbf{g}$ and $\mathbf{f}$. $\mathbf{f}$ could be any of the RWG basis functions $\mathbf{f}'_1(\mathbf{r}')$, $\mathbf{f}'_2(\mathbf{r}')$ or $\mathbf{f}'_3(\mathbf{r}')$ below and $\mathbf{g}$ could be any of the RWG weighting functions $\mathbf{f}_1(\mathbf{r})$, $\mathbf{f}_2(\mathbf{r})$ or $\mathbf{f}_3(\mathbf{r})$ or their $\hat{\mathbf{n}}$ cross versions $\mathbf{g}_1(\mathbf{r})$, $\mathbf{g}_2(\mathbf{r})$ or $\mathbf{g}_3(\mathbf{r})$ being $\hat{\mathbf{n}}$ the unitary normal vector of the triangle P corresponding to the outer integral.

$$\begin{aligned}
\mathbf{f}'_1(\mathbf{r}') &= \frac{|\mathbf{r}_4-\mathbf{r}_3|}{2A_q}(\mathbf{r}'-\mathbf{r}_4)\\
\mathbf{f}'_2(\mathbf{r}') &= \frac{|\mathbf{r}_5-\mathbf{r}_3|}{2A_q}(\mathbf{r}'-\mathbf{r}_5)\\
\mathbf{f}'_3(\mathbf{r}') &= \frac{|\mathbf{r}_4-\mathbf{r}_5|}{2A_q}(\mathbf{r}'-\mathbf{r}_3)\\
\mathbf{f}_1(\mathbf{r}) &= \frac{|\mathbf{r}_3-\mathbf{r}_2|}{2A_p}(\mathbf{r}-\mathbf{r}_1)\\
\mathbf{f}_2(\mathbf{r}) &= \frac{|\mathbf{r}_3-\mathbf{r}_1|}{2A_p}(\mathbf{r}-\mathbf{r}_2)\\
\mathbf{f}_3(\mathbf{r}) &= \frac{|\mathbf{r}_2-\mathbf{r}_1|}{2A_p}(\mathbf{r}-\mathbf{r}_3)\\
\mathbf{g}_1(\mathbf{r}) &= \hat{\mathbf{n}}\times\mathbf{f}_1(\mathbf{r})\\
\mathbf{g}_2(\mathbf{r}) &= \hat{\mathbf{n}}\times\mathbf{f}_2(\mathbf{r})\\
\mathbf{g}_3(\mathbf{r}) &= \hat{\mathbf{n}}\times\mathbf{f}_3(\mathbf{r})
\end{aligned} \tag{C.6}$$

After applying all the parametric transformations defined in section 5.4 and removing the constants $l_n/(2A)$ we finally obtain the following expressions for the possible basis and weighting functions:

$$
\begin{aligned}
\mathbf{f}'_1(\theta_p,\theta_q,\Psi,\Lambda) &\propto (\mathbf{r}_3-\mathbf{r}_4)+\sin(\Psi)(\boldsymbol{\alpha}_{e_4}\sin(\theta_q)-\boldsymbol{\alpha}_{e_5}\cos(\theta_q))\Lambda\\
\mathbf{f}'_2(\theta_p,\theta_q,\Psi,\Lambda) &\propto (\mathbf{r}_3-\mathbf{r}_5)+\sin(\Psi)(\boldsymbol{\alpha}_{e_4}\sin(\theta_q)-\boldsymbol{\alpha}_{e_5}\cos(\theta_q))\Lambda\\
\mathbf{f}'_3(\theta_p,\theta_q,\Psi,\Lambda) &\propto \sin(\Psi)(\boldsymbol{\alpha}_{e_4}\sin(\theta_q)-\boldsymbol{\alpha}_{e_5}\cos(\theta_q))\Lambda\\
\mathbf{f}_1(\theta_p,\theta_q,\Psi,\Lambda) &\propto (\mathbf{r}_3-\mathbf{r}_1)+\cos(\Psi)(\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda\\
\mathbf{f}_2(\theta_p,\theta_q,\Psi,\Lambda) &\propto (\mathbf{r}_3-\mathbf{r}_2)+\cos(\Psi)(\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda\\
\mathbf{f}_3(\theta_p,\theta_q,\Psi,\Lambda) &\propto \cos(\Psi)(\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda\\
\mathbf{g}_1(\theta_p,\theta_q,\Psi,\Lambda) &\propto \hat{\mathbf{n}}\times(\mathbf{r}_3-\mathbf{r}_1)+\cos(\Psi)(\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda\\
\mathbf{g}_2(\theta_p,\theta_q,\Psi,\Lambda) &\propto \hat{\mathbf{n}}\times(\mathbf{r}_3-\mathbf{r}_2)+\cos(\Psi)(\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda\\
\mathbf{g}_3(\theta_p,\theta_q,\Psi,\Lambda) &\propto \cos(\Psi)(\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_1}\sin(\theta_p)-\hat{\mathbf{n}}\times\boldsymbol{\alpha}_{e_2}\cos(\theta_p))\Lambda
\end{aligned}
\tag{C.7}
$$

Appendix D

Implementation of $N_i^\alpha(B,\theta,\Psi)$ functions

To compute the $N_i^\alpha(B,\theta,\Psi)$ in an efficient and compact way, we first introduce the following functions:

$$\begin{aligned}
\Delta^a(\theta,\Psi) &= \frac{2\sqrt{3}}{\cos(\Psi)(\sin(\theta)+\sqrt{3}\cos(\theta)+\tan(\Psi))}\\
\Delta^b(\theta,\Psi) &= \frac{\sqrt{3}}{\cos(\Psi)\sin(\theta)}\\
\Delta^c(\theta,\Psi) &= \frac{\sqrt{3}}{\sin(\Psi)}\\
\Delta^d(\theta,\Psi) &= \frac{\sqrt{3}}{\cos(\Psi)(\sin(\theta)-\sqrt{3}\cos(\theta))}.
\end{aligned} \tag{D.1}$$

The aimed functions can then be calculated in terms of two auxiliary functions $V_i\left(B,\Delta_1,\Delta_2\right)$ and $U_i\left(B,\Delta\right)$ (actually one, considering $U_i\left(B,\Delta\right) = V_i\left(B,\Delta,\Delta\right)$) which are defined in Tables D.1 and D.2, respectively, as

$$\begin{aligned}
N_i^a(B,\theta,\Psi) &= V_i\left(B,\Delta^a(\theta,\Psi),\Delta^c(\theta,\Psi)\right)\\
N_i^b(B,\theta,\Psi) &= V_i\left(B,\Delta^b(\theta,\Psi),\Delta^d(\theta,\Psi)\right)\\
N_i^c(B,\theta,\Psi) &= U_i\left(B,\Delta^c(\theta,\Psi)\right)\\
N_i^d(B,\theta,\Psi) &= U_i\left(B,\Delta^d(\theta,\Psi)\right).
\end{aligned} \tag{D.2}$$

Table D.1: Functions $V_i(B,\Delta_1,\Delta_2)$.

$$V_i(B,\Delta_1,\Delta_2)=\int_0^H M_i\left(\eta,\frac{\Delta_1}{1+H}(1+\eta),B\right)d\eta$$

$$+\int_H^1 M_i\left(\eta,\Delta_2(1-\eta),B\right)d\eta,\qquad H=1-\frac{\Delta_1}{\Delta_2}$$

$$a=jkB;\ H=1-\frac{\Delta_1}{\Delta_2};\ \Delta_3=\frac{\Delta_1}{1+H}$$

$$V_1=1;\ V_6=\frac{1}{2};\ V_7=\frac{1}{3}$$

$T_1^1=\frac{e^{-a\Delta_3}-e^{-a\Delta_1}}{a\Delta_3}$	$T_1^2=\frac{1-e^{-a\Delta_1}}{a\Delta_2}$
$T_2^1=\frac{T_1^1-He^{-a\Delta_1}}{a\Delta_3}$	$T_2^2=\frac{1-He^{-a\Delta_1}-T_1^2}{a\Delta_2}$
$T_3^1=\frac{2T_2^1-H^2e^{-a\Delta_1}}{a\Delta_3}$	$T_3^2=\frac{1-H^2e^{-a\Delta_1}-2T_2^2}{a\Delta_2}$
$T_4^1=\frac{3T_3^1-H^3e^{-a\Delta_1}}{a\Delta_3}$	$T_4^2=\frac{1-H^3e^{-a\Delta_1}-3T_3^2}{a\Delta_2}$
$V_2^1=T_1^1$	$V_2^2=T_1^2$
$V_3^1=\Delta_3(T_1^1+T_2^1)$	$V_3^2=\Delta_2(T_1^2-T_2^2)$
$V_8^1=T_2^1$	$V_8^2=T_2^2$
$V_9^1=T_3^1$	$V_9^2=T_3^2$
$V_{10}^1=\Delta_3(T_2^1+T_3^1)$	$V_{10}^2=\Delta_2(T_2^2-T_3^2)$
$V_{11}^1=\Delta_3(T_3^1+T_4^1)$	$V_{11}^2=\Delta_2(T_3^2-T_4^2)$
$V_{12}^1=\Delta_3(V_{10}^1+V_{11}^1)$	$V_{12}^2=\Delta_2(V_{10}^2-V_{11}^2)$
$V_4^1=\Delta_3(V_3^1+V_{10}^1)$	$V_4^2=\Delta_2(V_3^2-V_{10}^2)$
$V_5^1=\Delta_3(V_4^1+V_{12}^1)$	$V_5^2=\Delta_2(V_4^2-V_{12}^2)$

$$V_i=V_i^1+V_i^2\qquad i\in\{2,3,4,5,8,9,10,11,12\}$$

Table D.2: Functions $U_i(B, \Delta)$.

$$U_i(B, \Delta) = V_i(B, \Delta, \Delta) = \int_0^1 M_i(\eta, \Delta(1-\eta), B)d\eta$$

$$a = jkB$$

$T_1 = \frac{1-e^{-a\Delta}}{a\Delta}$	$U_3 = \Delta(T_1 - T_2)$	$U_{11} = \Delta(T_3 - T_4)$
$T_2 = \frac{1-T_1}{a\Delta}$	$U_6 = \frac{1}{2}$	$U_{12} = \Delta(U_{10} - U_{11})$
$T_3 = \frac{1-2T_2}{a\Delta}$	$U_7 = \frac{1}{3}$	$U_4 = \Delta(U_3 - U_{10})$
$T_4 = \frac{1-3T_3}{a\Delta}$	$U_8 = T_2$	$U_5 = \Delta(U_4 - U_{12})$
$U_1 = 1$	$U_9 = T_3$	
$U_2 = T_1$	$U_{10} = \Delta(T_2 - T_3)$	

Bibliography

[1] A. J. Poggio and E. K. Miller, *Integral equation solutions of three dimensional scattering problems.* in Computer Techniques for Electromagnetics, R. Mittra, Ed. Oxford, U.K.: Permagon, 1973.

[2] J. R. Mosig, R. C. Hall, and F. E. Gardiol, *Numerical analysis of microstrip patch antennas.* in Handbook of Microstrip Antennas, James and Hall, Eds. London: IEE-Peter Peregrinus, 1989.

[3] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and efficient algorithms in computational electromagnetics.* Boston, MA: Artech House, 2001.

[4] R. F. Harrington, *Field computation by moment methods.* New York: Macmillan, FL, Krieger, 1983.

[5] A. D. Yaghjian, "Augmented electric- and magnetic-field integral equations," *Radio Sci.*, vol. 16, no. 6, pp. 987–1001, 1981.

[6] Z. G. Qian and W. C. Chew, "Packaging modeling using fast broadband surface integral equation method," *Proc. IEEE 17th Topical Meeting on Electrical Performance of Electron. Packag.*, p. 347, 2008.

[7] M. Taskinen and P. Ylä-Oijala, "Current and charge Integral equation formulation," *IEEE Trans. Antennas Propagat.*, vol. 54, no. 1, pp. 58–67, Jan. 2006.

[8] Y. Saad, *Iterative Methods for Sparse Linear Systems.* Boston, MA: PWS, 1996.

[9] F. P. Andriulli, K. Cools, H. Bagci, F. Olyslager, A. Buffa, S. Christiansen, and E. Michielssen, "A Multiplicative Calderon Preconditioner for the Electric Field Integral Equation," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 8, pp. 2398–2412, Aug. 2008.

[10] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. 30, no. 5, pp. 409–418, May 1982.

[11] L. C. Trintinalia and H. Ling, "First order triangular patch basis functions for electromagnetic scattering analysis," *J. Electromagn. Waves Appl.*, vol. 15, pp. 1521–1537, 2001.

[12] Ö. Ergül and L. Gürel, "Linear-Linear Basis functions for MLFMA solutions of magnetic-field and combined-field integral equations," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 4, Apr. 2007.

[13] P. Ylä-Oijala, M. Taskinen, and S. Järvenpää, "Surface integral equation formulations for solving electromagnetic scattering methods with iterative methods," *Radio Science*, vol. 40, 2005.

[14] W. Wu, A. W. Glisson, and D. Kajfez, "A study of two numerical procedures for the Electric Field Integral Equation at low frequency," *Appl. Computat. Electromagn. Soc. J.*, vol. 10, no. 3, Nov. 1995.

[15] J. Lee, R. Lee, and R. J. Burkholder, "Loop Star Basis Functions and a Robust Preconditioner for EFIE Scattering Problems," *IEEE Trans. Antennas Propagat.*, vol. 51, no. 8, Aug. 2003.

[16] G. Vecchi, "Loop-star decomposition of basis functions in the discretization of the EFIE," *IEEE Trans. Antennas Propagat.*, vol. 47, no. 2, pp. 339–346, Feb. 1999.

[17] R. A. Wildman and D. S. Weile, "An accurate Broad-band method of moments using higher basis functions and Tree-Loop decomposition," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 11, Nov. 2004.

[18] A. G. Polimeridis and T. V. Yioultsis, "On the direct evaluation of weakly singular integrals in Galerkin mixed potential integral equation formulations," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 9, pp. 3011–3019, Sept. 2008.

[19] C. Schwab and W. L. Wendland, "On numerical cubatures of singular surface integrals in boundary element methods," *Numer. Math.*, vol. 62, pp. 343–369, 1992.

[20] S. Chakraborty and V. Jandhyala, "Evaluation of Green's function integrals in conducting media," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 12, pp. 3357–3363, Dec. 2004.

[21] H. X. Zhou, W. Hong, and G. Hua, "An accurate approach for the calculation of MoM matrix elements," *IEEE Trans. Antennas Propagat.*, vol. 54, no. 4, pp. 1185–1191, Apr. 2006.

[22] P. W. Fink, D. R. Wilton, and M. A. Khayat, "Simple and efficient numerical evaluation of near-hypersingular integrals," *IEEE Antennas Wireless Propagat. Lett.*, vol. 7, pp. 469–472, 2008.

[23] Ismatullah and T. F. Eibert, "Adaptive singularity cancellation for efficient treatment of near-singular and near-hypersingular integrals in surface integral equation formulations," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 1, pp. 274–278, Jan. 2008.

[24] S. Caorsi, D. Moreno, and F. Sidoti, "Theoretical and numerical treatment of surface integrals involving the free-space Green's function," *IEEE Trans. Antennas Propagat.*, vol. 41, no. 9, pp. 1296–1301, Sept. 1993.

[25] R. D. Graglia, "On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle," *IEEE Trans. Antennas Propagat.*, vol. 41, no. 10, pp. 1448–1455, Oct. 1993.

[26] T. F. Eibert and V. Hansen, "On the calculation of potential integrals for linear source distributions on triangular domains," *IEEE Trans. Antennas Propagat.*, vol. 43, no. 12, pp. 1499–1502, Dec. 1995.

[27] R. E. Hodges and Y. Rahmat-Samii, "The evaluation of MFIE integrals with the use of vector triangle basis functions," *Microwave Opt. Tech. Lett.*, vol. 14, no. 1, pp. 9–14, Jan. 1997.

[28] A. Herschlein, J. V. Hagen, and W. Wiesbeck, "Methods for the evaluation of regular weakly singular and strongly singular surface reaction integrals arising in method of moments," *ACES Journal*, vol. 17, no. 1, pp. 63–73, Mar. 2002.

[29] P. Ylä-Oijala and M. Taskinen, "Calculation of CFIE impedance matrix elements with RWG and $\hat{n}\times$RWG functions," *IEEE Trans. Antennas Propagat.*, vol. 51, no. 8, pp. 1837–1846, Aug. 2003.

[30] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala, "Singularity extraction technique for integral equation methods with higher order basis functions on plane triangles and tetrahedra," *Int. J. Numerical Methods Eng.*, vol. 58, pp. 1149–1165, Aug. 2003.

[31] A. Tzoulis and T. F. Eibert, "Review of singular potential integrals for method of moments solutions of surface integral equations," *Advances in Radio Science*, vol. 2, pp. 93–99, 2004.

[32] L. Gürel and Ö. Ergül, "Singularity of the magnetic-field integral equation and its extraction," *IEEE Antennas Wireless Propagat. Lett.*, vol. 4, pp. 229–232, 2005.

[33] I. Hänninen, M. Taskinen, and J. Sarvas, "Singularity subtraction integral formulae for surface integral equations with RWG, rooftop and hybrid basis functions," *Prog. Electromagn. Res. PIER*, vol. 63, pp. 243–278, 2006.

[34] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala, "Singularity subtraction technique for high-order polynomial vector basis functions on planar triangles," *IEEE Trans. Antennas Propagat.*, vol. 54, no. 1, pp. 42–49, Jan. 2006.

[35] G. Valerio, P. Baccarelli, P. Burghignoli, and A. Galli, "Comparative analysis of acceleration techniques for 2-D and 3-D Green's functions in periodic structures along one and two directions," *IEEE Trans. Microw. Theory Tech.*, vol. 55, pp. 1630–1643, June 2007.

[36] S. Singh, W. F. Richards, J. R. Zinecker, and D. R. Wilton, "Accelerating the convergence of series representing the free space periodic Green's functions," *IEEE Trans. Antennas Propagat.*, vol. 38, pp. 1958–1962, Dec. 1990.

[37] D. Shanks, "Non linear transformations of divergent and slowly convergent sequences," *J. Math. Phys.*, vol. 34, pp. 1–42, 1995.

[38] D. Levin, "Development of non-linear transformations for improving convergence of sequences," *Ins. J. Comput. Math.*, pp. 371–388, 1973.

[39] H. H. Homeier, "Scalar Levin-type sequence transformations," *Journal of Computational and Applied Mathematics*, vol. 122, pp. 81–147, 2000.

[40] M. Bressan and G. Conciauro, "Rapidly converging expressions for dyadic Green's functions of two-dimensional resonators of circular and rectangular cross-section," *Alta frequenza (Special issue on Applied Electromagnetics)*, vol. 52, no. 3, pp. 188–190, 1983.

[41] P. P. Ewald, “Die Berechnung Optischer und Elektrostatischen Gitterpotentiale,” *Ann. der Phys.*, vol. 64, pp. 253–287, 1921.

[42] V. G. Papanicolaou, “Ewald’s method revisited: rapidly convergent series representations of certain Green’s functions,” *J. Comp. Anal. Appl.*, vol. 1, no. 1, pp. 105–114, 1999.

[43] M. J. Park, J. Park, and S. Nam, “Efficient calculation of the Green’s fucntion for the rectangular cavity,” *IEEE Microwave Guided Wave Lett.*, vol. 8, no. 3, pp. 124–126, 1998.

[44] F. Marliani and A. Ciccolella, “Computationally efficient expressions of the dyadic Green’s function for rectangular enclosures,” *Progr. Electromagnet. Res.*, vol. 31, pp. 195–233, 2001.

[45] F. J. P. Soler, F. D. Q. Pereira, D. C. Rebenaque, A. A. Melcon, and J. R. Mosig, “A Novel Efficient Technique for the Calculation of the Green’s Functions in Rectangular Waveguides Based on Accelerated Series Decomposition,” *IEEE Trans. Antennas Propagat.*, vol. 56, no. 10, pp. 3260–3270, Oct. 2008.

[46] W. C. Chew, J.-M. Jin, C.-C. Lu, E. Michielssen, and J. Song, “Fast solution methods in electromagnetics,” *IEEE Trans. Antennas Propagat.*, vol. 45, no. 3, pp. 533–543, Mar. 1997.

[47] N. Morita, N. Kumagai, and J. Mautz, *Integral Equation Methods for Electromagnetics.* Boston, MA: Artech House, 1990.

[48] J. Song, C.-C. Lu, and W. C. Chew, “Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects,” *IEEE Trans. Antennas Propagat.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[49] S. Velamparambil and W. C. Chew, “Analysis and Performance of a Distributed Memory Multilevel Fast Multipole Algorithm,” *IEEE Trans. Antennas Propagat.*, vol. 53, no. 8, pp. 2719–2727, 2005.

[50] L. Gürel and Ö. Ergül , “Fast and accurate solutions of extremely large integral-equation problems discretised with tens of millions of unknowns,” *Electronics Letters*, vol. 43, no. 9, pp. 499–500, Apr. 2007.

[51] E. Michielssen and A. Boag, “A multilevel matrix decomposition algorithm for analyzing scattering from large structures,” *IEEE Trans. Antennas Propagat.*, vol. 44, no. 8, pp. 1086–1093, Aug. 1996.

[52] J. M. Rius, J. Parrón, E. Úbeda, and J. R. Mosig, “Multilevel matrix decomposition algorithm for analysis of electrically large electromagnetic problems in 3-D,” *Microwave and Optical Technology Letters*, vol. 22, no. 3, pp. 177–182, Aug. 1999.

[53] J. Parron, J. M. Rius, and J. R. Mosig, “Application of the multilevel matrix decomposition algorithm to the frequency analysis of large microstrip antenna arrays,” *Magnetics, IEEE Transactions on*, vol. 38, no. 2, pp. 721–724, Mar. 2002.

[54] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, “AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems,” *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, Sept./Oct. 1996.

[55] S. M. Seo and J. F. Lee, "A single-level low rank IE-QR algorithm for PEC scattering problems using EFIE formulation," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 8, pp. 2141–2146, Aug. 2004.

[56] S. Kapur and D. E. Long, "IES3: A fast integral equation solver for efficient 3-dimensional extraction," *Proc. ICCAD*, pp. 448–455, 1997.

[57] ——, "IES3: Efficient Electrostatic and Electromagnetic Simulation," *IEEE Computational Science and Engineering*, pp. 60–66, Oct.-Dec. 1998.

[58] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.*, vol. 86, no. 4, pp. 565–589, 2000.

[59] A. Heldring, J. M. Rius, and L. Ligthart, "New block ILU preconditioner scheme for numerical analysis of very large electromagnetic problems," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 337–340, Mar. 2002.

[60] A. Heldring, J. M. Rius, L. P. Ligthart, and A. Cardama, "Accurate numerical modeling of the TARA reflector system," *IEEE Trans. Antennas Propagat.*, vol. 52, no. 7, pp. 1758–1766, July 2004.

[61] K. Sertel and J. L. Volakis, "Incomplete LU preconditioner for FMM implementation," *Microw. Opt. Technol. Lett.*, vol. 26, no. 4, pp. 265–267, Aug. 2000.

[62] K. K. Mei and J. V. Bladel, "Scattering by Perfectly Conducting Rectangular Cylinders," *IEEE Trans. Antennas Propagat.*, vol. 11, no. 2, pp. 185–192, 1963.

[63] J. H. Richmond, "Digital Computer Solutions of the Rigorous Equations for Scattering Problems," *Proc. IEEE*, vol. 53, pp. 796–804, Aug. 1965.

[64] R. F. Harrington *et al.*, "Matrix Methods for field problems," *Proc. IEEE*, vol. 55, pp. 136–149, Feb. 1967.

[65] T. Itoh and J. R. Mosig, *Integral Equation Technique.* John Wiley & Sons, 1989.

[66] A. W. Glisson and D. R. Wilton, "Simple and Efficient Methods for Problems on Electromagnetic Radiation and Scattering from Surfaces," *IEEE Trans. Antennas Propagat.*, vol. 28, no. 5, 1980.

[67] E. Suter and J. R. Mosig, "A subdomain multilevel approach for the MOM analysis of large planar antennas," *Microwave and Optical Technology Letters*, vol. 26, pp. 270–277, 2000.

[68] L. Matekovits, G. Vecchi, M. Bercigli, and M. Bandinelli, "Synthetic-Functions Analysis of Large Aperture-Coupled Antennas," *IEEE Trans. Antennas Propagat.*, vol. 57, no. 7, pp. 1936–1943, July 2009.

[69] C. Delgado, E. Garcia, F. Catedra, and R. Mittra, "Generation of Characteristic Basis Functions Defined Over Large Surfaces by Using a Multilevel Approach," *IEEE Trans. Antennas Propagat.*, vol. 57, no. 4, pp. 1299–1301, Apr. 2009.

[70] C. Delgado, M. F. Catedra, and R. Mittra, "Efficient Multilevel Approach for the Generation of Characteristic Basis Functions for Large Scatters," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 7, pp. 2134–2137, July 2008.

[71] E. Garcia, C. Delgado, I. G. Diego, and M. F. Catedra, "An Iterative Solution for Electrically Large Problems Combining the Characteristic Basis Function Method and the Multilevel Fast Multipole Algorithm," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 8, pp. 2363–2371, Aug. 2008.

[72] F. Vipiana, G. Vecchi, and P. Pirinoli, "A Multiresolution System of Rao-Wilton-Glisson Functions," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 3, pp. 924–930, Mar. 2007.

[73] J. M. Rius, J. Parrón, A. Heldring, J. M. Tamayo, and E. Úbeda, "Fast Iterative Solution of Integral Equations With Method of Moments and Matrix Decomposition Algorithm - Singular Value Decomposition," *IEEE Trans. Antennas Propagat, special issue on Large and Multiscale Computational Electromagnetics*, vol. 56, no. 8, pp. 2314–2324, Aug. 1998.

[74] S. Kurz, O. Rain, and S. Rjasanow, "The adaptive cross-approximation technique for the 3D boundary-element method," *Magnetics, IEEE Transactions on*, vol. 38, no. 2, pp. 421–424, Mar. 2002.

[75] K. Zhao, M. N. Vouvakis, and J. F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *Electromagnetic Compatibility, IEEE Transactions on*, vol. 47, no. 4, pp. 763–773, Nov. 2005.

[76] J. F. Shaeffer, "LU factorization and solve of low rank electrically large MOM problems for monostatic scattering using the adaptive cross approximation for problem sizes to 1,025,101 unknowns on a PC workstation," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, 2007, pp. 1273–1276.

[77] A. Heldring, J. M. Tamayo, and J. M. Rius, "On the degrees of freedom in the interaction between sets of elementary scatterers," in *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, 2009, pp. 2511–2514.

[78] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Hopkins Fulfillment Service, 1996.

[79] W.-D. Li, W. Hong, and H.-X. Zhou, "An IE-ODDM-MLFMA Scheme With DILU Preconditioner for Analysis of Electromagnetic Scattering From Large Complex Objects," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 5, pp. 1368–1380, May 2008.

[80] M. S. Tong and W. C. Chew, "Super-hyper singularity treatment for solving 3D electric field integral equations," *Microwave Opt. Tech. Lett.*, vol. 49, no. 6, pp. 1383–1388, June 2007.

[81] ——, "On the near-interaction elements in integral equation solvers for electromagnetic scattering by three-dimensional thin objects," *IEEE Trans. Antennas Propagat.*, vol. 57, no. 8, pp. 2500–2506, Aug. 2009.

[82] L. J. Gray, J. M. Glaeser, and T. Kaplan, "Direct evaluation of hypersingular Galerkin surface integrals," *SIAM J. Sci. Comput.*, vol. 25, no. 5, pp. 1534–1556, 2004.

[83] L. J. Gray, A. Salvadori, A. V. Phan, and A. Mantic, “Direct evaluation of hypersingular Galerkin surface integrals. II,” *Electronic Journal of Boundary Elements*, vol. 4, no. 3, pp. 105–130, 2006.

[84] A. G. Polimeridis and J. R. Mosig, “Complete semi-analytical treatment of weakly singular integrals on planar triangles via the direct evaluation method,” *Int. J. Numerical Methods Eng.*, vol. 83, no. 12, pp. 1625–1650, Sept. 2010.

[85] D. A. Dunavant, “High degree efficient symmetrical gaussian quadrature rules for the triangle,” *International Journal for Num. Methods in Eng.*, vol. 21, pp. 1129–1148, 1985.

[86] A. G. Polimeridis and J. R. Mosig, “Evaluation of weakly singular integrals via generalized Cartesian product rules based on the double exponential formula,” *IEEE Trans. Antennas Propagat.*, vol. 58, no. 6, pp. 1980–1988, 2010.

[87] P. Persson and G. Strang, “A Simple Mesh Generator in MATLAB,” *SIAM REVIEW*, vol. 46, pp. 329–345, 2004.

[88] J. R. Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator,” *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148, pp. 203–222, 1996.

[89] ——, “Delaunay Refinement Algorithms for Triangular Mesh Generation,” *Computational Geometry: Theory and Applications*, vol. 22, pp. 21–74, 2002.

[90] P. A. Foteinos, A. N. Chernikov, and N. P. Chrisochoides, “Fully generalized two-dimensional constrained delaunay mesh refinement,” *SIAM J. Sci. Comput.*, vol. 32, pp. 2659–2686, 2010.

[91] E. Úbeda and J. M. Rius, “MFIE MoM-formulation with curl-conforming basis functions and accurate Kernel-integration in the analysis of perfectly conducting sharp-edged objects,” *Microwave and Optical Technology letters*, vol. 44, no. 4, 2005.

[92] ——, “Comments on “The use of Curl-conforming Basis functions for the Magnetic-Field Integral Equation”,” *IEEE Trans. Antennas Propagat.*, vol. 56, 2008.

[93] H. Takahasi and M. Mori, “Double Exponential Formulas for Numerical Integration,” *Publications of the Research Institute for Mathematical Sciences*, vol. 9, no. 3, pp. 721–741, 1974.

[94] A. G. Polimeridis, I. D. Koufogiannis, M. Mattes, and J. R. Mosig, “Efficient double exponential-based cubatures for the computation of weakly singular Galerkin inner products,” *IV Workshop on Integral Techniques for Electromagnetics, INTELECT 2010.*

[95] L. Tsai, “A numerical solution for the near and far fields of an annular ring of magnetic current,” *IEEE Trans. Antennas Propagat.*, vol. 20, no. 5, pp. 569–576, Sept. 1972.

[96] D. M. Pozar, *Microwave Engineering.* John Wiley & Sons, Inc., 1998.

Author Publications and Awards

Publications - Journal Papers

[APJ-1] J. M. Tamayo, A. Heldring and J. M. Rius, “Multilevel Adaptive Cross Approximation (MLACA),” submitted to *Antennas and Propagation, IEEE Transactions on.*

[APJ-2] A. G. Polimeridis, J. M. Tamayo, J. M. Rius and J. R. Mosig, “Fast and accurate computation of hyper-singular integrals in Galerkin surface integral equation formulations via the direct evaluation method,” submitted to *Antennas and Propagation, IEEE Transactions on.*

[APJ-3] J. M. Tamayo, S. López-Peña, M. Mattes, A. Heldring, J. M. Rius and J. R. Mosig, “A Recursive Acceleration Technique for Static Potential Green’s Functions of a Rectangular Cavity Combining Image and Modal Series,” accepted to *Microwave Theory and Techniques, IEEE Transactions on.*

[APJ-4] E. Úbeda, J. M. Tamayo and J. M. Rius, “Taylor-orthogonal basis functions for the discretization in Method of Moments of second kind integral equations,” submitted to *Microwave and Optical Technology Letters.*

[APJ-5] E. Úbeda, J. M. Tamayo, J. M. Rius and A. Heldring, “Zeroth-order Complete Discretizations of Integral-Equation formulations involving Conducting or Dielectric Objects at Very Low Frequencies,” submitted to *Antennas and Propagation, IEEE Transactions on.*

[APJ-6] A. Heldring, J. M. Tamayo, J. M. Rius, J. Parrón and E. Úbeda, “Multiscale Compressed Block Decomposition for Fast Direct Solution of Method of Moments Linear System,” accepted to *Antennas and Propagation, IEEE Transactions on.*

[APJ-7] A. Heldring, J. M. Rius and J. M. Tamayo, “Comments on ‘Fast Direct Solution of Method of Moments Linear System’,” *Antennas and Propagation, IEEE Transactions on*, vol. 58, no. 3, pp. 1015-1016, March 2010.

[APJ-8] J. M. Rius, J. Parrón, A. Heldring, J. M. Tamayo, and E. Úbeda, “Fast Iterative Solution of Integral Equations With Method of Moments and Matrix Decomposition Algorithm Singular Value Decomposition,” *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 8, pp. 2314-2324, Aug. 2008.

[APJ-9] A. Heldring, J. M. Rius, J. M. Tamayo, and J. Parrón, “Compressed Block-Decomposition Algorithm for Fast Capacitance Extraction,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 2, pp. 265-271, Feb. 2008.

[APJ-10] A. Heldring, J. M. Rius, J. M. Tamayo, J. Parrón and E. Úbeda, “Fast Direct Solution of Method of Moments Linear System,” *Antennas and Propagation, IEEE Transactions on*, vol. 55, no. 11, pp. 3220-3228, Nov. 2007.

Publications - Conference Papers

[APC-1] E. Úbeda, J. M. Tamayo and J. M. Rius, “Accurate discretization of the electric-magnetic field integral equation (EMFIE),” *IV Workshop Integral Techniques for Electromagnetics, INTELECT' 10*, Oct. 1, 2010.

[APC-2] J. M. Tamayo, A. G. Polimeridis, J. M. Rius and J. R. Mosig, “On the efficient evaluation of hyper-singular integrals in Galerkin surface integral equation formulations via the direct evaluation method,” *Antennas and Propagation Society International Symposium (APSURSI), 2010 IEEE*, pp. 1-4, 11-17 July 2010.

[APC-3] A. Padilla, J. Mateu, C. Collado, C. Ernst, J. M. Rius, J. M. Tamayo and J. M. O'Callaghan, “Comparison of lossy filters and predistorted filters using novel software,” *Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International*, pp. 1720-1723, 23-28 May 2010.

[APC-4] J. M. Tamayo, E. Úbeda, A. G. Polimeridis, J. M. Rius and J. R. Mosig, “Very accurate computation of the impedance elements on the discretization of the magnetic field integral equation with the orthogonal basis functions,” *VII Encuentro Ibérico de Electromagnetismo Computacional*, 2010.

[APC-5] A. Heldring, J. M. Tamayo, J. M. Rius and E. Úbeda, “Accelerated direct solution of the MoM-VIE for dielectric scatterers,” *VII Encuentro Ibérico de Electromagnetismo Computacional*, 2010.

[APC-6] A. Heldring, J. M. Rius and J. M. Tamayo, “Direct Mom solution of electrically large problems with N2 complexity,” *Antennas and Propagation (EuCAP), 2010 Proceedings of the Fourth European Conference on*, pp. 1-4, 12-16 April 2010.

[APC-7] J. M. Tamayo, A. G. Polimeridis, J. M. Rius, A. Heldring and J. R. Mosig, “Analysis of numerical integration in the evaluation of hyper-singular integrals in Galerkin surface integral equation formulations via the direct evaluation method,” *Antennas and Propagation (EuCAP), 2010 Proceedings of the Fourth European Conference on*, pp. 1-5, 12-16 April 2010.

[APC-8] J. M. Rius, J. A. Herrero, J. M. Tamayo, A. Heldring, E. Úbeda, J. Parrón, S. Lopez-Pena, S., A. G. Polimeridis, J. R. Mosig, H. Espinosa and A. Boag, “Software framework for integration of method of moments kernels with direct or iterative fast solvers,” *Antennas and Propagation (EuCAP), 2010 Proceedings of the Fourth European Conference on*, pp. 1-2, 12-16 April 2010.

[APC-9] E. Úbeda, J. M. Tamayo and J. M. Rius, "Orthogonal basis functions for the discretization of the Magnetic-field Integral Equation in the low frequency regime," *Antennas and Propagation (EuCAP), 2010 Proceedings of the Fourth European Conference on*, pp. 1-4, 12-16 April 2010.

[APC-10] J. M. Tamayo, A. Heldring and J. M. Rius, "Application of multilevel adaptive cross approximation (MLACA) to electromagnetic scattering and radiation problems," *Electromagnetics in Advanced Applications, 2009. ICEAA '09. International Conference on*, pp. 178-181, 14-18 Sept. 2009.

[APC-11] J. M. Rius, J. M. Tamayo, A. Heldring, E. Úbeda and J. Parrón, "Framework for integration of Fast Solvers in Method of Moments kernels," *Integral Equations in Electromagnetics*, Istanbul 2009.

[APC-12] E. Úbeda, J. M. Tamayo and J. M. Rius, "Zeroth-order divergence-complete discretizations of the EFIE at very low frequencies," *Computational Electromagnetics International Workshop, 2009. CEM 2009*, pp. 1-4, 20-23 July 2009.

[APC-13] J. M. Tamayo, E. Úbeda and J. M. Rius, "Novel Self-loop basis functions for the stability of the Linear-linear discretization of the Electric Field Integral Equation at very low frequencies," *Antennas and Propagation Society International Symposium, 2009. APSURSI '09. IEEE*, pp. 1-4, 1-5 June 2009.

[APC-14] J. M. Tamayo, A. Heldring and J. M. Rius, "Multilevel adaptive cross approximation (MLACA)," *Antennas and Propagation Society International Symposium, 2009. APSURSI '09. IEEE*, pp. 1-4, 1-5 June 2009.

[APC-15] A. Heldring, J. M. Tamayo and J. M. Rius, "On the degrees of freedom in the interaction between sets of elementary scatterers," *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, pp. 2511-2514, 23-27 March 2009.

[APC-16] J. M. Tamayo, A. Heldring and J. M. Rius, "Compression of the MoM matrix using macrobasis functions with a full-controlable accuracy," *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, pp. 171-175, 23-27 March 2009.

[APC-17] A. Heldring, J. M. Tamayo, E. Úbeda and J. M. Rius, "Progress in MoM Solution of Electromagnetic Scattering and Radiation at UPC AntennaLab," *VI Encuentro Ibérico de Electromagnetismo Computacional*, 2008.

[APC-18] H. Espinosa, A. Heldring, J. M. Tamayo and J. M. Rius, "Polynomial Interpolators of variable order for Method of Moments acceleration with the Multilevel Field Interpolation Algorithm (MLFIA)" , *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, 5-11 July 2008.

[APC-19] A. Heldring, J. M. Tamayo, J. M. Rius, J. Parrón and E. Úbeda, "Multiscale CBD for fast direct solution of MoM linear system," *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, pp. 1-4, 5-11 July 2008.

[APC-20] J. M. Rius, A. Heldring, J. M. Tamayo and J. Parrón, "The MDA-SVD Algorithm for Fast Direct or Iterative Solution of Discrete Integral Equations," *Antennas and Propagation, 2007. EuCAP 2007. The Second European Conference on*, pp. 1-8, 11-16 Nov. 2007.

[APC-21] A. Heldring, J. M. Rius, J. M. Tamayo, J. Rubio, J. Zapata, V. Volski, G. Vandenbosh, M. Mattes, A. Valero and J. Parrón, "Integration of UPC-UAB Fast Solvers into the Antenna Simulation Software of UNEX-UPM, KUL, UPV and EPFL ACE Partners," *Antennas and Propagation, 2007. EuCAP 2007. The Second European Conference on*, pp. 1-5, 11-16 Nov. 2007.

[APC-22] E. Úbeda, A. Heldring, J. M. Rius, H. Espinosa, J. M. Tamayo and J. Parrón, "Advances in Numerical Electromagnetics at UPC AntennaLab," *Computational Electromagnetics Workshop*, 2007.

[APC-23] A. Heldring, J. M. Rius, J. M. Tamayo and J. Parrón, "Multilevel MDA-CBI for fast direct solution of large scattering and radiation problems," *Antennas and Propagation Society International Symposium, 2007 IEEE*, pp. 5599-5602, 9-15 June 2007.

[APC-24] A. Heldring, E. Úbeda, J. M. Rius, H. Espinosa, J. M. Tamayo and J. Parrón, "Advances in Numerical Electromagnetics at UPC and UAB," *V Encuentro Ibérico de Electromagnetismo Computacional*, 2007.

[APC-25] H. Espinosa, A. Heldring, J. M. Tamayo, J. M. Rius and J. R. Mosig, "Multilevel field interpolation algorithm for large PEC objects," *Antennas and Propagation, 2006. EuCAP 2006. First European Conference on*, pp. 1-5, 6-10 Nov. 2006.

[APC-26] A. Heldring, J. M. Tamayo and J. M. Rius, "Fast direct solution of the method of moments linear system," *Antennas and Propagation, 2006. EuCAP 2006. First European Conference on*, pp. 1-4, 6-10 Nov. 2006.

[APC-27] J. M. Rius, A. Heldring, H. Espinosa, J. M. Tamayo, M. Mattes, A. Valero and J. Parrón, "Integration of UPC antennalab block-LU direct solver in MoM codes," *Antennas and Propagation, 2006. EuCAP 2006. First European Conference on*, pp. 1-7, 6-10 Nov. 2006.

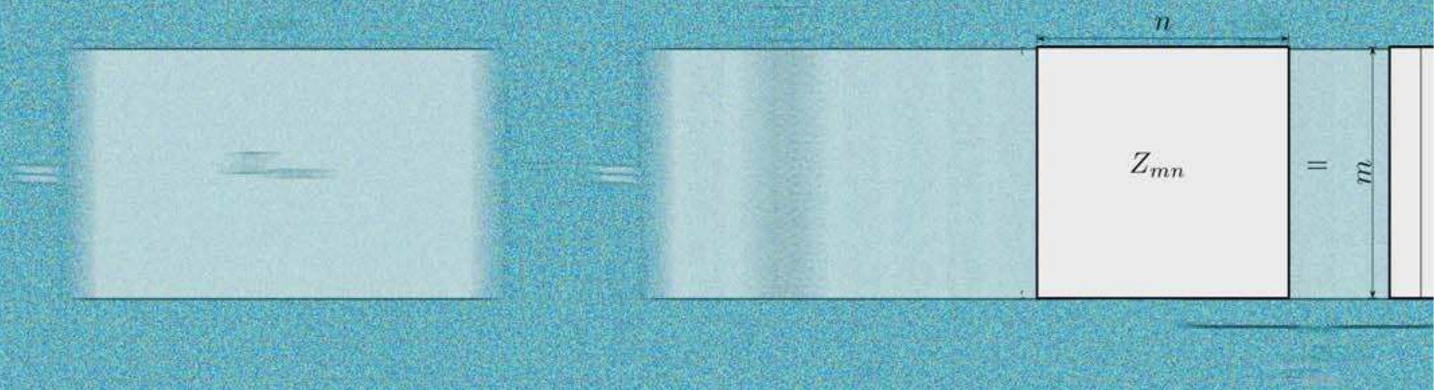
[APC-28] J. M. Rius, A. Heldring, J. M. Tamayo and J. Parrón, "New and more efficient formulation of MLMDA for arbitrary 3D antennas and scatterers," *Antennas and Propagation, 2006. EuCAP 2006. First European Conference on*, pp. 1-6, 6-10 Nov. 2006.

[APC-29] J. M. Rius, E. Úbeda, A. Heldring, H. Espinosa and J. M. Tamayo, "Algunas mejoras de la exactitud y eficiencia de la formulación de ecuaciones integrales para análisis de radiación y difracción ," *IV Encuentro Ibérico de Electromagnetismo Computacional*, 2005.

Awards

- Honorable Mention at the Student Contest in the IEEE Antennas and Propagation Society International Symposium, 2009. APSURSI '09. Related to the presentation with title "Multilevel adaptive cross approximation (MLACA)."

- Honorable Mention at the Student Contest in the IEEE Antennas and Propagation Society International Symposium, 2010. APSURSI '10. Related to the presentation with title "On the Efficient Evaluation of Hyper-Singular Integrals in Galerkin Surface Integral Equation Formulations via the Direct Evaluation Method"



n
Z_{mn}
=
m