UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior - Leganés

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

DOCTORAL THESIS

A NEW SOFTWARE SUITE FOR ELECTROMAGNETICS

AUTHOR

Daniel García Doñoro

ADVISORS

L. E. García Castillo and T. K. Sarkar

Tesis Doctoral: A NEW SOFTWARE SUITE
FOR ELECTROMAGNETICS

Autor: Daniel García Doñoro

Directores: D. Luis Emilio García Castillo
D. Tapan Kumar Sarkar

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores

Presidente:

Vocal:

Secretario:

acuerda otorgarle la calificación de

Leganés, a

A mis padres
A mi familia

ABSTRACT

In recent years, computational electromagnetics (CEM) techniques have become increasingly important with the rapid advancements in technology in areas such as electromagnetic compatibility, antenna analysis, radar cross section (RCS), cellular phone-human body interaction, design of electrical and medical devices, target recognition and lightning strike simulation.

Among a variety of numerical simulation tools existing in the commercial market, many are based on the method of moments (MoM), the finite-difference time-domain method (FDTD), and the finite element method (FEM). Also, they implement hybridization with high-frequency or asymptotic technique such as, physical optics (PO), the uniform geometrical theory of diffraction (UTD) and Multilevel Fast Multipole Algorithm (MLFMA) among others.

It is worth to note that many of the commercial simulation tools existing in the market has been born as numerical in-house codes in the academic sector. In this context, it is important to note the contribution of the research group guided by Prof. Tapan K. Sarkar (Syracuse University) to the CEM field during last decade. The development of a new electromagnetic solver based on MoM has been carried out in order to provide fast and accurate solutions of a wide range of electromagnetic problems, especially for the solution of electrically large and complex problems.

From other hand, the research group to which the author of the present Ph. D dissertation belongs has an important research line focused on the development of codes based on FEM. Then, the implementation of a FEM code makes possible the development of, not only an electromagnetic software based on an integral formulation of the electromagnetic problem, but a complete electromagnetic suite with also a differential formulation approach. Hence, the development of a new software suite for electromagnetics becomes the main objective of this Ph. D. dissertation.

The suite will be composed by a professional graphical user interface (GUI) and two solver modules based on MoM and FEM, respectively. The GUI will provide tools for an easy and quick simulation process, the parametrization of geometric models in terms of symbolic variables or the use of an automatic goal oriented optimizer.

The FEM module of the suite will present important unique features compared with other commercial softwares such as, the use of a novel iterative integral equation method for mesh truncation, the use of its own higher order set of basis functions and the use of parallel programming schemes from the beginning on its development. This module will also be able to perform the analysis of large antenna arrays using an infinite array approach. Although, the infinite array approach make uses of structures that are not a physically realistic, the analysis of this structures provides a reasonable good approximation with a less computing requirement than the analysis of the full problem.

Finally, taking advantage of the existence in the suite of two of the most important computational electromagnetics numerical techniques such as, MoM and FEM, the hybridization between them seems an appropriate choice to perform complex simulation where the use of these techniques alone may not be efficiently appropriate. Thus, a modular approach to combine MoM and FEM techniques for the analysis of large structures or finite arrays with complex radiating elements has been also performed.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PSEUDOCODES

CHAPTER 1

INTRODUCTION

1.1 Antecedents

Computational electromagnetics (CEM) deals with the science of solving Maxwell's equations with the numerical simulation of electromagnetic fields. It has become an indispensable tool for the analysis of electromagnetic problems because of the predictive power of Maxwell's equations: if these equations are solved correctly, the solution can predict experimental outcomes and design performances.

In recent years, CEM techniques have become increasingly important with the rapid advancements in technology in areas such as electromagnetic compatibility, antenna analysis, radar cross section (RCS), cellular phone-human body interaction, design of electrical and medical devices, target recognition and lightning strike simulation.

Among a variety of numerical simulation tools that provide a complete solution to Maxwell's equations, many are based on the method of moments (MoM) [1], the finite-difference time-domain method (FDTD) [2, 3, 4], and the finite element method (FEM) [5, 6, 7]. Other methods, such as the transmission-line method and the finite integration technique, can be identified as either a variation or an equivalent of one of the first three.

It is worth to note that many of the commercial simulation tools existing in the market has been born as numerical in-house codes in the academic sector. A survey of some of the most important commercial electromagnetic software is given below:

- **HFSS:** HFSS is a commercial software developed by ANSYS and based on FEM. The software is used for antenna design and design of complex RF electronic circuit elements including filters and transmission lines [8]. This commercial software was originally developed by Prof. Zoltan Cendes at Carnegie Mellon University.

- **CST STUDIO SUITE:** CST STUDIO SUITE is the main product of the German company Computer Simulation Technology [9]. The software comprises several solver modules based on different techniques including FEM, MoM, Multilevel Fast Multipole Algorithm (MLFMA) and Shooting Boundary Ray (SBR), each offering distinct advantages in their own domains. The flagship module of CST STUDIO SUITE is its time domain solver based on the Perfect Boundary Approximation (PBA) [10] with the Finite Integration Technique (FIT) [11, 12].

- **FEKO:** FEKO is a computational electromagnetics software product developed by the company group EM Software & Systems [13]. The software gathers different solvers based on pure techniques, such as MoM, FEM and FDTD and also contains various hybrid methods with acceleration techniques (MLFMA) and asymptotic methods such as, physical optics (PO) and the uniform geometrical theory of diffraction (UTD). FEKO also had an academic beginning and was originated in 1991 from research activities of Dr. Jakobus at the University of Stuttgart.

- **COMSOL Multiphysics:** COMSOL Multiphysics is a finite element simulation software package composed for various physic modules [14]. This software is able to perform multiphysics analysis making use of several of its modules in one simulation. The electromagnetic module is based on FEM with numerically stable edge elements. The software uses algorithms for preconditioning and iterative solutions of the resulting sparse equation systems. In 1986, COMSOL Multiphysics was started by graduate students at the Royal Institute of Technology (KTH) in Stockholm.

- **REMCOM:** Remcom is a US company formed in 1994 to develop and market electromanetic software. Its flagship product is the EM software named XFdtd [15]. This software is based on FDTD and is used in a wide variety of markets and applications, including antenna design, placement and analysis or microwave circuits among others. It is worth to note that this software includes support for GPU acceleration. Remcom company also offers a high frequency GTD/UTD based package for the design and analysis of antenna system on complex objects such vehicles and aircraft named XGtd [16].

- **ADS:** ADS is the flagship software package of the multinational company Agilent Technologies [17]. The software gathers a set of commercial solver such as, Momentum [18] (based on MoM) and the Agilent FEM Simulator Element, formerly EMDS G2. This simulator provides full wave 3D EM simulation capabilities based on FEM.

- **Momentum:** Momentum is a partial differential equation solver of Maxwell's equations based on MoM. It is a 3-D planar electromagnetic simulator used for passive circuit analysis [18]. It was originally developed by a Belgian company, Alphabit, a spinoff from the research center IMEC [19]. The company was acquired by Hewlett-Packard and later became part of Agilent Technologies, the current owners.

- **GEMS/EFIELD:** GEMS/EFIELD are two different commercial electromagnetic suites that gather solvers in both time and frequency domain including full wave solution (MoM, MLFMA, FDTD, FEM) [20, 21]. In the mid nineties, both softwares were started under the framework of the same large research project at the Royal Institute of Technology (KTH) in Stockholm, in close cooperation with end users at Saab and Ericsson companies.

- **WIPL-D:** WIPL-D is an electromagnetic software that use the MoM technique with higher order basis functions to perform the electromagnetic analysis [22]. This software also has a circuit simulator in order to provide results for microwave circuits. As happens with other commercial softwares, WIPL-D has an initial development in the academic sector.

- **GRASP:** GRASP is an electromagnetic software used to perform the accurate analysis of reflector antennas [23]. GRASP is the flagship software of the Danish company TICRA, founded in 1971 [24]. This software makes use of pure numerical techniques such as, MoM and high-frequency methods such as, physical optics (PO) and geometrical theory of diffraction (GTD) to perform the simulations.

- **CHAMP:** CHAMP is another software tool developed by the company TICRA. This tool is used to design horns and reflectors antennas with rotational symmetry [25]. CHAMP combines a Mode-Matching (MM) solver for the propagation internally in the horn with a highly-efficient Body-of-Revolution (BoR) MoM solver for the exterior parts, including reflectors and dielectric support material. The BoR solver was developed specifically for this application with the aim of optimizing speed and rendering optimization practical.

- **IE3D:** IE3D is a commercial software developed by the company Mentor Graphics (formerly Zeland Software) [26]. The current name of the software is HyperLynx 3D EM but it is better known by the name IE3D [27]. The software makes use of an integral formulation to solve the Maxwell's equations. The main applications of IE3D are antenna design, RFID design, package modeling and Monolithic Microwave Integrated Circuits (MMIC) design.

- **NewFasant:** NewFasant is an electromagnetic simulation software suite with different solvers based on the MONURBS technique and high-frequency techniques such as, GTD and PO [28]. The software also has acceleration techniques such as MLFMA. As other tools, NewFasant also started in the academic sector at University of Alcalá in Spain.

- **CEMWorks:** CEMWorks is a Canadian company that commercialize the electromagnetic software called Wave3D [29]. This software is based on broadband frequency-domain MoM analysis accelerated with error-controllable MLFMA. Wave3D is used in a wide range of industrial and academic applications such as, the analysis of aperture antennas, antenna positioning, design of antenna arrays or the scattering analysis of large vehicles.

- **MiCIAN:** MiCIAN is a German company that develops the electromagnetic software μWave [30]. This software is a design tool using the well-known fast and accurate MM technique. This method is particularly suitable for simulation and optimization of passive microwave systems and components, including antennas. The software avoids the use of time consuming 3D solvers wherever possible and to focus on applying MM and its derivatives instead. It is worth to note that a 3D FEM solver on element level is available within the μWave software for structures with very complex geometries or with features not feasible to be implemented in MM.

- **Tech-X Corporation:** Tech-X Corporation is a US company that offers different software packages for various physic and engineering applications [31]. The electromagnetic software of this company is called VSimEM [32]. This software uses the FDTD technique for solving electromagnetic problems for a variety of material types, yielding engineering outputs that can be used for design of electromagnetic devices.

As commented above, many of those commercial softwares have an academic beginning where research groups from different universities develop its own codes and, when they are competitive enough, are released to the market. Other interesting electromagnetics codes in the academic sector are described below:

- **openEMS:** openEMS is a free and open electromagnetic field solver based on the FDTD method. Matlab or Octave are used as an easy and flexible scripting interface [33]. OpenEMS is a project started by Thorsten Liebig at the laboratory for General and Theoretical Electrical Engineering (ATE), University of Duisburg-Essen in Germany.

- **LC:** LC is simulation tool for the analysis of the electromagnetic properties of electrical interconnects [34]. This tool uses the FDTD approach to analyze the electromagnetic problem. LC was orignally developed by Cray Research as an internal design tool. Like many projects, this one has a long roster of contributors from a variety of organizations: Northwestern University, University of Colorado and Los Alamos National Laboratory.

- **ERMES:** ERMES is freeware electromagnetic code based on FEM [35]. This software make use of a FEM formulation based on the regularized Maxwell equations. This code has been developed at Polytechnic University of Catalonia in Barcelona.

During last decade, the effort in the CEM field has been dedicated to provide a better user experience in the use of those softwares maintaining the mathematical approaches basically intact. One of the keys in the development of CEM tools in this time has been the accessibility to a new generation of computer processors. Nowadays, processors with several cores are common even in modern single-user laptop/desktop computers. Also, access to distributed computing have become quite affordable for research/development groups. Thus, it is common to have access to a small- or mid-size cluster consisting of several multi-core computer nodes. The parallelization of the solver in such computer systems allows to dramatically reduce the computation time and, at the same time, gives access to the distributed RAM memory of the whole cluster (in the order of hundreds of GBs). Under this scenario, many existing tools has needed to revamp their computational codes to run efficiently on this new generation of multi-core processors. Conversely, other codes has been developed from scratch taking into account parallel schemes in order to provide tools with a high parallel efficiency.

In this context, it is important to note the contribution of the research group guided by Prof. Tapan K. Sarkar (Syracuse University) to the CEM field during last decade. The development of a new electromagnetic solver based on MoM has been carried out in order to provide fast and accurate solutions of a wide range of electromagnetic problems, especially for the solution of electrically large and complex problems [36].

The solver was implemented from scratch making use of the latest computer technology and parallel paradigms to take advantage of the capabilities of the new generation of computer processors. Thus, efficient parallel algorithms were used to improve the performance and speed of the solver providing a powerful tool for analysis of electrically large objects composed of metallic and dielectric structures. However, the lack of a professional graphical user interface (GUI) made the solver tough to use for novel users. Thus, the development of a professional GUI that allows a friendly and easy use of the MoM solver is needed in order to create a competitive electromagnetic software. This GUI should support tools for an easy and quick simulation process, the parametrization of geometric models in terms of symbolic variables or the use of an automatic goal oriented optimizer among others.

From other hand, the research group to which the author of the present Ph. D dissertation belongs has an important research line focused on the development of codes based on FEM. Then, the implementation of a FEM code makes possible the development of, not only an electromagnetic software based on an integral formulation of the EM problem, but a complete electromagnetic suite with also a differential formulation approach. Hence, the development of a new software suite for electromagnetics becomes the main objective of this Ph. D. dissertation.

1.2 Objectives

As commented above, the main objective of the present Ph. D. dissertation is the development of a new software suite for electromagnetics. The suite will be composed by a professional GUI and two solver modules based on MoM and FEM, respectively. The list of the objectives for this Ph. D. dissertation may be summarized as follows:

- The development of a graphical framework to support tools for an easy and quick simulation process is required to make a competitive electromagnetic software. Also, the parametrization of geometric models in terms of symbolic variables or the use of an automatic goal oriented optimizer are necessary tools in order to complete this graphical framework. Thus, an important objective of this Ph. D. dissertation is the development of this graphical framework where to integrate the MoM solver implemented under the guidance of Prof. Tapan K. Sarkar. Section 1.2.1 gives further details about this objective.

- The development of a differential solver module based on FEM is another important objective of the present Ph. D. dissertation. This module presents important unique features compared with other commercial softwares such as, the use of a novel iterative integral equation method for mesh truncation, the use of its own higher order set of basis functions and the use of parallel programming schemes from the beginning on its development. More detail about this objective are given in section 1.2.2.

- The development of a solver module for the analysis of large antenna arrays using an infinite array approach is also another objective of the present Ph. D. dissertation. Although, the infinite array approach make uses of structures that are not a physically realistic, the analysis of this structures provides a reasonable good approximation with a less computing requirement than the analysis of the full problem. Furthermore, fast techniques approaches such as Macro Basis Functions (`MBFs`) use the infinite array solution as the basis brick for their approaches. To carried out this objective the implementation of the Periodic Boundary Condition (`PBCs`) in the `FEM` module of the suite is required. Further information about this objective of the dissertation is given in section 1.2.3

- Taking advantage of the existence of two of the most important `CEM` numerical techniques in the suite such as, `MoM` and `FEM`, another objective of this Ph. D. dissertation is to perform the hybridization between both solver modules. However, due to the difference between the formulation setup and the basis functions employed by both solvers, several change in the formulation of the `FEM` module are needed together with a study of the projection between the basis functions of both codes. Section 1.2.4 gives further details about this objective.

1.2.1 Graphical framework and optimizer development

The development of a graphical framework where to integrate the integral equation solver implemented under the guidance of Prof. Tapan K. Sarkar is an important objective of this Ph. D. dissertation. Together with this development, the implementation of a goal oriented optimizer to achieve an automatic improvement of the results is another important objective. Thus, the graphical framework should support the capability to define the geometry model in term of symbolic variables.

Before gives a detailed description about the contents of the graphical framework and the optimizer, a brief introduction about the electromagnetic formulation, the basis functions used by the MoM solver, and the parallel implementation in code is given next.

- **Electromagnetic formulation:** The solver is based on the solution of Surface Integral Equations (SIEs) in the frequency domain for electric and magnetic currents over dielectric boundary surfaces and electric currents over perfect electric conductors (PECs). Specifically, the solver makes uses of a general form of the Poggio-Miller-Chang-Harrington-Wu (PMCHW) formulation [37, 38]. It is worth noting that if the boundary surface of two different regions is a PEC, the magnetic currents are equal to zero at the boundary surface and thus, the equations degenerates into the electric field integral equation (EFIE).

  The solver is able to handle inhomogeneous dielectrics categorized by a combination of various homogeneous dielectrics. Therefore, any composite metallic and dielectric structure can be represented as an electromagnetic system consisting of a finite number of finite-size linear, homogeneous and isotropic regions situated in an unbounded linear, homogeneous and isotropic environment.

- **Higher-order basis functions:** The electric and magnetic currents are approximated by higher-order polynomials, which reduce the number of unknowns compared with the rational piece-wise basis functions. The code makes use of truncated cones for wires and bilinear patches to characterize other surfaces. Higher-order polynomials over these type of elements are used as basis functions over larger subdomain segments and patches. Polynomial expansions for the basis functions over large subdomains lead to a good approximation of the current distributions over large surfaces using approximately 20 unknowns per wavelength squared of surface area. This number is much lower than that for the piecewise Rao-Wilton-Glisson (RWG) basis functions, and thus, the use of polynomial basis functions over larger subdomains reduce significantly the number of unknowns.

- **Parallel processing and out-of-core techniques:** The parallel implementation of an integral equation code involves parallel matrix filling followed by a parallel solution of the dense matrix equation. Load-balancing in terms of data and CPU operations is crucial to achieve speedups with a large number of processors.

The parallel implementation is achieved by using MPI (Message Passing Interface) [39]. MPI generates a logical rectangular process grid which is assumed to map matrix data blocks onto processors. Specifically, a block-cyclic matrix distribution is used among processors as ScaLAPACK does. ScaLAPACK library [40] is used to solve the matrix system of equations. The matrix solution is based on the LU factorization algorithm

An out-of-core version of the parallel solver has been implemented in order to break the limit imposed by the available RAM in the system. The out-of-core solver uses hard disk as source of memory. It partitions the matrix in *slabs*, the number of slabs being dependent on the relation between available hard disk storage, the RAM and the number of processors. At a rough level, it can be said that for each slab the algorithm is exactly the same as the in-core version. Obviously, the main difference is that, once a slab has been processed the data is written to disk instead of keeping it in RAM. Although the data access to disk is much slower than the access to RAM memory, the degradation in performance of the out-of-core solver with respect to the in-core may be reduced up to 20%−30%. This is achieved by overlapping the disk access operations related with the future computations with the current computational process.

Coming back to the development of the graphical framework, the main idea is to develop a new GUI focused on electromagnetics. Furthermore, the definition of geometrical models with symbolic (non-numeric) variables is a fundamental requirement for the suite in order to enable the use of automatic goal oriented optimizations. Thereby, the graphical framework shall be developed from scratch to provide new important features such as, the parametrization of the geometry model in term of symbolic variables, the use of an automatic goal oriented optimizer, definition windows for electromagnetic variables, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite among others.

As commented above, the parametrization of the geometry model in term of symbolic variables enables the use of automatic optimization processes to reach a given goal. Thus, a new automatic goal oriented optimizer shall be developed to endow the suite with a powerful tool in the design of electromagnetic structures.

Finally, some optimization examples will be analyzed in order to demonstrate the performance and capabilities of the automatic goal oriented optimizer and, therefore, of the graphical framework of the suite.

It is worth to note that a user guide shall be written in order to provide to future users a document where to explain the details about how to perform a simulation through the suite. The user manual will describe step by step from the creation of the geometry model to the visualization of the results.

1.2.2 Finite Element Module

The development of a solver module based on `FEM` is another important objective of the present Ph. D. dissertation. The main idea of the development of this module is to complement the integral equation approach of the suite with a new differential electromagnetic solver. This new module shall be implemented from scratch making use of parallel paradigms in order to run efficiently from small laptop to high performance clusters with many `CPU` cores.

The module will use its own set of curl-conforming higher-order elements to approximate the solution of the Maxwell's equations [7, 41, 42]. These elements implement the first family of Nédélec curl-conforming elements proposed in 1980, [43]. It is worth noting that these elements are the appropriate choice for the discretization of the electric and magnetic fields due to provide tangential continuity across element interfaces. Furthermore, amount the mesh truncation techniques appeared in the literature, the module shall use of a boundary integral equation which provides a (numerically) exact radiation boundary condition. These unique features difference the `FEM` module developed in the present Ph. D. dissertation from other implementations.

Before the module may be used to provide solutions for real world problems, the accuracy of its results must be verified. This verification shall be performed employing the Method of Manufactured Solution (`MMS`). The basic idea of this technique is to manufacture an exact solution to some equation by solving the problem backwards. Thus, the method can be used to provide evidence that the code is correct and correctly implemented.

Finally, the validation of the results for real world application will be carried out in order to proof that the module is ready for research use. Numerous simulation for different electromagnetic problems will be performed comparing the result with established code (as commercial softwares) or directly with measurement.

1.2.3 Analysis of infinite periodic structures

Another objective of this Ph. D. dissertation is to develop a solver module for the `FEM` analysis of large antenna arrays using the infinite array approach. The infinite approach chosen in this dissertation is based on the use of the so-called Periodic Boundary Conditions (`PBCs`). Thus, the implementation of the `PBCs` in the `FEM` module of the suite shall be performed. In addition to the use of the `PBCs`, a truncation of the computational domain along the non-periodic direction is needed to perform the analysis for open scattering and radiation problems. Then, the modification of the truncation techniques supported by the `FEM` module shall be done enabling, together with the implementation of the `PBCs`, the analysis of infinite periodic structure by the suite.

It is worth to note that one of the truncation techniques supported by the `FEM` module makes use of a boundary integral equation. In this technique, the exterior infinite domain is truncated by an integral equation representation of the exterior field which is calculated using the Green's function $G(r, r_s)$. In the case of infinite periodic structures the previous Green's function must be replaced by the appropriate periodic Green's function. The main constrain of this approach is how extremely slow to converge is the periodic Green's function making the numerical calculation very difficult and computationally expensive. Thus, an acceleration technique is required in order to improve the converge rate of the periodic Green's function. Among the acceleration techniques existing in the literature, the Ewald's transformation has been chosen to accelerate the series since, this method is considered as the reference method for the efficient numerical calculation of the periodic Green's function [44, 45]. Hence, the derivation of the Ewald representation for the two-dimensional periodic Green's function shall be implemented in order to accelerate its calculation. Finally, analysis of some real structures in order to validate the results of the suite solving infinite periodic structures will be performed.

1.2.4 FEM-MoM hybridization

To perform the hybridization between the `FEM` and `MoM` modules of the suite a previous task is required in order to take a decision about the convenience of the hybridization. This task consists of, from one side, the study of the projection between the basis function of both codes in order to perform the connection of the polynomial approximation of the solution. From other side, the current variational formulation of the `FEM` module must be modified. This modification resides in the fact that the unknowns of a standard `MoM` formulation are the electric and magnetic currents over dielectric boundary surfaces and the electric currents over perfect electric conductors. However, the standard variational formulation of `FEM` only contains one of the previous unknowns. It is important to note that the `FEM` unknowns are field unknowns but they may be easily transformed to current unknowns performing an easy rotation of the field on the surface. Thus, the modification of the variational formulation of the `FEM` module is needed to provide unknowns for the dual field in a explicit way (H-field/E-field for E-field/H-field formulation, respectively)

1.3 Dissertation structure

A summary of the structure of the present Ph. D. dissertation is given in this section. The document has been structured in six chapters and two appendix as follows:

- Chapter 1: Introduction

  This initial chapter provides an overview of the antecedents and the objectives of the this Ph. D. dissertation. Details about the organization of this document are also given.

- Chapter 2: Graphical framework and optimizer development

  This chapter describes the development of the graphical framework of the suite from a technical point of view. Details about important tools such as, the definition of the models in terms of symbolic variables, the implementation of an automatic goal oriented optimizer, the automatic online update of the suite or the easy access to the menus through a navigation tree are given. Also, some numerical examples are described in order to illustrate the performance of the automatic goal oriented optimizer of the suite.

- Chapter 3: Finite Element Module

  This chapter provides a depth description of the Finite Element module of the suite. Details about the variational formulation employed and the basis functions used to approximate the solution are given in this chapter. Verification tests and numerical results are also provided in order to demonstrate the accuracy and performance of the `FEM` module of the suite.

- Chapter 4: Study of the hybridization of FEM-MoM techniques

  This chapter presents a modular approach to combine `MoM` and `FEM` techniques for the analysis of large structures or finite arrays with complex radiating elements. Details about the methodology of the approach are given in this chapter. Also, details about the variational formulation of this modular approach and a study of the connection between the basis functions of the `MoM` and `FEM` modules of the suite are described.

- Chapter 5: Analysis of infinite periodic structures

  This chapter gives a detailed description about the analysis of infinite periodic structures in the `FEM`. Also, details about the implementation of Periodic Boundary Conditions on `FEM` and the acceleration of the Periodic Green's function for the analysis of infinite structures are given. Finally, some numerical example are provided in order to demonstrate the capabilities and performance of the suite solving these type of problems.

- Chapter 6: Conclusions and future research lines

  This chapter provides the final conclusions of this Ph. D. dissertation, as well as a brief summary of the future research work that may be done to improve the performance and capabilities of the suite. This chapter also gathers a complete list of books, international journals and conference contribution that the work developed in this Ph.D dissertation has led.

- Appendix A: Solving Electromagnetic Problems in the suite

  This appendix provides a user manual about how to solve electromagnetic problems using the suite. The manual is based on the use of the `FEM` module of the suite but its use may be extensible to the `MoM` module. Detailed about the most important tools for the creation of the geometry or for the generation of the mesh are given. The manual is completed providing an example of the analysis from scratch of a simple waveguide model.

- Appendix B: Ewald representation for periodic Green's function

  This appendix gathers all the expression related to the Ewald representation for periodic Green's function as a summary. Due to the complexity and extension of the calculation of the first and the second derivative of the Ewald series the expressions of both derivatives are collected in this appendix.

CHAPTER 2

GRAPHICAL FRAMEWORK AND OPTIMIZER DEVELOPMENT

A detailed description of the development of the graphical user interface (GUI) is given in this chapter. The GUI is based on a general purpose pre and postprocessor called GiD that has been developed at CIMNE [46]. GiD has been designed to provide an easy and fast way to interconnect the in-house softwares developed in research groups with its powerful GUI. A brief description about the capabilities of GiD are given in section 2.1.

In the particular case of the suite, the main idea is to develop a new GUI focused only on electromagnetics, instead of developing a general purpose pre and postprocessor as GiD is. Furthermore, the definition of geometrical models with symbolic (non-numeric) variables is a fundamental requirement for the suite in order to enable the use of automatic goal oriented optimizations. Thereby, a complete framework has been developed from scratch to provide new important features to GiD such as, the parametrization of the geometry model in term of symbolic variables, the use of an automatic goal oriented optimizer, definition windows for electromagnetic variables, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite among others. Further details about this framework are given in section 2.2.

As commented above, the parametrization of the geometry model in term of symbolic variables enables the use of automatic optimization processes to reach a given goal. Thus, a new automatic goal oriented optimizer has been developed to endow the suite with a powerful tool in the design of electromagnetic structures. Section 2.3 describes the details of the implementation of this optimizer and how it has been integrated in GiD.

Finally, section 2.4 shows some optimization examples in order to demonstrate the performance and capabilities of the automatic goal oriented optimizer and, therefore, of the `GUI` of the suite.

2.1 Pre-postprocessor GiD

GiD is a universal, adaptive and user-friendly pre and postprocessor for numerical simulations in science and engineering. It has been designed to cover all the common needs in the numerical simulations field from pre to post-processing tasks: geometrical modeling, effective definition of analysis data, meshing, data transfer to analysis software, as well as the visualization of numerical results [46]. GiD main characteristics are:

- **Universal:** GiD is able to generate the information required for the analysis of any problem in science and engineering.
- **Adaptive:** GiD is very easy to adapt to any numerical simulation code, offering the possibility to customize its input and output data in order to made them compatible with an existing in-house software. Figure 2.1 shows a diagram block of a typical interaction between GiD and an in-house solver.
- **User-friendly:** GiD offers many different tools to facilitate the input data preparation and results visualization to the user.

GiD allows to the users the creation of very complex geometry definitions featuring the widely used `NURBS` surfaces. Also, it offers compatibility with the most important `CAD` software packages such as, `IGES`, `DXF`, `STL`, `3DS`, `RHINO` and so on.

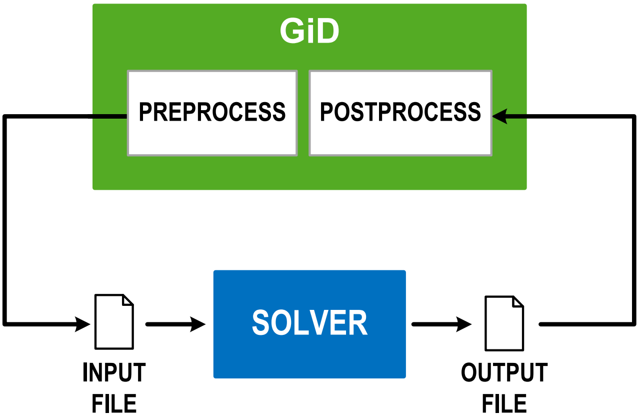


Figure 2.1: Typical interaction process with GiD

Another important feature of GiD is the possibility to generate large meshes in a fast and efficient manner, using several algorithms, for surfaces and volumes. Several element types are also supported such as triangles, quadrilaterals, circles, hexahedras, prisms, tetrahedras or spheres. Thereby, GiD offers an excellent and easy way to interconnect the in-house softwares developed in research group with its powerful GUI.

The integration of these in-house softwares with GiD may be performed through two different integration levels depending on the needs and capabilities of the user. GiD provides a basic integration level where programming knowledge is not required or, conversely, an advanced integration level where users has the full control of look and behavior of GiD with the consequent use of script programming languages.

- **Basic integration level:** The integration can be done for any user regardless of his/her programming knowledge. Only a couple of text files, using an easy keyword system, should be written describing the user's problem properties (conditions, materials, etc.) and GiD will automatically create the corresponding windows, allowing the end user to manage the data of the problem: assign or modify conditions, draw properties over model and so on. More information about this integration method may be found in GiD customization manuals [46].

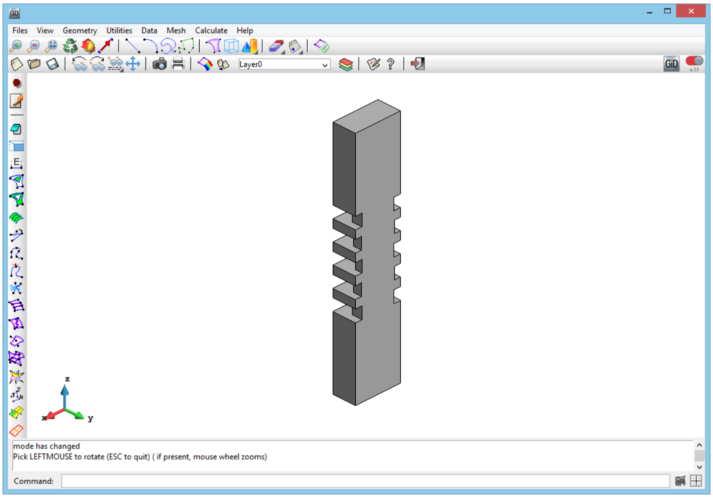
- **Advanced integration level:** This method offers many more possibilities, including an appealing view of the managed data and a display of the problem and group data during pre and postprocess. The advanced integration, with full control of look and behavior of GiD, is made possible by the use of `TCL/TK` scripting language. These advanced customization features provide a way to create your own connection with in-house or commercial numerical simulation codes.

In the particular case of this suite, the advanced integration level of GiD has been chosen as the development model for the `GUI`. The main idea is to develop a new `GUI` focused only on electromagnetics, instead of be a general purpose pre and postprocessor as GiD is. Thereby, a full framework has been developed to provide new important features to GiD such as, new definition windows for electromagnetic variables, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite among others.

To give the reader a first idea about the depth of the changes provided by the framework, figure 2.2 shows the graphical aspect of the `GUI` before and after the application of the framework. A total change in the graphical aspect may be appreciated, where the position of the top menu in figure 2.2(a) is changed to the right on the new `GUI` and the navigation tree appears on the left in figure 2.2(b) among other changes.

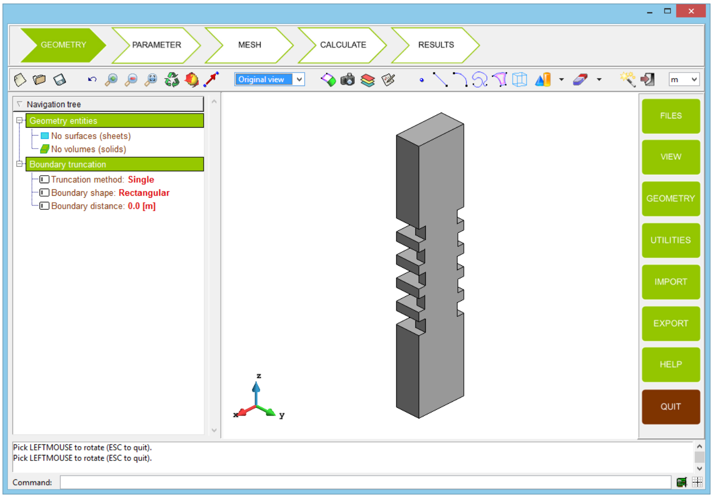
It is worth to note that one of the main important features provided by this new framework is the capability to define the geometry model in term of symbolic variables. The definition and/or modification of the geometry in GiD (as it happens in other pre-processors) must be performed by using real or integer numbers. For instance, the coordinates of a point, or the vector used to define a translation operation, must be real numbers. That is, the use of symbolic variables is not supported directly by GiD, but in contrast, it is supported by the suite thanks to this new framework. Under this scenario, the symbolic variables tool becomes a very important contribution of the present dissertation to GiD.

The framework has been implemented using the `TCL-TK` language (the same language used to program/customize GiD) and it acts as an upper layer between GiD itself and the users. Figure 2.3 shows a block diagram of the interaction between users and the suite.



mode has changed
Pick LEFTMOUSE to rotate (ESC to quit) (if present, mouse wheel zooms)

(a) Before framework application



Navigation tree
- Geometry entities
  - No surfaces (sheets)
  - No volumes (solids)
- Boundary truncation
  - Truncation method: Single
  - Boundary shape: Rectangular
  - Boundary distance: 0.0 [m]

Pick LEFTMOUSE to rotate (ESC to quit).
Pick LEFTMOUSE to rotate (ESC to quit).

(b) After framework application

Figure 2.2: Graphical aspect before and after framework application

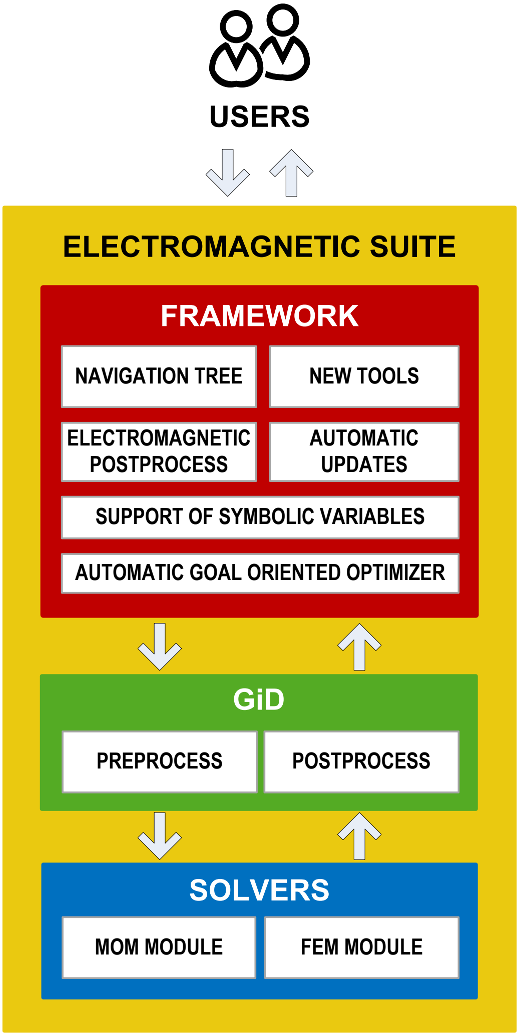
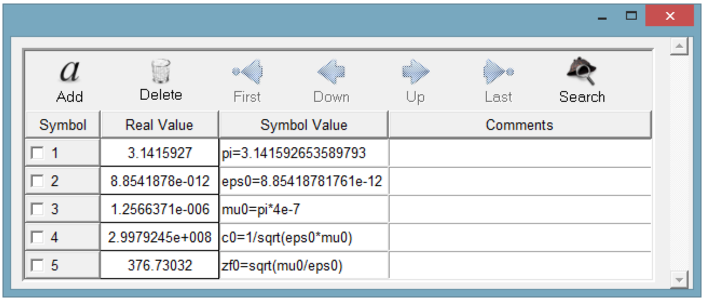


Figure 2.3: Block diagram of the interaction between users,the developed framework, GiD and the solver modules

The framework appears in red with some of the most important tools provided by it. The framework is the part of the `GUI` that interacts with the users receiving their requests. GiD appears in green and interacts with the framework. Finally, the `MoM` module and the `FEM` module of the suite are included in blue.



Symbol	Real Value	Symbol Value	Comments
□ 1	3.1415927	pi=3.141592653589793	
□ 2	8.8541878e-012	eps0=8.85418781761e-12	
□ 3	1.2566371e-006	mu0=pi*4e-7	
□ 4	2.9979245e+008	c0=1/sqrt(eps0*mu0)	
□ 5	376.73032	zf0=sqrt(mu0/eps0)	

Figure 2.4: Window that controls the symbolic variables

2.2 Suite framework

As commented in the previous section, the framework has been implemented using TCL-TK language and it acts as an upper layer between GiD itself and the users. The framework provides important features such as, a new graphical aspect, the capability to define the geometry model in term of symbolic variables, new windows to define the electromagnetic properties, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite. Details about the implementation of these tools are given in this section.

2.2.1 Implementation of symbolic variables tool

The symbolic variables tool acts between the user, the symbolic variables and GiD. The definition and/or modification of the geometry in GiD must be performed by using real or integer numbers. For instance, the coordinates of a vector used to define a translation operation or the control points of a NURBS line must be real numbers.

In this context, it is very important to create a controlled environment where the users may define and use their symbolic variables. Then, the developer may know when the users are defining or using a symbolic variable and what features of GiD are going to employ. To create this controlled environment, a window that allows the users to create, delete and/or modify the symbolic variables has been developed first. Thus, the developer has access to a list with the relationship between the symbolic variables and its real values. Figure 2.4 shows an screenshot of the symbolic variables window of the suite.

The next step is to provide the users the necessary facilities to use these symbolic variables in GiD. For this reason, a full set of commands have been written in order to act between the users and GiD. The framework detects the GiD command that is being called by the user and it exchanges the symbolic variables for their real value before calling the corresponding GiD command. For instance, lets imagine that a user wants to move a point from its original position to another. GiD offers a command called `MovePoint` that receives the identifier of the point to be moved and the real coordinates of the new location. The user invokes the `MovePoint` command by clicking on the corresponding entry of the menu. Then, the user selects the desired point to be moved and types the new location using some of the symbolic variable defined through the symbols window commented above. At this moment, the framework is able to exchange the symbolic variables by their real value checking the symbol list and calls the `MovePoint` command but using the corresponding real numbers required by GiD. Pseudocode 2.1 shows how to exchange the symbolic value of a variable by its real value in `TCL-TK`.

```
#Define two symbols as example
set a 2; set b 5
#Define one expression as example
set expression "c=a+b"

#Replace the symbol (a) by its real value (2)
regsub -all -nocase -- "\\m a \\M" $expression \
    $real_value_for_a $exchange
#Assign real expression to the variable
set expression $exchange #(c=2+b)
#Replace the symbol (b) by its real value (5)
regsub -all -nocase -- "\\m b \\M" $expression \
    $real_value_for_b $exchange
#Assign real expression to the variable
set expression $exchange #(c=2+5)

#Return the real value of the symbol c
return [expr $expression]
```

Pseudocode 2.1: `TCL-TK` code that exchanges the symbolic value of a variable by its real value

Some other operations such as, changing the radius of an sphere or the length of a cylinder requires a more complex implementation of the tool. In this case, when the user creates an object (sphere, cylinder, cone and so on) the framework marks the entities that conforms that object. Then, the framework knows the entities that it should change if the user modifies any parameter of the object. If, for instance, the radius of an sphere is controlled by the symbolic variable a and the user changes the value of that variable, the framework checks the objects that use the symbolic variable a and regenerates the geometry of those objects. This process is performed in two steps; firstly, the framework deletes the entities corresponding to the objects that use the symbolic variable a, and following, the framework creates new objects with the corresponding new values. This process is performed internally and completely hidden for the users that only notice a change in the geometry of the objects in the screen.

It is worth to note that this task requires a list with the entities of the geometry that have been parametrized and that list must be updated any time the user performs a change in the geometry. For this reason, the commented controlled environment is very important in order to avoid errors when the framework interacts with GiD.

A list with the most important commands of GiD that support the symbolic variables thanks to the framework is shown below:

- `MovePoint:` By using this command, an existing point is selected and moved to another position. The framework receives the identifier of the point and the three coordinates of the new location.

- `CreateLine:` This command creates a new `NURBS` line. The framework receives the coordinates of the two end points for straight lines, or a list with the control points for `NURBS` lines.

- `ParamEquations:` Parametric equations defining lines and surfaces may contain symbolic variables enabling automatic changes in the shape of the lines or surfaces. The framework regenerates the lines/surfaces as it was describe previously when the symbolic variables changes their values.

- `CreateObjects:` This command is used to create new predefined objects. The objects are already parametrized and symbolic variables may be used to control their geometry. Several objects are supported, such as, spheres, cylinders, cones, prisms and rectangles.

  `Spheres:` In the case of the spheres, the radius is the only variable utilized to control the geometry.

  `Cylinders:` The length and the radius of the cylinder are the variables used to control the geometry in this case.

  `Cones:` In the case of the cones, the radius and the length are the variables utilized to control the geometry.

  `Prisms:` The width of the base and the length of prism are the variable employed to control the geometry.

- `Move:` This command is used to move the desired entities of the geometry. Several operations are supported, such as, translation, rotation and scale.

  `Translation:` This operation performs the movement of entities from one point to another.

  `Rotation:` This operation performs the rotation of entities in any direction (defined through a rotation axis) with a desired angle.

  `Scale:` This operation is used to scale the selected entities in a factor.

- `Copy:` This command is used to copy the desired entities of the geometry. The operations supported by this command are translation, rotation and scale.

  `Translation:` This operation performs a copy of the selected entities from one point to another.

  `Rotation:` This operation performs a copy of the selected entities rotating them in any direction (defined through a rotation axis) with a desired angle.

  `Scale:` This operation performs a copy of the selected entities scaling them in a factor.

2.2.2 Implementation of new windows

The new windows provided by the framework, as for example, the one used by the symbolic variables tool have been implemented using appropriate `TCL-TK` commands.

The first step to create a window in `TCL-TK` is to call the command `toplevel`. This command creates and manipulates an instance of a toplevel entity where to add the rest of the `TCL-TK` widgets that will conform the window. Lines 1−5 of pseudocode (2.2) shows how to define an instance of the toplevel widget.

The next step is to provide functionality to the window creating some `TCL-TK` widgets as frames, labels or text entry boxes. The commands `ttk::frame`, `ttk::label` and `ttk::entry` create and manipulate those widgets. In this case, lines 7−16 of pseudocode shows how to define a frame with a label and an entry inside.

Finally, some buttons are created to accept or cancel the changes performed in the entry box. Lines 18−29 give details about how to create these buttons. Figure 2.5 illustrates the window that may be created following the pseudocode (2.2).

2.2.3 Implementation of the navigation tree

The navigation tree of the suite has been created using the `TCL-TK` package `treectrl 2.4.1`. This package provides the tools to create and manipulate hierarchical multicolumn widgets. An example of a navigation tree is shown in figure 2.6 where the elements of the tree may be appreciated. The `treectrl` command is used to define the root widget where to add the items of the tree. Pseudocode (2.3) shows how to define a navigation tree with a vertical scrollbar.

Once the tree has been created, the definition of the style of the items of the tree is required. The first step to create a new style is to define the elements that will conform it. For instance, lets define a style with a border, an image and a text box (see lines 1−6 of pseudocode 2.4).

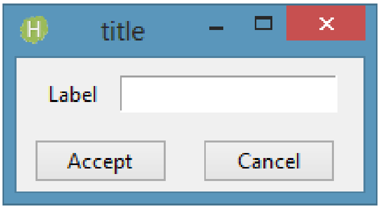


Figure 2.5: Example of a TCL-TK window

```
#Definition of the toplevel window
toplevel .gid.window_name -class OverlayWindow
#Definition of the window title
wm title .gid.window_name "window title"
wm iconname .gid.window_name "window title"
...
#Create a frame where to add new widgets
set f [frame .gid.window_name.frame]
#Create a label
label $f.label -text "label"
#Create an entry box where to insert data
entry $f.entry -textvariable variable_data
#Print the frame into the window
grid $f -sticky nsew
#Print the label and the entry
grid $f.label $f.entry -sticky nsew
...
#Create a frame where to place the buttons
set f [frame .gid.window_name.fbuttons]
#Create the accept button
button $f.accept -text "Accept" \
    -command [accept_procedure]
#Create the cancel button
button $f.cancel -text "Cancel" \
    -command [cancel_procedure]
#Print the button frame into the window
grid $f -sticky nsew
#Print the buttons
grid $f.accept $f.cancel -sticky nsew
```

Pseudocode 2.2: Definition of a top window in TCL-TK

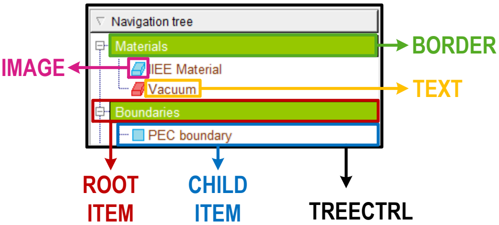


Figure 2.6: Elements of the navigation tree

The next step is to create the style with the command `style create` and add the elements that conform the style (see lines 8–12). Finally, the layout of the style is created with the command `style layout` (see lines 14–21).

The last step is to create the items that will conform the tree using the command `item create`. For example, lets add to the navigation tree a root item called `Materials` and one child item called `Vacuum`. Lines 1–11 of pseudocode (2.5) create the root item `Materials`. The item is created in line 2 calling the command `item create`. Then, the desired style is assigned to this item (see line 4) and the label `Materials` is set (see lines 6–8). Finally, line 11 shows how the item is added as root to the tree. Lines 13–26 create the child item `Vacuum`. As it occurs with the root item, lines 13–20 create the item, select the desired style and configure the label. In this case, an image is also configured as may be appreciated in lines 21–23. Finally, the item is added as child to the root item (see line 26).

```
#Navigation tree definition
treectrl .tree -yscrollcommand {.treevs set}
#Vertical scrool bar definition
scrollbar .treevs -command {.tree yview}
#Print the tree and the scroolbar in the screen
grid .tree   -row 0 -column 0 -sticky nsew
grid .treevs -row 0 -column 1 -sticky ns
```

Pseudocode 2.3: Definition of a navigation tree

```
#Create the border element of the style
.tree element create border_name border
#Create the image element of the style
.tree element create image_name image
#Create the text element of the style
.tree element create text_name text

#Definition of the name of the style
.tree style create style_name
#Definition of the element of the style
.tree style elements style_name \
    {border_name image_name text_name}

#Create the layout of the style
.tree style layout style_name border \
    -union {image text} -iexpand nsew

#Create the layout of the image
.tree style layout style_name image
#Create the layout of the text
.tree style layout style_name text
```

Pseudocode 2.4: Definition of the style of a tree item

2.2.4 Implementation of the online update tool

The online update tool of the suite has been implemented using the `TCL-TK` packages `http` and `tls`. The `http` package provides tools to create and manipulate connections through the `HTTP` protocol. The `tls` package supports the mechanism to make the previous connections safe.

The tool starts by connecting to the server and detecting if a new version of the suite is available. Pseudocode (2.6) shows how the tool establishes a secure connection to the server (lines 1–6). Then, the tool waits for the server response (lines 8–9) and gets the `http` code of the connection. If the code is different from 200, the connection is not correct and an error is returned (lines 10–17). If the connection is correct, the string with the version available in the server is translated to an ASCII format and compared with the current version of the software.

```
#Definition of a root item
set root_item [.tree item create -button auto]
#Definition of the style of the item
.tree item style set $root_item 0 root_style

#Configuration of the text element of the style
.tree item element configure $root_item 0 \
    text_name -text "Materials"

#Add the item to the tree as root item
.tree item lastchild root $root_item

#Definition of a child item
set child_item [.tree item create -button auto]
#Definition of the style of the item
.tree item style set $child_item 0 child_style

#Configuration of the text element of the style
.tree item element configure $child_item 0 \
    text_name -text "Vacuum"
#Configuration of the image element of the style
.tree item element configure $child_item 0 \
    img_name -image [image create photo img.png]

#Add the item to the root as child item
.tree item lastchild $root_item $child_item
```

Pseudocode 2.5: Creation of the items of the tree

If they are different, the tool asks if the user wants to update the software (see lines 19–34). The new version is compressed in a ZIP file located in the server. If the user wants to update the software, the tool downloads the `ZIP` file, decompresses its content and, finally, deletes the `ZIP` file. Pseudocode (2.7) illustrates this step, where lines 1 – 23 show how the tool downloads the `ZIP` file from the server. Line 25 checks if the response of the server is correct. Once the response is correct, the downloaded file is closed and the `http` token is cleaned (lines 30–33). Finally, the tool decompresses the file (lines 34–36) and deletes the update file (lines 37–38).

```
#Register the port for secure connection
::http::register https 443 ::tls::socket

#Get the version of the suite from the server
set token [http::geturl "$ip_server/version" \
    -binary 1 -timeout 20000]

#Wait for the server response
::http::wait $token
#Get the http code
set ncode [::http::ncode $token]

#Diferent from 200, the result is not correct
if {$ncode != 200} {
    set error "Error connecting to server..."
    return
}

#Translate the version to ASCII
binary scan [::http::data $token] \
    SSS val1 val2 val3

#Obtain the version available in the server
set httpversion "${val1}.${val2}.${val3}"

#Clean the http token
::http::cleanup $token

#Compare the versions
if {$httpversion != $current_version} {

    #Update the software??
    ...
}
```

Pseudocode 2.6: TCL-TK code that detects if there is a new version of the suite available to be downloaded

```
#Open the file where to save the results
if {[catch {set out [open "file.zip" w]} ops]} {
    #Problems retrieving the file from server
    return
}

#Gets the file
if {[catch {set token [http::geturl "file.zip" \
    -channel $out -binary 1]} oops]} {
    #Problems retrieving the file from server
    ...
    #Close the file
    close $output
    #Delete the file
    file delete [file join "file.zip"]
    #End task
    return
}

#Wait for the token
::http::wait $token
#Get the http code
set ncode [::http::ncode $token]
#Diferent from 200, the result is not correct
if {$ncode != 200} {
    set error "Error connecting to server..."
    return
}

#Close the file
close $output
#Clean the token
::http::cleanup $token
#Decompress the update
runExe run "unzip.exe -x -o -U file.zip" \
    -blocking false -timeout 3600
#Delete the update file
file delete [file join "file.zip"]
```

Pseudocode 2.7: TCL-TK code that updates the version of the suite

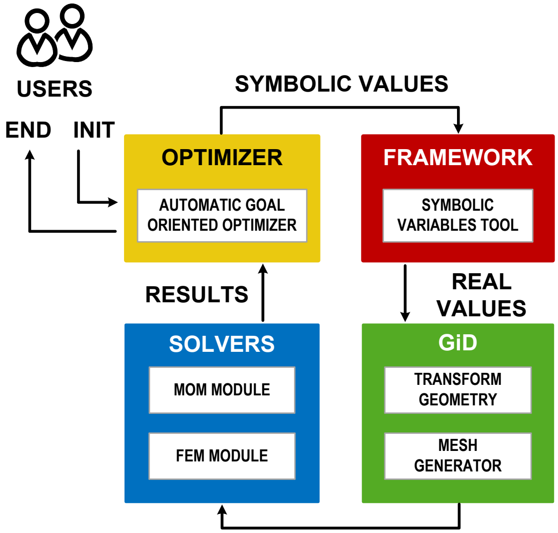


Figure 2.7: Optimizer flowchart

2.3 Automatic Goal Oriented Optimizer

An automatic goal oriented optimizer has been developed to equip the suite with a powerful tool in the design of electromagnetic structures. It is worth to note that the development of the optimizer is the main reason of the implementation of the commented symbolic variables tool. Users can use the optimizer to automatically adjust the designated model parameters such as model-element coordinates, object length, and similar quantities to achieve an improvement of the results like maximum gain and low side lobes.

2.3.1 Flowchart of the optimizer

Before using the optimizer, the user needs to define the model in term of symbolic variables with the appropriate tools provides by the framework. Further details about these tools were given in Section 2.2. Once the model is defined in terms of symbols, the optimizer can drive, without the user intervention, the modification of the model (geometry dimensions, material constants, etc) by changing the actual values of the symbols used to define the model. The flowchart of the optimizer is depicted in figure 2.7.

The optimizer calls the solver with the initial values of the symbols. If the goal is satisfied, the optimization process stops; otherwise, another iteration is started with new values of the optimization variables given by the optimizer. For that purpose, the symbolic variables tool of the framework exchanges the symbols for their real values and executes the needed operations (within GiD) to modify the geometry of the model (and also other parameters, as material constants, and so on). Once the geometry is modified, a new mesh is generated using the GiD mesh generator. The next step is to call the solver to in order to perform the simulation. These results are given again to the optimizer module and the process continues until either the goal is satisfied or the maximum number of iterations predetermined by the user is reached.

Several optimization methods are available in the suite, such as, Powell's Method [47], Simplex Method [48] and Particle Swarm Optimization (PSO) [49].

2.3.2 Fitness function

The fitness function of the optimization (i.e., difference between the simulated results and the given criteria) is calculated using the formula:

$$FF(\mathbf{x}) = \frac{w_i}{\alpha_i} f_i(\mathbf{x}), i = 1, 2, ...n \quad \text{with} \quad f_i(\mathbf{x}) = \sqrt{\sum_{k=1}^{num} (G_k(\mathbf{x}) - G_0)} \tag{2.1}$$

where FF is the total fitness value and f_i is the fitness value for i-th criterion. The total fitness value is the addition of fitness value for each criterion that needs to be normalized and weighted. $\mathbf{x}$ are the variables to be optimized. There are n criteria in total, and each criterion is associated with a normalization factor α_i and a weight w_i. The weight is used to differentiate the priority of the criteria, and the normalization factor is used to balance the values for different levels of criteria (e.g. the gain and/or the impedance). The fitness value for each criterion is calculated using the fitness value $f_i(\mathbf{x})$. For the i-th criterion, the optimizer stores the results that satisfy the i-th criterion and computes the error between the collected value and the objective value. For example, if the criterion is the optimization of the radiation pattern within a certain angular range, then there may be some intermediate results that meet the given criterion within the angle of interest and they will be stored by the optimizer. Within the stored results, some of which satisfy the given criterion will not be computed for the error between the stored value and the objective value. In this way, the optimization efforts will be focused on the results that do not satisfy the criterion. $G_k(\mathbf{x})$ stands for the stored results that does not satisfy the criterion. num is the number of results being stored and computed for the value of the cost function, and G_0 is the target value of the i-th criterion.

When the fitness function is optimized, the computed pattern becomes closer to the objective pattern. Thus, the goal of the optimizer is to use optimization algorithms to decrease the total fitness function value.

2.4 Numerical results

To illustrate the capabilities and performance of the suite some examples are included in this section. It is worth to note that, in the current version of the suite, the optimizer is only linked to the MoM module of the suite. In future developments, the optimizer will be linked also with the FEM module. The main capabilities and features of the MoM module of the suite were already introduced in Chapter 1. An example of its performance is shown below.

The bistatic analysis of an apache helicopter as the shown in figure 2.8 has been carried out. The helicopter has been illuminated by an incident plane wave at 800 MHz with $\theta\theta-$polarization along the y$-$axis. The helicopter model at that frequency has an electrical size of $47.2\,\lambda$, $38.93\,\lambda$ and $10.13\,\lambda$, which makes the electromagnetic analysis a challenge. The wheels and the blades of the helicopter have been considered as dielectric to make a realistic model. A dielectric constant $\epsilon_r = 4.5$ has been chosen since the values of ϵ_r for different materials, mainly carbon and glass fibers, are around 4.5.

The simulation has been performed on high performance computing cluster formed by 35 nodes with a total of 560 AMD CPUs: 16 CPU cores on each node and 4 GB RAM per core, and a total of RAM approximately equal to 2.24 TB. No hard disk storage is available for computations.

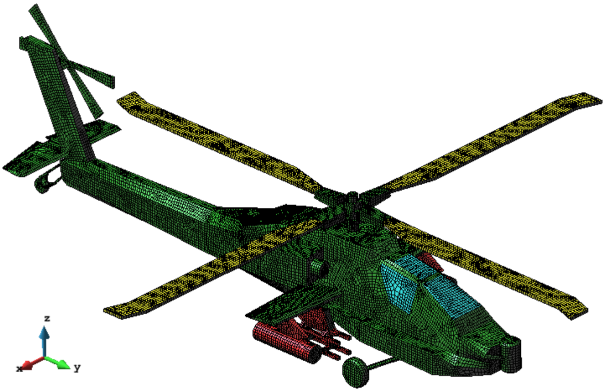


Figure 2.8: Model of apache helicopter

The total number of degrees of freedom for the accurate modeling of this problem was around 255000 (half in the case of no dielectric parts in the model). The total `RAM` memory needed to solve this problem was 968.94 `GB` and the simulation took 9.67 hours. Note that the degrees of freedom considered in this case are regarding to the `MoM` module of the suite. The computed results for azimuth angle are shown in figure 2.9. The main lobe of the `RCS` appears in the tail of the helicopter ($\phi = 270°$), while the incident plane wave is coming from ($\phi = 90°$)

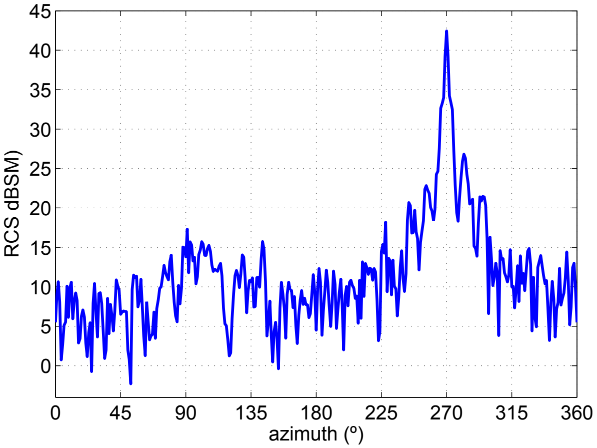
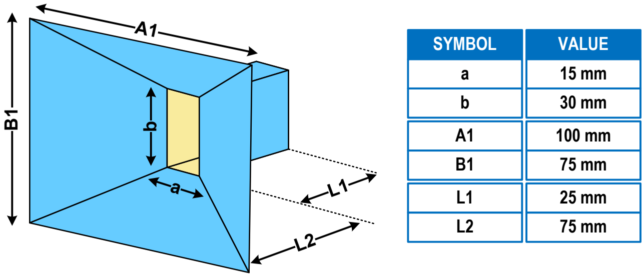


Figure 2.9: Bistatic `RCS` of real helicopter at 800 `MHz` in azimuth



SYMBOL	VALUE
a	15 mm
b	30 mm
A1	100 mm
B1	75 mm
L1	25 mm
L2	75 mm

Figure 2.10: Horn antenna before optimization

The following examples illustrate the optimization process of two antennas. The first one shows the optimization process of the dimensions of a horn antenna to obtain a certain gain along the broadside direction. The second example consists of the optimization of the radiation pattern of a slotted waveguide array. In particular, the goal is to maximize the main love and minimize the sidelobe levels (SLLs).

2.4.1 Optimization of a horn antenna

The aim of this optimization process is to obtain a horn antenna with a 17 dB gain along the broadside direction. The working frequency in this case is 2.5 GHz. Figure 2.10 shows the dimensions of the horn antenna before starting the optimization process. The 3D representation of the computed gain pattern is shown in figure 2.11. It is observed that the initial gain of the antenna is 11.74 dB. In this case, the computed gain must be enhanced more than 5 dB by changing the dimensions of the horn antenna. The Powell's Method is chosen as optimization method. The maximum number of iterations is set to 500.

Figure 2.12 shows the dimensions of the horn antenna after 298 iterations at which the desired goal has been reached. It is observed how the optimizer has changed the dimensions of the horn antenna. The 3D computed gain pattern of the antenna after 298 iterations is shown in figure 2.13. A maximum gain of 17.53 dB, which satisfies the desired goal, is observed. The number of unknowns for this example is 900 and the total time used in the optimization process is 16.5 minutes (3.2 seconds per iteration) using a single desktop computer with 4 cores at 2.4 GHz and 3.2 GB of memory in the simulation.

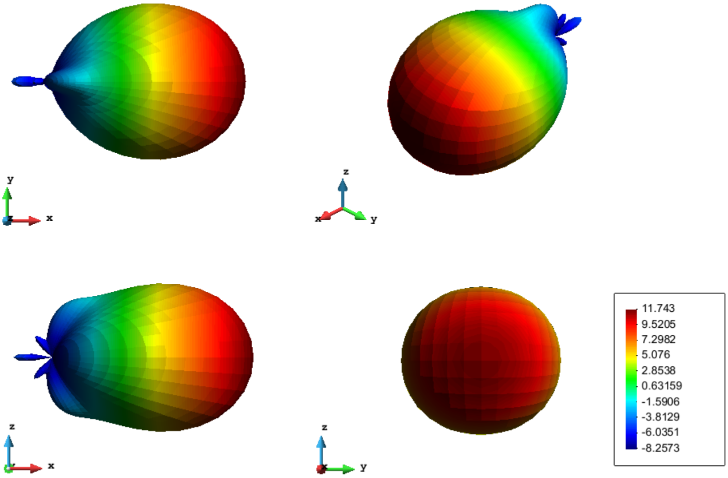
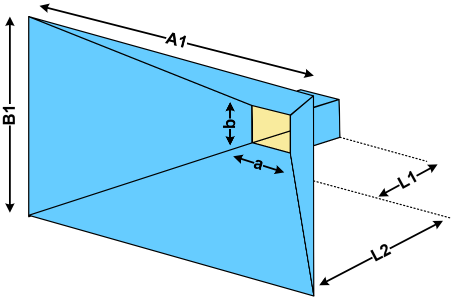


Figure 2.11: Horn antenna before optimization



SYMBOL	VALUE
a	20.5 mm
b	22.0 mm
A1	200.1 mm
B1	129.5 mm
L1	46.8 mm
L2	370.4 mm

Figure 2.12: Horn antenna before optimization

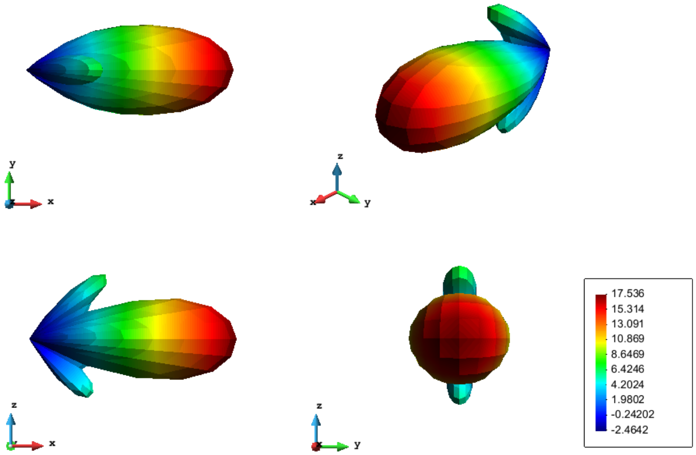


Figure 2.13: Horn antenna after optimization

2.4.2 Optimization of slotted waveguide array

The slotted waveguide antenna array is formed by the combination of 10 single slotted waveguides. Each single waveguide has 10 narrow-wall slots, with dimensions of 22.86 mm by 10.16 mm, and wall thickness of 1.00 mm. The whole length of the waveguide is 266.58 mm. Figure 2.14 gives the geometry of the inclined slot. The inclined angle of the slots is set to be θ, with cutting depth h (measured from the inner wall), and the width of the slots is w. A small dipole inside the waveguide is used as the excitation, as shown in figure 2.15. The ends of the waveguides are shorted, with the feeding dipole placed at 0.25 λ_g from one end (λ_g is the guided wavelength). The operating frequency of interest is 9.375 GHz.

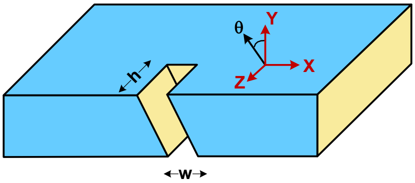


Figure 2.14: Geometry of the inclined slot

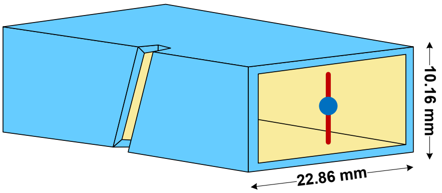


Figure 2.15: Model of the excitation of the waveguide

From array theory, a 20 dB Taylor distribution is used to determine the feeding of the waveguide array to achieve the desired radiation pattern. There is a $0.9\,\pi$ phase difference between the feeds of the adjacent radiating waveguides.

The symbol list to be optimized contains 20 symbols in total. There are 10 slots on the wall of each waveguide, where 10 symbols are used to compute the inclined angle θ for each of the ten slots, and another 10 symbols represent the cutting depth h for each slot. Each waveguide in the array has the same dimensions except for alternating slot incline angles as shown in figure. Hence, a total of 20 variables need to be optimized. The goal is to optimize the radiation pattern of the slotted waveguide array. Due to the phase differences between the feeds, the direction of the main lobe is 5° toward the y-axis measured from the z-direction, with corresponds to $\phi = 90°$, $\theta = 5°$. The goal is to maximize the main love and minimize the sidelobe levels (`SLLs`).

After 100 iterations, the optimization finishes as the iteration number reaches the predefined limitations. In figure 2.16 which shows the xz-plane, it is seen that the main lobe remains roughly the same after optimization; however, the largest sidelobes at angles $\theta = 5°$ and $\theta = -15°$ are suppressed by about 3 dB. The solid and dashed indicators mark the highest sidelobe for the initial and optimized results, respectively. It is easy to see that the `SLL` performance is improved by about 3 dB. Figure 2.17 shows the yz-plane, and similarly, the solid and dashed indicators show the highest sidelobe and the mainlobe values for the radiation pattern before and after optimization, respectively. It can be seen that both the mainlobe and the largest sidelobe decrease by roughly 1 dB. The pattern has been improved in the xz-plane, whereas in the yz-plane, it remains relatively the same in terms of the mainlobe to peak sidelobe performance.

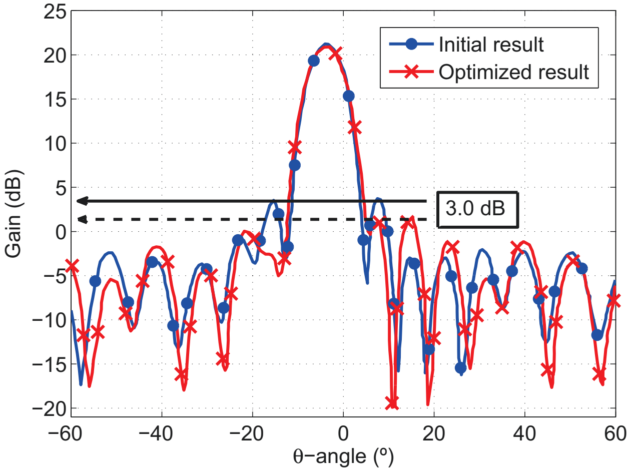


Figure 2.16: Radiation pattern of the slotted waveguide array after optimization (xz-plane)

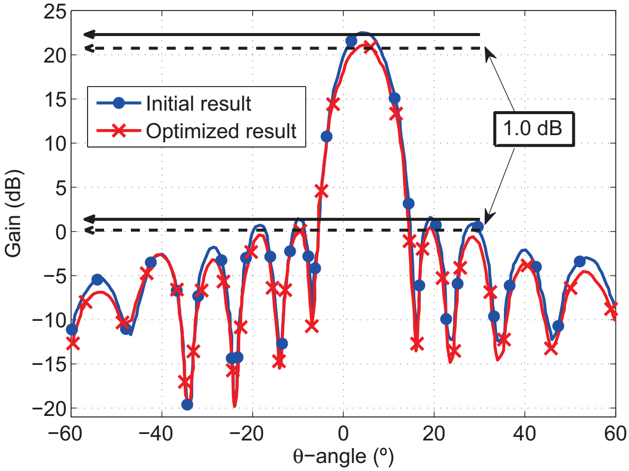


Figure 2.17: Radiation pattern of the slotted waveguide array after optimization (yz-plane)

2.5 Conclusions

A detailed description of the development of the graphical user interface (`GUI`) has been given in this chapter. The `GUI` has been based on a general purpose pre and postprocessor called GiD. The main idea has been to develop a new `GUI` focused only on electromagnetics, instead of developing a general purpose pre and postprocessor as GiD is. Thereby, a full framework has been developed to provide new important features to the software GiD, such as, a symbolic variables tool that enables the parametrization of models using symbols, new definition windows for electromagnetic variables, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite among others.

Also, an automatic goal oriented optimizer has been developed to endow the suite with a very powerful tool in the design of electromagnetic structures. Users can use the optimizer to automatically adjust the designated model parameters and improve the results like maximum gain and low side lobes.

CHAPTER 3

FINITE ELEMENT MODULE

A detailed description of the Finite Element module of the suite is given in this chapter. The module is based on the Finite Element Method (FEM) and makes use of edge-based elements to solve the Maxwell equations, not only for the electric field, but also for the magnetic field. The variational formulation used by the module is introduced in Section 3.1.

The use of FEM for the analysis of radiation/scattering problems requires the truncation of infinite space into a finite computational domain. This truncation can be accomplished by introducing an artificial surface to enclose the antenna. However, to emulate the original free-space environment, the artificial truncation surface should absorb as much of the radiated field as possible in order to reduce any artificially reflected fields back to the computational domain. Amount the mesh truncation techniques appeared in the literature, the module makes use of a boundary integral equation which provides a (numerically) exact radiation boundary condition. Details about this truncation technique are given in Section 3.2.

In order to approximate the solutions for the vector wave equations arising from Maxwell's equations, the module uses edge-based elements. Particularly, the finite domain is discretized into tetrahedral curl-conforming elements of second-order belonging to the Nédélec family.

These elements are the appropriate choice for the discretization of the electric and magnetic fields due to provide tangential continuity across element interfaces. Section 3.3 provides a brief escription about these elements and the set of interpolation polynomials used by the FEM module of the suite.

The system of equations given by FEM provides very sparse matrix that have to be solve using advanced sparse solvers. The implementation of the module lets the choice of four different sparse solvers, such as, MUMPS [50], HSL [51], UHM and PARDISO [52]. Furthermore, the module is completely parallelized and able to run from small laptop to high performance clusters with many CPU cores. Further details about the how the module was implemented may be found in Section 3.4.

Before a computer code can be used to provide solutions for real world problems, the accuracy of its results must be verified. A very powerful way to perform this verification is to employ the Method of Manufactured Solution (MMS). The basic idea of this technique is to manufacture an exact solution to some equation by solving the problem backwards. Thus, the method can be used to provide evidence that the code is correct and correctly implemented. Some of the verification tests carried out are documented in Section 3.5

Finally, the validation of the results for real world application is required in order to proof that the module is ready for research use. Thus, numerous simulation for different electromagnetic problems have been performed comparing the result with established code (as commercial softwares) or directly with measurement. Section 3.6 gathers some results of real applications where a great agreement may be found in all the cases.

3.1 Variational Formulation

The module is based on the so-called Finite Element - Iterative Integral Equation Evaluation (FE-IIEE) method. This technique uses the well-known Finite Element Method (FEM) to model the interior finite domain Ω^{FEM} of the problem using a Cauchy (Robin) type of boundary condition for the mesh truncation boundary.

Table 3.1: Formulation magnitudes and parameters

	$\mathbf{V}$	$\bar{\bar{f}}_r$	$\bar{\bar{g}}_r$	h	$\mathbf{O}$	$\mathbf{L}$	$\Gamma_{\mathtt{D}}$	$\Gamma_{\mathtt{N}}$
Form. $\mathbf{E}$	$\mathbf{E}$	$\bar{\bar{\mu}}_r$	$\bar{\bar{\epsilon}}_r$	η	$\mathbf{J}$	$\mathbf{M}$	$\Gamma_{\mathtt{PEC}}$	$\Gamma_{\mathtt{PMC}}$
Form. $\mathbf{H}$	$\mathbf{H}$	$\bar{\bar{\epsilon}}_r$	$\bar{\bar{\mu}}_r$	$\frac{1}{\eta}$	$\mathbf{M}$	$-\mathbf{J}$	$\Gamma_{\mathtt{PMC}}$	$\Gamma_{\mathtt{PEC}}$

The exterior infinite domain is truncated by an integral equation representation of the exterior field. Then, the solution is obtained through an iterative process in which the residual of the radiation boundary condition on the mesh truncation boundary is updated.

It is worth noting that the exterior infinite domain can be truncated with the mentioned Iterative Integral Equation Evaluation Method (see Section 3.2 for further details), or, conversely, it can be truncated by using the typical Cauchy (Robin) type of boundary condition (see equation 3.6). Thus, the electromagnetic problem may be reduced to a common FEM analysis.

The algebraic system of equations that characterize the interior finite domain $\Omega^{\mathtt{FEM}}$ can be obtained in terms of the electric ($\mathbf{E}$) or magnetic ($\mathbf{H}$) field through

$$\boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) - k_0^2 \, \bar{\bar{g}}_r \, \mathbf{V} = -j \, k_0 \, \eta_0 \, \mathbf{O} - \boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \mathbf{L} \right) \tag{3.1}$$

where $\mathbf{V}$ denotes the magnitude to be solve depending on the formulation employed, k_0 is the wavenumber in vacuum and the right hand side of the equation is the source term due to the presence of impressed electric and/or magnetic currents within $\Omega^{\mathtt{FEM}}$. Table 3.1 shows the different magnitudes involved in the $\mathbf{E}$ and $\mathbf{H}$ formulations. For simplicity, the source term is renamed to $\mathbf{q}$

$$\mathbf{q} = -j \, k_0 \, \eta_0 \, \mathbf{O} - \boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \mathbf{L} \right) \tag{3.2}$$

modifying the equation (3.1) to

$$\boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) - k_0^2 \, \bar{\bar{g}}_r \, \mathbf{V} = \mathbf{q} \tag{3.3}$$

Equation (3.3) provides the behavior of electromagnetic fields in all domains, but in particular, in the interior domain $\Omega^{\texttt{FEM}}$. In addition to the above differential equation, boundary conditions on the contour of the domain $\Omega^{\texttt{FEM}}$ are required to make the electromagnetic solution unique. The boundary conditions considered are the homogeneous Dirichlet condition over Perfect Electric Conductor (**E** formulation) and Perfect Magnetic Conductor (**H** formulation), the homogeneous Neumann condition over Perfect Magnetic Conductor (**E** formulation) and Perfect Electric Conductor (**H** formulation) and the Cauchy condition over the external boundary of the domain $\Omega^{\texttt{FEM}}$. The equations of these boundary conditions are given by

$$\hat{\mathbf{n}} \times \mathbf{V} = 0 \quad \text{over } \Gamma_{\texttt{D}} \quad \text{Dirichlet} \tag{3.4}$$

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f_r}}^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) = 0 \quad \text{over } \Gamma_{\texttt{N}} \quad \text{Neumann} \tag{3.5}$$

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f_r}}^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) + \gamma \, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} = \boldsymbol{\Psi} \quad \text{over } \Gamma_{\texttt{C}} \quad \text{Cauchy} \tag{3.6}$$

where $\Gamma_{\texttt{D}}$, $\Gamma_{\texttt{N}}$ and $\Gamma_{\texttt{C}}$ are the boundaries where the Dirichlet, Neumann and Cauchy conditions must be applied, respectively, and $\hat{\mathbf{n}}$ is the outward unit vector to the boundary surface.

Following the traditional methodology of `FEM`, the vectorial equation (3.3) must be converted to a variational equation. The variational expression consists of the application of a weighted-integral form instead of using the original formulation of the problem. In this case, the Galerkin Weighted Residual Method is used to obtain the mentioned weighted-integral form. Lets define the *residual* associated to equation (3.3) as

$$\mathbf{R} = \boldsymbol{\nabla} \times \left(\bar{\bar{f_r}}^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) - k_0^2 \, \bar{\bar{g_r}} \, \mathbf{V} - \mathbf{q} \tag{3.7}$$

and then, applying the Galerkin Method, the weighted-integral form is

$$\begin{aligned} R_W = \iiint_\Omega \mathbf{W} \cdot \mathbf{R} \, dV = \iiint_\Omega \mathbf{W} \cdot \boldsymbol{\nabla} \times \left(\bar{\bar{f_r}}^{-1} \boldsymbol{\nabla} \times \mathbf{V} \right) dV \\ - k_0^2 \iiint_\Omega \mathbf{W} \cdot \bar{\bar{g_r}} \, \mathbf{V} \, dV - \iiint_\Omega \mathbf{W} \cdot \mathbf{q} \, dV \end{aligned} \tag{3.8}$$

where the testing functions **W** belong to the vectorial space $\mathbf{H}(\texttt{curl})_0$, the basis functions **V** belong to the vectorial space $\mathbf{H}(\texttt{curl})$ and $dV = dx\, dy\, dz$ is the differential volume.

$$\mathbf{H}(\texttt{curl}) = \{\mathbf{V} \in L^2,\, \boldsymbol{\nabla} \times \mathbf{V} \in L^2\} \tag{3.9}$$

$$\mathbf{H}(\texttt{curl})_0 = \{\mathbf{W} \in \mathbf{H}(\texttt{curl}),\, \hat{\mathbf{n}} \times \mathbf{W} = 0\,\mathrm{on}\,\Gamma_{\texttt{D}}\} \tag{3.10}$$

Applying the following vectorial identities

$$\boldsymbol{\nabla} \cdot (\mathbf{A} \times \mathbf{B}) \;=\; \mathbf{B} \cdot \boldsymbol{\nabla} \times \mathbf{A} - \mathbf{A} \cdot \boldsymbol{\nabla} \times \mathbf{B} \tag{3.11}$$

$$(\mathbf{A} \times \mathbf{B}) \cdot \hat{\mathbf{n}} \;=\; -\mathbf{A} \cdot (\hat{\mathbf{n}} \times \mathbf{B}) \tag{3.12}$$

and the Gauss theorem

$$\iiint_{\Omega} \boldsymbol{\nabla} \cdot \mathbf{A}\, d\Omega = \underset{\Gamma}{\oiint} \mathbf{A} \cdot \hat{\mathbf{n}}\, d\Gamma \tag{3.13}$$

the double rotational term of equation (3.8) is given by

$$\begin{aligned} \iiint_{\Omega} \mathbf{W} \cdot \boldsymbol{\nabla} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) dV = \iiint_{\Omega} \boldsymbol{\nabla} \times \mathbf{W} \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) dV \\ + \underset{\Gamma}{\oiint} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \end{aligned} \tag{3.14}$$

where Γ is the boundary of the volume $\Omega^{\texttt{FEM}}$ and $\hat{\mathbf{n}}$ is the outward unit vector for each point of Γ.

Substituting equation (3.14) in (3.8), the *residual* has the given below terms

$$\begin{aligned} R_W = \iiint_{\Omega} (\boldsymbol{\nabla} \times \mathbf{W}) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) dV - k_0^2 \iiint_{\Omega} \mathbf{W} \cdot \bar{\bar{g}}_r \mathbf{V} dV \\ - \iiint_{\Omega} \mathbf{W} \cdot \mathbf{q}\, dV + \underset{\Gamma}{\oiint} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \end{aligned} \tag{3.15}$$

The boundary integral of equation (3.15) may be simplified applying the boundary conditions introduced in (3.4), (3.5) and (3.6). For simplicity, lets consider the boundary integral alone

$$\underset{\Gamma}{\oiint} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \tag{3.16}$$

The boundary of the problem is divided in three disjoint surfaces $\Gamma \equiv \Gamma_{\mathtt{D}} \cup \Gamma_{\mathtt{N}} \cup \Gamma_{\mathtt{C}}$ based on the boundary condition assigned to each one: homogeneous Dirichlet (3.4) in $\Gamma_{\mathtt{D}}$, homogeneous Neumann (3.5) in $\Gamma_{\mathtt{N}}$ and Cauchy (3.6) in $\Gamma_{\mathtt{C}}$. Thus, equation (3.16) may be rewritten as

$$\oiint_{\Gamma} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] d\Gamma = -\iint_{\Gamma_{\mathtt{D}}} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) dS$$
$$+\iint_{\Gamma_{\mathtt{N}}} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS + \iint_{\Gamma_{\mathtt{C}}} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \tag{3.17}$$

The imposition of the Dirichlet boundary condition over the integral (3.17) depends on the vectorial spaces used for testing function $\mathbf{W}$ and the basis functions $\mathbf{V}$. In this case, the term $\hat{\mathbf{n}} \times \mathbf{W}$ over the boundary surface $\Gamma_{\mathtt{D}}$ is directly zero due to the application of the condition given by (3.10) over $\Gamma_{\mathtt{D}}$. Then, the integral corresponding to the Dirichlet boundary is

$$\iint_{\Gamma_{\mathtt{D}}} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) dS = 0 \tag{3.18}$$

The homogeneous Neumann condition may be imposed directly over the boundary integral resulting also zero.

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right) = 0 \Rightarrow \iint_{\Gamma_{\mathtt{N}}} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS = 0 \tag{3.19}$$

As it occurs with the case of the homogeneous Neumann boundary condition, the Cauchy condition may be imposed directly over the boundary of $\Omega^{\mathtt{FEM}}$ obtaining

$$\iint_{\Gamma_{\mathtt{C}}} \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS = \iint_{\Gamma_{\mathtt{C}}} \mathbf{W} \cdot \boldsymbol{\Psi}\, dS$$
$$+ \gamma \iint_{\Gamma_{\mathtt{C}}} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \mathbf{V})\, dS \tag{3.20}$$

Substituting in equation (3.15), the equations (3.17, 3.18, 3.19, 3.20), the *residual* is defined in the whole FEM domain as function of the boundary conditions, the source of the problem and the energy terms

$$R_W = \iiint_\Omega \boldsymbol{\nabla} \times \mathbf{W} \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) dV - k_0^2 \iiint_\Omega \mathbf{W} \cdot \bar{\bar{g}}_r \, \mathbf{V} \, dV$$
$$- \iiint_\Omega \mathbf{W} \cdot \mathbf{q} \, dV + \iint_{\Gamma_\mathtt{C}} \mathbf{W} \cdot \boldsymbol{\Psi} \, dS + \gamma \iint_{\Gamma_\mathtt{C}} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \mathbf{V}) \, dS \tag{3.21}$$

The variational equation (3.21) is very common in FEM and also is used in multiple methods for the resolution of electromagnetic problems. The module makes use of this equation on each of the sub-domains in which the interior finite domain $\Omega^{\mathtt{FEM}}$ is divided.

The current version of the FEM module supports two different interior excitations, such as the rectangular waveguide port and the coaxial waveguide port and one exterior excitation, such as plane waves. The source term due to the presence of impressed electric and/or magnetic currents is supported in the formulation but, in the current version of the module, is not provided as interior excitation. Future developments of the module will provide impressed electric and/or magnetic currents as interior excitations. The following subsections describe how the rectangular waveguide port, the coaxial waveguide port and the exterior plane wave excitation have been implemented in the module.

3.1.1 Rectangular waveguide port

This waveguide port is a mono-mode boundary condition where the desired analytic field is placed (TE/TM modes). The residual of equation (3.6) applied on the rectangular waveguide for the mode $\mathtt{TE}_{10}$ is

$$\boldsymbol{\Psi}_{\mathtt{TE}_{10}} = -2\,\gamma\,(\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}_{\mathtt{TE}_{10}}) \tag{3.22}$$

where γ is the propagation constant of the mode $\mathtt{TE}_{10}$, $\hat{\mathbf{n}}$ is the outward unit vector to the boundary surfaces and $\mathbf{V}_{\mathtt{TE}_{10}}$ is the imposed $\mathtt{TE}_{10}$ mode depending on the formulation used. Then, particularizing the expressions of the $\mathtt{TE}_{10}$ mode for the electric field formulation, the value of the field $\mathbf{E}_{\mathtt{TE}_{10}}$ referred to the local coordinate system of the waveguide (see figure 3.1) is given by

$$\mathbf{E}_{\mathtt{TE}_{10}} = -\frac{1}{\sqrt{2}} \frac{\omega \mu_0 \pi}{k_c^2 \, a} \, \mathtt{sin} \left(\frac{\pi \xi}{a} \right) \hat{\eta} \tag{3.23}$$

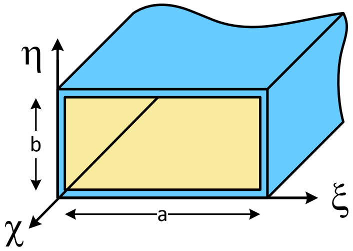


Figure 3.1: Local coordinate system of rectangular waveguide port

where μ_0 is the vacuum permeability, a is the width of the rectangular waveguide, k_c is the cut-off wavenumber of the waveguide and ξ is the coordinate of the local ξ-axis of the waveguide where the field is calculated. It is worth noting that the factor that multiplies the field $\mathbf{E}_{\mathtt{TE}_{10}}$ is used to normalize the input power of the port to one Watt.

3.1.2 Coaxial waveguide port

The waveguide port of the coaxial transmission line is a mono-mode boundary condition where the fundamental TEM mode is excited. The residual of equation (3.6) applied on this waveguide port is

$$\mathbf{\Psi}_{\mathtt{TEM}} = -2\,\gamma\;(\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}_{\mathtt{TEM}}) \tag{3.24}$$

where γ is the propagation constant of the TEM mode, $\hat{\mathbf{n}}$ is the outward unit vector to the boundary surfaces and $\mathbf{V}_{\mathtt{TEM}}$ is the imposed TEM field depending on the formulation used. For instance, for the electric field formulation, the imposed TEM field referred to the local coordinate system of the coaxial transmission line (see figure 3.2) is given by

$$\mathbf{E}_{\mathtt{TEM}} = \sqrt{\frac{\eta}{2\pi\; ln\,(b/a)}}\left(\frac{1}{r}\right)\hat{\mathbf{r}} \tag{3.25}$$

where η is the characteristic impedance of the coaxial port medium, a is the coaxial inner radius, b is the coaxial exterior radius, r is the radial coordinate and $\hat{\mathbf{r}}$ is the unitary vector of r. As it occurs with the rectangular waveguide port, this waveport is also normalized to one Watt of input power.

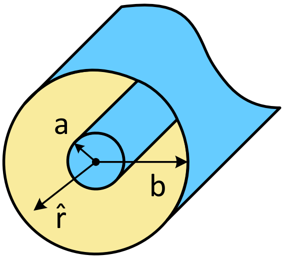


Figure 3.2: Local coordinate system of the coaxial waveguide port

3.1.3 Exterior plane wave excitation

Other way to excite the electromagnetic problem is using exterior incident sources as plane waves, cylindrical waves or spherical waves. The current version of the module only supports plane waves as exterior excitations, but other types of wave excitations may be easily incorporated. The residual of equation (3.6) applied on this excitation is

$$\mathbf{\Psi}_{\mathtt{PLW}} = -2\,\gamma\,(\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}_{\mathtt{PLW}}) \tag{3.26}$$

where γ is the propagation constant of the exterior medium of the problem, $\hat{\mathbf{n}}$ is the outward unit vector to the boundary surfaces and $\mathbf{V}_{\mathtt{PLW}}$ is the incident field depending on the formulation used. For instance, the incident field for the electric field formulation is the electric field of the desired plane wave

$$\mathbf{E}_{\mathtt{PLW}} = \mathbf{E_0}\, e^{-j\mathbf{k}\cdot\mathbf{r}} \tag{3.27}$$

where E_0 is the polarization vector, $\mathbf{k}$ is the propagation vector of the wave and $\mathbf{r}$ is the position vector where the incident field is calculated.

3.2 Iterative Integral Equation Evaluation Method

The Iterative Integral Equation Evaluation Method (`FE-IIEE`) is a mesh truncation technique where the original infinite domain is divided in two overlapping domains: a finite `FEM` domain ($\Omega^{\mathtt{FEM}}$) bounded by the surface S and the infinite domain exterior to the auxiliary boundary $S^{'}$ [**?**].

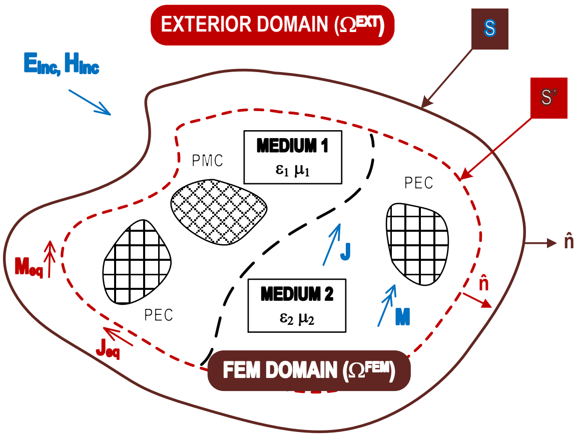


Figure 3.3: Typical setup of a open single-region problem

Thus, the overlapping region is limited by $S^{'}$ and S. The method makes use of a integral equation representation of the field exterior to $S^{'}$ obtaining the solution through an iterative process in which the residual of the radiation boundary condition on the mesh truncation boundary is updated.

This truncation method is a very powerful technique since the distance from the auxiliary boundary $S^{'}$ to the boundary surface S is usually small, typically in the range of 0.05 λ to 0.2 λ. Thus, the FEM domain can be truncated very close to the source of the problem reducing the number of unknowns of the analysis while the sparsity of the FEM matrices is retained.

As it was commented above, the solution is obtaining through an iterative process in which the residual of the radiation boundary condition on the mesh truncation boundary is updated. Thus, the convergence of the method needs to be considered in order to obtain an accurate solution. Convergence studies have been done during the last years, for details see [53], concluding that conformal shapes, as convex boundaries, have a good convergence rates even, when the distance between $S^{'}$ and S is very small (≤ 0.01 λ). However, a clear pattern about how the method diverges or converges depending on the size of the overlapping region cannot be obtained for the case of non-convex truncation boundaries.

It is worth to note that electromagnetic problems involving multiple structures, such as, scatterer objects or antennas, can be solved by this method using a single domain truncation, but often it is more efficient to separate the single domain into several subdomains, separated by free-space, and to solve the set of subdomains iteratively. Thus, a single-region or a multi-region `FE-IIEE` method must be considered.

3.2.1 Single-region FE-IIEE

Figure 3.3 depicts a typical setup problem of a single-region truncation. The method starts computing the sparse system of equations that variational equation (3.21) gives on each of the sub-domains in which the interior finite domain $\Omega^{\mathtt{FEM}}$ is divided. The boundary condition used over the surface S is the following Cauchy (Robin) boundary condition

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) + j\,k\,\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} = \boldsymbol{\Psi} \ \text{ over } \Gamma_{\mathtt{S}} \tag{3.28}$$

where $\Gamma_{\mathtt{S}}$ stands for the part of the boundary domain on S, $\hat{\mathbf{n}}$ is the outward unit vector to S and k is the wavenumber of the medium exterior to $S^{'}$, assumed to be homogeneous. This system of equations may be expressed in partitioned form as follows:

$$\begin{bmatrix} K_{II} & K_{IS} \\ K_{SI} & K_{SS} \end{bmatrix} \begin{bmatrix} g_I \\ g_S \end{bmatrix} = \begin{bmatrix} b_I \\ b_\Psi \end{bmatrix} \tag{3.29}$$

where the sub-indexes S and I refer to the degrees of freedom g associated to S and those associated to nodes in the interior of S, respectively. Thus, the right hand side term $\{b_I\}$ corresponds to the interior current sources $\mathbf{J}$ and $\mathbf{M}$ and the inward waves impressed at the ports. The term $\{b_\Psi\}$ is related to the residual $\boldsymbol{\Psi}$ of the boundary condition at the truncating boundary S. Once the system of equation is obtained the algorithm to solve the problem is as follows:

- **Step 1:** An initial value of $\boldsymbol{\Psi}$, denoted as $\boldsymbol{\Psi}^0$, is assumed. Specifically, $\boldsymbol{\Psi}^0$ is zero for radiation problems and $\boldsymbol{\Psi}^0 = \boldsymbol{\Psi}^{\mathtt{inc}}$ for scattering problems where $\boldsymbol{\Psi}^{\mathtt{inc}}$ is the value of equation (3.28) when the magnitude $\mathbf{V}$ is replaced by the corresponding incident field. Then, the right had side term b_Ψ^0 is computed.

- **Step 2:** The FEM system of equations (3.29) is solved. After that, fields on $S^{'}$ are calculated in order to compute the electric and magnetic current densities $\mathbf{J}_{\texttt{eq}}$ and $\mathbf{M}_{\texttt{eq}}$ of the equivalent exterior problem.

- **Step 3:** The field, and its curl, over S radiated by the equivalent currents $\mathbf{J}_{\texttt{eq}}$ and $\mathbf{M}_{\texttt{eq}}$ are calculated. The fields radiated by the FEM region, $\mathbf{V}_{\texttt{FE-IIEE}}$ and its curl $(\boldsymbol{\nabla}\times\mathbf{V})_{\texttt{FE-IIEE}}$, are computed using the integral expressions

$$\begin{aligned}\mathbf{V}_{\texttt{FE-IIEE}} = & \oiint_{S'} \left(\mathbf{L}_{\texttt{eq}} \times \boldsymbol{\nabla} G\right)\, dS^{'} \\ & - jkh \oiint_{S'} \left[\mathbf{O}_{\texttt{eq}} \left(G + \frac{1}{k^2}\,\boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G\right)\right] dS^{'} \qquad (3.30)\end{aligned}$$

$$\begin{aligned}(\boldsymbol{\nabla}\times\mathbf{V})_{\texttt{FE-IIEE}} = & \, jkh \oiint_{S'} \left(\mathbf{O}_{\texttt{eq}} \times \boldsymbol{\nabla} G\right)\, dS^{'} \\ & - \oiint_{S'} \left[\mathbf{L}_{\texttt{eq}} \left(k^2\, G + \boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G\right)\right] dS^{'} \qquad (3.31)\end{aligned}$$

where h is the inmitance of the homogeneous medium (see table 3.1), and G denotes the Green's function for a homogeneous medium which typically is the free space

$$G \equiv G\left(r, r'\right) = \frac{e^{jk(r-r')}}{4\pi|r-r'|} \qquad (3.32)$$

It is worth noting that the methodology is also valid for non homogeneous exterior regions (as those with infinite metallic planes, layered media, and so on) by using the corresponding integral expression representation of the field and the Green's functions of the exterior region.

- **Step 4:** A new value of $\boldsymbol{\Psi}$, ($\boldsymbol{\Psi}^{i+1}$ in general) is computed by introducing the values of the fields $\mathbf{V}_{\texttt{FE-IIEE}}$ and $(\boldsymbol{\nabla}\times\mathbf{V})_{\texttt{FE-IIEE}}$ in equation 3.28 where i means the iteration number.

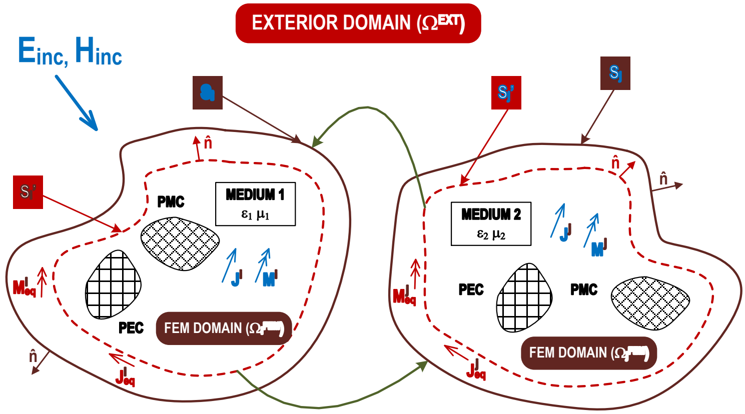


Figure 3.4: Typical setup of a open multi-region problem.

- **Step 5:** The error between $\mathbf{\Psi}^{i+1}$ and $\mathbf{\Psi}^{i}$ is calculated. If the error is greater than an error threshold, the method will start again for step 2 (using $\mathbf{\Psi}^{i+1}$ as the new residual function); otherwise the iteration process finishes. The error in $\mathbf{\Psi}$ is measured in a weighted L_2-norm

$$\text{error}_i = \frac{\left\| \mathbf{\Psi}^i - \mathbf{\Psi}^{i-1} \right\|_2}{\left\| \mathbf{\Psi}^i \right\|_2} \tag{3.33}$$

Thus, as it was mentioned above a (numerically) exact radiation boundary condition is imposed (allowing the external boundary to be placed close to the sources) while the sparsity of the FEM matrices is retained. It is worth noting that the numerical cost of the second and subsequent iterations is very small since the factorization of the FEM matrix must be performed only once at the first iteration (if direct solvers are used).

3.2.2 Multi-region FE-IIEE

Figure 3.4 depicts a typical setup problem of a multi-region truncation. In this case, each region is bounded by its own auxiliary surface $S_i^{'}$ and the exterior boundary S_i. As the single-region case, the method starts computing the sparse system of equations given by the variational equation (3.21) on each of the sub-domains in which the interior finite domain Ω^{FEM} is divided.

The main difference between both methods occurs in the third step of the algorithm where the field, and its curl, radiated by the equivalent currents over the exterior boundary S_i are calculated. The equivalent currents of each auxiliary surface $S_i^{'}$ are used to calculate the fields radiated by the `FEM` region using the integral expressions

$$\mathbf{V}_{\texttt{FE-IIEE}} = \sum_{i=0}^{N} \oiint_{S_i^{'}} \left(\mathbf{L}^i_{\mathsf{eq}} \times \boldsymbol{\nabla}\, G\right)\, dS_i^{'} - jkh \oiint_{S_i^{'}} \left[\mathbf{O}^i_{\mathsf{eq}} \left(G + \frac{1}{k^2}\, \boldsymbol{\nabla}\, \boldsymbol{\nabla}\, G\right)\right] dS_i^{'} \quad (3.34)$$

$$(\boldsymbol{\nabla} \times \mathbf{V})_{\texttt{FE-IIEE}} = \sum_{i=0}^{N} jkh \oiint_{S_i^{'}} \left(\mathbf{O}^i_{\mathsf{eq}} \times \boldsymbol{\nabla}\, G\right)\, dS_i^{'} - \oiint_{S_i^{'}} \left[\mathbf{L}^i_{\mathsf{eq}} \left(k^2\, G + \boldsymbol{\nabla}\, \boldsymbol{\nabla}\, G\right)\right] dS_i^{'} \quad (3.35)$$

where N is the total number of regions, h is the inmitance of the homogeneous medium as the single region case and G denotes the Green's function for a homogeneous medium (see equation 3.32).

Once the fields radiated by the `FEM` region are calculated, a new value of $\boldsymbol{\Psi}$ is computed and the error between the previous and the current iteration is obtained. If the error is greater than an error threshold, the method will start again for step 2 (as single region); otherwise the iteration process finishes.

3.3 Basis Functions

In order to approximate the field in the elements of the mesh, a set of interpolation polynomials are chosen. Specifically, vector finite elements have shown to be the appropriate choice for the approximation of electromagnetic field quantities. Two families of vector elements may be distinguished: div-conforming and curl-conforming elements, providing continuity across element interfaces in the sense of the divergence

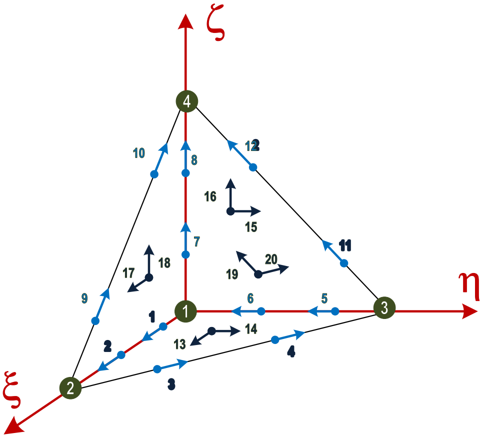


Figure 3.5: Second-order Nédélec tetrahedron

operator (i.e., normal continuity) and curl operator (i.e., tangential continuity), respectively. Thus, div-conforming elements are suitable for the discretization of the electric and magnetic inductions, **D** and **B**, whereas curl-conforming elements are appropriate to approximate the electric and magnetic fields, **E** and **H**.

The curl-conforming elements were proposed by Nédélec in 1980, [43], extending the divergence-conforming elements of Raviart and Thomas defined to solve second-order elliptic problems in `2D` [54]. It is worth to note that different types of curl-conforming elements have appeared since Nédélec's 1980 paper. It may distinguished between mixed order element and polynomial complete elements. Polynomial complete elements have the same order of approximation along any direction, however, mixed order elements provide a constrained representation of the vector field leading to a different order of polynomial approximation along one direction than along the others.

Among the curl-conforming elements appeared in the literature, the `FEM` module of this suite makes use of the Nédélec curl-conforming elements proposed in [7, 41]. Specifically, the finite domain is discretized into tetrahedral curl-conforming elements of second-order. Figure 3.5 shows the representation of the space of the function for the second-order Nédélec tetrahedron.

The space of functions for the second-order element is denoted as $\mathcal{R}^2$, which is the space of vector polynomials of order 2 that satisfy certain constraints (i.e., the so-called Nédélec constraints). Equation (3.36) shows this space of functions with 20 independent coefficients that have to be determined. Thus, the number of degrees of freedom of the tetrahedral element of second-order is 20, and 20 is also the number of vector basis function belonging to $\mathcal{R}^2$ that should be chosen $\mathbf{N}_j, (j = 1...20)$.

$$\mathcal{R}^2 \equiv \left\{ \begin{array}{c} a_1 + a_2x + a_3y + a_4z + Dy^2 - Fxy - Gxz + Hz^2 + Jyz \\ b_1 + b_2x + b_3y + b_4z - Dxy - Eyz + Fx^2 + Iz^2 - Jxz + Kxz \\ c_1 + c_2x + c_3y + c_4z + y^2 + Gx^2 - Hxz - Iyz - Kxy \end{array} \right\} \tag{3.36}$$

The vector basis functions $\mathbf{N}_j$ are obtained from the definition of the degrees of freedom (dof) of a finite element. The dof definition is a fundamental issue in order to understand how boundary conditions should be imposed. Specifically, the vector basis functions are obtained by imposing the interpolatory character of the basis functions with respect to the definition of the degrees of freedom

$$g_i\left(\mathbf{N}_j\right) = \delta_{ij}, \quad i = 1...20 \tag{3.37}$$

where g_i stands for the i-th functional defining the i-th dof of the element. Note that equation (3.37) represents a system of equations for each $\mathbf{N}_j$ basis function where the unknowns are the coefficients mentioned above. Once the coefficients are obtained for $j = 1...20$ the basis functions are completely determined. The definition of the dof functionals of the Nédélec tetrahedral curl-confor-ming element of second-order is as follows: 12 *dofs* associated to the six edges of the tetrahedron given by

$$\int_{edge} (\mathbf{u} \cdot \hat{\tau})\, q \, dl \quad \forall q \in P_2 \tag{3.38}$$

and 8 dof associated to the four faces of the tetrahedron given by

$$\int_{face} (\hat{\mathbf{n}} \times \mathbf{u}) \cdot \mathbf{q} \, ds \quad \forall q \in (P_1)^2 \tag{3.39}$$

where $\mathbf{u}$ stands for the vector unknown, $(P_k)^n$ for the space of polynomials of order k in n-dimensions, $\hat{\tau}$ for the unit vector along the direction of the edge and $\hat{\mathbf{n}}$ for the unit vector normal to the face.

It is worth noting that the previous definitions of degrees of freedom are not ready to be used in a finite element code. They must be discretized. The discretization is performed by means of the choice of a basis for each one of the polynomial spaces, P_2, $(P_1)^2$ appearing in equations (3.38) and (3.39). In this particular case, the module makes use of Lagrange basis for the polynomial spaces P_2 and $(P_1)^2$. Thus, the practical `FEM` *dof* definition is made in terms of *momentums* (or different order) over the adequate components of the unknown $\mathbf{u}$.

Specifically, the components involved in the definition of the degrees of freedom associated to the boundary of the element are the components tangential to that boundary. Thus, the tangential continuity between elements may be easily imposed in the `FEM` assembly procedure. However, care must be taken with the local definitions of the vector quantities involved in the *dof* definition, specially when the basis functions are obtained in the parent element. In the case of the present `FEM` module, the basis functions are obtained in the parent element and they are transformed to the real element using the inverse of the jacobian matrix.

3.4 Implementation Details

The system of equations given by `FEM` provides very sparse matrices that have to be solve using advanced sparse solvers. There are many types of these solvers in both academic and commercial sectors with very similar capabilities and performance. The implementation of the module allows the use of four of these sparse solver, such as `MUMPS` [50], `HSL` [51], `UHM` and `PARDISO` [52].

The module also provides a parallel implementation where one can utilize all the `CPU` cores available on single computers or on several compute nodes of a high performance computing (`HPC`) cluster. The module has been developed to run efficiently on any type of computer ranging from small laptops to `HPC` clusters. The parallel implementation is achieved by using `MPI` (Message Passing Interface) [39] and multi-thread execution within each `MPI` process by calling appropriate math kernel libraries. Figure 3.6 shows the block diagram of the implementation of the module.

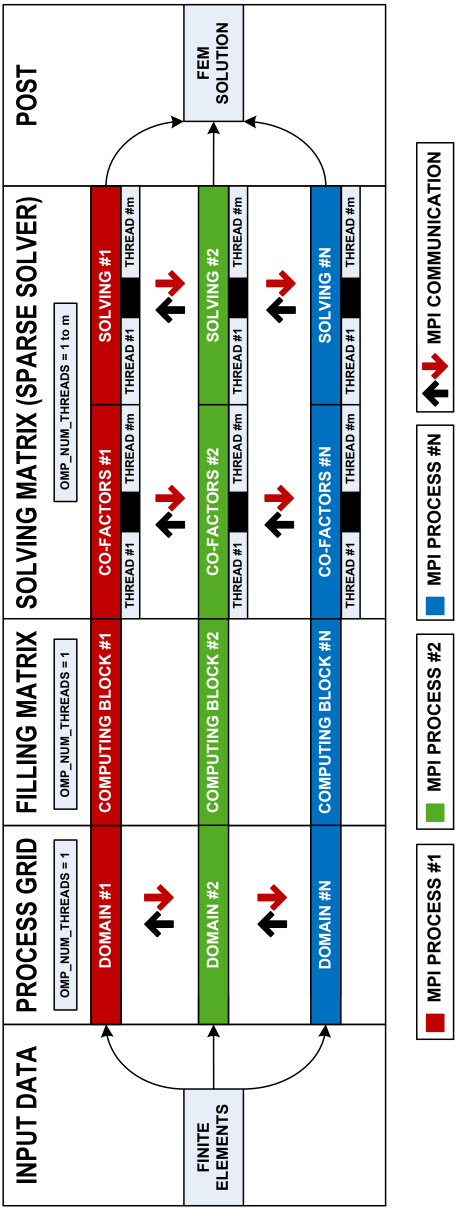


Figure 3.6: Block diagrams of the implementation of the module

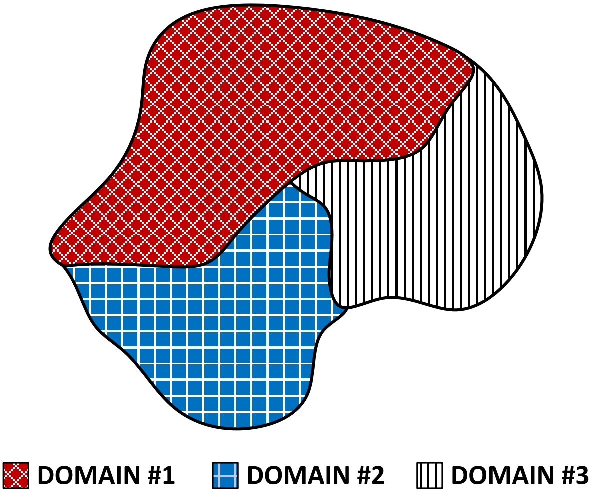


Figure 3.7: Arbitrary distribution of the finite elements for a parallel execution with 3 `MPI` processes

A logical `MPI` process grid is generated which is used to distribute the finite elements of the problem into domains. Thus, every `MPI` process computes its portion of the `FEM` matrix concurrently using a distributed memory system (see the filling matrix step in figure 3.6). An arbitrary distribution of the finite elements supposing a `MPI` grid of 3 processes is shown in figure 3.7. The elements of each domain (distributed by colors) are processed by the corresponding `MPI` process.

Once the computation of the `FEM` matrix is finished, the sparse solver factorizes the matrix using multi-frontal methods (note that the sparse solver also computes concurrently on different cores as it is shown in figure 3.6). At the end of this task, the solver is ready to solve the system of equations using back substitution methods. Finally, the solution is sent to the main process using `MPI` communication and given to the user.

It is worth noting that while the sparse solver is factorizing the FEM matrix, `MPI` communication is required between the processes involved (see the black/red arrows in figure 3.6). Also, `MPI` communication is needed to solve the system of equations, once the matrix has been factorized.

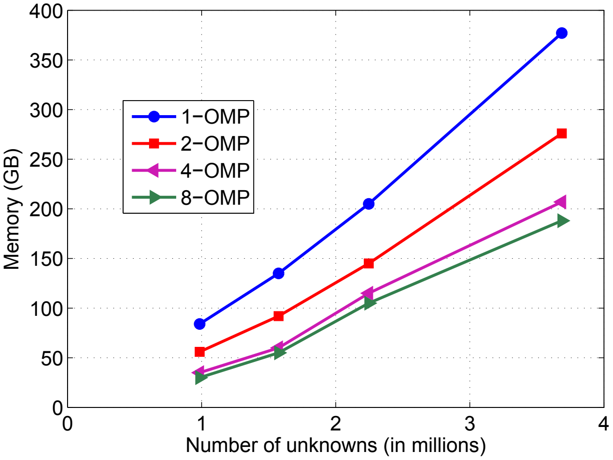


Figure 3.8: Memory used during the factorization and solving phases with MUMPS as sparse solver for different problem sizes

A multi-thread execution within each MPI process may be achieved by using appropriate math kernel libraries as BLACS and level 3 BLAS routines [?]. Sparse solvers make use of these types of math kernel libraries to improve the efficiency of multi-frontal methods. The number of OMP threads per MPI process is selected by a system variable called OMP_NUM_THREADS that can be easily modified by the users.

Figures 3.8 − 3.9 show the memory and the computational time used for the sparse solver during the factorization and solving phases when the number of unknowns of the problem is increased. Different values of the variable OMP_NUM_THREADS have been employed in order to study the efficiency and versatility of the multi-thread execution in the module. The sparse solver used in both cases is MUMPS and the number of cores employed in each simulation is 56. Figure 3.8 shows great differences in terms of memory used for the same number of unknowns. In the case of 1-OMP thread per MPI process the memory used is almost double than the case of 4/8-OMP threads. Similar behavior may be appreciated in figure 3.9 regarding the computational elapsed time during the factorization and solving phases. In this case, the differences between a simulation using 4-OMP threads and the others are even more pronounced than for the memory case.

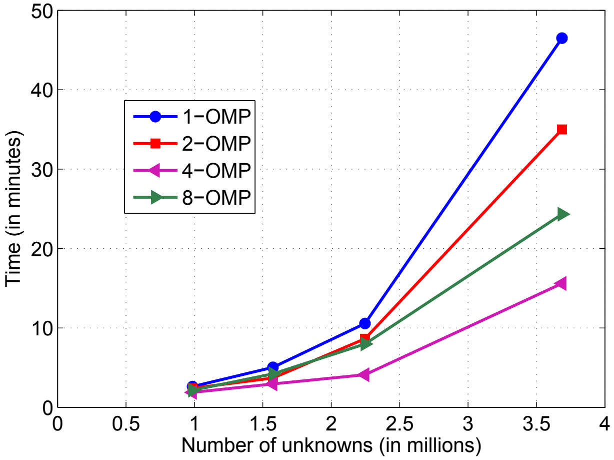


Figure 3.9: Computational time expended during the factorization and solving phases with `MUMPS` as sparse solver for different problem sizes

This study demonstrates that a multi-thread execution with a configuration of 4-`OMP` threads per `MPI` process is much more efficient in terms of memory and computational elapsed time than a single-thread execution. Hence, an important future work line in the development of the suite will be the support of this type of multi-thread execution in the whole module and not only in the sparse solver part. Figure 3.10 shows the block diagram of a future implementation of the module.

Figures 3.11 − 3.12 show the speedup of the module for the factorization and solving phases and for an iteration of the `FE-IIEE` method, respectively. The speedup on the ordinate axis represents the ratio between the execution time of a benchmark using one process (sequential execution) and the execution time of the same benchmark using n processes. A linear speedup is representative of a good scalability of the code. The ideal situation is when the slope of the linear speedup is one, meaning that the execution time of the parallel version is n times lower than the sequential execution time. In figure 3.11, it is observed that the slopes of the speedup curves is not one, but is around 0.50 for both in-core and out-of-core versions. Typically, the parallel performance of a sparse solver is between 55-60%, so an admissible speedup is obtained.

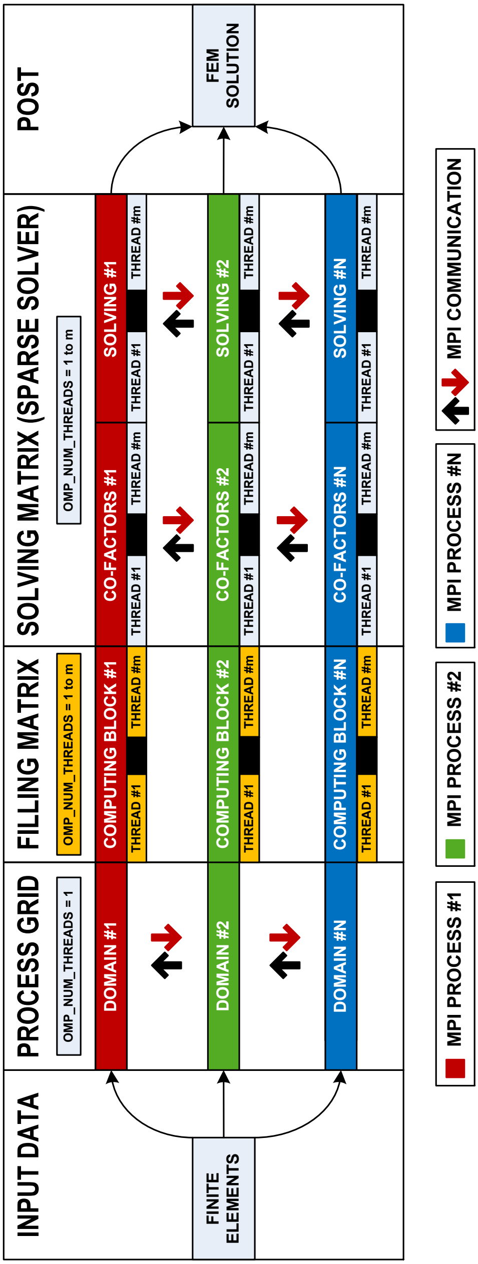


Figure 3.10: Block diagrams of a future implementation of the module

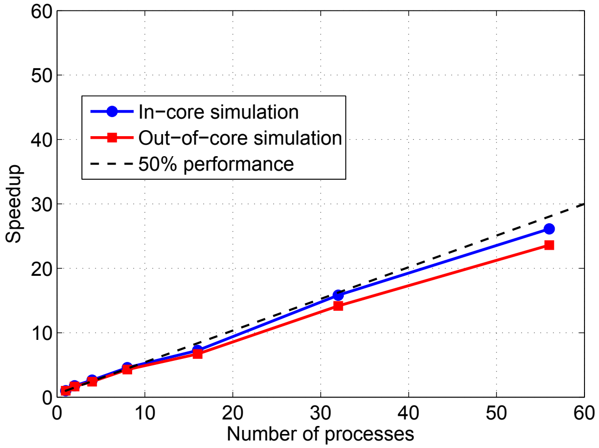


Figure 3.11: Speedup vs number of processes during the factorization and solving phases using `MUMPS` as sparse solver

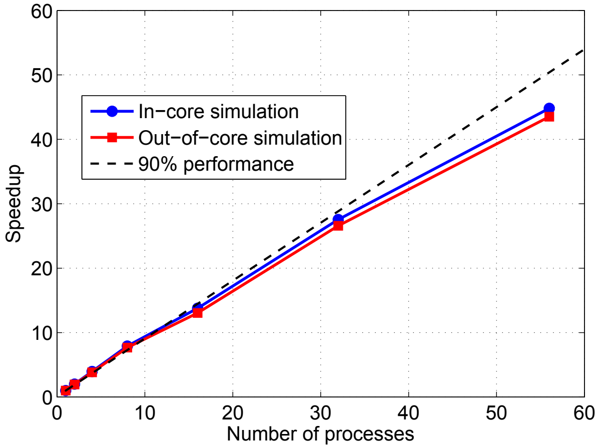


Figure 3.12: Speedup vs number of processes for the `FE-IIEE` method

Figure 3.12 shows how the parallel performance of the `FE-IIEE` method code for both versions is around 90%, which is near to the ideal parallel performance in practice. The parallelization of the `FE-IIEE` method is very straightforward, since just the splitting of two loops between the `MPI` processes of the grid is required. Furthermore, no communication between those `MPI` processes is needed during the computation of the method with the consequent advantages in the parallel performance.

3.5 Module Verification

The accuracy of the results given by a computer code must be verified before it can be used to provide solutions for real world problems. Nowadays there are many techniques that engineers use to verify their codes such as comparison methods in which one code is compared to an established code or set of codes that solve similar problems (typically used for validation tests), the Method of Exact Solutions (MES) or the Method of Manufactured Solutions (MMS).

In this particular case, the FEM module was tested using the Method of Manufactured Solutions. The basic idea of this technique is to manufacture an exact solution to some equation by solving the problem backwards. Suppose one is solving a differential equation of the form

$$D\,u = f \tag{3.40}$$

where D is the differential operator, u is the solution and f is a source term. The idea is to manufacture a solution u and then applies the equation to find the source term f. The manufactured solution should be a solution to the fully set of interior equations in order to test as much code as possible.

In order to measure the error given by the module, different relative and maximum errors have been defined, such as, the relative field error $\xi_{\mathtt{REL}}$, the maximum field error $\xi_{\mathtt{MAX}}$ and its curl components, $(\boldsymbol{\nabla} \times \xi)_{\mathtt{REL}}$ and $(\boldsymbol{\nabla} \times \xi)_{\mathtt{MAX}}$. The following expressions show how these magnitudes have been defined

$$\xi_{\mathtt{REL}} = \frac{\left\|\mathbf{V}_{\mathtt{FEM}} - \mathbf{V}_{\mathtt{analytic}}\right\|_2}{\left\|\mathbf{V}_{\mathtt{analytic}}\right\|_2} \tag{3.41}$$

$$\xi_{\mathtt{MAX}} = \frac{\left\|\mathbf{V}_{\mathtt{FEM}} - \mathbf{V}_{\mathtt{analytic}}\right\|_\infty}{\left\|\mathbf{V}_{\mathtt{analytic}}\right\|_\infty} \tag{3.42}$$

$$(\boldsymbol{\nabla} \times \xi)_{\mathtt{REL}} = \frac{\left\|(\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{FEM}}) - (\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{analytic}})\right\|_2}{\left\|(\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{analytic}})\right\|_2} \tag{3.43}$$

$$(\boldsymbol{\nabla} \times \xi)_{\mathtt{MAX}} = \frac{\left\|(\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{FEM}}) - (\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{analytic}})\right\|_\infty}{\left\|(\boldsymbol{\nabla} \times \mathbf{V}_{\mathtt{analytic}})\right\|_\infty} \tag{3.44}$$

Then, lets apply the manufactured solution technique to solve equation (3.1) and measure the error given by the module. Two types of functions are used as manufactured solution, such as polynomial functions belonging to the vectorial space of the basis functions and non-polynomial functions as complex exponentials. The main reason to use polynomials function belonging to the vectorial space of the basis functions is because the relative/maximum errors previously defined should be numerically zero for meshes formed only by straight tetrahedral elements. In other words, the module should be able to represent the polynomial proposed as manufactured solution with an error near to the machine precision. That condition can be used to provide evidence that the code is correct and correctly implemented. For other types of functions or meshed with the presence of curved elements, the module should be able to represent the manufactured solution with an error that decays as the discretization of the FEM domain contains a lower geometrical error.

The following subsections show the results of applying the MMS for a polynomial function belonging to the vectorial space of the basis functions and a complex exponential in a cube, a cylinder and a sphere.

3.5.1 Polynomial solution

These verification tests make use of polynomials belonging to the vectorial space of the basis function as manufactured solution. The full set of polynomials valid for this verification tests is 20 and any of the present in the vectorial space shown in equation (3.36) can be used.

Although the code was tested using the full set of polynomials, only the results of one of them are shown in order to reduce the extension of this document. The behavior of the code in all the tests was very similar obtaining the same results in term of the relative ($\xi_{\texttt{REL}}$) and maximum error ($\xi_{\texttt{MAX}}$) and, its curl components, $(\boldsymbol{\nabla} \times \xi)_{\texttt{REL}}$ and $(\boldsymbol{\nabla} \times \xi)_{\texttt{MAX}}$. The choice to illustrate the verification process using the Manufactured Solution Method is

$$\mathbf{V}_{\texttt{analytic}} = \left\{ \begin{array}{c} y^2 \\ -xy \\ 0 \end{array} \right\} \tag{3.45}$$

As commented above, the error given by the code for these type of manufactured solutions should be numerically zero for meshes formed only by straight tetrahedrons. Thus, the expected result for a cube, where all the elements are straight tetrahedrons, is an error numerically zero.

Results for several mesh discretization of a cube

Table 3.2 gathers the relative/maximum errors for different mesh discretization of a cube. Both relative and maximum errors follow the pattern described above and the error is numerically zero in all the cases. It is worth nothing that the error increases when the structure presents a finer mesh discretization (it means a higher number of tetrahedrons in the table). This behavior is completely expected and it is produced due to the numerical noise accumulated when the computer performs the numerical operations. If the mesh contains a higher number of tetrahedrons, the computer needs to perform a higher number of operations and the numerical noise accumulated is higher. In this case, as the errors is numerically zero, this behavior is more appreciable.

Figure 3.13(a) shows the magnitude of the analytic solution $\|\mathbf{V}_{\texttt{analytic}}\|$ for the finest mesh discretization of the cube. Figure 3.13(b) shows the solution given by the FEM module for the same component $\|\mathbf{V}_{\texttt{FEM}}\|$. The error between both solutions is shown in Fig. 3.13(c) where an uniform and very low error may be appreciated.

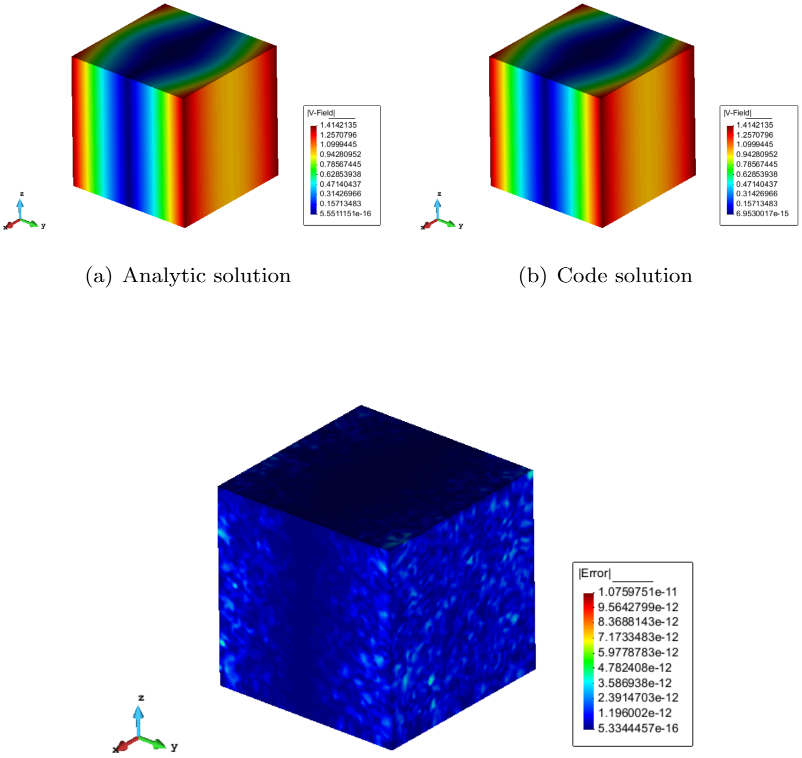
Results for several mesh discretization of a cylinder

In this case, as the cylinder has curved elements around its boundary, the errors are not numerically zero, since the FEM basis functions for curved elements are no longer polynomials. Thus, the errors given by the FEM module for this case should decay as a better geometrical approximation is achieved, i.e., when the size of the mesh discretization is smaller.

Table 3.3 gathers the relative/maximum errors for different mesh discretization of a cylinder. Both relative and maximum errors describe the behavior commented above giving a first evidence that the FEM module is correctly implemented also for curved elements. Figure 3.14(a) shows a 3D representation of the magnitude of the analytic solution $\|\mathbf{V}_{\texttt{analytic}}\|$ for the smallest mesh discretization of the cylinder. Figure 3.14(b) shows the solution given by the FEM code for the same component $\|\mathbf{V}_{\texttt{FEM}}\|$.

Table 3.2: Relative/maximum errors for several mesh sizes of a cube using a polynomial as manufactured solution

Tetrahedrons	Unknowns	$\xi_{\mathtt{REL}}$	$\xi_{\mathtt{MAX}}$	$(\nabla \times \xi)_{\mathtt{REL}}$	$(\nabla \times \xi)_{\mathtt{MAX}}$
28	280	8.787e^{-15}	8.328e^{-15}	3.708e^{-15}	8.988e^{-15}
2283	15706	1.086e^{-13}	2.623e^{-13}	6.866e^{-15}	4.011e^{-14}
7237	48450	2.492e^{-13}	2.853e^{-12}	9.577e^{-15}	5.257e^{-14}
16150	107084	4.322e^{-13}	3.461e^{-12}	1.241e^{-14}	9.483e^{-14}
79747	518772	1.330e^{-12}	1.015e^{-11}	2.105e^{-14}	2.575e^{-13}
109733	712062	1.655e^{-12}	1.075e^{-11}	2.340e^{-14}	3.029e^{-13}



(a) Analytic solution

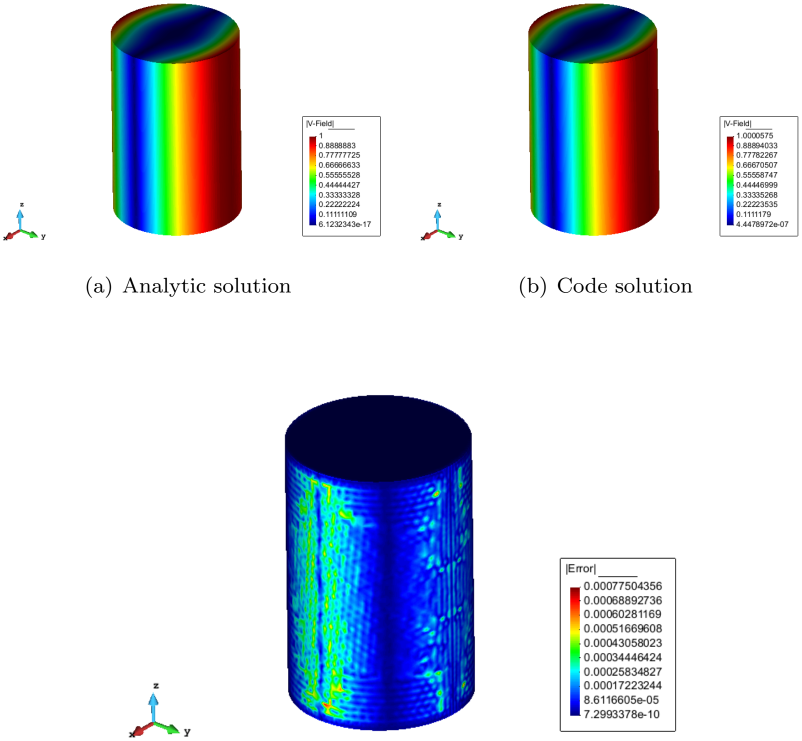
(b) Code solution

(c) Error between analytic and code solution

Figure 3.13: `3D` representation of the polynomial solution over a cube

Table 3.3: Relative/maximum errors for several mesh sizes of a cylinder using a polynomial as manufactured solution

Tetrahedrons	Unknowns	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
162	1316	$1.556e^{-2}$	$3.473e^{-2}$	$2.220e^{-2}$	$7.028e^{-2}$
4321	29370	$1.897e^{-4}$	$1.549e^{-3}$	$9.689e^{-4}$	$2.174e^{-2}$
11820	78718	$5.327e^{-5}$	$1.007e^{-3}$	$3.673e^{-4}$	$2.385e^{-2}$
36191	238554	$7.509e^{-5}$	$3.169e^{-3}$	$1.271e^{-4}$	$4.986e^{-3}$
73014	477110	$3.427e^{-5}$	$1.207e^{-3}$	$5.890e^{-5}$	$2.467e^{-3}$
112707	733462	$2.369e^{-5}$	$7.750e^{-4}$	$3.911e^{-5}$	$2.247e^{-3}$



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 3.14: 3D representation of the polynomial solution over a cylinder

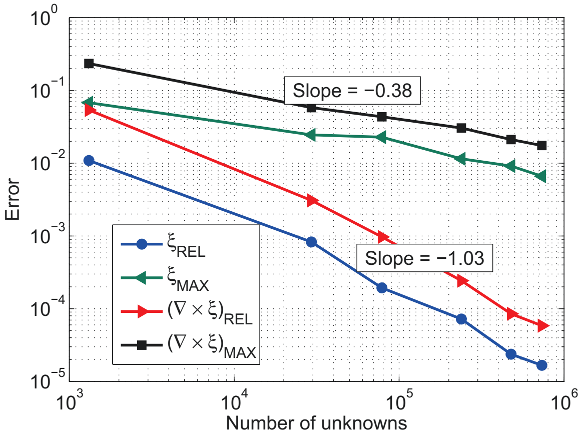


Figure 3.15: Convergence rate of the error over a sphere using a polynomial function as manufactured solution

The error between both solutions is shown in Figure 3.13(c), but unlike the error over the cube, the error in the cylinder is not uniform and it is localized over the curved boundary of the structure. It is worth nothing that the error is not evenly distributed along the curved boundary, but it is focused on some parts. The main reason of this behavior is due to the existence of a non-homogeneous mesh along the curved boundary and the error is located where the size of the discretization is larger.

Results for several mesh discretization of a sphere

As it occurs with the cylinder, the errors over the sphere are localized along the curved boundary of the structure. In this case, the basis functions are not able to approximate a polynomial solution without errors either, so, as explained above, a mesh discretization with a lower geometrical error will provide a lower error. Table 3.4 gathers the relative/maximum errors for six different mesh discretization of a sphere.

Both relative and maximum errors describe the behavior commented above confirming that the `FEM` module gives correct results for curved elements. Furthermore, the convergence rate of the error when the number of unknowns of the problem increase, is shown in figure 3.15. The relative error for both components decays linearly with an approximate slope of -1.03, while the maximum error decays approximately with a slope of -0.38.

The convergence rate of the error is typically given by the expression $\alpha\,\mathtt{N}^{\mathtt{slope}}$, where α is a constant and N is the total number of unknowns. Thus, the errors provided by the code in this case can be calculated using the following expressions

$$\begin{aligned} \xi_{\mathtt{REL}} \leq&\ \alpha_{\mathtt{REL}}\,\mathtt{N}^{-1.03} & \xi_{\mathtt{MAX}} \leq&\ \alpha_{\mathtt{MAX}}\,\mathtt{N}^{-0.38} \\ (\nabla \times \xi)_{\mathtt{REL}} \leq&\ \beta_{\mathtt{REL}}\,\mathtt{N}^{-1.03} & (\nabla \times \xi)_{\mathtt{MAX}} \leq&\ \beta_{\mathtt{MAX}}\,\mathtt{N}^{-0.38} \end{aligned} \tag{3.46}$$

Figure 3.16(a) shows a 3D plot of the magnitude of the analytic solution $|\mathbf{V}_{\text{analytic}}|$ for the finer mesh discretization of the sphere. Figure 3.16(b) shows the solution given by the code for the same component $|\mathbf{V}_{\text{FEM}}|$. The error between both solutions is shown in figure 3.16(c) where a concentration of error in a small part of the sphere can be appreciated. This type of error is due to the bad shape of the finite element located in that part of the sphere. Considering the rest of the sphere, the behavior of the error is correct and a mesh discretization with a lower geometrical error provided a lower relative error.

3.5.2 Non-polynomial solution

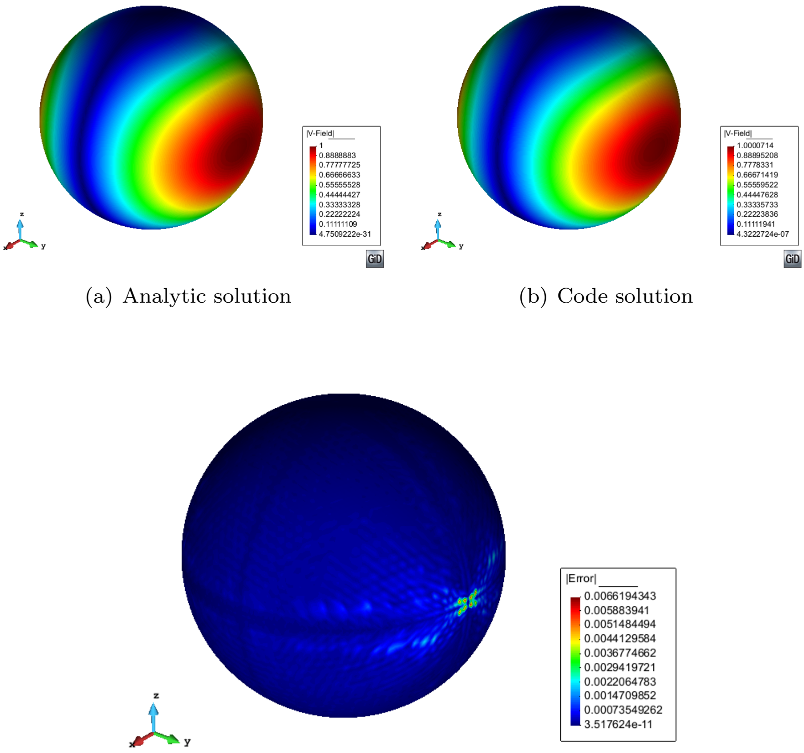
The use of polynomials as manufactured solution is a valid test to provide evidence that the code is correct and correctly implemented but, other tests are needed to assume that the solutions given by the code are correct and accurate for real world problems. Thus, new manufactures solutions should be found to finish the verification of the code. Maybe the most useful non-polynomial solution for testing an electromagnetic code is the complex exponential function. In this case, a complex exponential function with the following properties has been used

$$\mathbf{V}_{\text{analytic}} = \mathbf{E}_{\text{pol}}\, e^{-jk_0(\mathbf{k}_p \cdot \mathbf{r})} \tag{3.47}$$

where $\mathbf{E}_{\text{pol}}$ is the polarization vector, $\mathbf{k}_p$ is the propagation vector and $\mathbf{r}$ is the vector with the real coordinates where the function is evaluated. The exponential function has been polarized in both θ and ϕ components and incident angles are $\phi = 30°$ and $\theta = 20°$. In this case, the module should be able to represent the manufactured solution with an error that decays as the discretization of the FEM domain contains a lower geometrical error.

Table 3.4: Relative/maximum errors for several mesh sizes of a sphere using a polynomial as manufactured solution

Tetrahedras	Unknowns	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
88	710	$1.090e^{-2}$	$6.705e^{-2}$	$5.373e^{-2}$	$2.342e^{-1}$
1986	13958	$8.282e^{-4}$	$2.486e^{-2}$	$3.078e^{-3}$	$5.805e^{-2}$
9069	61096	$1.937e^{-4}$	$2.279e^{-2}$	$9.719e^{-4}$	$4.358e^{-2}$
26929	177838	$7.201e^{-5}$	$1.146e^{-2}$	$2.435e^{-4}$	$3.745e^{-2}$
77892	507630	$2.376e^{-5}$	$9.128e^{-3}$	$8.532e^{-5}$	$2.111e^{-2}$
113390	736650	$1.631e^{-5}$	$6.619e^{-3}$	$5.858e^{-5}$	$2.413e^{-2}$



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 3.16: 3D representation of the polynomial solution over a sphere

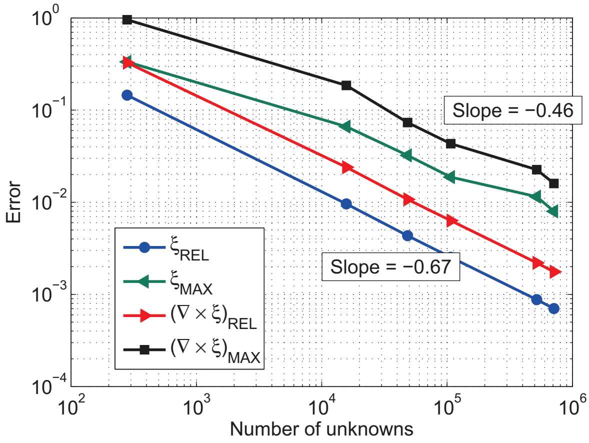


Figure 3.17: Convergence rate of the error over a cube using a complex exponential as manufactured solution

Results for several mesh discretization of a cube

Table 3.5 gathers the relative/maximum errors for different mesh discretization of the cube. As the manufactured solution is a complex exponential, an error close to the machine precision should not be expected in this case. Both relative and maximum errors decay as the number of element is increased, confirming the expected behavior. Figure 3.17 shows how the errors decay when the number of unknowns of the problem increases. For this test, the relative errors decay approximately with a slope of -0.67 and the maximum errors with a slope of -0.46. It is worth to note that the theoretical expression of the convergence rate of the relative error for this test is given by

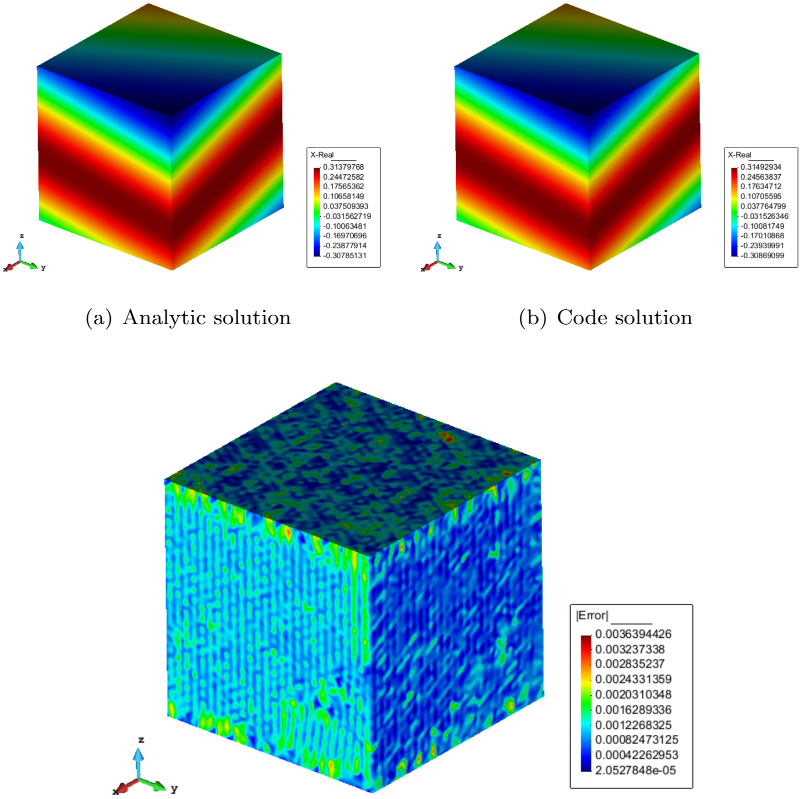
$$\xi_{\mathtt{REL}} \leq \alpha\, \mathtt{N}^{-2/3} \tag{3.48}$$

which matches exactly with the results given by the module. Details about this theoretical expression are provided in [7].

Figure 3.18(a) shows a 3D representation of the analytic solution (real part of x-component) for the smallest mesh discretization of the cube. Figure 3.18(b) shows the solution given by the FEM code for the same component. The error between both solutions is shown in figure 3.13(c) where a distributed error around the cube may be appreciated.

Table 3.5: Relative/maximum errors for several mesh sizes of a cube using an exponential function as manufactured solution

Tetrahedras	Unknowns	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
28	280	$1.453e^{-1}$	$3.336e^{-1}$	$3.267e^{-1}$	$9.548e^{-1}$
2283	15706	$9.597e^{-3}$	$6.620e^{-2}$	$2.400e^{-2}$	$1.855e^{-1}$
7237	48450	$4.340e^{-3}$	$6.223e^{-2}$	$1.076e^{-2}$	$7.335e^{-2}$
16150	107084	$2.531e^{-3}$	$1.876e^{-2}$	$6.332e^{-3}$	$4.322e^{-2}$
79747	518772	$8.773e^{-4}$	$1.150e^{-2}$	$2.203e^{-3}$	$2.259e^{-2}$
109733	712062	$7.021e^{-4}$	$7.970e^{-3}$	$1.759e^{-3}$	$1.598e^{-2}$



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 3.18: 3D representation of the exponential solution over a cube

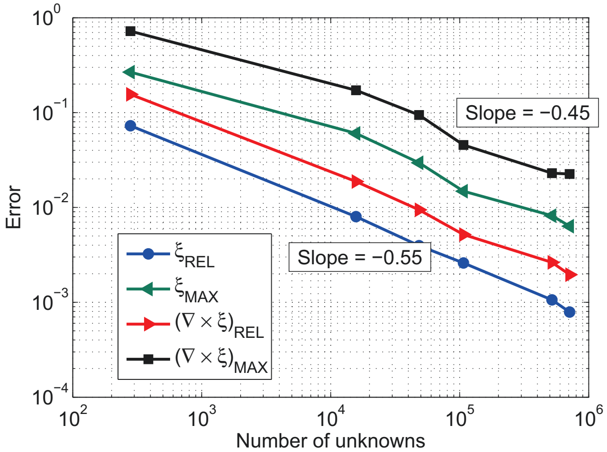


Figure 3.19: Convergence rate of the error over a cylinder using a complex exponential as manufactured solution

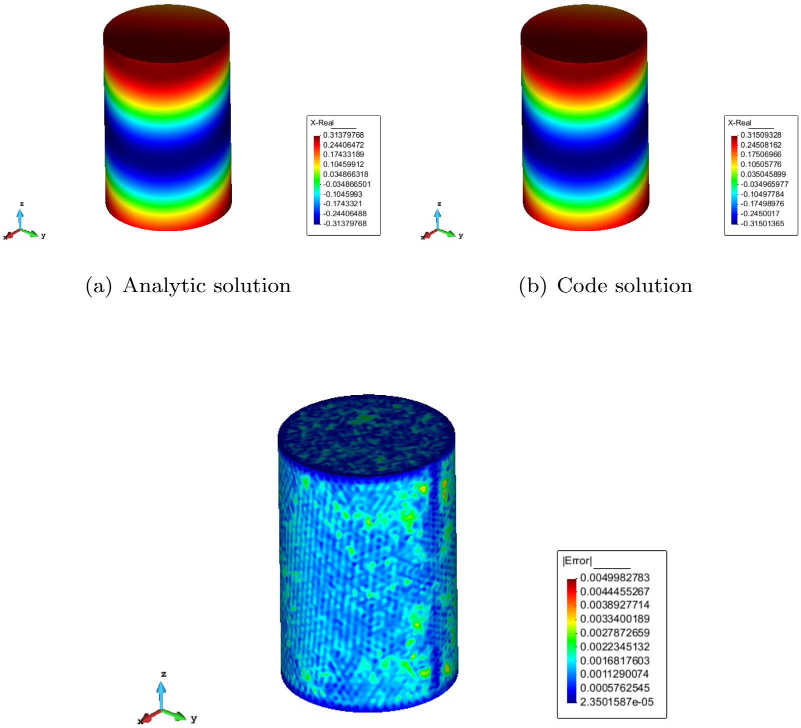
Results for several mesh discretization of a cylinder

Table 3.6 gathers the relative/maximum errors for different mesh discretization of the cylinder. As it happens in the previous case, the errors decay as the number of elements is increased. The convergence rate of the errors when the size of the problem is larger (a higher number of unknowns) is shown in figure 3.19. In this case, the slope of the maximum errors is very similar to the cube test (around -0.45), while the slope corresponding to the relative errors is slightly lower (-0.55). The slope of the relative errors differs slightly from the theoretical value ($-\frac{2}{3}$), which is expected, since the presence of curved elements in the geometry may introduce some deviation in the error.

Figure 3.20(a) shows a 3D representation of the analytic solution (real part of x-component) for the finer discretization of the cylinder. Figure 3.20(b) shows the solution given by the code for the same real component. The error between both solutions is presented in figure 3.13(c), where a smooth distribution of the error may be appreciated. Tt is worth noting that the elements located near the edges of the cylinder have lower errors since those elements present a smaller mesh discretization with consequent lower geometric error.

Table 3.6: Relative/maximum errors for several mesh sizes of a cylinder using an exponential function as manufactured solution

Tetrahedras	Unknowns	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
162	1316	$7.256e^{-2}$	$2.668e^{-1}$	$1.558e^{-1}$	$7.223e^{-1}$
4321	29370	$8.008e^{-3}$	$9.036e^{-2}$	$1.875e^{-2}$	$1.716e^{-1}$
11820	78718	$3.957e^{-3}$	$4.953e^{-2}$	$9.447e^{-3}$	$9.443e^{-2}$
36191	238554	$1.699e^{-3}$	$1.485e^{-2}$	$4.176e^{-3}$	$3.552e^{-2}$
73014	477110	$1.061e^{-3}$	$1.198e^{-2}$	$2.632e^{-3}$	$2.301e^{-2}$
112707	733462	$7.883e^{-4}$	$6.339e^{-3}$	$1.954e^{-3}$	$2.255e^{-2}$



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 3.20: 3D representation of the exponential solution over a cylinder

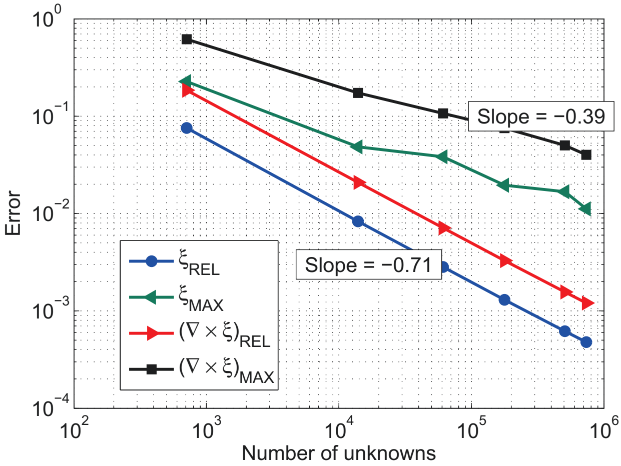


Figure 3.21: Convergence rate of the error over a sphere using a complex exponential as manufactured solution

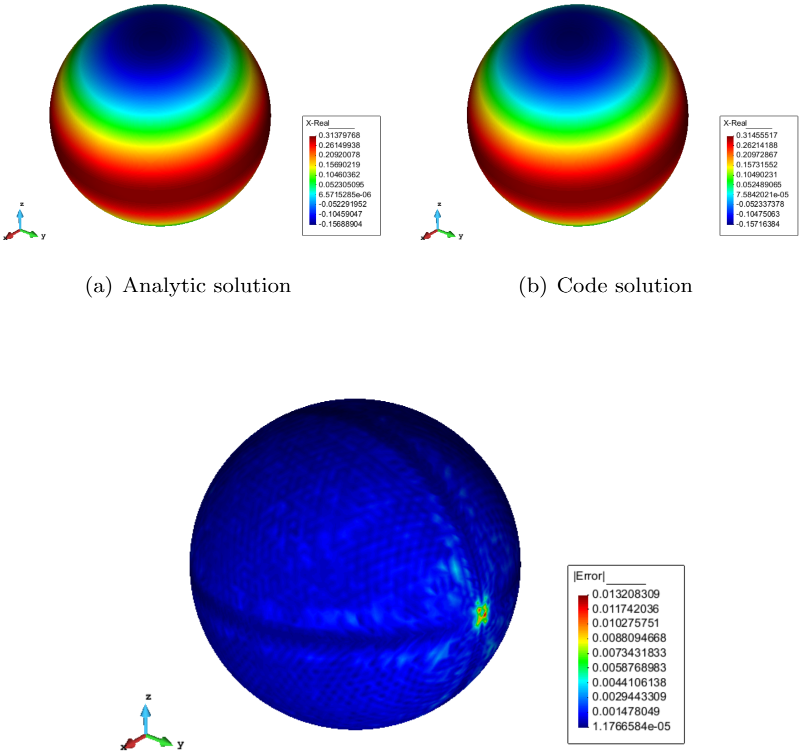
Results for several mesh discretization of a sphere

Table 3.7 gathers the relative/maximum errors for six different discretization of a sphere. As it happens in the previous tests, the errors decay according to a better mesh discretization of the structure. Figure 3.21 shows how the errors have a linear downgrade when the number of unknown of the problem is larger. The slopes of the graphs are similar to the other verification tests, around -0.71 for the relative errors and approximately -0.39 for the maximum errors. In this case, the slope of the relative error is again very close to the theoretical value providing evidences that the code is correctly implemented.

Figure 3.22(a) shows the analytic solution for the smallest discretization of the sphere Figure 3.22(b) shows the solution given by the code for the same component. The error between both solutions is shown in figure 3.22(c) where the same concentration of errors than the one presented on the polynomial test can be appreciated. As it was commented before, this type of error is due to the bad shape of the finite element located in that part of the sphere. Considering the rest of the sphere, the behavior of the error is correct and a mesh discretization with a lower geometrical error provided a lower relative error.

Table 3.7: Relative/maximum errors for several mesh sizes of a sphere using an exponential function as manufactured solution

Tetrahedras	Unknowns	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
88	710	7.570e^{-2}	2.269e^{-1}	1.841e^{-1}	6.175e^{-1}
1986	13958	8.299e^{-3}	4.844e^{-2}	2.086e^{-2}	1.735e^{-1}
9069	61096	2.823e^{-3}	3.826e^{-2}	7.133e^{-3}	1.069e^{-1}
26929	177838	1.300e^{-3}	1.953e^{-2}	3.282e^{-3}	7.543e^{-2}
77892	507630	6.185e^{-4}	1.676e^{-2}	1.565e^{-3}	5.006e^{-2}
113390	736650	4.772e^{-4}	1.118e^{-2}	1.205e^{-3}	5.310e^{-2}



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 3.22: 3D representation of the exponential solution over a sphere

3.6 Numerical Results

Once the implementation of the code have been verified using the Method of Manufactured Solutions, the simulation of real world application is required in order to proof that the code is ready for research use. The numerical results of these real world problems has been compared with established codes (as commercial softwares) and measurement to provide evidences that the code can be used for many type of simulations.

The platform used to run all the simulation documented here has been a desktop computer with 2-way quad-core processors Intel Xeon E5620 (2.4 GHz, 12 Mb cache, 5.86 GT/s), 32 GB RAM and 1 TB of hard disk.

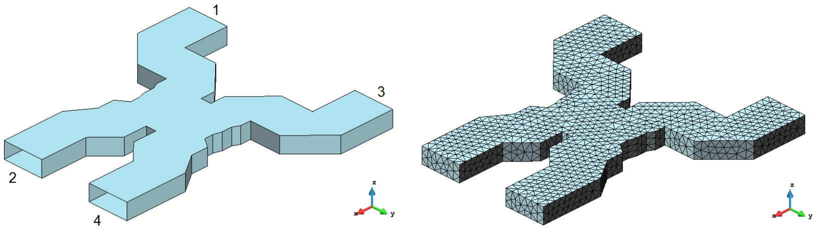
Typically, users can find three main types of problems in electromagnetism, namely, waveguide, scattering and radiation problems. Thus, the examples presented in this section have covered all these areas.

3.6.1 Waveguide problems

The simulations performed to validate the results of the FEM module for waveguide problems has covered different structures starting from simple X-band waveguides to complex waveguide filters with dielectric resonators or iris waveguide bandpass filters. In order to avoid the extension of this document only three examples has been attached: a power combiner working at 20 GHz, a filter with one dielectric resonator and a filter with four dielectric resonators working on the range of 11-12 GHz.

Power combiner

This example consists of the analysis of a standard WR42 waveguide combiner with a four-port combining junction. Figure 3.23 shows the details of the combiner. Each waveguide is 420 mils wide and 170 mils high. This type of waveguide combiner is used to combine the output power of two 20 GHz solid state power amplifiers (SSPA) with a very compact size and low insertion loss. The outputs of the SSPAs are fed into ports 2 and 4 of the waveguide with a 90° out-of-phase separation to steer the output power of the amplifiers to port 1. Port 3 of the waveguide is the isolated port where the impedance mismatch at the output (port 1) is absorbed.



(a) Geometry model (b) Mesh model

Figure 3.23: Model of the power combiner

Figure 3.24 shows a `3D` representation of the electric field (in particular, real part of z-component) when port 1 is excited. The behavior commented above, where port 2 and 4 of the waveguide are 90° out-of-phase is clearly appreciated in the figure.

The working frequency of this example is 20 `GHz` providing a model with an electrical size of 3.8 λ by 0.6 λ by 4.2 λ. The number of elements in the mesh is 17,338 tetrahedrons and the number of unknowns corresponding to that number of elements is 120,006. The memory used to solve this problem was 1.3 `GB` and the simulation took 10 seconds.

The scattering parameters of the combiner have been calculated using the established commercial software `HFSS` based also on `FEM`. Table 3.8 gathers the comparison between the scattering parameters given by both softwares obtaining a good agreement between them.

Table 3.8: Scattering parameters of the waveguide combiner at 20 `GHz`

	HFSS	**SUITE-FEM**		**HFSS**	**SUITE-FEM**
S_{11}	0.024 ∠ -117.08°	0.023 ∠ -118.65°	S_{21}	0.696 ∠ 85.11°	0.699 ∠ 86.31°
S_{12}	0.696 ∠ 85.11°	0.699 ∠ 86.31°	S_{22}	0.097 ∠ 174.71°	0.097 ∠ 175.32°
S_{13}	0.097 ∠ 85.65°	0.097 ∠ 87.67°	S_{23}	0.704 ∠ -4.13°	0.707 ∠ -3.45°
S_{14}	0.704 ∠ -4.15°	0.707 ∠ -3.45°	S_{24}	0.023 ∠ 24.91°	0.024 ∠ 23.40°
S_{31}	0.097 ∠ 85.65°	0.097 ∠ 87.67°	S_{41}	0.704 ∠ -4.15°	0.707 ∠ -3.45°
S_{32}	0.704 ∠ -4.13°	0.707 ∠ -3.44°	S_{42}	0.023 ∠ 24.91°	0.024 ∠ 23.40°
S_{33}	0.024 ∠ -117.08°	0.023 ∠ -118.21°	S_{43}	0.696 ∠ 85.15°	0.699 ∠ 86.31°
S_{34}	0.696 ∠ 85.15°	0.699 ∠ 86.31°	S_{44}	0.097 ∠ 174.77°	0.097 ∠ 175.31°

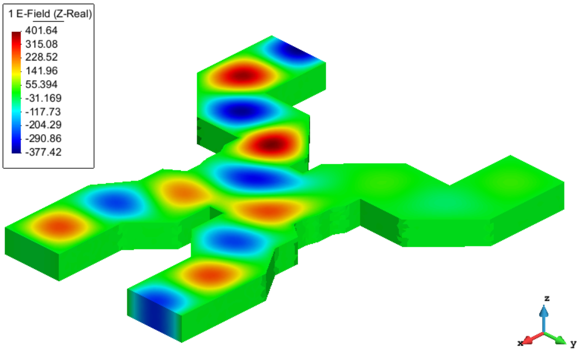


Figure 3.24: 3D plot of real part of z-component for electric field

Filter with one dielectric resonator

The following example has consisted of the simulation of a dielectric resonator coupled trough slots to two rectangular waveguide sections (see figure 3.25). The dielectric resonator consists of a high permittivity and low losses dielectric material. The high contrast between the value of the permittivity of the dielectric and the surrounding medium (air) makes it behave as a cavity concentrating the electromagnetic field. The dielectric resonator has a selective frequency response by itself. Note that it is placed on top of a pedestal which consists of another circular dielectric (this time of much lower permittivity). Details of this structure are given in [55].

The analysis of the filter has been focused between 11.1 GHz and 11.4 GHz where the first resonance appears. This resonance has a field singularity, so it is a challenge for numerical methods to calculate its exact frequency location. Figure 3.26 shows the frequency response of the filter calculated using a semi-analytic method [56], the commercial software HFSS, the FEM module of the suite and an in-house FEM code with *hp*-adaptivity [?]. The figure shows the difficult to obtain the exact location of the resonance since the only code that match with the semi-analytic method is the in-house *hp*-adaptivity code. HFSS gives a 20 MHz shift from the location of the resonance and the FEM module a shift of 30 MHz.

It is worth noting that an adaptive mesh is required to calculate the location of the resonance. The in-house *hp*-adaptivity code makes use

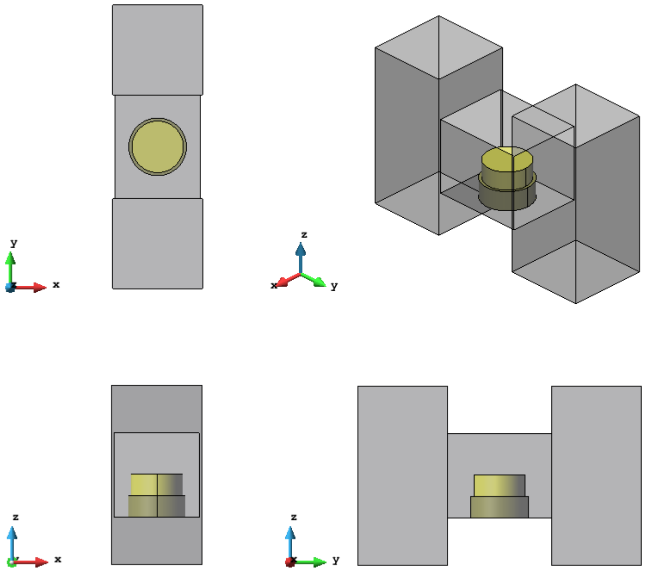


Figure 3.25: Slot coupled dielectric loaded resonator

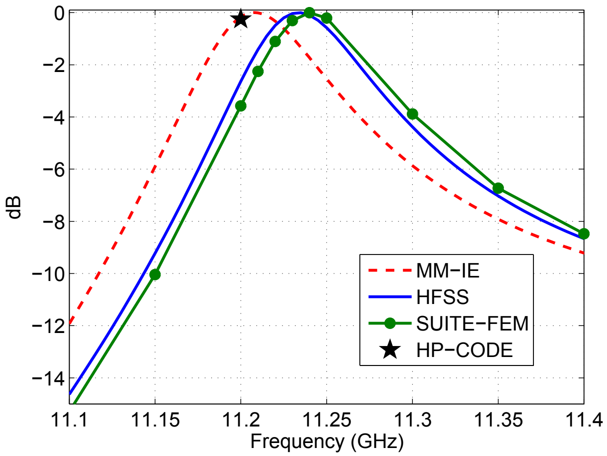


Figure 3.26: Frequency response of the filter for the first resonance

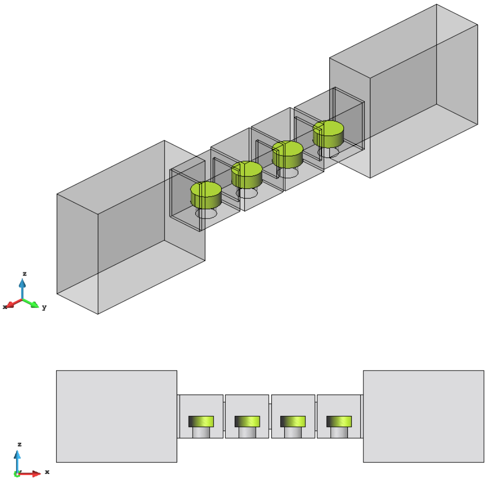


Figure 3.27: Slot coupled dielectric loaded resonator

of an adaptive method that changes, at the same time, the order of the approximation (p refinement) and the size of the elements of the mesh (h refinement with hanging nodes). `HFSS` also makes use of an adaptive method but only regular h refinements without hanging nodes were made modifying the size of the elements of the mesh.

In the case of the `FEM` module, the refinement of the mesh was made manually, since an adaptive mesh process is not available in this version of the suite. The number of elements of the mesh to achieve the results shown above was 102,057 tetrahedrons providing 653,510 unknowns. At the maximum frequency the electrical size of the model was $0.36\,\lambda$ by $0.72\,\lambda$ by $1.14\,\lambda$. The problem was solved in an out-of-core way requiring almost 13 `GB` of memory and 11 minutes per frequency.

Filter with four dielectric resonators

A filter composed of four aligned resonators coupled by rectangular slots has been considered for this validation test. Figure 3.27 shows the model of this filter. The inputs are realized in a rectangular waveguide `WR75` where a = 19.05 mm and b = 9.52 mm. Details about the dimensions of the resonant cavities and the permittivity of the dielectric resonators may be found in [57].

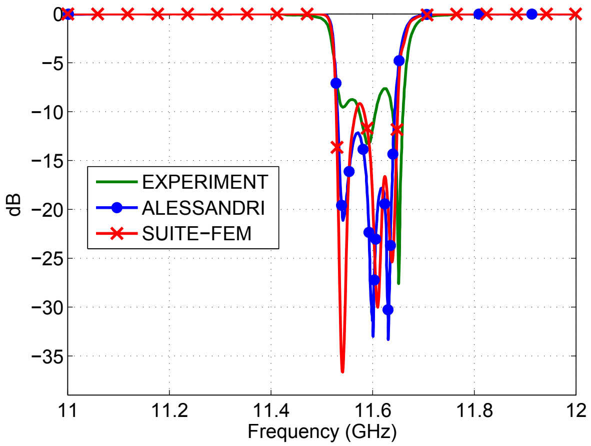


Figure 3.28: Comparison of S_{11} parameter for the filter with four dielectric resonators

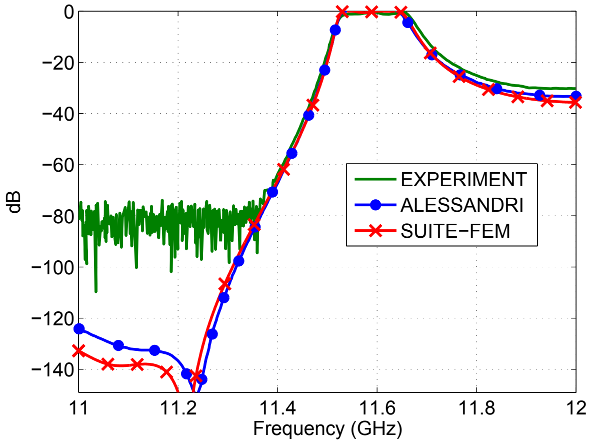


Figure 3.29: Comparison of S_{12} parameter for the filter with four dielectric resonators

The response of the filter without the tuning screws is compared with the analysis done by the `FEM` module and by the `EFIE` method provided in [57]. Figures 3.28 and 3.29 shows the comparison between the results presenting a very good agreement between the simulated `EFIE` and simulated `FEM`. Also a good agreement of both the theoretical simulations with the experimental results may be found.

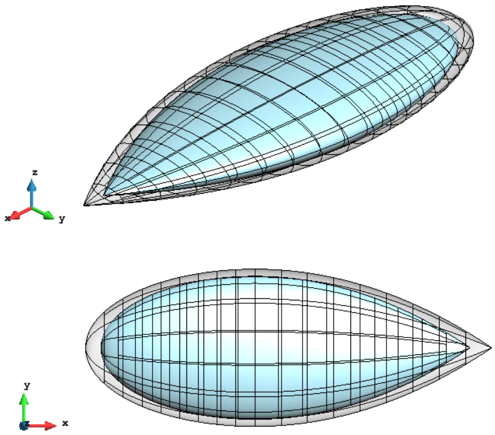


Figure 3.30: NASA almond with conformal truncation boundary

The number of finite elements used to achieve the results shown above was 85,158 tetrahedrons providing 548,896 unknowns. The electric size of the model was 0.38 λ by 0.76 λ by 3.54 λ at 12 GHz. This problem was also solved using the out-of-core version of the module requiring around 10 GB of memory and 8.5 minutes per frequency.

3.6.2 Scattering results

The simulations carried out to validate the results of the FEM module for scattering problems have consisted of the analysis of RCS targets made of different materials. This document contains the analysis of some of them, such as, the monostatic analysis of the well-known metallic NASA almond and the monostatic analysis of the NASA almond with a lossy dielectric coating.

Metallic NASA almond

The monostatic analysis of a metallic NASA almond at 1190 MHz has been considered in this example. The parametric equations that define the geometry of the NASA almond are well-known and available in the literature (for instance, [58]). The model of the NASA almond used in the simulation is shown in figure 3.30 where a conformal truncation boundary may be appreciated.

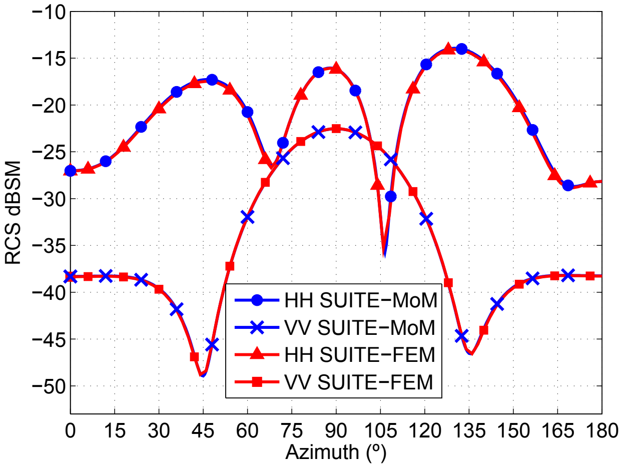


Figure 3.31: Comparison of RCS of NASA almond at 1.19 GHz

The results given by the FEM module have been compared with the measurements provided by [58] and also with the results given by the MoM module of the suite. Two polarization has been considered in the analysis: horizontal or $\phi\phi$-polarization and vertical or $\theta\theta$-polarization. Figure 3.31 shows the comparison between the simulated results for the horizontal and the vertical polarization, respectively. An excellent agreement in both polarizations is observed despite the fact the differences between both numerical techniques.

The number of finite elements used in the discretization of the NASA almond was 13,590 tetrahedrons, which provide 96,862 unknowns. The electrical size of the model at 1190 MHz was 0.44 λ by 0.70 λ by 1.31 λ. In this case, the in-core version of the FEM module was used to perform the simulation requiring around 2.8 GB of memory. The total time needed to complete the monostatic simulation was 28 minutes, but it is worth noting that 182 excitations were analyzed in total.

Coated NASA almond

The monostatic analysis of a coated NASA almond has been carried out. Here, only the $\theta\theta$-polarization has been considered. The incident frequency is 3 GHz and the thickness of the coating is 10 mm. The model of the coated NASA almond used for the simulation is shown in figure 3.32.

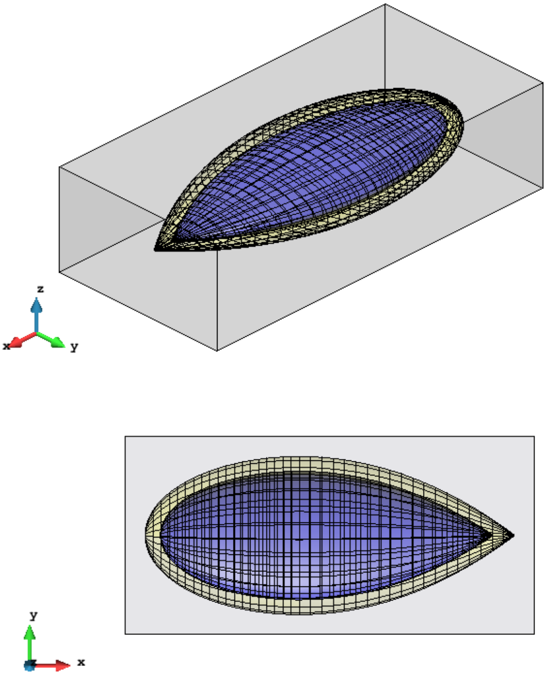


Figure 3.32: Two views of the coated NASA almond model

For this analysis, as there is no measurement to compare the results, the FEM simulations have been compared with the results provided by the MoM module of the suite. The simulations have consisted of the analysis of the coated NASA almond using a lossy dielectric shell ($\epsilon_r = 3 - 2i, \mu_r = 2 - i$). Figure 3.33 shows the comparison between the results given by the FEM module and the MoM module of the suite. An excellent agreement is observed despite the fact of the use of such different numerical techniques.

The total finite elements used in the discretization of the model was 73,299 tetrahedrons, which provide 491,002 unknowns. The electrical size of the model at 3000 MHz was 1.12 λ by 1.77 λ by 3.32 λ. The simulation was carried out using the out-of-core version of the FEM module requiring around 9.8 GB of memory. The total time needed to complete the simulation was 1 hour and 48 minutes analyzing 91 excitations in total.

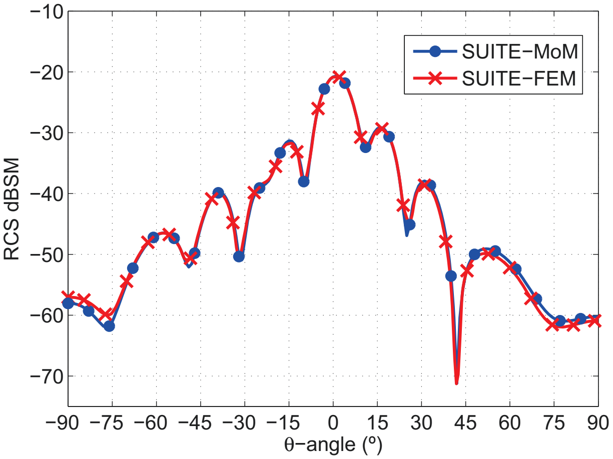


Figure 3.33: RCS results for the coated NASA almond at 3 GHz

3.6.3 Radiation results

The simulations performed to validate the results of the FEM module for radiation problems has consisted of the analysis of different kind of antennas ranging form simple metallic horn antennas to complex helical antennas or planar antennas with different substrates. The antennas documented in this document are a metallic helical antenna working in the range of 200 MHz to 800 MHz and a circular patch antenna with a resonance at 2.4 GHz.

Helical antenna

This example has consisted of the analysis of a helical antenna in the range of 200 MHz to 800 MHz. The parametric equations that define the geometry of the helix are give

$$x = r\,\mathtt{cos}(t) \qquad y = r\,\mathtt{sin}(t) \qquad z = \frac{rt}{2\,\pi} \tag{3.49}$$

where t is the parametric variable ranging from $0 \leq t \leq 38$ cm and r is the radius of the helix (5 cm in this case). The antenna is mounted over a circular ground plane 12.5 cm of radius. Different views of the antenna are shown in figure 3.34.

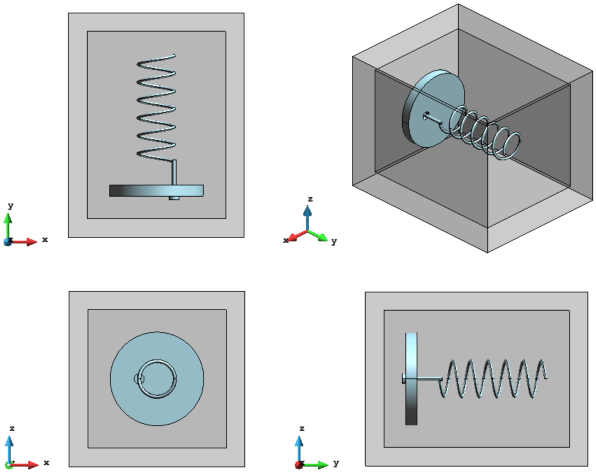


Figure 3.34: Different views of the model of the helical antenna

The analysis of the antenna has been performed using the commercial software `CST` and the `FEM` module of the suite. The comparison of the S_{11} parameter from 200 `MHz` to 800 `MHz` is shown in figure 3.35. The results of both softwares closely match each other. Two different cuts ($\theta = 90°$ and $\phi = 90°$) of the directivity at 600 `MHz` are shown in figures 3.36 and 3.37. Again a good agreement is observed in the results of both softwares.

The electrical size of the model at the maximum frequency of the analysis was λ by λ by $1.33\,\lambda$. The total finite elements used in the discretization of the model was 43,547 tetrahedrons, which provide 288,102 unknowns. The simulation was carried out using the out-of-core version of the `FEM` module requiring around 5.4 `GB` of memory. The total time needed to complete the simulation was 4.3 minutes per frequency.

Circular patch antenna

This example has consisted of the analysis of a circular patch antenna operating at 2.4 `GHz`. The dimensions of the substrate are 60 mm by 60 mm by 0.7 mm with a relative permittivity of $\epsilon_r = 2.3$. The radius of the circular patch is 23.2 mm and the antenna is fed by a coaxial cable. Some screenshots of the antenna are shown in figure 3.38.

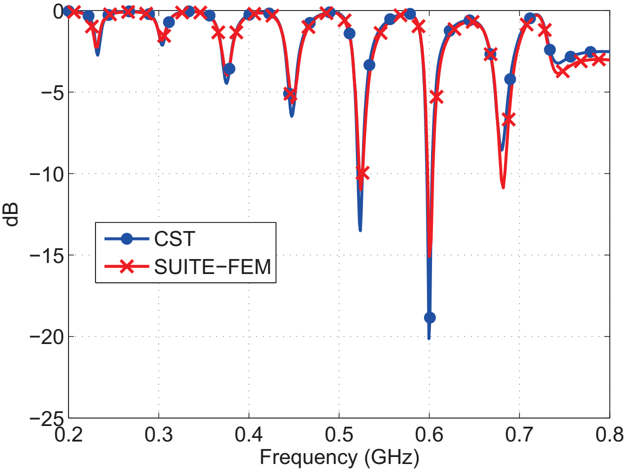


Figure 3.35: Comparison of the S_{11} parameter of the helical antenna from 200 MHz to 800 MHz

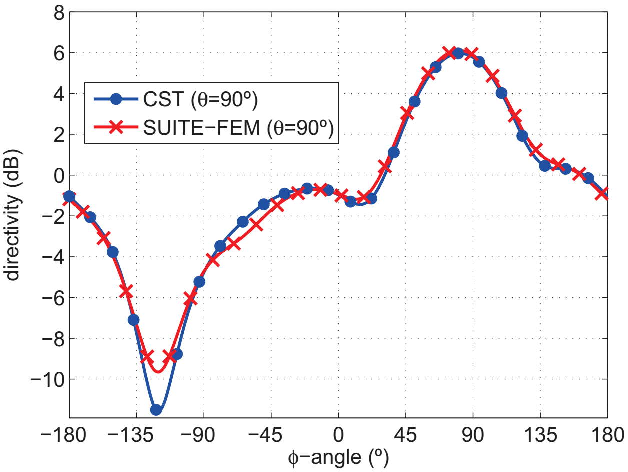


Figure 3.36: Comparison of the directivity for $\theta = 90^{\circ}$ cut at 600 MHz

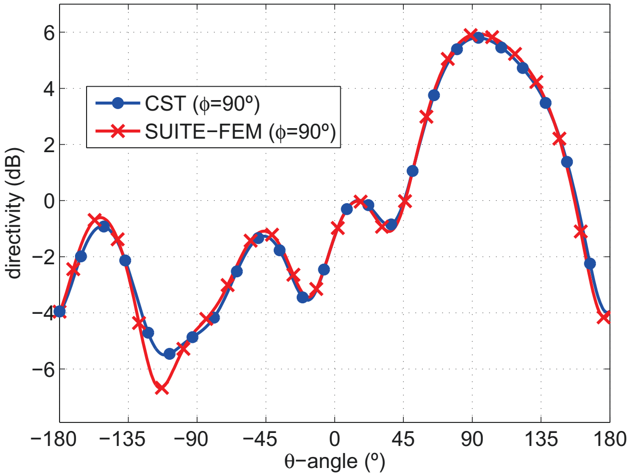


Figure 3.37: Comparison of the directivity for $\phi = 90^{\circ}$ cut at 600 MHz

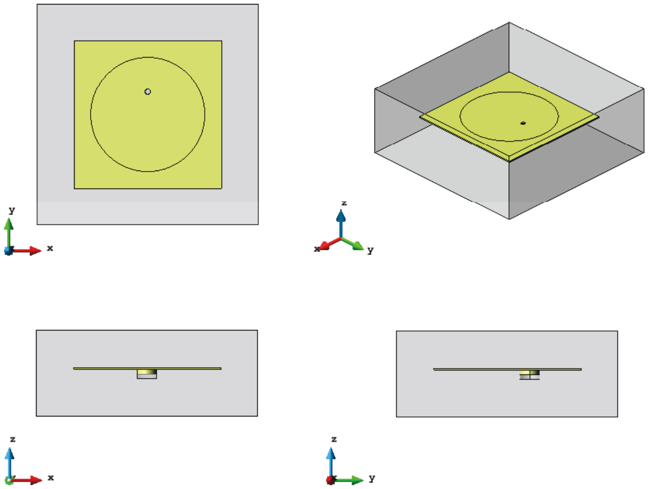


Figure 3.38: Different views of the model of the circular patch antenna

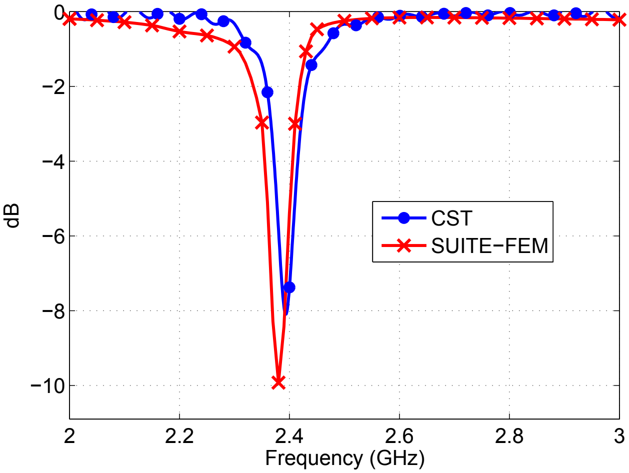


Figure 3.39: Comparison of the S_{11} parameter of the patch antenna

The results given by the FEM module has been compared with the results provided by the commercial software CST. The comparison of the S_{11} parameter from 2 GHz to 3 GHz is shown in figure 3.39. In this case, a good agreement is observed, although a slight frequency shift may be appreciated between both results. Two different cuts ($\theta = 90°$ and $\phi = 90°$) of the directivity are shown in figure 3.40. The results of both softwares closely match each other.

The electrical size of the model at the maximum frequency of the analysis was $0.9\,\lambda$ by $0.9\,\lambda$ by $0.35\,\lambda$. The total finite elements used in the discretization of the model was 26,894 tetrahedrons, which provide 174,494 unknowns. The simulation was carried out using the out-of-core version of the FEM module requiring around 3.7 GB of memory. The total time needed to complete the simulation was 3.6 minutes per frequency.

3.7 Conclusions and future research lines

The Finite Element module of the suite has been presented in this chapter. The module is based on the so-called Finite Element - Iterative Integral Equation Evaluation (FE-IIEE) method which makes use of a boundary integral equation providing a (numerically) exact radiation boundary condition.

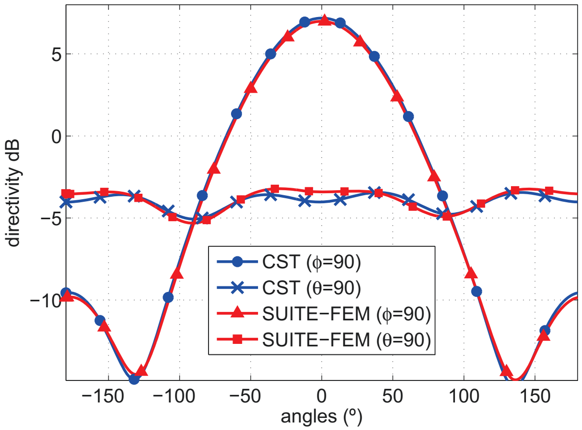


Figure 3.40: Comparison of the directivity for different 2D cuts at a operating frequency of 2.4 GHz

The module allows the use of four different sparse solver and provides a parallel implementation where one can utilize all the CPU cores available on single computers or on several compute nodes of a high performance computing (HPC) cluster.

The accuracy of the results given by the module has been verified applying the Method of Manufactured Solutions to different verification tests obtaining excellent results. Also, the numerical results of real world problems has been compared with established codes (as commercial softwares) and measurement providing evidences that the code can be used for many type of simulations with very good performance and scalability in mid-size HPC clusters.

CHAPTER 4

STUDY OF THE HYBRIDIZATION OF FEM-MOM TECHNIQUES

Taking advantage of the existence in the suite of two of the most important computational electromagnetics numerical techniques such as, the Method of Moments (MoM) and the Finite Element Method (FEM), the hybridization between them seems an appropriate choice to perform complex simulation where the use of these techniques alone may not be efficiently appropriate.

It is well-known that MoM is a powerful technique for the simulation of radiation/scattering problems of both metallic and dielectric structures. However, the method presents some small inaccuracies for geometric models containing complex non-homogeneous dielectric materials. Conversely, FEM offers a great flexibility in the treatment of complex structures, even in the case of those non-homogeneous dielectric materials. Thereby, it seems an appropriate decision to hybridize both techniques joining their advantages.

This chapter presents a modular approach to combine MoM and FEM techniques for the analysis of large structures or finite arrays with complex radiating elements. The approach consists of performing the FEM analysis of each structure and to define the equivalent electric and magnetic currents on the boundaries of the structure.

Then, the interaction between the structures is computed by using MoM and the global FEM-MoM system of equation is assembled. Finally, the system is solved and the analysis is completed. Details about the methodology of this approach are given in section 4.1.

The numerical implementation of this modular approach requires the modification of the current variational formulation of the FEM module of the suite. This modification resides in the fact that the unknowns of a standard MoM formulation are the electric and magnetic currents over dielectric boundary surfaces and electric currents over perfect electric conductors. However, the standard variational formulation of FEM problems only has unknowns for one of the corresponding electromagnetic fields (E-Field or H-Field). Thus, the modification of the FEM formulation is required to provide unknowns also for the corresponding dual field (H-Field or E-Field) in an explicit way. Details about the modification of the FEM formulation are given in section 4.2.

Furthermore, due to the difference between the basis functions used by both solvers, the definition of the equivalent electric and magnetic current on the boundaries of the antenna is not straightforward. Note that the boundaries of the antenna is where both methods share the unknowns and those need to be expressed in terms of the same basis functions. Thereby, a study of the projection between the basis functions of both modules is required in order to connect the unknowns between both codes. In the particular case of this Ph. D. dissertation, a study between the well-known Rao-Wilton-Glisson (RWG) div-conforming basis functions [59], typically used in MoM, and the the Nédélec triangular curl-conforming basis functions of second-order is performed. Details about this study are given in section 4.3.

4.1 Methodology of the approach

The methodology of the modular approach proposed in this Ph. D. dissertation to combine the FEM and MoM techniques is described below. Figure 4.1 depicts a setup problem where the analysis of complex objects is performed through FEM, the full metallic objects are analyzed through MoM and the interaction between them is modeled by using MoM. The methodology of the approach is as follows:

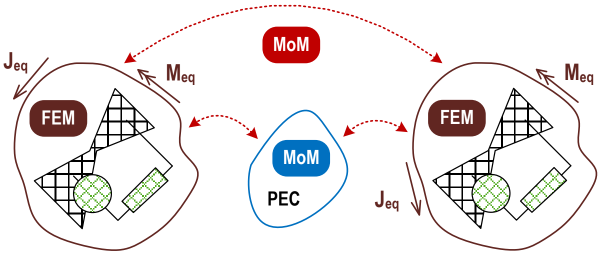


Figure 4.1: Setup problems of the modular approach between the FEM code and the MoM code

- **Step 1:** The FEM analysis of each object is performed and the unknowns (field $\mathbf{V}$ and its dual field $\mathbf{V}_d$) corresponding to the boundaries (S) of the object are extracted through the application of the Schur complement to the FEM matrix with respect to those boundary unknowns. Thus, a reduce system of equations is obtained where only the unknowns of the boundaries of the objects are contained. This system of equations may be expressed as follows:

$$\begin{bmatrix} \bar{\bar{A}}_{(1)} & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \bar{\bar{A}}_{(n)} \end{bmatrix} \begin{bmatrix} \mathbf{g}_{(1)} \\ \vdots \\ \mathbf{g}_{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{(1)} \\ \vdots \\ \mathbf{b}_{(n)} \end{bmatrix} \tag{4.1}$$

  where the matrix $\bar{\bar{A}}$ refers to the Schur complement of the the FEM matrix of the n-th object, $\mathbf{g}$ corresponds to the degrees of freedom associated to the n-th object and the term $\mathbf{b}$ is related to the field on the boundaries of the n-th object.

  Note that if the FEM discretization of the objects is the same, as for instance happens in the case of the radiating elements of arrays, the FEM analysis is only performed in one object since the system of equations is equivalent for the others ,i.e., $\bar{\bar{A}}_{(1)} = \cdots = \bar{\bar{A}}_{(n)}$.

- **Step 2:** The interaction between the fields radiated by the objects is computed using the MoM technique. A new system of equations is obtained where the impedance matrices of the objects involved in the analysis are computed. This system of equations may be expressed as follows:

$$\begin{bmatrix} \bar{\bar{Z}}_{(11)} & \cdots & \bar{\bar{Z}}_{(1n)} \\ \vdots & \ddots & \vdots \\ \bar{\bar{Z}}_{(n1)} & \cdots & \bar{\bar{Z}}_{(nn)} \end{bmatrix} \begin{bmatrix} \mathbf{g}_{(1)} \\ \vdots \\ \mathbf{g}_{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{(1)} \\ \vdots \\ \mathbf{c}_{(n)} \end{bmatrix} \tag{4.2}$$

where the matrix $\bar{\bar{Z}}$ refers to the impedance matrix of the interaction between the i-th and j-th object, $\mathbf{g}$ corresponds to the degrees of freedom associated to the n-th object and the term $\mathbf{c}$ is related to the field on the boundaries of the n-th object.

- **Step 3:** Then, the continuity of the fields on the boundary of the objects is applied obtaining the global FEM-MoM system of equations. It is worth to note that, at this point, the FEM matrix $\bar{\bar{A}}$ must be treated to be connected with the MoM formulation. This connection is performed through the projection of the FEM basis functions into the MoM basis functions (further details about these projections are given in section 4.3).

$$\begin{bmatrix} \bar{\bar{Z}}_{(11)} - \bar{\bar{A}}_{(1)} & \cdots & \bar{\bar{Z}}_{(1n)} \\ \vdots & \ddots & \vdots \\ \bar{\bar{Z}}_{(n1)} & \cdots & \bar{\bar{Z}}_{(nn)} - \bar{\bar{A}}_{(n)} \end{bmatrix} \begin{bmatrix} \mathbf{g}_{(1)} \\ \vdots \\ \mathbf{g}_{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{(1)} \\ \vdots \\ \mathbf{b}_{(n)} \end{bmatrix} \tag{4.3}$$

- **Step 4:** The system of equations is ready to be solved by using the appropriated dense solver (for instance, SCALAPACK).

- **Step 5:** The unknowns residing on the boundaries of the objects are solved through the solution of the previous system of equations. The last task is to get the solution for the unknowns of the interior FEM region. To do this, the coefficients associated to the MoM basis functions has to be transformed to the corresponding FEM basis functions. This transformation is performed through a representation of the MoM basis functions into the FEM basis functions. (further details about these representations are given in section 4.3.3).

- **Step 6:** Once the transformation is performed, the solution for the interior FEM region is obtained and the analysis is completed.

This approach, where the FEM technique is used to analyze each object and the MoM technique is used to calculate the interaction between them, entails the modification of the current variational formulation of the FEM module. This is due to the fact that, the degrees of freedom (*dofs*) of a standard MoM formulation are associated to both electric and magnetic currents over dielectric boundary surfaces and electric currents over perfect electric conductors. However, the *dofs* of the current variational formulation of the FEM module are only associated to one electromagnetic field (E-field or H-field). Thus, the modification of the FEM formulation is needed to provide *dofs* associated also to the dual field (H-field or E-field, respectively) in an explicit way. The following section describe the modification performed in the variation formulation of the FEM module to be able to provide the matrix $\bar{\bar{A}}$.

It is worth to note that the equivalent electric and magnetic currents on the boundaries of the object may be defined with the help of Macro Basis Functions (MBFs). It is generally accepted that, as long as the elements are small with respect to the wavelength, high accuracy can be achieved with relatively very few MBFs, which leads to a dramatic reduction of the number of unknowns when the elements of the array are complex.

4.2 Variational Formulation

As commented above, the current variational formulation of the FEM module has to be modified in order to provide the unknowns corresponding to the dual field $\mathbf{V}_d$ in an explicit way. Thus, the final system of equations will contains unknowns for both electric and magnetic fields enabling the connection with the MoM formulation.

Lets start from the current variational formulation of the FEM module of the suite. The weighted-integral form is given by

$$\begin{aligned}&\iiint_\Omega \boldsymbol{\nabla} \times \mathbf{W} \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) dV - k_0^2 \iiint_\Omega \mathbf{W} \cdot \bar{\bar{g}}_r \, \mathbf{V} \, dV \\ &+ \gamma \iint_{\Gamma_C} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \mathbf{V}) \, dS = \iiint_\Omega \mathbf{W} \cdot \mathbf{q} \, dV - \iint_{\Gamma_C} \mathbf{W} \cdot \boldsymbol{\Psi} \, dS \end{aligned} \tag{4.4}$$

where $\mathbf{V}$ denotes the magnitude to be solve depending on the formulation employed, k_0 is the wavenumber in vacuum, $\mathbf{q}$ is the source term due to the presence of impressed electric and/or magnetic currents in the interior `FEM` domain and $\boldsymbol{\Psi} = \boldsymbol{\Psi}^{\mathtt{inc}} + \boldsymbol{\Psi}^{\mathtt{scat}}$. It is worth to note that the calculation of the value corresponding to $\boldsymbol{\Psi}^{\mathtt{inc}}$ is straightforward. $\boldsymbol{\Psi}^{\mathtt{inc}}$ is the result of substituting the value of the corresponding incident field $\mathbf{V}^{\mathtt{inc}}$ in equation (4.5). However, the value of the term $\boldsymbol{\Psi}^{\mathtt{scat}}$ depends on the scattered field $\mathbf{V}^{\mathtt{scat}}$ on the boundary of the object (see equation 4.6). This field is not known a priori and its calculation is required in order to compute $\boldsymbol{\Psi}^{\mathtt{scat}}$.

$$\boldsymbol{\Psi}^{\mathtt{inc}} = \hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}^{\mathtt{inc}} \right) + \gamma\, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}^{\mathtt{inc}} \tag{4.5}$$

$$\boldsymbol{\Psi}^{\mathtt{scat}} = \hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}^{\mathtt{scat}} \right) + \gamma\, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}^{\mathtt{scat}} \tag{4.6}$$

A first way to calculate $\boldsymbol{\Psi}^{\mathtt{scat}}$ is to introduce it as an unknown in the variational formulation of the problem. As equation (4.6) shows, the value of $\boldsymbol{\Psi}^{\mathtt{scat}}$ is calculated using the tangential component of the scattered field $\mathbf{V}^{\mathtt{scat}}$. Hence, the `FEM` formulation, adding this new unknown, is given by

$$\begin{aligned} &\iiint_\Omega \boldsymbol{\nabla} \times \mathbf{W} \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) dV - k_0^2 \iiint_\Omega \mathbf{W} \cdot \bar{\bar{g}}_r\, \mathbf{V}\, dV \\ &+ \gamma \iint_{\Gamma_{\mathrm{C}}} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \mathbf{V})\, dS - \iint_{\Gamma_{\mathrm{C}}} (\hat{\mathbf{n}} \times \mathbf{W}) (\hat{\mathbf{n}} \times \boldsymbol{\Psi}^{\mathtt{scat}})\, dS = \\ &\qquad \iiint_\Omega \mathbf{W} \cdot \mathbf{q}\, dV - \iint_{\Gamma_{\mathrm{C}}} \mathbf{W} \cdot \boldsymbol{\Psi}^{\mathtt{inc}}\, dS \end{aligned} \tag{4.7}$$

It is worth to note that $\boldsymbol{\Psi}^{\mathtt{scat}} = 0$ if the `FEM` domain is truncated using the first-order absorbing boundary condition (`ABC`). In the case of the `FE-IIEE` method, the scattered fields are calculated through the corresponding boundary integral equation and the equation (4.6) is evaluated to compute $\boldsymbol{\Psi}^{\mathtt{scat}}$. Hence, to complete the system of equation a new equation has to be added:

$$-\iint_{\Gamma_{\mathrm{C}}} (\hat{\mathbf{n}} \times \mathbf{W}) (\hat{\mathbf{n}} \times \boldsymbol{\Psi}^{\mathtt{scat}}) = 0 \qquad \texttt{for ABC} \tag{4.8}$$

$$\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}_{\texttt{FE-IIEE}} \right) + \gamma \, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}_{\texttt{FE-IIEE}} - \iint_{\Gamma_{\mathrm{C}}} (\hat{\mathbf{n}} \times \mathbf{W}) \, (\hat{\mathbf{n}} \times \boldsymbol{\Psi}^{\texttt{scat}}) = 0 \qquad \texttt{for FE-IIEE} \quad (4.9)$$

where $\mathbf{V}_{\texttt{FE-IIEE}}$ and its curl $(\boldsymbol{\nabla} \times \mathbf{V})_{\texttt{FE-IIEE}}$, are computed using the corresponding integral expressions.

This system of equations retains the sparsity of the FEM matrices if the first-order ABC is used as truncation method. However, if the FE-IIEE method is used a large dense block is added to the sparse matrix with the consequence difficulties for the sparse solver during the solving process. This first approach have the advantage that the Schur complement may be applied to the resulting matrix and it can be used directly to perform the hybridization with MoM. However, the disadvantage of the approach is that a big amount of RAM memory is needed even for small problem making the simulation of large problems prohibitive in terms of memory capabilities. Thus, this approach was discarded due to this problem.

A second approach to calculate $\boldsymbol{\Psi}^{\texttt{scat}}$ is to perform the FEM analysis of the object using the current variational formulation. In this way, the value of $\boldsymbol{\Psi}^{\texttt{scat}}$ is obtained directly after applying the FE-IIEE method. The advantage of this approach is that the sparsity of the FEM matrices is retained and previous memory problem is avoided. However, the FEM analysis of the object is required before the hybridization with the consequence extra computing time needed per simulation. In the particular case of this Ph. D. dissertation, this second approach is chosen.

At this point, the degrees of freedom corresponding to the dual function $\mathbf{V}_d$, are still not included in the FEM system of equations. The definition of the function $\boldsymbol{\Psi}$ (over S boundary) given by equation (4.10) is taken and the Galerkin Weighted Residual Method is used to obtain a weighted-integral form (note that the value of $\boldsymbol{\Psi}$ is known at this point since, following the second approach described above, $\boldsymbol{\Psi}^{\texttt{inc}}$ and $\boldsymbol{\Psi}^{\texttt{scat}}$ has been calculated previously).

$$\boldsymbol{\Psi} = \hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) + \gamma \, \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V} \qquad (4.10)$$

Applying the Galerkin method to the equation (4.10), the weighted-integral form is given by

$$\oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \boldsymbol{\Psi} = \oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] \\ + \gamma \oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{V}) \quad (4.11)$$

Note that the testing function used in the Galerkin method is rotated $(\hat{\mathbf{n}} \times \mathbf{W})$ with respect to the testing functions used in the current variational formulation of the module. Now, the rotational of the unknown field $\boldsymbol{\nabla} \times \mathbf{V}$ is expressed in terms of the dual field $\mathbf{V}_d$ using the Maxwell's equations:

$$\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} = -jk_0 h \, \mathbf{V}_d \quad (4.12)$$

Substituting equation (4.12) into equation (4.11), the weighted-integral form is given by

$$\oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \boldsymbol{\Psi} = -jk_0 h \oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot (\hat{\mathbf{n}} \times \mathbf{V}_d) \\ - \gamma \oiint_{\Gamma_S} (\hat{\mathbf{n}} \times \mathbf{W}) \cdot \mathbf{V} \quad (4.13)$$

Finally, the FEM system of equations is assembled conforming the global FEM matrices. These matrices contain the field unknowns $\mathbf{V}$ for the interior FEM region and the field unknowns V and the dual field unknowns $\mathbf{V}_d$ for the boundary S of the objects. Then, Shur complement is applied to these FEM matrices and the corresponding $\bar{\bar{A}}_{(n)}$ matrix is obtained.

4.3 Projections between basis functions

As commented in the introduction of this chapter, a study of the projection between the basis functions used to approximate the solution in the FEM and MoM modules is needed in order to connect the unknowns shared by both codes. It is worth to note that due to the differences

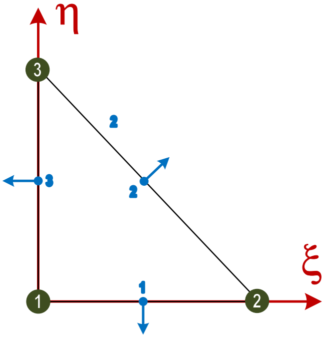


Figure 4.2: Abstract representation of the RWG basis function

between the basis functions employed by both modules (Nédélec tetrahedral curl-conforming basis functions of second-order for FEM and higher-order quadrilateral basis functions for MoM) a simpler study has been considered first. This study has been consisted of the projections between the well-known Rao-Wilton-Glisson (RWG) div-conforming basis functions [59], typically used in MoM, and the the Nédélec triangular curl-conforming basis functions of second-order.

Before to proceed with the study of the projection between these basis functions, a brief description of the Nédélec triangular curl-conforming basis functions of second-order and the RWG div-conforming basis functions is given.

4.3.1 Nédélec triangular curl-conforming basis functions

Equation (4.14) shows the space of functions of the second-order Nédélec triangle with 8 independent coefficients that have to be determined (see section 3.3 for further details). Thus, the number of degrees of freedom of the triangular element of second-order is 8, and 8 is also the number of vector basis function belonging to $\mathcal{R}^2$ that should be chosen $\mathbf{W}_j, (j = 1...8)$. Figure 4.2 shows the representation of the space of the function for the second-order Nédélec triangle.

$$\mathcal{R}^2 \equiv \left\{ \begin{array}{c} a_1 + a_2 x + a_3 y + Cy^2 - Dxy \\ b_1 + b_2 x + b_3 y - Cxy + Dx^2 \end{array} \right\} \tag{4.14}$$

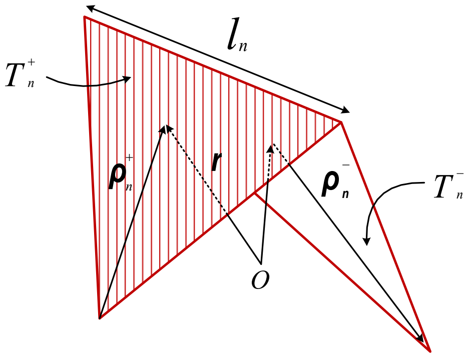


Figure 4.3: A pair of triangular basis functions

The definition of the degrees of freedom (dof) of the Nédélec triangular curl-conforming element of second-order is as follows: 6 $dofs$ associated to the three edges of the triangle given by

$$\alpha_i = \int_{edge} (\mathbf{u} \cdot \hat{\tau})\, q\, dl \quad \forall q \in P_2 \tag{4.15}$$

and 2 $dofs$ associated to the triangle given by

$$\alpha_i = \int_{face} (\hat{\mathbf{n}} \times \mathbf{u}) \cdot \mathbf{q}\, ds \quad \forall q \in (P_1)^2 \tag{4.16}$$

4.3.2 RWG basis functions

This section gives a brief description of the well-known Rao-Wilton-Glisson (`RWG`) triangular basis functions. These basis functions are associated to the edges of the triangle and have a normal component to the corresponding edge. As figure 4.3 shows, for any triangle pair, T_n^+ and T_n^-, with area A_n^+ and A_n^-, respectively, sharing a common edge l_n, the basis function can be written as

$$\mathbf{w}_\mathtt{n}(\mathbf{r}) = \begin{cases} \dfrac{l_n}{2A_n^+}\,\boldsymbol{\rho}_n^+ & \mathbf{r} \in T_n^+ \\[2ex] \dfrac{l_n}{2A_n^-}\,\boldsymbol{\rho}_n^- & \mathbf{r} \in T_n^- \\[2ex] 0 & \texttt{otherwise} \end{cases} \tag{4.17}$$

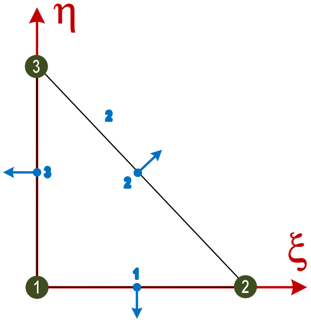


Figure 4.4: Abstract representation of the `RWG` basis function

where $\boldsymbol{\rho}_n^+ = \mathbf{r} - \mathbf{r}_n^+$ is the vector directed from the free vertex of the triangle T_n^+ to the observation point, and $\boldsymbol{\rho}_n^- = \mathbf{r}_n^- - \mathbf{r}$ is the vector directed from the observation point to the free vertex of the triangle T_n^-. The basis function is zero outside the two adjacent triangles T_n^+ and T_n^-. The current has no component normal to the boundary (which excludes the common edge) of the surface formed by the triangular pair T_n^+ and T_n^-, and hence there are no line charge along this boundary.

The representation of the `RWG` space of the functions is illustrated in figure 4.4. The relationship between the `RWG` div-conforming triangular basis functions and the Nédélec curl-conforming triangular basis functions is a simple rotation of the basis function counterclockwise. This rotation may be obtain by computing the vectorial product between the functions and the normal vector of the triangle. In other words, the relationship is given by the following expression

$$\hat{\mathbf{n}} \times \mathbf{w}_j \approx \mathbf{W}_j \qquad \texttt{or} \qquad \mathbf{W}_j \times \hat{\mathbf{n}} \approx \mathbf{w}_j \tag{4.18}$$

It is worth to note that the Nédélec space is richer than `RWG` space and the `RWG` basis space is contained into the Nédélec space:

$$\hat{\mathbf{n}} \times \mathbf{H}(\texttt{div})_1 \subset \mathbf{H}(\texttt{curl})_2 \tag{4.19}$$

The definition of the degrees of freedom (dof) of the `RWG` div-conforming element is as follows: 3 $dofs$ associated to the three edges of the triangle given by

$$\alpha_i = \int_{edge} (\mathbf{u} \times \hat{\mathbf{n}})\, \hat{\tau} q\, dl = \int_{edge} (\mathbf{u} \cdot \hat{\mathbf{n}}_\tau)\, q\, dl \tag{4.20}$$

Table 4.1: α coefficients for the reference triangle

	edge 1 ($\mathbf{w}_1^{\texttt{RWG}}$)	**edge 2** ($\mathbf{w}_2^{\texttt{RWG}}$)	**edge 3** ($\mathbf{w}_3^{\texttt{RWG}}$)
edge 1 ($\mathbf{W}_1^{\texttt{NED}}$)	0.5000	0.0000	0.0000
edge 1 ($\mathbf{W}_2^{\texttt{NED}}$)	0.5000	0.0000	0.0000
edge 2 ($\mathbf{W}_3^{\texttt{NED}}$)	0.0000	0.5000	0.0000
edge 2 ($\mathbf{W}_4^{\texttt{NED}}$)	0.0000	0.5000	0.0000
edge 3 ($\mathbf{W}_5^{\texttt{NED}}$)	0.0000	0.0000	0.5000
edge 3 ($\mathbf{W}_6^{\texttt{NED}}$)	0.0000	0.0000	0.5000
face ($\mathbf{W}_7^{\texttt{NED}}$)	-0.3333	0.1666	0.1666
face ($\mathbf{W}_8^{\texttt{NED}}$)	-0.1666	-0.1666	0.3333

4.3.3 Representation of RWG space into Nedelec space

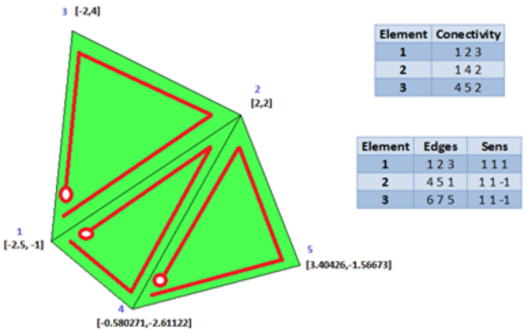
The objective here is to calculate the coefficients that represent the RWG div-conforming basis functions into the Nédélec triangular curl-conforming basis functions. For that, the following equation must be solved finding the value of the unknowns α_i

$$\mathbf{w}_j = \sum_{i=1}^{8} \alpha_i \left(\mathbf{W}_i \times \hat{\mathbf{n}}\right) \qquad \texttt{for each} \quad \mathbf{w}_{j=1\ldots3} \tag{4.21}$$

where $\mathbf{w}_j$ are the RWG functions and $\mathbf{W}_i$ are the Nédélec function.

There are different approaches that can be used to solve the previous equation, as for example, applying the Galerkin Method or using the definition of degrees of freedom of Nédélec. Both approaches have been used to calculate the α_i coefficients that represent the RWG into the Nédélec functions for the reference triangle. It is worth to note that all approaches provide identical coefficients. Thus, the representation of the RWG basis functions into the Nedelec basis functions is unequivocal. Table 4.1 gathers the α_i coefficients for the reference triangle case.

These results provide a correct local projection. For each edge, the $\mathbf{w}_j$ function associated to that edge gives trace only on the edge function $\mathbf{W}_i$ of the same edge. Also, it gives trace on the face function $\mathbf{W}_{7,8}$. For



Element	Conectivity
1	1 2 3
2	1 4 2
3	4 5 2

Element	Edges	Sens
1	1 2 3	1 1 1
2	4 5 1	1 1 -1
3	6 7 5	1 1 -1

Figure 4.5: Triangular mesh used for the numerical example

example, the representation of the first RWG basis function in the Nedelec basis function only uses the coefficients of the first column, so the Nedelec basis functions involved in the representation are $\mathbf{W}_1$,$\mathbf{W}_2$,$\mathbf{W}_7$ and $\mathbf{W}_8$.

Numerical example

The aim of this example is to approximate a given vectorial function by using the RWG basis functions and the representation of those basis function into the Nédélec space. The approximation of a given vectorial function through the previous set of basis functions is given by the following expression

$$\mathbf{f} \approx \sum_{i=1}^{3} \alpha_i \mathbf{w}_i \qquad \Longleftrightarrow \qquad \hat{\mathbf{n}} \times \mathbf{f} \approx \sum_{i=1}^{8} g_i \mathbf{W}_i \tag{4.22}$$

where $\mathbf{f}$ is the given vectorial function, $\mathbf{w}_i$ are the RWG basis functions, $\mathbf{W}_i$ are the Nédélec basis functions, α_i are the coefficients that represent the function in term of RWG and g_i are the coefficients that represent the function in terms of Nédélec basis functions.

Figure 4.5 shows the mesh used in this numerical example. It contains three random triangles with 2 shared edges. This document only contains the approximation of one of the functions that belongs to the RWG vectorial space, but this test was performed with several functions of that space getting similar results.

Table 4.2: Global coefficients for the approximation with RWG

Coefficients	Value
edge 1 (α_1)	-3.000
edge 2 (α_2)	12.000
edge 3 (α_3)	12.000
edge 4 (α_4)	5.9477
edge 5 (α_5)	4.0619
edge 6 (α_6)	9.7983
edge 7 (α_7)	9.9420

The coefficients of the RWG basis function to approximate the function $\{x, y\}$ are calculated using the definition of degrees of freedom of RWG:

$$\alpha_i = \int_{edge} (\mathbf{u} \times \hat{\mathbf{n}})\, \hat{\tau} q \, dl = \int_{edge} (\mathbf{u} \cdot \hat{\mathbf{n}}_\tau)\, q \; dl \tag{4.23}$$

This integral is evaluated in every edge of the mesh obtaining the α coefficients shown in table 4.2. The error of the approximation using the previous coefficients for the function $\{x, y\}$ is 4.440e-16. This error may be considered as a numerical zero error approximation.

The coefficients that represent the RWG div-conforming basis functions into the Nédélec triangular curl-conforming basis functions are calculated for each triangle. Later those coefficients are globally assembled obtaining the global $[\alpha]$ matrix used to calculate the coefficients that will approximate the vectorial function by using the Nédélec triangular curl-conforming basis functions. In the case of this numerical example, the $[\alpha]$ matrix of each triangle has been calculate by using the definition of degrees of freedom of Nédélec. Tables 4.3−4.5 gathers the coefficients of those matrices.

Once the matrix has been assembled, the RWG coefficients shown in table 4.2 are multiplied by the global assembled matrix obtaining the represented coefficients in terms of the Nédélec basis functions. The error of the approximation for the function $\{-y, x\}$ is 3.1405e-016. This error indicates that the approach employed to represent the RWG basis function into Nédélec triangular curl-conforming basis functions is correct.

Table 4.3: α coefficients for the first element of the mesh

	edge 1 $(\mathbf{w}_1^{\mathtt{RWG}})$	**edge 2** $(\mathbf{w}_2^{\mathtt{RWG}})$	**edge 3** $(\mathbf{w}_3^{\mathtt{RWG}})$
edge 1 $(\mathbf{W}_1^{\mathtt{NED}})$	0.5000	0.0000	0.0000
edge 1 $(\mathbf{W}_2^{\mathtt{NED}})$	0.5000	0.0000	0.0000
edge 2 $(\mathbf{W}_3^{\mathtt{NED}})$	0.0000	0.5000	0.0000
edge 2 $(\mathbf{W}_4^{\mathtt{NED}})$	0.0000	0.5000	0.0000
edge 3 $(\mathbf{W}_5^{\mathtt{NED}})$	0.0000	0.0000	0.5000
edge 3 $(\mathbf{W}_6^{\mathtt{NED}})$	0.0000	0.0000	0.5000
face 1 $(\mathbf{W}_7^{\mathtt{NED}})$	-1.1666	1.3333	-0.1666
face 1 $(\mathbf{W}_8^{\mathtt{NED}})$	-0.5833	-0.8333	1.4166

Table 4.4: α coefficients for the second element of the mesh

	edge 4 $(\mathbf{w}_4^{\mathtt{RWG}})$	**edge 5** $(\mathbf{w}_5^{\mathtt{RWG}})$	**edge 1** $(\mathbf{w}_1^{\mathtt{RWG}})$
edge 4 $(\mathbf{W}_9^{\mathtt{NED}})$	0.5000	0.0000	0.0000
edge 4 $(\mathbf{W}_{10}^{\mathtt{NED}})$	0.5000	0.0000	0.0000
edge 5 $(\mathbf{W}_{11}^{\mathtt{NED}})$	0.0000	0.5000	0.0000
edge 5 $(\mathbf{W}_{12}^{\mathtt{NED}})$	0.0000	0.5000	0.0000
edge 1 $(\mathbf{W}_2^{\mathtt{NED}})$	0.0000	0.0000	0.5000
edge 1 $(\mathbf{W}_1^{\mathtt{NED}})$	0.0000	0.0000	0.5000
face 2 $(\mathbf{W}_{13}^{\mathtt{NED}})$	-1.2685	0.2315	1.0371
face 2 $(\mathbf{W}_{14}^{\mathtt{NED}})$	-1.1801	-1.0700	-0.1101

Table 4.5: α coefficients for the third element of the mesh

	edge 6 ($\mathbf{w}_6^{\mathtt{RWG}}$)	**edge 7** ($\mathbf{w}_7^{\mathtt{RWG}}$)	**edge 5** ($\mathbf{w}_5^{\mathtt{RWG}}$)
edge 6 ($\mathbf{W}_{15}^{\mathtt{NED}}$)	0.5000	0.0000	0.0000
edge 6 ($\mathbf{W}_{16}^{\mathtt{NED}}$)	0.5000	0.0000	0.0000
edge 7 ($\mathbf{W}_{17}^{\mathtt{NED}}$)	0.0000	0.5000	0.0000
edge 7 ($\mathbf{W}_{18}^{\mathtt{NED}}$)	0.0000	0.5000	0.0000
edge 5 ($\mathbf{W}_{12}^{\mathtt{NED}}$)	0.0000	0.0000	0.5000
edge 5 ($\mathbf{W}_{11}^{\mathtt{NED}}$)	0.0000	0.0000	0.5000
face 3 ($\mathbf{W}_{19}^{\mathtt{NED}}$)	-1.3630	0.9426	0.4204
face 3 ($\mathbf{W}_{20}^{\mathtt{NED}}$)	0.1960	-1.0941	0.8981

4.3.4 Projection of Nedelec space into RWG space

The objective in this case is to calculate the coefficients that projects the Nédélec triangular curl-conforming basis functions into the RWG basis functions. For that, the following equation must be solved finding the value of the unknowns β_i

$$\mathbf{W}_j \times \hat{\mathbf{n}} = \sum_{i=1}^{3} \beta_i \left(\mathbf{w}_i\right) \qquad \texttt{for each} \quad \mathbf{W}_{j=1\ldots 8} \tag{4.24}$$

where $\mathbf{W}_i$ are the Nédélec function and $\mathbf{w}_j$ are the RWG functions.

As the previous case, there are different approaches that can be used to solve this problem such as, solving a Galerkin system of equations of using the definition of degrees of freedom of Nedelec. The main problem, in this case, is that all the approaches provide different matrix β. The Nedelec space is not contained in the RWG space, so there are different ways to perform the projection. The different matrices for the reference triangle are shown in tables 4.6−4.8.

The first two β matrices provide a non-local projection because the interior functions $W_{7,8}^{NED}$ give non-null trace on the RWG space elements. The last matrix provides a correct local projection as the matrix α. Although, the first two matrices β provide a non-local projection they cannot be discarded a priori before calculating the error given by all the projections.

Table 4.6: β coefficients obtained by using Galerkin Method

	$\mathbf{W}_1^{\texttt{NED}}$	$\mathbf{W}_2^{\texttt{NED}}$	$\mathbf{W}_3^{\texttt{NED}}$	$\mathbf{W}_4^{\texttt{NED}}$	$\mathbf{W}_5^{\texttt{NED}}$	$\mathbf{W}_6^{\texttt{NED}}$	$\mathbf{W}_7^{\texttt{NED}}$	$\mathbf{W}_8^{\texttt{NED}}$
$\mathbf{w}_1^{\texttt{RWG}}$	1/6	7/6	-1/3	-1/3	-7/6	-1/6	0.0	2.0
$\mathbf{w}_2^{\texttt{RWG}}$	-4/15	-28/15	8/15	8/15	28/15	4/15	12/5	22/5
$\mathbf{w}_3^{\texttt{RWG}}$	-2/15	28/30	4/15	4/15	28/30	2/15	6/5	4/5

Table 4.7: β coefficients obtained by using transpose α matrix

	$\mathbf{W}_1^{\texttt{NED}}$	$\mathbf{W}_2^{\texttt{NED}}$	$\mathbf{W}_3^{\texttt{NED}}$	$\mathbf{W}_4^{\texttt{NED}}$	$\mathbf{W}_5^{\texttt{NED}}$	$\mathbf{W}_6^{\texttt{NED}}$	$\mathbf{W}_7^{\texttt{NED}}$	$\mathbf{W}_8^{\texttt{NED}}$
$\mathbf{w}_1^{\texttt{RWG}}$	1/2	1/2	0.0	0.0	0.0	0.0	-1/3	-1/6
$\mathbf{w}_2^{\texttt{RWG}}$	0.0	0.0	1/2	1/2	0.0	0.0	1/6	-1/6
$\mathbf{w}_3^{\texttt{RWG}}$	0.0	0.0	0.0	0.0	1/2	1/2	1/3	1/6

Table 4.8: β coefficients obtained by using the definition of $dofs$

	$\mathbf{W}_1^{\texttt{NED}}$	$\mathbf{W}_2^{\texttt{NED}}$	$\mathbf{W}_3^{\texttt{NED}}$	$\mathbf{W}_4^{\texttt{NED}}$	$\mathbf{W}_5^{\texttt{NED}}$	$\mathbf{W}_6^{\texttt{NED}}$	$\mathbf{W}_7^{\texttt{NED}}$	$\mathbf{W}_8^{\texttt{NED}}$
$\mathbf{w}_1^{\texttt{RWG}}$	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
$\mathbf{w}_2^{\texttt{RWG}}$	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
$\mathbf{w}_3^{\texttt{RWG}}$	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0

4.4 Conclusions

This chapter has presented a modular approach to combine MoM and FEM techniques for the analysis of large structures or finite arrays with complex radiating elements. The approach have consisted of performing the FEM analysis of each structure and to define the equivalent electric and magnetic currents on the boundaries of the structure. Then, the interaction between the structures is computed by using MoM and the global FEM-MoM system of equation is assembled. Finally, the system is solved and the analysis is completed. It is worth to note that the approach has not been totally implemented but some previous works have been done.

One of these works has been the modification of the current variational formulation of the FEM module of the suite. This modification has been to provide unknowns for the corresponding dual field (H-Field or E-Field) in an explicit way. The standard variational formulation of FEM problems only has unknowns for one of the corresponding electromagnetic fields (E-Field or H-Field) and the previous modification is required in order to connect both FEM and MoM formulation.

Furthermore, a study of the projection between the basis functions of both modules has been done in order to connect the unknowns between both codes. The boundaries of the antenna is where both methods share the unknowns and those need to be expressed in terms of the same basis functions. In the particular case of this Ph. D. dissertation, a study between the well-known Rao-Wilton-Glisson (RWG) div-conforming basis functions and the the Nédélec triangular curl-conforming basis functions of second-order has been performed.

CHAPTER 5

ANALYSIS OF INFINITE PERIODIC STRUCTURES

The electromagnetic analysis of periodic structures is of great importance in modern radar and communication systems. Accurate prediction of structure performance using numerical methods not only reduces the development cost and design timeline but also provides invaluable physical insight to design engineers [6].

The main problem of using numerical methods for computations involving large finite structures is that the use of higher working frequencies of modern radars and communication systems makes the computation, despite the constant enhancements in computer power, a challenge, due to the large electrical sizes of these structures.

One way to approach the electromagnetic analysis of large arrays is to solve the full problem using a pure numerical technique, such as, the Method of Moment (`MoM`) [60] or the Finite Element Method (`FEM`) [?]. It is worth to note that acceleration techniques, such as, the Fast Multipole Method (`FMM`) [?] or the Adaptive Integral Method (`AIM`) [?] are also widely used in the analysis for large structure. However, since these methods try to solve the full problem, their memory requirements are prohibitive and make the simulation a great challenge.

Another family of approaches is to make use of the infinite structure analysis, and apply post-processing techniques to correct the border effects of the structure. This analysis helps to understand the behavior of elements in the central region of an electrically large array, since they have similar active impedance characteristics as that of an element in an infinite array. Also, the infinite array results are applied to predict the mutual coupling between the elements in an array environment or the embedded element pattern that includes mutual coupling effects can be determined directly from the infinite array results. Thus, although they are not a physically realistic problem, the infinite array analysis provides a reasonable good approximation with a less computing requirement than the analysis of the full problem. Furthermore, fast techniques approaches such as Macro Basis Functions (`MBFs`) use the infinite array solution as the basis brick for their approaches.

In this chapter, the analysis of structures that are infinitely periodic in the two-dimensional xy-plane is carried out. These structures can be analyzed using a single unit cell where the field at one periodic surface is related to the field at the opposite parallel surface through a simple phase shift. This approach is based on the use of the so-called Periodic Boundary Conditions (`PBCs`). Details about these boundary conditions and its implementation on `FEM` are given in Section 5.1.

In addition to the use of the `PBCs`, a truncation of the computational domain along the non-periodic direction (the z-direction in this case) is needed to perform the analysis for open scattering and radiation problems. Thus, the modification of the truncation techniques implemented in the `FEM` module is required in order to support the analysis of infinite structures. Details about this modification are discussed in Section 5.2.

Finally, the analysis of some real structures is performed in Section 5.3. There, examples of the performance and the capabilities of the suite solving infinite periodic structures are provided.

5.1 Periodic Boundary Condition on FEM

As commented in the previous introduction, the analysis of infinite periodic structures can be reduced to a single unit cell by applying appropriate periodic boundary conditions (`PBCs`). For simplicity, the finite

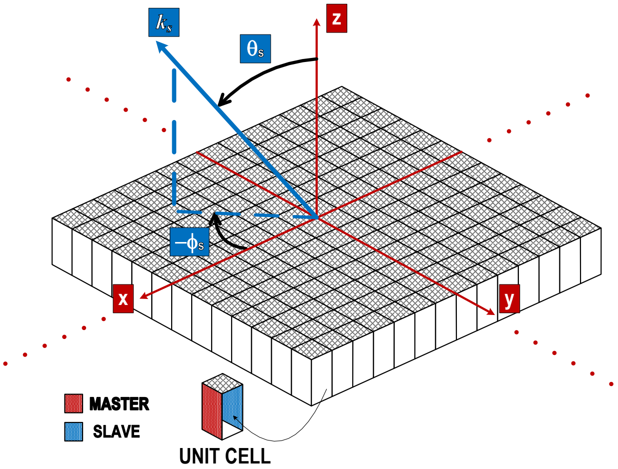


Figure 5.1: Infinite periodic array in the xy-plane that extends to infinity in four directions.

element analysis consider in this section is performed on infinite periodic structures located in two-dimensional xy-plane as the one shown in figure 5.1. The unit cell is also shown in the figure.

It is worth to note that is convenient that opposing surfaces in the unit cell have identical surface meshes since, if these meshes are not identical, non-conformal variational formulations or projections over a common surface grid are required in order to analyze the infinite problem. The use of these techniques imply the modification of the `FEM` module with the consequence extra developments. In this particular case, unit cells with hexahedrical shapes and identical surface meshes for the opposing surfaces are considered. With such a volumetric mesh, the infinite periodic structure can be generated repeating the unit cell mesh in the x- and y- directions. The opposite surfaces are related by pairs where one surface is called *"master"* and the opposite is called *"slave"*. Figure 5.1 shows an example of the master and the slave surfaces of a unit cell. Thus, a identical distribution of the *dofs* is obtained between a master and its corresponding slave surface, and also, between the surfaces of adjacent unit cells. Figure 5.2 shows a two dimensional cut of a unit cell where identical meshes for the master surface $\mathcal{X}_1$ and slave surface $\mathcal{X}_2$ may be appreciated.

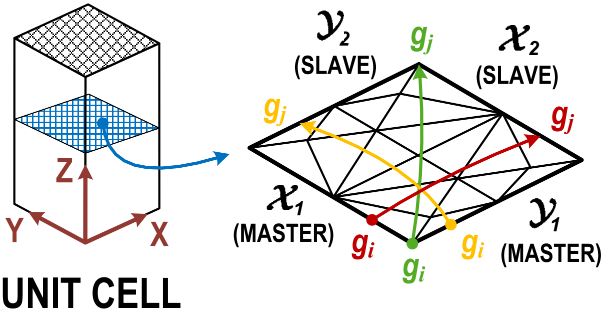


Figure 5.2: Two-dimensional finite element mesh for a unit cell

According to the Floquet theorem [61], if g_i is a *dof* associated to the master boundary $\mathcal{X}_1$, the corresponding *dof* g_j on the slave boundary $\mathcal{X}_2$ (red relationship between *dofs* on figure 5.2) is given by

$$g_j = g_i \, e^{-j\Psi_{\mathrm{x}}} \qquad \texttt{with} \qquad \Psi_{\mathrm{x}} = k_x \, \mathtt{D}_{\mathrm{x}} \tag{5.1}$$

where $\mathtt{D}_{\mathrm{x}}$ is the unit cell spacing in the x- and k_x is the corresponding propagation vector with (θ_s, ϕ_s) being the scan angle of the structure.

$$k_x = k_0 \, \mathtt{sin} \, \theta_s \mathtt{cos} \, \phi_s \tag{5.2}$$

Conversely, if g_i is a *dof* associated to the master boundary $\mathcal{Y}_1$, the corresponding *dof* g_j associated to the slave surface $\mathcal{Y}_2$ (orange relationship between *dofs* on figure 5.2) is given by

$$g_j = g_i \, e^{-j\Psi_{\mathrm{y}}} \qquad \texttt{with} \qquad \Psi_{\mathrm{y}} = k_y \, \mathtt{D}_{\mathrm{y}} \tag{5.3}$$

where $\mathtt{D}_{\mathrm{y}}$ is the unit cell spacing in the y- and k_y is the corresponding propagation vector with (θ_s, ϕ_s) being the same scan angle as before.

$$k_y = k_0 \, \mathtt{sin} \, \theta_s \mathtt{sin} \, \phi_s \tag{5.4}$$

Finally, the expression of a *dof* g_j associated to the corner intersected by the slave boundaries $\mathcal{X}_2$ and $\mathcal{Y}_2$ may be considered an application of both previous conditions (5.1) and (5.3) at the same time over a *dof* g_i associated to the corner intersected by the master boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$ (green relationship between *dofs* on figure 5.2).

$$g_j = g_i \, e^{-j(\Psi_{\mathrm{x}} + \Psi_{\mathrm{y}})} \tag{5.5}$$

Hence, the expression of the *dofs* associated to the slave surface (g_j*'s*) is related with the *dofs* associated to the master surface (g_i*'s*) through a phase-shit term Ψ. In summary, the relationship between g_i and g_j for the different boundaries conforming a unit cell is given by

$$g_j = g_i \, e^{-j\Psi_{ij}} \tag{5.6}$$

where the phase-shift term Ψ_{ij} is given by

$$\Psi_{ij} = \begin{cases} k_x \, \mathtt{D_x} & \forall \, g_i \in \mathcal{X}_1, \; g_j \in \mathcal{X}_2 \\ k_y \, \mathtt{D_y} & \forall \, g_i \in \mathcal{Y}_1, \; g_j \in \mathcal{Y}_2 \\ k_x \, \mathtt{D_x} + k_y \, \mathtt{D_y} & \forall \, g_i \in \mathcal{X}_1 \cap \mathcal{Y}_1, \; g_j \in \mathcal{X}_2 \cap \mathcal{Y}_2 \end{cases} \tag{5.7}$$

In accordance to the formulation presented in Chapter 3, the weak-form representation of the vector wave equation is given by

$$\begin{aligned} \iiint_\Omega (\boldsymbol{\nabla} \times \mathbf{W}) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) dV - k_0^2 \iiint_\Omega \mathbf{W} \cdot \bar{\bar{g}}_r \, \mathbf{V} dV \\ = \iiint_\Omega \mathbf{W} \cdot \mathbf{q} \, dV - \oiint_\Gamma \mathbf{W} \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \end{aligned} \tag{5.8}$$

where Γ denotes surfaces that encloses Ω. The corresponding finite element matrix equation can be written as

$$[K] \; \{g\} = \{b\} + \{h\} \tag{5.9}$$

where the ij-th coefficient of the $[K]$ matrix is given by

$$K_{ij} = \iiint_\Omega (\boldsymbol{\nabla} \times \mathbf{W}_i) \cdot \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}_j \right) dV - k_0^2 \iiint_\Omega \mathbf{W}_i \cdot \bar{\bar{g}}_r \mathbf{V}_j \, dV \tag{5.10}$$

and the i-th coefficient of the right hand side is given by

$$b_i = \iiint_\Omega \mathbf{W}_i \cdot \mathbf{q} \;\; dV, \quad h_i = - \oiint_\Gamma \mathbf{W}_i \cdot \left[\hat{\mathbf{n}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \tag{5.11}$$

It is worth noting that the surfaces belonging to the periodic directions do not have a known boundary condition and the surface integral of the previous equation (h_i) must be treated. The approach used in this thesis consists of the following.

Lets consider a periodic structure with a unit cell as the one shown in figure 5.2. If g_i resides inside the unit cell, its basis function has no tangential component on any side surfaces. Thus, the value of the integral h_i for all the *dofs* inside the unit cell is zero. Now considering a g_i with a basis function that has a tangential component on the surface $\mathcal{X}_1$, the matrix equation can be written as

$$\sum_{l=1}^{N} K_{il}\, g_l = b_i + h_i \qquad \forall\, g_i \in \mathcal{X}_1 \tag{5.12}$$

where

$$h_i = \oiint_{\mathcal{X}_1} \mathbf{W}_i \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \tag{5.13}$$

since $\hat{\mathbf{n}} = -\hat{\mathbf{x}}$ on $\mathcal{X}_1$. As the opposing surfaces have identical surface meshes, the matrix equation corresponding to a g_j with basis function that has a tangential component only on $\mathcal{X}_2$ can be written as

$$\sum_{l=1}^{N} K_{jl}\, g_l = b_j + h_j \qquad \forall\, g_j \in \mathcal{X}_2 \tag{5.14}$$

where

$$h_j = -\oiint_{\mathcal{X}_2} \mathbf{W}_j \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \tag{5.15}$$

Both h_i and h_j are unknown terms. However, because of the periodic condition for the *dofs* given in equation (5.1), the relationship between the integrals h_i and h_j can be defined as

$$h_j = -h_i\, e^{-j\Psi_\mathrm{x}} \tag{5.16}$$

Hence, the surface integral h_i in equation (5.12) can be canceled by multiplying the equation (5.14) by the factor $e^{j\Psi_\mathrm{x}}$ and adding the resulting equation to (5.12). This yields a new matrix equation for the g_i belonging to $\mathcal{X}_1$ as

$$\sum_{l=1}^{N} \left[K_{il} + K_{jl}\, e^{j\Psi_\mathrm{x}}\right] g_l = b_i + b_j\, e^{j\Psi_\mathrm{x}} \qquad \forall\, g_i \in \mathcal{X}_1 \tag{5.17}$$

The treatment for a g_i where its basis function has a tangential component only on $\mathcal{Y}_1$ is the same than the previous case, yielding the matrix equation

$$\sum_{l=1}^{N} \left[K_{il} + K_{jl}\, e^{j\Psi_{\mathrm{y}}}\right] g_l = b_i + b_j\, e^{j\Psi_{\mathrm{y}}} \qquad \forall\, g_i \in \mathcal{Y}_1 \tag{5.18}$$

It is interesting to look at a g_i associated to the corner edge intersected by the boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$, whose basis function has a tangential component on both $\mathcal{X}_1$ and $\mathcal{Y}_1$. In this case the equation is given by

$$\sum_{l=1}^{N} K_{il}\, g_l = b_i + h_i \qquad \forall\, g_i \in \mathcal{X}_1 \cap \mathcal{Y}_1 \tag{5.19}$$

where

$$\begin{aligned} h_i = h_i^{(x)} + h_i^{(y)} = & \oiint_{\mathcal{X}_1} \mathbf{W}_i \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \\ & + \oiint_{\mathcal{Y}_1} \mathbf{W}_i \cdot \left[\hat{\mathbf{y}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \end{aligned} \tag{5.20}$$

This unknown term h_i can be eliminated by the equations for the corresponding h's terms of the three other corner edges of the unit cell. For the corresponding g_m associated to the corner edge intersected by the boundaries $\mathcal{Y}_1$ and $\mathcal{X}_2$, its equation is given by

$$\sum_{l=1}^{N} K_{ml}\, g_l = b_m + h_m \qquad \forall\, g_m \in \mathcal{Y}_1 \cap \mathcal{X}_2 \tag{5.21}$$

with

$$\begin{aligned} h_m = h_m^{(x)} + h_m^{(y)} = & -\oiint_{\mathcal{X}_2} \mathbf{W}_m \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \\ & + \oiint_{\mathcal{Y}_1} \mathbf{W}_m \cdot \left[\hat{\mathbf{y}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V}\right)\right] dS \end{aligned} \tag{5.22}$$

For the corresponding g_n associated to the corner edge intersected by the boundaries $\mathcal{X}_1$ and $\mathcal{Y}_2$, its equation is given by

$$\sum_{l=1}^{N} K_{nl}\, g_l = b_n + h_n \qquad \forall\, g_n \in \mathcal{X}_1 \cap \mathcal{Y}_2 \tag{5.23}$$

with

$$\begin{aligned} h_n = h_n^{(x)} + h_n^{(y)} = & \oiint_{\mathcal{X}_1} \mathbf{W}_n \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \\ & - \oiint_{\mathcal{Y}_2} \mathbf{W}_n \cdot \left[\hat{\mathbf{y}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \end{aligned} \tag{5.24}$$

Finally, for the corresponding g_j associated to the corner edge intersected by the boundaries $\mathcal{X}_2$ and $\mathcal{Y}_2$, its equation is given by

$$\sum_{l=1}^{N} K_{jl}\, g_l = b_j + h_j \qquad \forall\, g_j \in \mathcal{X}_2 \cap \mathcal{Y}_2 \tag{5.25}$$

with

$$\begin{aligned} h_j = h_j^{(x)} + h_j^{(y)} = & - \oiint_{\mathcal{X}_2} \mathbf{W}_j \cdot \left[\hat{\mathbf{x}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \\ & - \oiint_{\mathcal{Y}_2} \mathbf{W}_j \cdot \left[\hat{\mathbf{y}} \times \left(\bar{\bar{f}}_r^{\,-1} \boldsymbol{\nabla} \times \mathbf{V} \right) \right] dS \end{aligned} \tag{5.26}$$

Using the periodic boundary condition for the *dofs* given in equation (5.6), the relation between the surface integrals is given by

$$h_m^{(x)} = -h_i^{(x)} e^{-j\Psi_{\mathrm{x}}}, \qquad h_n^{(y)} = -h_i^{(y)} e^{-j\Psi_{\mathrm{y}}} \tag{5.27}$$

$$h_j^{(y)} = -h_m^{(y)} e^{-j\Psi_{\mathrm{y}}}, \qquad h_j^{(x)} = -h_n^{(x)} e^{-j\Psi_{\mathrm{x}}} \tag{5.28}$$

Thus, the equation of a g_i associated to the corner edge intersected by the boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$ can be found multiplying the equation (5.21) by the factor $e^{j\Psi_{\mathrm{x}}}$, the equation (5.23) by the factor $e^{j\Psi_{\mathrm{y}}}$, the equation (5.25) by $e^{j(\Psi_{\mathrm{x}}+\Psi_{\mathrm{y}})}$ and adding the resulting equations to (5.19), yielding

$$\sum_{l=1}^{N} \left[K_{il} + K_{ml}\, e^{j\Psi_x} + K_{nl}\, e^{j\Psi_y} + K_{jl}\, e^{j(\Psi_x+\Psi_y)} \right] g_l$$
$$= b_i + b_m e^{j\Psi_x} + b_n e^{j\Psi_y} + b_j e^{j(\Psi_x+\Psi_y)} \qquad \forall\, g_i \in \mathcal{X}_1 \cap \mathcal{Y}_1 \quad (5.29)$$

The previous equations (5.17), (5.18) and (5.29) show how the system of equations is modified in order to eliminate the unknown terms h_i. This can be interpreted as the connection of the *dof* associated to the master surfaces with the corresponding *dof* associated to the slave surfaces. The periodic boundary conditions (PBCs), given by equation (5.6), have not been imposed at this point yet. The imposition is performed by substituting the original equations of the *dof* associated to the slave surfaces by equation (5.6).

Although the previous modifications seem quite complicated, the numerical implementation is straightforward. The following section describes how the PBCs may be incorporated in numerical codes as, for instance, the FEM module of the suite.

5.1.1 Numerical implementation

As mentioned above, the numerical implementation of PBCs on FEM is limited to a slight modification on the system of equations of the problem. The algorithm to impose the PBCs on FEM is as follows:

- **Step 1:** The sparse system of equations given by the variational formulation described in Section 3.1 is computed. At this step, no considerations about PBCs are taken and the standard FEM formulation is used. Figure 5.3 shows a system of equations where the coefficients regarding to the i-th *dof* (associated to the master boundary $\mathcal{X}_1$) are marked in red. The coefficients corresponding to the j-th *dof* (associated to the slave boundary $\mathcal{X}_2$) are marked in blue.

- **Step 2:** Once the system of equations is obtained, the modification over the equations of the *dofs* associated to the master boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$ is applied. For helping the reader to follow this step, lets focus on the modification of the *dofs* associated to the boundary $\mathcal{X}_1$ first. For simplicity, the matrix equation corresponding to those *dofs* is written here again

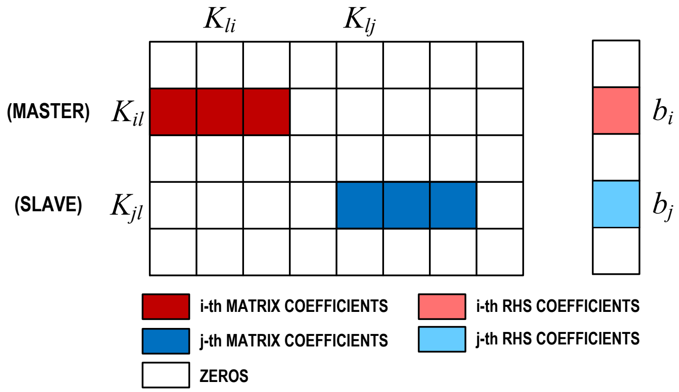


Figure 5.3: Equations of two opposite *dofs* before applying the PBCs

$$\sum_{l=1}^{N} \left[K_{il} + K_{jl}\, e^{j\Psi_{\mathrm{x}}}\right] g_l = b_i + b_j\, e^{j\Psi_{\mathrm{x}}} \qquad \forall\, g_i \in \mathcal{X}_1 \tag{5.30}$$

where the i-index corresponds to the coefficients of the i-th *dof* associated to the master surface $\mathcal{X}_1$ and the j-index corresponds to the coefficients of the opposite *dof* associated to the slave surface $\mathcal{X}_2$.

It is worth noting that those coefficients are already computed in the previous step of the algorithm (marked in red/blue in figure 5.3, respectively). Thus, the modification of the system of equation consists of a displacement of the coefficients of the j-th equation into the i-th equation but multiplied by the corresponding phase-shift term. In the particular case considered first, the phase-shift between the *dofs* of $\mathcal{X}_1$ and $\mathcal{X}_2$ is $e^{j\Psi_{\mathrm{x}}}$. Figure 5.4 shows the displacement of the coefficients of the j-th equation (marked in blue/brown in the figure) into the i-th equation. The coefficients marked in green/purple in the i-th equation are already multiplied by the corresponding phase-shit term.

After the displacement is performed, the j-th equation is completely removed from the system. Figure 5.5 shows the state of the system of equations at this point of the algorithm. See how all the coefficients corresponding to the j-th equation have been removed from the system.

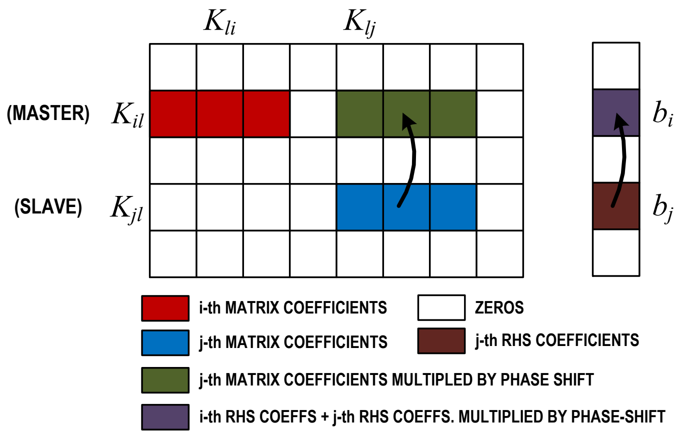


Figure 5.4: Modification of the i-th equation of the system

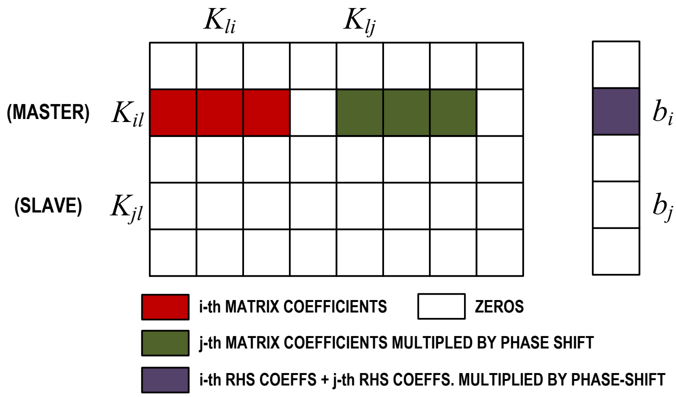

K_{li}
K_{lj}
(MASTER) K_{il}
(SLAVE) K_{jl}
b_i
b_j
i-th MATRIX COEFFICIENTS
ZEROS
j-th MATRIX COEFFICIENTS MULTIPLED BY PHASE SHIFT
i-th RHS COEFFS + j-th RHS COEFFS. MULTIPLIED BY PHASE-SHIFT

Figure 5.5: j-th equation is removed from the system

The methodology employed to modify the equations of the *dofs* associated to the master boundary $\mathcal{Y}_1$ is exactly the same as previously. In this case, the modification of the system of equations is performed using the matrix equation (5.18), instead of equation (5.17). Furthermore, the corresponding phase-shift between the *dofs* of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ is $e^{j\Psi_{\mathrm{y}}}$. Figure 5.4 may also be used to show the displacement of the coefficients of the j-th equation (marked in blue/brown in the figure) into the i-th equation for this case.

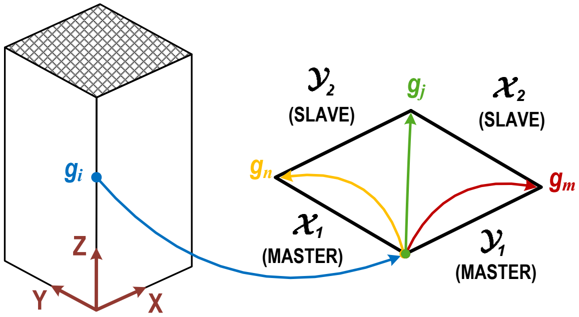


Figure 5.6: Relationships of *dofs* associated to the corner intersected by the master boundaries

- **Step 3:** The modification of the equations associated to *dofs* residing on the corner intersected by the master boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$ is a slightly different than the previous cases, since three displacements are needed instead of one. Note that the *dofs* associated to the corner intersected by these master boundaries have relationships with the two slave boundaries ($\mathcal{X}_2$ and $\mathcal{Y}_2$) and with the corner intersected by those slave boundaries (see figure 5.6).

  The corresponding matrix equation governing the modification of the system is given by equation (5.29). Figure 5.7 shows the displacements of the coefficients of the j-th, m-th and n-th equations into the i-th equation for this particular case. Each displacement is multiplied by the corresponding phase-shift term.

- **Step 4:** As commented before, once all the displacements have been performed, the equations associated to all the slave *dofs*, regardless of the boundary where they reside, are completely removed from the system of equations. Figure 5.8 illustrates an example of the state of the system of equations at this point of the algorithm.

  To complete the numerical implementation of the `PBCs` on `FEM`, the periodic relationship between the slave and the master *dofs* described according the Floquet theorem (see page 118) is added to the system of equations. For simplicity, the equation corresponding to the relationship of the *dofs* associated to the boundaries $\mathcal{X}_1$ and $\mathcal{X}_2$ is written here again

$$g_j = g_i\, e^{-j\Psi_{\mathrm{x}}} \qquad \texttt{with} \qquad \Psi_{\mathrm{x}} = k_x\, \mathtt{D}_{\mathrm{x}} \tag{5.31}$$

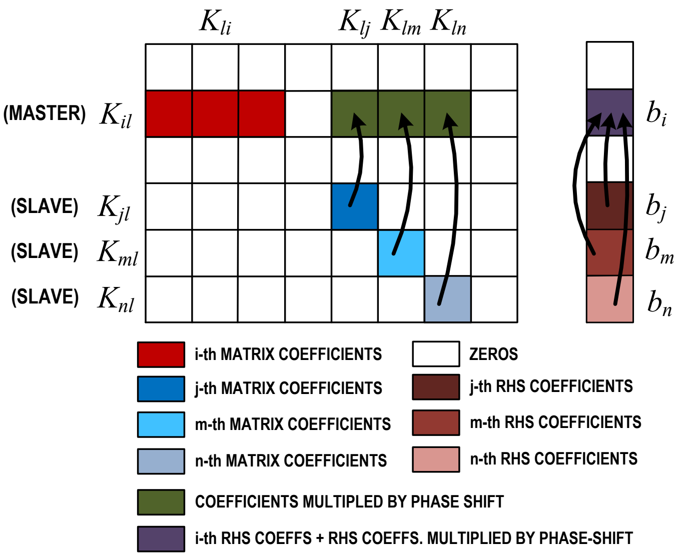


Figure 5.7: Modification of the system of equations for *dofs* residing on the corner intersected by the master boundaries

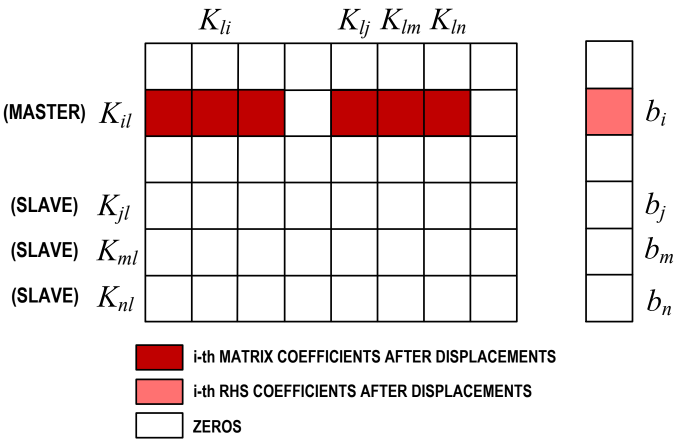


Figure 5.8: State of the system of equation

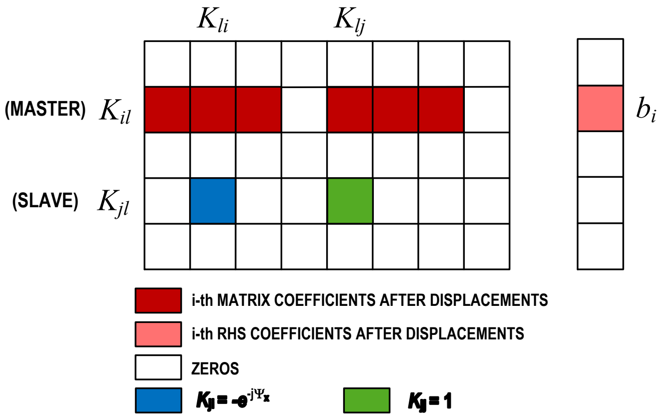


Figure 5.9: Adding periodic relationship between the slave and the master *dofs* to the system of equations

where g_i is the i-th *dof* located on the master surface $\mathcal{X}_1$ and g_j is the opposing *dof* located on the slave surface $\mathcal{X}_2$. The periodic relationship between the *dofs* residing on the boundaries $\mathcal{Y}_1$ and $\mathcal{Y}_2$ is governing by equation (5.3), while the relationship between the *dofs* residing on the corner intersected by the master boundaries $\mathcal{X}_1$ and $\mathcal{Y}_1$ and the corner intersected by the slave boundaries $\mathcal{X}_2$ and $\mathcal{Y}_2$ is given by equation (5.5). Figure 5.9 shows an example about how equation (5.31) is added to the system of equations.

- **Step 5:** Once the addition of the new set of equations is complete, the application of the `PBCs` over the structure is finished and the system of equation is ready to be solved by the sparse solver.

The algorithm presented above can be used for the implementation of the `PBCs` on any numerical code based on `FEM`. Particularizing to the `FEM` module of the suite, the algorithm has been easily implemented using an efficient management of the memory. Typically, sparse solvers store the system of equations by using four large vectors that contain the coefficients of the matrix (vector `A`), the corresponding rows of these coefficients (vector `I`), the corresponding columns (vector `J`) and the coefficients of the right hand side (vector `RHS`). Furthermore, sparse solvers automatically assemble the coefficients of the system of equations by adding the corresponding values of the vector `A` which have the same

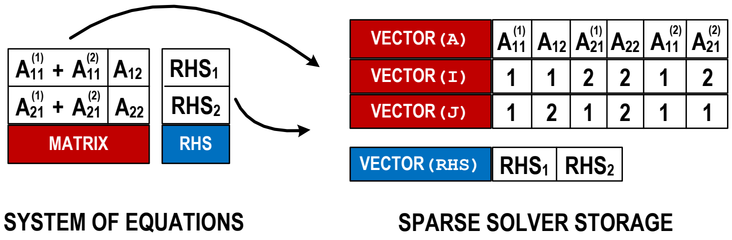


Figure 5.10: Internal storage of sparse solver for a system of equations

rows and columns in vectors I and J. Figure 5.10 illustrates the internal storage of a simple system of equations in the four vectors previously commented. Note how the coefficient $\mathtt{A}_{11}$ of the matrix is stored in two different locations of the vector A (first and fifth), while the vectors I and J contains the same values for those locations. Subsequently, the sparse solver assembles them internally.

Thereby, the displacement of the matrix coefficients of a j-th equation into a i-th equation of the system (steps 2−3) is performed with a slight change in the vector I that contains the rows of the matrix. In other words, the row contained in the vector I for all the matrix coefficients associated to the j-th equation (see vector in purple in figure 5.11) is changed by the row of the i-th equation (see vector in orange in figure 5.11). In addition, the corresponding matrix coefficients of the j-th equation, stored in the vector A, are multiplied by the phase-shift. Then, the displacement is complete. Regarding to the right hand side of the system, the coefficients of the j-th equation, stored in the vector RHS, are multiplied by the phase-shift and added manually to the coefficient of the i-th equation. Finally, the new set of equations corresponding to the step 4 of the algorithm is added to the vectors A, I and J. Thus, the application of the PBCs in the FEM module is completed.

5.1.2 Verification tests

The accuracy of the results given by a computer code, specially when new features and capabilities are implemented, must be verified before the code is ready for research use. In this case, the Method of Manufactured Solutions (MMS) has been employed to provide evidence that the numerical implementation of the PBCs inside the FEM module has been correctly performed. Further details about MMS are given in Section 3.5.

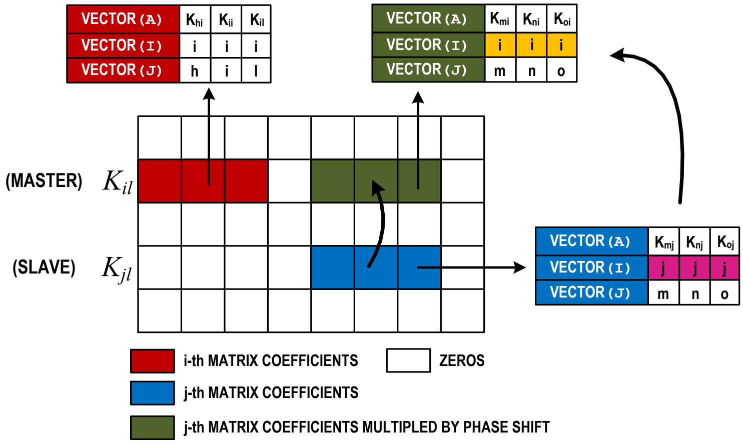


Figure 5.11: Modification of the *i*-th equation of the system using the internal data of sparse solvers

A complex exponential function is used as manufactured solution to compute the relative/maximum errors given by the code when `PBCs` are applied. It is worth to note that, in the case of `PBCs`, the manufactured solution must satisfy also the periodic condition for the *dofs* given in equation (5.6). Thus, complex exponential functions become a nice choice of smooth solution to perform this validation test. The expression of the function used as manufactured solution is given by

$$\mathbf{E}_{\text{analytic}} = \mathbf{E}_{\text{pol}}\, e^{-jk_0(\mathbf{k}_p\cdot\mathbf{r})} \tag{5.32}$$

where $\mathbf{E}_{\text{pol}}$ is the polarization vector, $\mathbf{k}_p$ is the propagation vector and $\mathbf{r}$ is the vector with the real coordinates where to evaluate the function. The exponential function has been polarized in both θ and ϕ components and the incident angles are $\phi = 30^\circ$ and $\theta = 20^\circ$. Thereby, the scan angles that determinate the phase shift between the corresponding pairs of *dofs* are $\phi_s = 210^\circ$ and $\theta_s = 20^\circ$.

The structure used for these validation tests consists of a cube where the surfaces perpendicular to the x- and y-directions are marked with `PBCs` (see the boundaries on figure 5.12). Then, the cube may be considered as the unit cell of a infinite periodic structure where the periodicity directions are x- and y-, respectively.

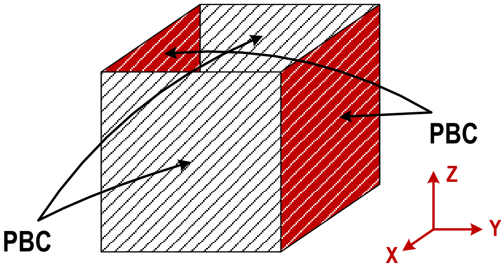


Figure 5.12: Structure used for the verification test

Table 5.1 gathers the relative/maximum errors for different mesh discretization of the cube. Both relative and maximum errors decay as the number of element is increased. This behavior is exactly the expected providing evidences that the implementation of the `PBCs` in the `FEM` module is correct. Figure 5.13 shows the convergence rate of the error when the number of unknowns of the problem increases. The relative errors decay approximately with a slope of -0.68 and the maximum errors with a slope of -0.54. It is worth to note that the theoretical slope of the convergence rate of the relative error for this test $(-2/3)$ matches exactly with the results given by the `FEM` module (see Chapter 3). Details about this theoretical slope are provided in [7].

Figure 5.14(a) shows the imaginary part of the x-component of the analytic solution `imag-x` $\left(\mathbf{V}_{\texttt{analytic}}\right)$ for the finest mesh discretization of the cube. Figure 5.14(b) shows the solution given by the `FEM` module for the same component `imag-x` $(\mathbf{V}_{\texttt{FEM}})$. The error between both solutions is shown in figure 5.14(c) where an uniform and low error may be seen.

5.2 Infinite domain truncation methods on the suite

Typically, infinite periodic structures have a two-dimensional periodicity and are unbounded in the third non-periodic direction. Thus, a mesh truncation technique is needed in order to perform the simulation. In other words, if the original problem domain is infinite in the z-direction, a mesh truncation technique is needed above the unit cell and another below the unit cell (if necessary) to make the computational domain finite in the z-direction.

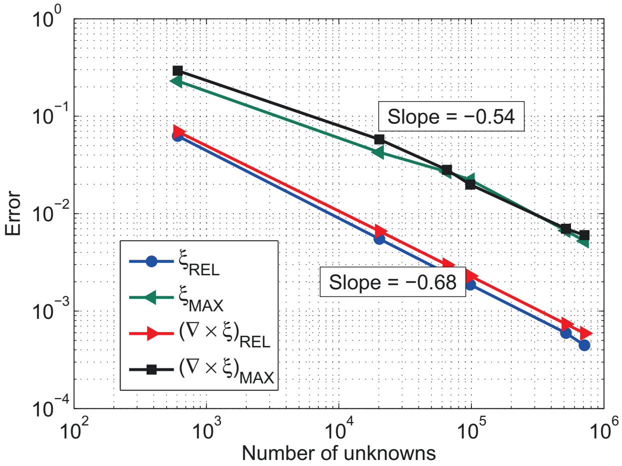


Figure 5.13: Convergence rate of the error over a cube using a complex exponential as manufactured solution

As described in Chapter 3, the `FEM` module supports a first-order absorbing boundary condition (`ABC`) or a boundary integral equation (`FE-IIEE`) as mesh truncation techniques. The standard first-order `ABC` for the upper truncation surface is given by

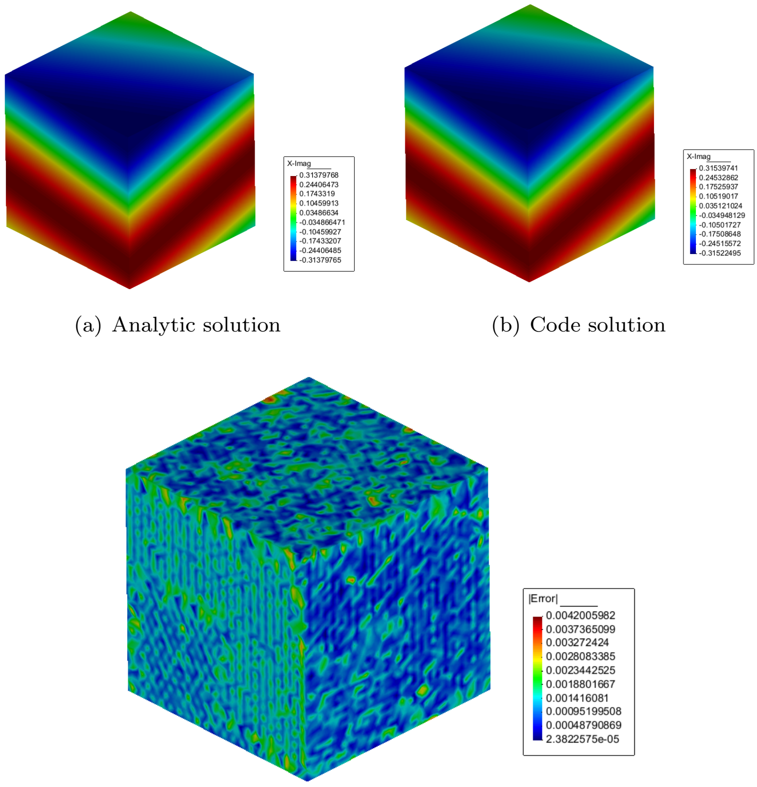
$$\hat{\mathbf{z}} \times (\boldsymbol{\nabla} \times \mathbf{V}) + jk_0 \, (\hat{\mathbf{z}} \times \hat{\mathbf{z}} \times \mathbf{V}) - \boldsymbol{\Psi}^{\mathbf{inc}} = 0 \qquad (5.33)$$

where $\boldsymbol{\Psi}^{\mathbf{inc}}$ is the result of evaluating the first two terms of the previous equation with $\mathbf{V} = \mathbf{V}^{\mathbf{inc}}$.

The implementation of this boundary condition in the finite element formulation is straightforward. However, this condition is only satisfied for waves propagating along the z-direction (i.e., only absorbs waves propagating along $\theta_s = 0$), and it has a significant reflection for waves propagating in other directions. This reflection may produce disturbances when analyzing infinite periodic structures in which the solution contains significant wave components travelling in other directions different from the z-direction. Figure 5.15 shows the residual of equation (5.33) for waves propagating along different directions ranging from $0^\circ \leq \phi_s \leq 360^\circ$ and from $0^\circ \leq \theta_s \leq 90^\circ$. The incident frequency is 100 MHz and $(0, 0, 1)$ is the observation point where equation (5.33) is evaluated. The figure shows clearly how the condition absorbs the waves

Table 5.1: Relative/maximum errors for several mesh sizes of a cube using an exponential function as manufactured solution

Tetrahedrons	**Unknowns**	ξ_{rel}	ξ_{max}	$(\nabla \times \xi)_{rel}$	$(\nabla \times \xi)_{max}$
75	608	$6.568e^{-2}$	$2.293e^{-1}$	$6.946e^{-2}$	$2.931e^{-1}$
2948	20194	$5.501e^{-3}$	$4.262e^{-2}$	$6.620e^{-3}$	$5.777e^{-2}$
9825	65454	$2.439e^{-3}$	$2.687e^{-2}$	$2.983e^{-3}$	$2.812e^{-2}$
14802	98138	$1.867e^{-3}$	$2.221e^{-2}$	$2.290e^{-3}$	$1.989e^{-2}$
79378	516466	$5.965e^{-4}$	$6.710e^{-3}$	$7.351e^{-4}$	$6.995e^{-3}$
108421	713252	$5.547e^{-4}$	$5.232e^{-3}$	$5.919e^{-4}$	$6.015e^{-3}$



(a) Analytic solution

(b) Code solution

(c) Error between analytic and code solution

Figure 5.14: 3D representation of the exponential solution

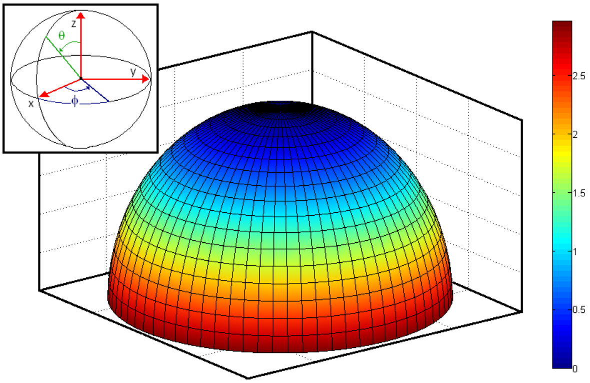


Figure 5.15: Standard first-order absorbing boundary condition values for waves propagating in different (θ_s, ϕ_s) directions

propagating along the z-direction (values in blue in the figure) and how, as the angle θ_s is increased, the condition does not absorb the incident waves (values changing from blue to red in the figure). Thus, a more appropriate boundary condition is needed to analyze infinite periodic structures using the first-order absorbing boundary conditions.

Following the procedure described in [6, Section 9.1.2], a modified absorbing boundary condition is obtained which will absorb perfectly a plane wave propagating in the (θ_s, ϕ_s) direction. This modified first-order absorbing boundary condition is given by

$$\hat{\mathbf{z}} \times (\boldsymbol{\nabla} \times \mathbf{V}) + jk_0 \cos\theta_s \, (\hat{\mathbf{z}} \times \hat{\mathbf{z}} \times \mathbf{V}) - \frac{jk_0}{\cos\theta_s} \mathbf{k}_t^s \, (\mathbf{k}_t^s \cdot \mathbf{V}) - \boldsymbol{\Psi}^{\mathbf{inc}}_{\boldsymbol{\theta_s},\boldsymbol{\phi_s}} = 0 \tag{5.34}$$

where $\mathbf{k}_t^s = \sin\theta_s \cos\phi_s \, \hat{x} + \sin\theta_s \sin\phi_s \, \hat{y}$ with (θ_s, ϕ_s) being the scan angles of the array and $\boldsymbol{\Psi}^{\mathbf{inc}}_{\boldsymbol{\theta_s},\boldsymbol{\phi_s}}$ is the result of evaluating the first three terms of equation (5.34) with $\mathbf{V} = \mathbf{V}^{\mathbf{inc}}$.

Figure 5.16 shows the residual of equation (5.34) for the waves propagating in the same directions than the previous figure. The frequency of the plane waves and the observation point where the equation is evaluated are also the same than the case of the first-order absorbing boundary condition. The figure shows how the modified condition absorbs perfectly any plane wave propagating in the (θ_s, ϕ_s) direction (except waves propagating along $\theta_s = 90°$ where a indetermination in one term of the equation is found).

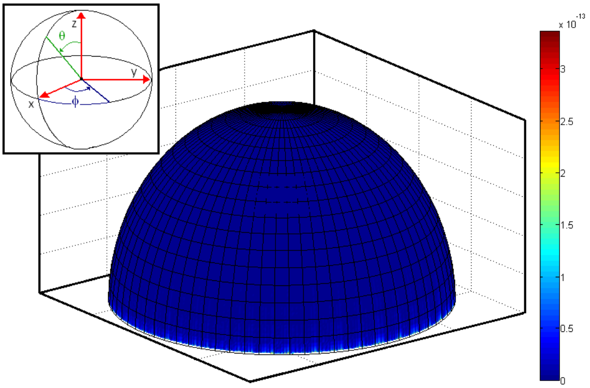


Figure 5.16: Modified first-order absorbing boundary condition values for waves propagating in different (θ_s, ϕ_s) directions

5.2.1 FE-IIEE method for infinite periodic structures

Another way supported by the `FEM` module to truncate the mesh is by using the Iterative Integral Equation Evaluation method. As it was described in Section 3.2, the exterior infinite domain is truncated by an integral equation representation of the exterior field which is calculated using the Green's function $G(r, r_s)$ in free-space[1]

$$G(\mathbf{r}, \mathbf{r}_s) = \frac{e^{-jk_0 R}}{4\pi R} \tag{5.35}$$

where $R = |\mathbf{r} - \mathbf{r}_s|$ is the distance between the source point and the observation point.

In the case of infinite periodic structures the previous Green's function must be replaced by the appropriate periodic Green's function $G_p(\mathbf{r}, \mathbf{r}_s)$ in free space. Assuming a periodic structure in the xy-plane (as the one shown in figure 5.1), the (m, n) cell of the structure is obtained by shifting the $(0, 0)$ cell through the relation

$$\rho_{mn} = m\,\mathtt{D_x}\,\hat{x} + n\,\mathtt{D_y}\,\hat{y} \tag{5.36}$$

where $\mathtt{D_x}$ and $\mathtt{D_y}$ are the periodic distances in the x- and y-directions.

[1] The `FE-IIEE` method supports not only the Green's function in free space but also others such as, the half space Green's function or the periodic Green's function.

Thus, the periodic Green's function $G_p(\mathbf{r}, \mathbf{r}_s)$ in the spatial domain has the form

$$G_p(\mathbf{r}, \mathbf{r}_s) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} e^{-j(k_x m \mathtt{D_x} + k_y n \mathtt{D_y})} \frac{e^{-jk_0 R_{mn}}}{4\pi R_{mn}} \tag{5.37}$$

where

$$k_x = k_0 \,\mathtt{sin}\,\theta_s \,\mathtt{cos}\,\phi_s \qquad k_y = k_0 \,\mathtt{sin}\,\theta_s \,\mathtt{sin}\,\phi_s \tag{5.38}$$

with θ_s and ϕ_s as the scan angles that determinate the phase-shift between the different adjacent cells and R_{mn} is the distance between the source in the corresponding cell and the observation point. This distance is given by

$$R_{mn} = \sqrt{\left(x - x' + m\,\mathtt{D_x}\right)^2 + \left(y - y' + n\,\mathtt{D_y}\right)^2 + \left(z - z'\right)^2} \tag{5.39}$$

where x', y' and z' are the coordinates of the source in the unit cell.

An accurate and efficient evaluation of the series given in (5.37) is of fundamental importance for the analysis of structures using boundary integral equations. The main constrain of those series is its slow convergence rate for the free space case. Equation (5.37) is extremely slow to converge (for arbitrary d, the number of terms having magnitude 10^{-d} is of order 10^{+2d}) making the numerical evaluation of the series difficult and computationally expensive. Figure 5.17 shows an example about how extremely slow to converge is equation (5.37).

A unit cell of 1 m by 1 m placed in the xy-plane has been considered. Equation (5.37) has been calculated by increasing the number of adjacent unit cells in both x- and y-directions. The red line shows the converged magnitude of the equation for an arbitrary observation point along the z-axis. After consider more than 2000 units cells along each direction, the magnitude of the periodic Green's functions is still not accurate enough, which gives an idea about how slow to converge is the series.

Many techniques exist for accelerating slowly-convergent series, such as the Euler Transformation [62, Equation 3.6.27], the Shank Transformation [63], the Poisson Transformation, the Ewald Transformation [64, 65] or the Kummer Transformation [62, Equation 13.1.27]. A survey of them

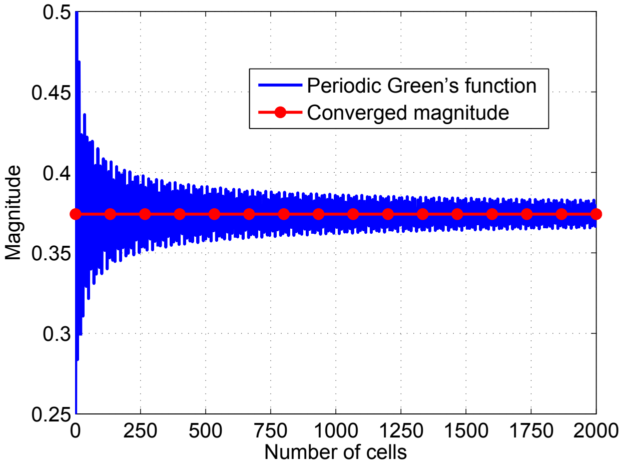


Figure 5.17: Comparison between the value of the series given in (5.37) with the converged magnitude

and their use in evaluating periodic sums of three-dimensional points sources is given in [66]. Among these techniques the one chosen to accelerate the series in the `FEM` module has been the Ewald's transformation since, this method has been considered in the literature as the reference method for the efficient numerical calculation of the periodic Green's function [44, 45].

Ewald Transformation

Ewald transformation was originally derived in [64] for the computation of three-dimensional lattice potentials. It splits the series representation of the lattice potential, and likewise that of the periodic Green's function, into a sum of two series of different types from that of the original series, which are both exponentially convergent.

This section provides a detailed derivation of the Ewald representation for the two-dimensional periodic Green's function. The Ewald transformation starts from the spatial domain representation of the periodic Green's function given in equation (5.37) and makes use of the identity

$$\frac{e^{-jk_0R_{mn}}}{R_{mn}} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{\left(-R_{mn}^2 s^2 + \frac{k_0^2}{4s^2}\right)} ds \tag{5.40}$$

where s is a complex variable. The periodic Green's function can be written in two parts by using the previous identity and splitting the path integration at the parameter $\mathtt{E}$ as

$$G_p(\mathbf{r},\mathbf{r}_s) = G_{p1}(\mathbf{r},\mathbf{r}_s) + G_{p2}(\mathbf{r},\mathbf{r}_s) \tag{5.41}$$

where $G_{p1}(\mathbf{r},\mathbf{r}_s)$ is given by

$$G_{p1}(\mathbf{r},\mathbf{r}_s) = \frac{1}{4\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} e^{-j(k_x m\mathtt{D_x} + k_y n\mathtt{D_y})} \times \frac{2}{\sqrt{\pi}}\int_0^{\mathtt{E}} e^{\left(-R_{mn}^2 s^2 + \frac{k_0^2}{4s^2}\right)} ds \tag{5.42}$$

and $G_{p2}(\mathbf{r},\mathbf{r}_s)$ is given by

$$G_{p2}(\mathbf{r},\mathbf{r}_s) = \frac{1}{4\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} e^{-j(k_x m\mathtt{D_x} + k_y n\mathtt{D_y})} \times \frac{2}{\sqrt{\pi}}\int_{\mathtt{E}}^{\infty} e^{\left(-R_{mn}^2 s^2 + \frac{k_0^2}{4s^2}\right)} ds \tag{5.43}$$

with k_x, k_y and R_{mn} as the ones used in equation (5.37). For the integral in equation (5.43), Ewald transformation applies directly. More precisely, using the identity [62, Equation 7.4.34],

$$\frac{2}{\sqrt{\pi}}\int_{\mathtt{E}}^{\infty} e^{\left(-R_{mn}^2 s^2 + \frac{k_0^2}{4s^2}\right)} ds = \frac{1}{2R_{mn}}\left[e^{-jk_0 R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E} - \frac{jk}{2\mathtt{E}}\right) + e^{jk_0 R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E} + \frac{jk}{2\mathtt{E}}\right)\right] \tag{5.44}$$

where $\mathtt{erfc}$ is the complementary error function, the integral can be written as

$$G_{p2}(\mathbf{r},\mathbf{r}_s) = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \frac{e^{-j(k_x m\mathtt{D_x} + k_y n\mathtt{D_y})}}{R_{mn}} \times \sum_{\pm}\left[e^{\pm jk_0 R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}}\right)\right] \tag{5.45}$$

which is essentially a "modified" spatial-domain portion of the periodic Green's function. The summation over ± is a shorthand notation for the right hand side of equation (5.44) and it will be used along this chapter.

Making use of the Poisson transformation, or alternatively the procedure in [65, 44], the integral in equation (5.42) is finally transformed to

$$G_{\mathtt{p1}}(\mathbf{r},\mathbf{r}_s) = \frac{e^{j(k_x\xi+k_y\eta)}}{8\mathtt{D_x}\mathtt{D_y}} \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \frac{1}{\alpha_{mn}} \times \sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right) e^{\pm 2\,\alpha_{mn}\,\zeta}\right] e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}}+\frac{n\eta}{\mathtt{D_y}}\right)} \tag{5.46}$$

where

$$\xi = x - x' \qquad \eta = y - y' \qquad \zeta = z - z' \tag{5.47}$$

and

$$\alpha_{mn}^2 = \left(\frac{\pi m}{\mathtt{D_x}}\right)^2 + \left(\frac{\pi n}{\mathtt{D_y}}\right)^2 + \left(\frac{\pi m}{\mathtt{D_x}}\right)k_x + \left(\frac{\pi n}{\mathtt{D_y}}\right)k_y + \frac{1}{4}\left(k_x^2 + k_y^2 - k_0^2\right) \tag{5.48}$$

Thus, equations (5.45) and (5.46) are combined to give the numerical value of the Green's function. The complementary error function which appears in both series makes them converge rapidly. The parameter E controls the convergence rate. As E becomes larger, the spatial series (5.45) converges faster, while the spectral series (5.46) converges slower. The optimum parameter is the one that makes the two series converge at the same rate, so that equal numbers of terms are required in the calculation of both series. By analysis of the behavior of the series terms, the optimum parameter $\mathtt{E_{opt}}$ is found to be [44]

$$\mathtt{E_{opt}} = \sqrt{\frac{\pi}{\mathtt{D_x}\mathtt{D_y}}} \tag{5.49}$$

Choosing this value for E and adjusting the summation limits so that the most dominant terms are kept, in almost all practical cases it is sufficient to include only three adjacent cells.

Numerical implementation

The numerical implementation of the FE-IIEE algorithm for infinite periodic structures is conceptually straightforward, since just a slight modification of the Green's function used to calculate the radiated field by the FEM region is required.

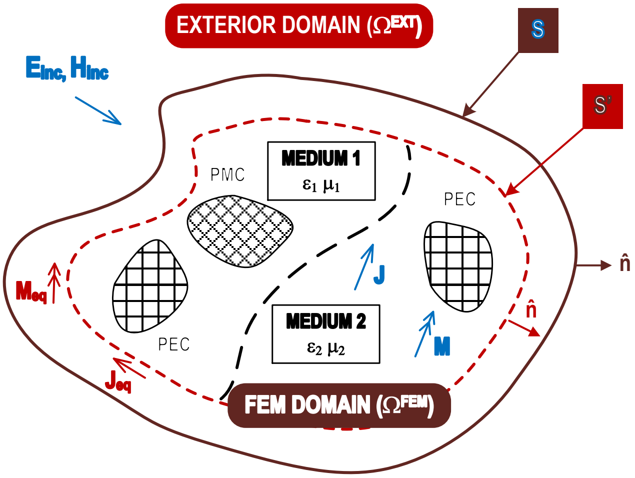


Figure 5.18: Typical setup of a open single-region problem

In this sense, the third step of the `FE-IIEE` algorithm requires the calculation of the field, and its curl, over S radiated by the equivalent currents $\mathbf{J}_{\texttt{eq}}$ and $\mathbf{M}_{\texttt{eq}}$. Figure 5.18 shows a sketch where all the component involved in the `FE-IIEE` algorithm may be appreciated. The fields radiated by the `FEM` region, $\mathbf{V}_{\texttt{FE-IIEE}}$ and its curl $(\boldsymbol{\nabla} \times \mathbf{V})_{\texttt{FE-IIEE}}$, are computed using the integral expressions

$$\mathbf{V}_{\texttt{FE-IIEE}} = \oiint_{S'} \left(\mathbf{L}_{\texttt{eq}} \times \boldsymbol{\nabla}\, G\right)\, dS^{'} \\ - jkh \oiint_{S'} \left[\mathbf{O}_{\texttt{eq}} \left(G + \frac{1}{k^2}\, \boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G\right)\right]\, dS^{'} \quad (5.50)$$

$$(\boldsymbol{\nabla} \times \mathbf{V})_{\texttt{FE-IIEE}} = jkh \oiint_{S'} \left(\mathbf{O}_{\texttt{eq}} \times \boldsymbol{\nabla}\, G\right)\; dS^{'} \\ - \oiint_{S'} \left[\mathbf{L}_{\texttt{eq}} \left(k^2\, G + \boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G\right)\right]\, dS^{'} \quad (5.51)$$

where one can see that G denotes the Green's function. In the particular case of the infinite periodic structures, G is the corresponding Ewald representation of the two-dimensional periodic Green's function G_p.

It is worth noting that the first derivative ($\boldsymbol{\nabla}\, G$ term) and the second derivative ($\boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G$ term) of the Ewald series are required to compute the fields radiated by the FEM region. Due to the complexity and extension of the calculation of those derivatives, the expressions of both terms $\boldsymbol{\nabla}\, G$ and $\boldsymbol{\nabla}\,\boldsymbol{\nabla}\, G$ are attached in the appendix B.

Thereby, the FE-IIEE algorithm used to solve the infinite periodic problems is exactly the same as the non-periodic case except for the calculation of the fields radiated by the FEM region.

Something to consider about the use of the FE-IIEE method is that it is able to absorb reflection for waves propagating simultaneously in various directions and not only for waves propagating in a given (θ_s, ϕ_s) direction. This feature is very useful for the analysis of finite arrays when several floquets modes are excited at the same time. Therefore, the initial value of $\boldsymbol{\Psi}$ (see step 1 of the FE-IIEE algorithm in section 3.2.1) may be calculated regardless of the absorbing boundary condition (ABC) used, either applying the first-order or the modified ABC commented previously. The only difference between the use of those conditions in the method resides in the number of iterations needed to reach the (numerically) exact radiation boundary condition. A study of this behavior is detailed in the first example of section 5.3.

Ewald verification tests

Before starting with the simulation of real problems, the verification of the values given by the Ewald representation described in previous sections has been performed. One way to verify whether the value given by the Ewald representation in equations (5.45) and (5.46) is correct, is to compare that value with the one given directly evaluating the periodic Green's function in (5.37) when the number of cell tends to infinity.

The unit cell considered has been a square cell ranging from $0 \leq x \leq 1\,\mathrm{m}$ and from $0 \leq y \leq 1\,\mathrm{m}$ with a source point located at $(0.98, 0.30, 0.0)\,\mathrm{m}$. The observation point has been placed at (0.68,0.71,0.6) m and the working frequency has been 100 MHz. The periodic Green's function in (5.37) has been evaluated by increasing the number of adjacent unit cells until 2000 in both x- and y-directions. Figure 5.19 shows the comparison between the magnitude of the periodic Green's function and the Ewald representation.

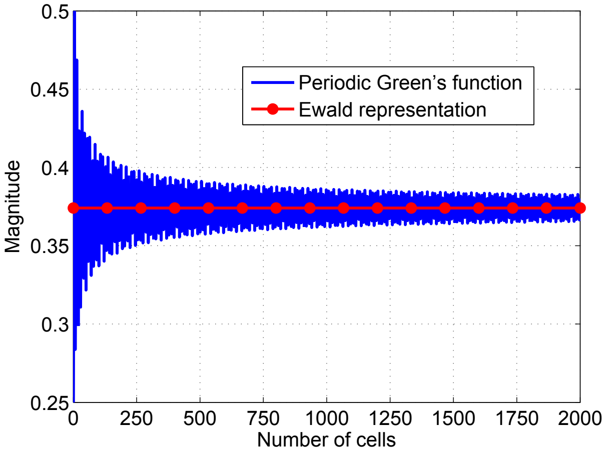


Figure 5.19: Comparison between the module of the Ewald representation and the periodic Green's function

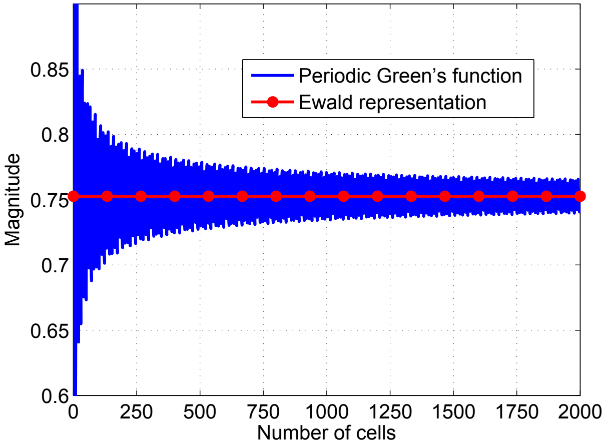


Figure 5.20: Comparison between the module of the first derivative of the Ewald representation and the periodic Green's function

The Ewald representation converges using 3 adjacent units cells; many more than 2000 cells are needed to reach an acceptable value of the periodic Green's function. Figure 5.20 shows the comparison between the magnitude of the first derivative of the periodic Green's function and the magnitude of the first derivative of the Ewald representation. The

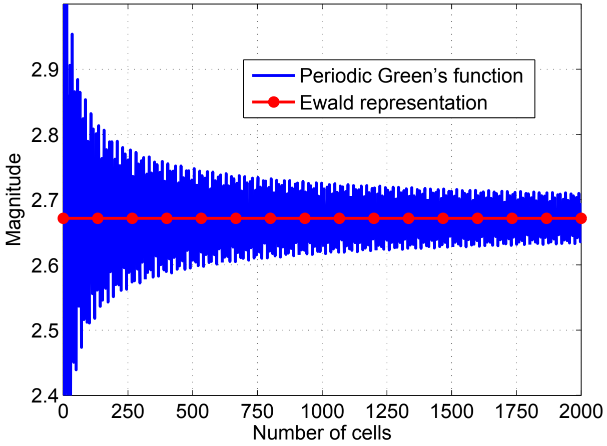


Figure 5.21: Comparison between the module of the second derivative of the Ewald representation and the periodic Green's function

equations involved in the calculation of the first derivative of the Ewald representation are documented in the appendix B, in particular, equations (B.8)-(B.13). In this case, the Ewald representation also converges using 3 adjacent units cells providing an accurate value. The comparison between the magnitude of the second derivative of the periodic Green's function and the magnitude of the second derivative of the Ewald representation is shown in figure 5.21. Appendix B contains the equations of the second derivative of the Ewald representation, in particular, equations (B.15)-(B.32). The Ewald representation also provides an accurate value in this case.

5.3 Numerical Results

To illustrate the capabilities of the suite solving infinite periodic structures some examples has been analyzed. The first example has consisted of the analysis of an infinite ground plane. It is worth to note that this example has been used as validation test since the result of this analysis may be compare with an analytic solution. The second example has consisted of the analysis of a microstrip patch phased array using the infinite analysis approach. The results of this analysis have been compared with the ones given by the `MoM` module of the suite for an 11 x 11 finite array.

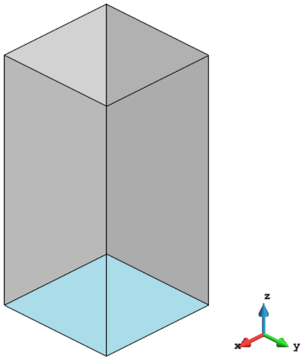


Figure 5.22: Unit cell model used for the first test

5.3.1 Analysis of infinite ground plane

The following example performs several analysis of an infinite ground plane under a plane wave excitation. Furthermore, the different mesh truncation techniques supported by the FEM module have also been used in order to check their accuracy and performance in the analysis of infinite structures.

Normal incidence

In this first test, the infinite ground plane has been illuminated by a plane wave with $\phi = 0°$ and $\theta = 0°$ as incident angles. The plane wave has been polarized in both ϕ- and θ-components and the working frequency has been set to 300 MHz.

The unit cell consider for this test has been an hexahedron with 0.25 m long by 0.25 m wide by 0.5 m high. The ground plane has been placed in the xy-plane. It is worth noting that the solution to this problem is a standing wave (SW) with a wavelength of 0.5 m. Thus, it is easy to see if the results given by the FEM module are correct, since a complete period of the SW should be appreciated along the z-axis of the unit cell. Figure 5.22 shows the unit cell model used in this test.

The non-periodic direction (z-axis) has been truncated using a first-order absorbing boundary condition (ABC). The expression of the modified ABC

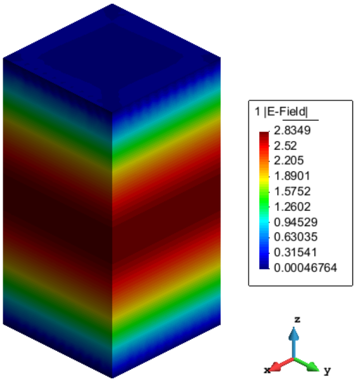


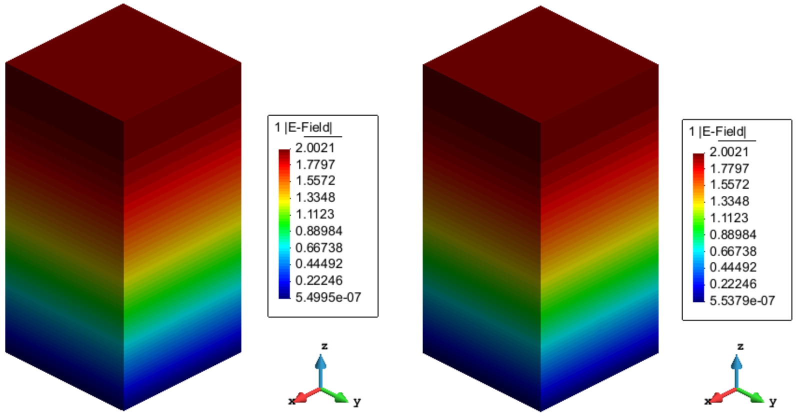
Figure 5.23: |`E-Field`| for the first test

for the scan angle $\theta_s = 0°$ matches with the first-order `ABC`. Thereby, this test has been analyzed only using the first-order `ABC` alone and in combination with `FE-IIEE` method as truncation techniques. Note that the initial value of $\mathbf{\Psi}$ for the `FE-IIEE` method has been calculated using the first-order `ABC`.

Figure 5.23 shows the magnitude of the E-field when the non-periodic direction of the unit cell is truncated by using the first-order `ABC` alone. A complete period of the standing wave may be appreciated as it was expected. The analysis of the unit cell when the non-periodic direction is truncated by using the `FE-IIEE` method has been also performed. In this case, the initial value of $\mathbf{\Psi}$ is already numerically exact, since that value is calculated by using the first-order `ABC`. Under this scenario, the `FE-IIEE` method should reach the solution in one iteration with an error numerically zero. The error given by the method in this test has been `5.23772E-08` which confirms the expected behavior. It is worth to note that the results given by both truncation techniques are indistinguishable. Hence, this first validation test is concluded to be satisfactory.

Oblique incidence

In this case, the infinite ground plane has been illuminated by a plane wave coming from $\phi = 20°$ and $\theta = 60°$. The plane wave has been polarized in the ϕ-component and the working frequency has been set



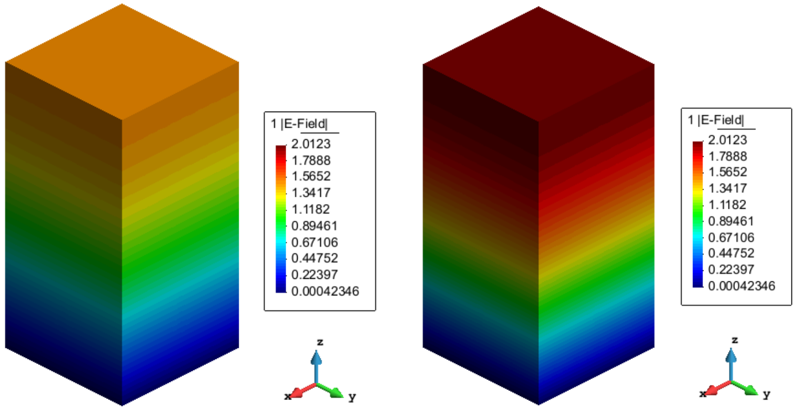
(a) Modified `ABC` truncation technique (b) Modified `ABC` using `FE-IIEE` method

Figure 5.24: Comparison of the |`E-Field`| for the second test

to 300 `MHz`. The unit cell consider for this test has been the same as the used in the previous example. The solution to this problem is a `SW` but with a wavelength of 1.0 m, instead of 0.5 m. Thus, a half period of the `SW` should be appreciated along the z-axis of the unit cell.

The non-periodic direction (z-axis) of this example has been truncated using the modified `ABC` first. Figure 5.24(a) shows the magnitude of the E-field when the non-periodic direction of the unit cell is truncated with this technique alone. The half standing wave commented above may be appreciated. The analysis of the unit cell when the non-periodic direction is truncated by using the modified `ABC` in combination with the `FE-IIEE` method has been also carried out. As it occurs in the previous case, the initial value of $\mathbf{\Psi}$ is numerically exact, since the modified `ABC` is used to calculate that value. Then, the `FE-IIEE` method should reach the solution in one iteration with an error numerically zero. The error given by the method in this second test has been `8.45770E-09` which confirms the commented behavior. Hence, this second validation test also may be considered satisfactory.

The first-order `ABC` has been used to truncate the non-periodic direction of the unit cell for the next test. It is worth to note, that the plane wave



(a) 1^{st}-ABC truncation technique (b) 1^{st}-ABC using FE-IIEE method

Figure 5.25: Comparison of the |E-Field| for the second test

used in this example does not satisfy the first-order ABC and significant reflections appear in this case. Figure 5.25(a) shows the magnitude of the E-field where the maximum value of the SW has a substantial reduction due to the reflection produces by the first-order ABC. However, applying the FE-IIEE method the reflection is canceled as it may be seen in figure 5.25(b). The initial value of $\mathbf{\Psi}$ is not accurate in this case, but the FE-IIEE method is able to cancel the reflections up to a relative error of 4.58628E-05 in 7 iterations.

5.3.2 Microstrip patch phased array

An 11 x 11 microstrip patch phased array has been analyzed in this example using the infinite array approach. The array is printed on a substrate $\epsilon_r = 2.67$ and is housed in a 520 mm x 580 mm x 7 mm cavity in a ground plane [67] as illustrated in figure 5.26. The dimensions of each patch element are 30 mm x 35.6 mm and the gaps between any two neighbor elements are 14 mm along both length and width directions. The first analysis has consisted of the simulation of the isolated radiating element of the array with both MoM and FEM modules of the suite comparing their results. Figure 5.27 shows the comparison of the radiation pattern between both modules. A good agreement is appreciated.

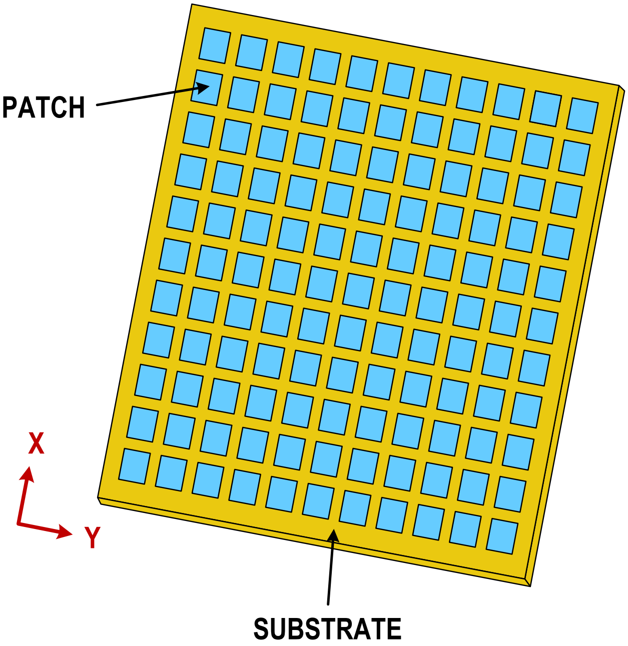


Figure 5.26: Perspective view of the 11 x 11 microstrip patch array

The next analysis has consisted of the simulation of different unit cells of the array using the FEM module and the PBCs. The distance between the elements of the array has been increased from $0.31\,\lambda$ to $0.37\,\lambda$ by increasing the distance of the PBCs to the patch. Figure 5.28 shows a comparison of the S_{11} parameter of the unit cells where the influence of the mutual coupling between the elements is clearly observed. The S_{11} parameter of the isolated radiating element of the array computed using the FEM module of the suite has been also included in the figure.

The final analysis has consisted of the simulation of the real 11 x 11 array using the infinite array approach. The distance between elements in this analysis has been $0.37\,\lambda$. The radiation pattern computed by the FEM module using the infinite array approach has been multiplied by the array factor and the results has been compared with the ones given by the MoM module of the suite for the whole array. Figure 5.29 shows the comparison of the results where a good agreement is appreciated.

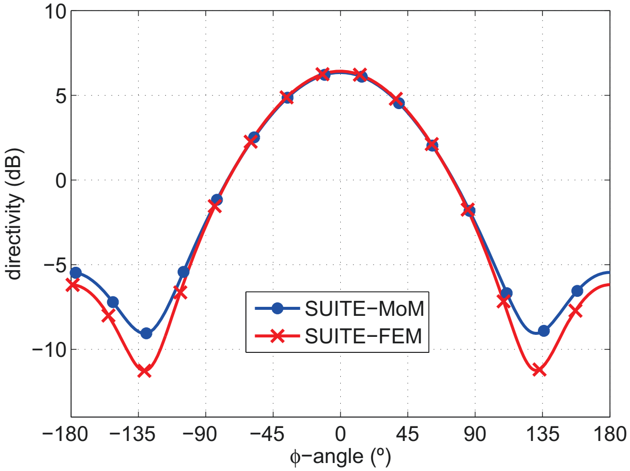


Figure 5.27: Comparison of radiation pattern for isolated element

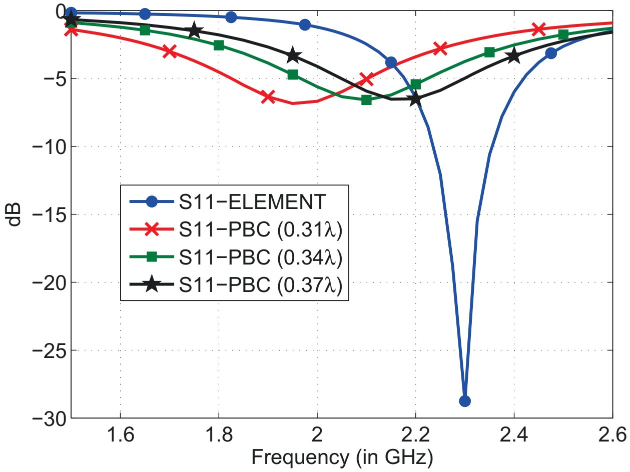


Figure 5.28: Comparison of S_{11} parameter

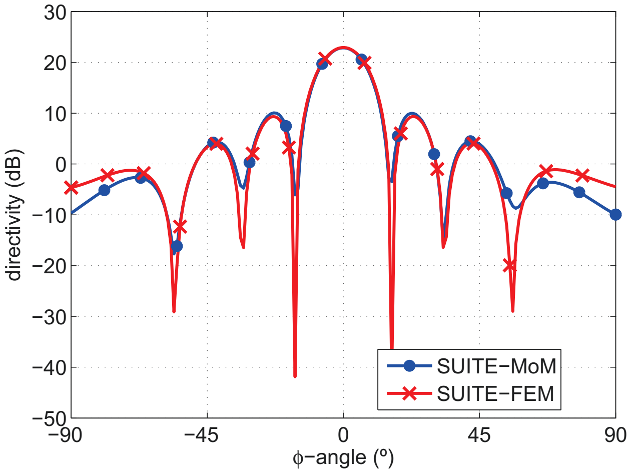


Figure 5.29: Comparison of the radiation pattern of the whole array

5.4 Conclusions

The FEM analysis of structures that are infinitely periodic in the two dimensional xy-plane has been carried out in this chapter. The approach employed to perform this analysis has been based on the use of the so-called Periodic Boundary Conditions (PBCs). The formulation to include the PBCs on numerical codes such as the FEM module of the suite, has been described in detail in Section 5.1. An algorithm to perform the numerical implementation of the PBCs on these numerical codes has been presented in this chapter. Technical details about the practical implementation of this algorithm in sparse solvers have also been given.

The modification of the truncation techniques implemented in the FEM module have been performed in order to support the analysis for open scattering and radiation problems. The standard first-order absorbing boundary condition (ABC) has been modified in order to provide a more appropriate boundary condition that absorbs waves propagating in any direction and not only propagating along $\theta_s = 0$. A validation test of this modified ABC has been performed in order to demonstrate the validity of this boundary condition.

In the case of the FE-IIEE method, the Green's function $G(r, r_s)$ in free-space has been modified by an accelerated representation of the two-dimensional periodic Green's function. The acceleration technique used in this Ph. D. dissertation has been the Ewald Transformation that converges using 3 adjacent units cells; many more than 2000 cells are needed to reach an acceptable value of the periodic Green's function. It is important to note that the FE-IIEE method is able to absorb reflection for waves propagating simultaneously in various directions and not only for waves propagating in a given (θ_s, ϕ_s) direction. This feature is very useful for the analysis of finite arrays when several floquets modes are excited at the same time. Finally, the analysis of some real structures has been carried out in order to validate the capabilities of the suite solving infinite periodic structures.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH LINES

6.1 Final conclusions

The work developed in this Ph. D dissertation has consisted of the development of a new software suite for electromagnetics. This suite contains important features, such as, a professional graphical framework that provides a friendly graphical user interface, two powerful solver based on integral (MoM) and differential (FEM) approaches or the possibility to analyze large array structures by using infinite array approaches. It is worth to note that an hybridization of the two solver existing in the suite has been considered. A previous study has been performed in the present Ph. D dissertation deriving the hybrid variational formulation and the connection between the basis functions of both solvers.

Chapter 2 has presented a detailed description of the development of the graphical framework developed in this Ph. D dissertation. The graphical framework provides a GUI based on the general purpose pre and postprocessor called GiD. The main idea has been to develop a new GUI focused only on electromagnetics, instead of developing a general purpose pre and postprocessor as GiD is. Thereby, the framework provides an easy and quick way to perform the simulation process in five steps: geometry creation, electromagnetic setup, mesh generation, simulation and results visualization. Also, new important features has been provided to the software GiD, such as, a symbolic variables tool that enables the parametrization of models using symbols, new definition windows for electromagnetic variables, a complete navigation tree to facilitate the access to the menus or automatic online updates for the suite among others. The development of an automatic goal oriented optimizer has been other of the objectives of this Ph. D dissertation. In this way, a powerful tool in the design of electromagnetic structures has been implemented to complement the suite. Users can use the optimizer to automatically adjust the designated model parameters and improve the results like maximum gain and low side lobes by using three different optimization algorithms such as, Powell's Method, Simplex Method and Particle Swarm Optimization (PSO).

Chapter 3 has presented the FEM module of the suite. The module implements the so-called Finite Element - Iterative Integral Equation Evaluation (FE-IIEE) method which employs a boundary integral equation that provides a (numerically) exact radiation boundary condition. The module has been developed to support the use of four different sparse solver and a parallel implementation where the CPU cores available on single computers or on several compute nodes of a high performance computing (HPC) cluster may be utilized. Furthermore, the accuracy of the results given by the module has been verified applying the Method of Manufactured Solutions (MMS) to different verification tests obtaining excellent results. Finally, in this chapter, the simulations of real world problems has been compared with the results of commercial softwares and measurement providing evidences that the code can be used for many type of simulations with very good performance and scalability in mid-size HPC clusters.

Chapter 5 provides a description of the FEM analysis of structures that are infinitely periodic in the two dimensional xy-plane. The approach employed to perform this analysis has been based on the use of the so-called Periodic Boundary Conditions (PBCs). The formulation to include the PBCs on numerical codes, such as, the FEM module of the suite, has been described in detail. An algorithm to perform the numerical implementation of the PBCs on these numerical codes has been presented in this chapter. Technical details about the practical implementation of this algorithm in sparse solvers have also been given. The modification of the truncation techniques implemented in the FEM module have been performed in order to support the analysis for open scattering and radiation problems. The standard first-order absorbing boundary condition (ABC) has been modified in order to provide a more appropriate boundary condition that absorbs waves propagating in any direction and not only propagating along $\theta_s = 0$. In the case of the FE-IIEE method, the Green's function $G(r, r_s)$ in free-space has been modified by an accelerated representation of the two-dimensional periodic Green's function. The acceleration technique used in this Ph. D. dissertation has been the Ewald Transformation. Finally, the analysis of some real structures has been carried out in order to validate the capabilities of the suite solving infinite periodic structures.

Chapter 4 has presented a modular approach to combine MoM and FEM techniques for the analysis of large structures or finite arrays with complex radiating elements. The approach have consisted of performing the FEM analysis of each structure and to define the equivalent electric and magnetic currents on the boundaries of the structure. Then, the interaction between the structures is computed by using MoM and the global FEM-MoM system of equation is assembled. Finally, the system is solved and the analysis is completed. It is worth to note that the approach has not been totally implemented but some previous works have been done.

One of these works has been the modification of the current variational formulation of the FEM module of the suite. This modification has been to provide unknowns for the corresponding dual field (H-Field or E-Field) in an explicit way. The standard variational formulation of FEM problems only has unknowns for one of the corresponding electromagnetic fields (E-Field or H-Field) and the previous modification is required in order to connect both FEM and MoM formulation. Furthermore, a study of the projection between the basis functions of both modules has been done in order to connect the unknowns between both codes. The boundaries of the antenna is where both methods share the unknowns and those need to be expressed in terms of the same basis functions. In the particular case of this Ph. D. dissertation, a study between the well-known Rao-Wilton-Glisson (RWG) div-conforming basis functions and the the Nédélec triangular curl-conforming basis functions of second-order has been performed.

6.2 Future research lines

As commented previously, the work developed in the present Ph. D dissertation has been focused on the development of a new software suite for electromagnetics. The current version of the suite presents important features that have been already described along this document, but future improvement may be implemented as the one shown in this section.

Regarding to the graphical framework, the link of the automatic goal oriented optimizer with the `FEM` module is a future work of great importance in order to complement the suite with a powerful tool in the design of electromagnetic structure. Other future improvements may be the implementation of postprocessing `RCS` tecniques as Synthetic-aperture radar (`SAR`) or the support of new time domain modules (`TDIE` or `FDTD`)

It is worth to note that the performance of the Finite Element module are very competitive in a single frequency simulation. However, due to the absence of an adaptive frequency sweep (as for instance, a numerical interpolation using the Cauchy Method [68]), the simulation for a range of frequencies may be slower than other codes that supports an adaptive frequency sweep. Thus, the implementation of the Cauchy Method inside the module is considered as a future research work. Also, the implementation of an adaptive method for mesh refinement is considered as a future research work in order to provide more accurate meshes with a less number of elements reducing the current simulation time.

Regarding the analysis of infinite periodic structures, some future work may be performed as for instance, an improvement in the performance of the calculation of the Ewald representation or the calculation of S-parameters of any source of an infinite array.

Finally, an intensive future work should be done regarding the `FEM-MoM` hybridization. A study between the higher-order quadrilateral basis function and the Nédélec triangular curl-conforming basis functions of second order is required in order to perform the hybridization. Also, the connection between both codes should be checked by using the appropriate verification tools, as for instance, the Method os Manufactures Solutions (`MMS`). Once, this process has been done, the simulation of real world problems should be carried out in order to demonstrate the performance and capabilities of the codes for those kinds of simulations.

6.3 Publications

The work developed in this Ph. D dissertation has led to several journal, book and conference contributions. The complete list is detailed below:

Books (2):

- Yu Zhang and Tapan K. Sarkar, *"Parallel Solution of Integral Equation Based EM Problems in the Frequency Domain"*, John Wiley & Sons, Inc., Hoboken, New Jersey, June-2009. Contributors: **Daniel Garcia Doñoro**, Hongsik Moon, Mary Taylor and Robert A. van de Geijn
- Yu Zhang, Tapan K. Sarkar, Xun Wang Zhao, **Daniel Garcia Doñoro**, Weixin Zhao, Magdalena Salazar Palma and Sio Weng Ting, *"Higher Order Basis Based Integral Equation Solver (HOBBIES)"*, John Wiley & Sons, Inc., Hoboken, New Jersey, June-2012.

Articles in International Journals (7):

- R. Otin, L. E. Garcia Castillo, I. Martinez Fernandez and **Daniel Garcia Doñoro**, *"Computational Performance of a Weighted Regularized Maxwell Equation Finite Element Formulation"*, Progress in Electromagnetics Research-PIER, Vol. 136, Pages 61−77, Jan−2013
- Ying Yan, Yu Zhang, Chang-Hong Liang, Hui Zhao and **Daniel Garcia Doñoro**, *"RCS Computation by Parallel MoM Using Higher Order Basis Functions"*, International Journal of Antennas and Propagation, Vol. 2012, Jan−2012
- **Daniel Garcia Doñoro**, I. Martinez Fernandez, L. E. Garcia Castillo, Y. Zhang and T.K. Sarkar, *"RCS Computation Using a Parallel in-core and Out-of-core Direct Solver"*, Progress in Electromagnetics Research-PIER, Vol. 118, Pages 505−525, Jul−2011
- Yu Zhang, Xun-Wang Zhao, **Daniel Garcia Doñoro**, Sio-Weng Ting and Tapan K. Sarkar, *"Parallelized Hybrid Method with Higher-Order MoM and PO for Analysis of Phased Array Antennas on Electrically Large Platforms"*, IEEE Transactions on Antennas and Propagation, Vol. 58, Issue 12, Pages 4110−4115, Dec-2010

- Hong Wei Zhang, Xun Wang Zhao, Yu Zhang, **Daniel Garcia Doñoro**, Weixin Zhao, Chang-Hong Liang, *"Analysis of a Large Scale Narrow-wall Slotted Waveguide Array by Parallel MoM Out-of-core Solver Using the Higher Order Basis Functions"*, Journal of Electromagnetic Waves and Applications, Vol. 24, Issue 14−15, Pages 1953−1965, Nov-2010

- Xun Wang Zhao, Yu Zhang, Hong Wei Zhang, **Daniel Garcia Doñoro**, Sio Weng Ting, Tapan K. Sarkar and Chang-Hong Liang, *"Parallel MoM-PO Method with Out-of-core Technique for Analysis of Complex Arrays on Electrically Large Platforms"*, Progress In Electromagnetics Research PIER, Vol. 108, Pages 1-21, Sep-2010

- **Daniel Garcia Doñoro**, Luis Emilio Garcia-Castillo and Ignacio Gomez-Revuelto, *"An Interface for an hp-Adaptive Finite Element Package Using GiD"*, Finite Elements in Analysis and Design, Vol. 46, Pages 328-338, Jan-2010

International Conference Proceedings (22):

- **Daniel Garcia Doñoro**, Y. Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar-Palma, *"HOBBIES EM-Suite 2014"*, GiD Convention 2014, Barcelona (Spain), Jul−2014

- **Daniel Garcia Doñoro**, I. Martinez Fernandez, L. E. Garcia Castillo and M. Salazar-Palma, *"A Higher Order Finite Element Method Electromagnetic Simulator"*, 12th International Workshop on Finite Element for Microwave Engineering, Chengdu (China), May−2014

- T. K. Sarkar, M. Salazar Palma, **Daniel Garcia Doñoro** and Hongsik Moon, *"Solution of a Million by Million Complex Matrix Equation by Gaussian Elimination with Partial Pivotting Using Parallel Out-of-core Solvers"*, 8th European Conference on Antennas and Propagation, The Hague (The Netherlands), Apr−2014

- M. Salazar Palma, **Daniel Garcia Doñoro**, T. Sarkar, Y. Zhang, H. Moon, and Sio Wei Ting, *"Advantage of Using a Higher Order Basis for the Solution of Large Electromagnetic Field problems"*, International Wireless Symposium 2014, Xi'an (China), March−2014

- Yu Zhang, H Moon, **Daniel Garcia Doñoro**, T. K. Sarkar, M. Salazar Palma and Sio Wei Ting, *"The Art of Parallelization in Solving Large Electromagnetic Field Problems Out-Of-Core"*, International Wireless Symposium 2014, Xi'an (China), March−2014

- **Daniel Garcia Doñoro**, M. Salazar Palma, T. K. Sarkar, Y. Zhang, Hongsik Moon and Sio Wei Ting, *"Use of Optimization in Designing Complex Electromagnetic Radiating Structures"*, International Wireless Symposium 2014, Xi'an (China), March−2014

- T. K. Sarkar, Y. Zhang, **Daniel Garcia Doñoro**, Hongsik Moon, M. Salazar Palma and Sio Wei Ting, *"Solving Large Complex Problems Using A Higher Order Basis: Parallel Out-of-core Integral Equation Solvers Involving A Million Unknowns"*, International Wireless Symposium 2014, Xi'an (China), March−2014

- L. Garcia Castillo, I. Gomez Revuelto, **Daniel Garcia Doñoro**, I. Martinez Fernandez and M. Salazar Palma, *"Recent Developments Regarding Finite Element Methods at the Radiofrequency Group of Universidad Carlos III de Madrid"*, Iberian Meeting on Computational Electromagnetics, Denia (Spain), May− 2013

- **Daniel Garcia Doñoro**, Y. Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"Higher Order Basis Based Integral Equation Solver: HOBBIES"*, Iberian Meeting on Computational Electromagnetics, Denia (Spain), May−2013

- **Daniel Garcia Doñoro**, L. E. Garcia Castillo, S. Llorente-Romano and Y. Zhang, *"EMC Analysis using a Higher Order Basis Based Integral Equation Solver"*, Computational Electromagnetics for EMC 2013, Granada (Spain), March−2013

- **Daniel Garcia Doñoro**, Y. Zhang, W. Zhao, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"Very Large Scale Computational Electromagnetics"*, 6th Conference on Advances and Applications of GiD, Monograph CIMNE, Barcelona (Spain), May−2012

- Ying Yan, Yu Zhang, Xun Wang Zhao, Chong Hong Liang, **Daniel Garcia Doñoro** and Tapan K. Sarkar, *"Parallel Computation of Radar Cross Section of Target with Coatings"*, 5th European Conference on Antennas and Propagation, Rome (Italy), April−2011

- Y. Zhang, Sio Ting, **Daniel Garcia Doñoro**, Xun Wang Zhao, M. Salazar Palma and Tapan K. Sarkar, *"Stability Analysis of a Parallel Higher Order Basis Based Integral Equation Solver (HOBBIES) on a Cluster with 512 AMD CPU Cores"*, 5th European Conference on Antennas and Propagation, Rome (Italy), April−2011

- Xun-Wang Zhao, Y. Zhang, **Daniel Garcia Doñoro**, T. K. Sarkar and Sio Ting, *"Parallel Higher Order Method of Moments for Accurate Analysis of Antenna Radome Platform System"*, 5th European Conference on Antennas and Propagation, Rome (Italy), April−2011

- **Daniel Garcia Doñoro**, W. Zhao, Y. Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"Automatic Goal Oriented Optimization Using Parallel Higher Order Basis Based Integral Equation Solver"* 5th European Conference on Antennas and Propagation, EuCAP 2011, Rome (Italy), April−2011

- W. Zhao, **Daniel Garcia Doñoro**, Y. Zhang, Xun Wang Zhao and T. K. Sarkar, *"Optimization of Radiation Pattern for Narrow wall Slotted Waveguide Arrays Using HOBBIES"*, International Symposium on Antennas and Propagation, Macao (China), Nov−2010

- W. Zhao, Y. Zhang, **Daniel Garcia Doñoro** and T. K. Sarkar, *Optimizing Narrow wall Slotted Waveguide Arrays Using HOBBIES"*, 2010 IEEE International Symposium on Antennas and Propagation, Toronto (Canada), July−2010

- **Daniel Garcia Doñoro**, W. Zhao, Yu Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"HOBBIES: A new electromagnetic simulator"*, 2010 IEEE International Symposium on Antennas and Propagation, Toronto (Canada), July−2010

- **Daniel Garcia Doñoro**, W. Zhao, Y. Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"HOBBIES: Electromagnetic Simulator Using GiD"*, 5th Conference on Advances and Applications of GiD, Monograph CIMNE, Barcelona (Spain), May−2010

- **Daniel Garcia Doñoro**, Weixin Zhao, Y. Zhang, T. K. Sarkar, L. E. Garcia Castillo and M. Salazar Palma, *"Higher Order Basis Based Integral Equation Solver with Automatic Goal Oriented Optimiza-tion"*, 14th Biennial IEEE Conference on Electromagnetic Field Computation 2010, Chicago (USA), May−2010

- W. Zhao, Arijit De, **Daniel Garcia Doñoro**, Y. Zhang and T. K. Sarkar, *"Antenna Optimization by Using NEWUOA"*, 2009 IEEE International Symposium on Antennas and Propagation, Charleston (USA), June−2009

- Yu Zhang, Tapan K. Sarkar, H. Moon, Mary C. Taylor, **Daniel Garcia Doñoro**, and M. Salazar Palma, *"Parallel MoM Simulation of Complex EM Problems"*, 2009 IEEE International Symposium on Antennas and Propagation, Charleston (USA), June−2009

APPENDIX A

SOLVING ELECTROMAGNETIC PROBLEMS IN THE SUITE

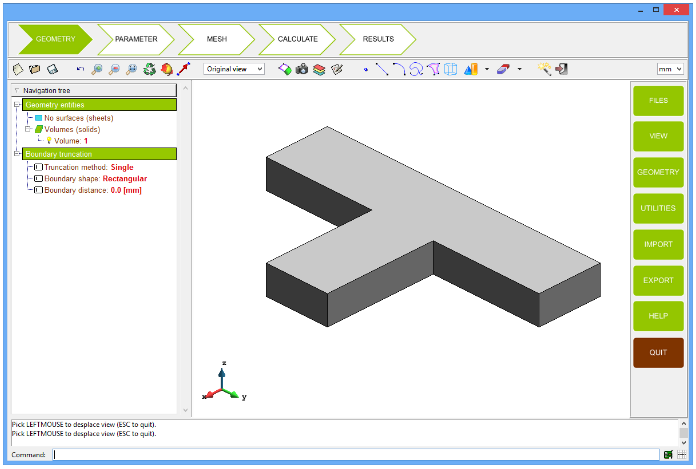
This appendix describes how to use and set up the suite for the simulation of electromagnetic problems. It is worth to note that the `FEM` module of the suite has been used to elaborate this appendix but its use is extensible also to the `MoM` module. The simulation of a problem is performed in five easy steps which are described in the following sections.

A.1 Step 1: Geometry modeling

The first step is to generate the geometry model of the structure. The suite makes use of the tools provided by GiD in order to facilitate the creation of the geometry model. Figure A.1 shows the graphical aspect of the suite for this first step. A summary of the most important geometry tools is given in the following section.

A.1.1 Basic tools

It is worth to note that GiD constructs the geometry model in a hierarchical mode. This means that entities of a higher level such as surfaces are constructed over entities of a lower level such as lines. This section gathers how to create entities from the lower to the higher level.



Navigation tree
- Geometry entities
  - No surfaces (sheets)
  - Volumes (solids)
    - Volume: 1
- Boundary truncation
  - Truncation method: Single
  - Boundary shape: Rectangular
  - Boundary distance: 0.0 [mm]

```
Pick LEFTMOUSE to desplace view (ESC to quit).
Pick LEFTMOUSE to desplace view (ESC to quit).
```

Command:

Figure A.1: Graphical aspect of the geometry step of the suite

- **Points**: Individual points are created by entering each point in any of the following ways:

  1. Picking the coordinates in the screen with the mouse.
  2. Typing the coordinates of the points in the command line.

- **Straight lines**: A straight line is created by entering the coordinates of the two end points or the identifiers of the two end points. The steps to create a straight line are:

  1. Click on `Geometry` ⇒ `Create` ⇒ `straight line` menu, or click on the icon ╲ in the toolbar.
  2. Enter the coordinates of the end points of the line. The key sequence `CTRL+a` enables the selection mode where existing points may be selected to create the line. The points may be selected by the mouse or the identifier of those points may be typed on the command line.
  3. Press `ESC` key to finish the line creation process.

- `NURBS` **lines**: The `NURBS` line is created by entering the interpolated points of the curve, either by adding new points or by selecting existing ones. The steps to create a `NURBS` line are:

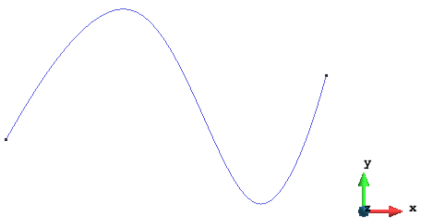


Figure A.2: Examples of `NURBS` line

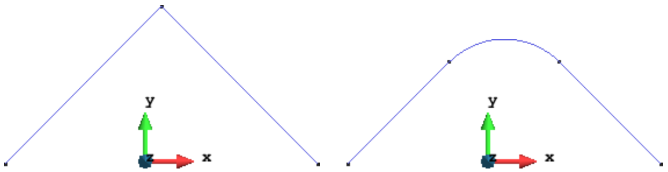
1. Click on `Geometry` ⇒ `Create` ⇒ `NURBS line` menu, or click on the icon in the toolbar.
2. Enter two o more points to create a `NURBS` line that is a cubic polynomial passing through all the points.
3. Press `ESC` key to finish the line creation process (an example of a `NURBS` line is illustrated in figure A.2).

- **Arcs**: An arc is created by entering the coordinates of three points (`By 3 points`) or by entering a radius and the two tangent lines at the arc's ends (`Fillet curves`). The steps to create an arc by using three points are as follows:

  1. Select the option `Geometry` ⇒ `Create` ⇒ `Arc` ⇒ `By 3 points` or click on the icon in the toolbar.
  2. Enter three points to create an arc line. One can also select existing points to create arcs by using the key sequence `CTRL+a`.
  3. Press `ESC` key to finish the line creation process.

  The basic steps to create an arc by using a radius and the two tangent lines at the arc's ends are as follows:

  1. Select the menu `Geometry` ⇒ `Create` ⇒ `Arc` ⇒ `Fillet curves`.
  2. Enter the radius in the command line, and then select two lines that share one common point to create two tangent lines.
  3. Press `ESC` key to finish the line creation process. An arc is created and the two lines are modified to be tangent and continuous with this new arc (see figure A.3 for further details).

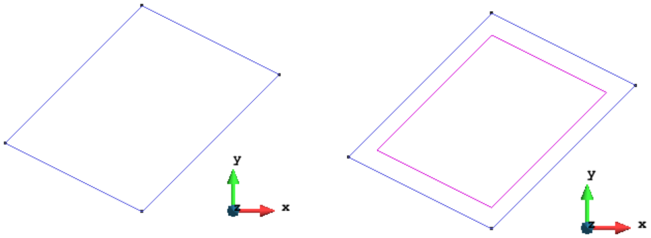


(a) Two lines (b) Arc created by the two lines

Figure A.3: Examples of a fillet curve

- `NURBS` **Surfaces:** The `NURBS` surfaces are created by selecting its contour lines. The steps to create a `NURBS` surface are as follows:

  1. Select the menu `Geometry` ⇒ `Create` ⇒ `NURBS surface` ⇒ `By contour` or click on the icon in the toolbar.
  2. Select the lines that will conform the contour of the surface.
  3. Press `ESC` key to finish the surface creation process. Figure A.4 shows an example of a `NURBS` surface.

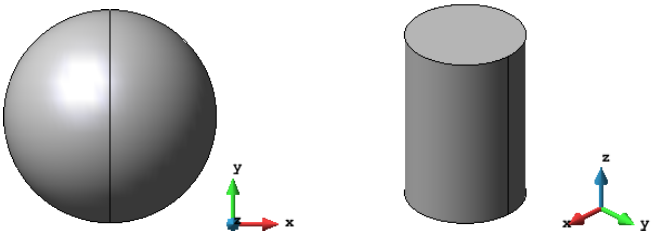


(a) Contour lines (b) NURBS surface

Figure A.4: Creation of a `NURBS` surface

- **Volumes:** The volumes are created by selecting its boundary surfaces. The steps to create a volume are as follows:

  1. Click on `Geometry` ⇒ `Create` ⇒ `Volume` ⇒ `By contour` or click on the icon in the toolbar.
  2. Select the surfaces that will conform the volume.
  3. Press `ESC` key to finish the volume creation process.



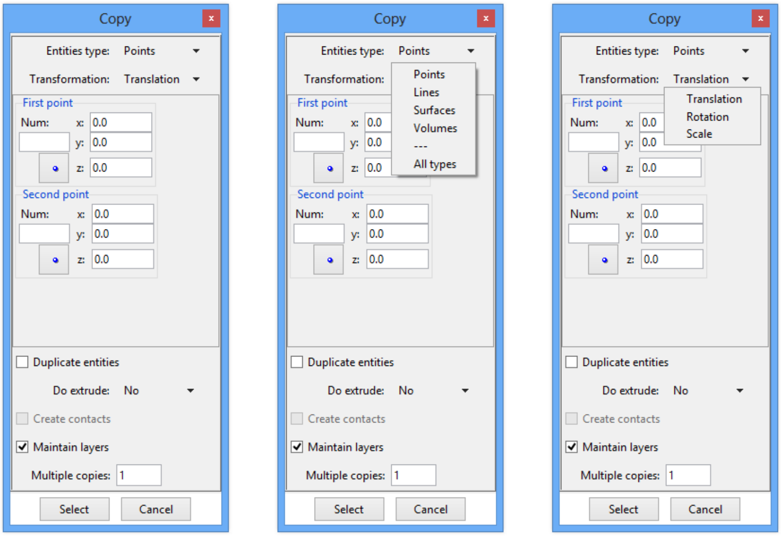
(a) Sphere (b) Cylinder

Figure A.5: Example of objects

- **Objects:** Several predefined objects such as rectangles, polygons, circles, sphere, cylinders, cones, prism and torus are available. When an object is created, the suite requires information about the center and the normal vector first. The coordinates of the center may be selected by clicking on the screen, by typing the coordinates in the command line or by selecting an existing point. To enter the normal vector, the suite displays a window where the vector may be selected through one of the three main axes or by typing the coordinates of a point. The steps to create a volume are as follows:

  1. Click on `Geometry` ⇒ `Create` ⇒ `Objects` or click on the icon in the toolbar.
  2. Select the object that will be created.
  3. Enter the center and the normal vector of the object.
  4. Enter the required information such as the radius or the length of the object. This information depends on the selected object.
  5. Press `ESC` key to finish the object creation process. Figure A.5 shows an example of several objects.

- **Copy/Move**: These tools allow to select a group of entities and perform a copy/movement using an operation, either `translation`, `rotation` or `scale`. The entity types include points, lines, surfaces and volumes. The copy/move window is shown in figure A.6(a), while the drop-down menu showing the entities is given in figure A.6(b) and the drop-down menu for the transformation operation is illustrated in figure A.6(c).



Copy

Entities type: Points

Transformation: Translation

First point

Num: x: 0.0 y: 0.0 z: 0.0

Second point

Num: x: 0.0 y: 0.0 z: 0.0

□ Duplicate entities

Do extrude: No

□ Create contacts

☑ Maintain layers

Multiple copies: 1

(a) window (b) entities (c) operations

Figure A.6: Screenshot of the copy/move window

The copy/move tools are found by clicking on the menu `Utilities` ⇒ `Copy/Move`. The transformation operation are described below:

1. `Translation:` This operation performs a translation from one point to another. Relative movements can be obtained by defining the first point as (0, 0, 0) and considering the second point as the translation vector.
2. `Rotation:` This operation performs a rotation of the geometry. Two points defines the axis of rotation and its orientation. The rotation angle in degrees is also required; it can be positive and negative. The direction of rotation is defined by the right hand rule. Figure A.7 shows an example of the rotation operation.
3. `Scale:` This operation performs a scale of the geometry. This operation is defined by a manipulation center point and a vector scale factor for x, y and z-axis. A scale factor greater than one increases the size, while a scale factor less than one decreases the size. If the scale factor is negative, a change of sign in the coordinates will be performed. Figure A.8 shows an example of the scale operation.

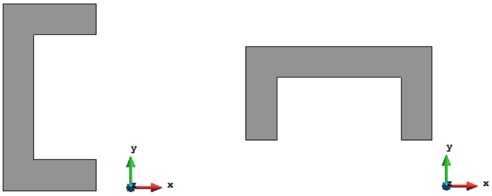


Figure A.7: Example of the rotation operation

A.1.2 Example

A tutorial about the creation of the geometry model of a x-band waveguide tee junction (see figure A.11) is given in this section. The units of the model are expressed in mm. The steps to create the complete model of the tee junction are detailed below.

1. Select mm as geometry units in the top-right combobox.
2. Click on `Geometry` ⇒ `Create` ⇒ `straight line` menu, or click on the icon ╲ in the toolbar.
3. Enter the coordinates of the points that will conform the base of the tee junction. The coordinates of the points are (-11.43, 50.81, 0.0), (-11.43, -50.81, 0.0), (11.43, -50.81, 0.0) (11.43, -11.43, 0.0), (50.81, -11.43, 0.0), (50.81, 11.43, 0.0), (11.43, 11.43, 0.0), (11.43, 50.81, 0.0) and, finally enter the initial point (-11.43, 50.81, 0.0).
4. Click on Join to use the existing point.
5. Press `ESC` key to finish the line creation process. Figure A.9 shows the lines that conforms the base of the tee junction.

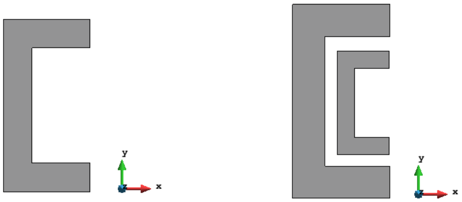
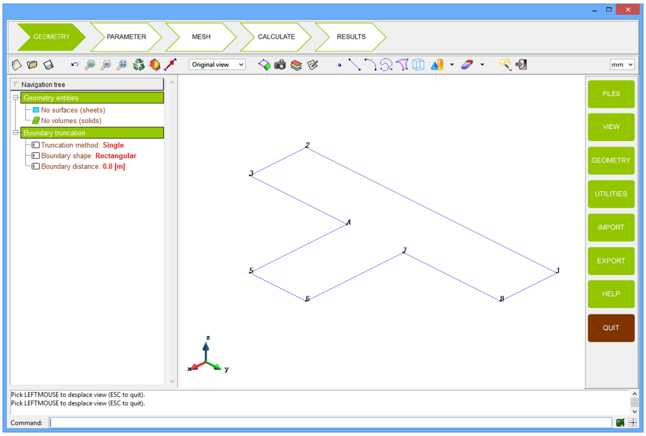


Figure A.8: Example of the scale operation



Navigation tree
- Geometry entities
  - No surfaces (sheets)
  - No volumes (solids)
- Boundary truncation
  - Truncation method: Single
  - Boundary shape: Rectangular
  - Boundary distance: 0.0 [m]

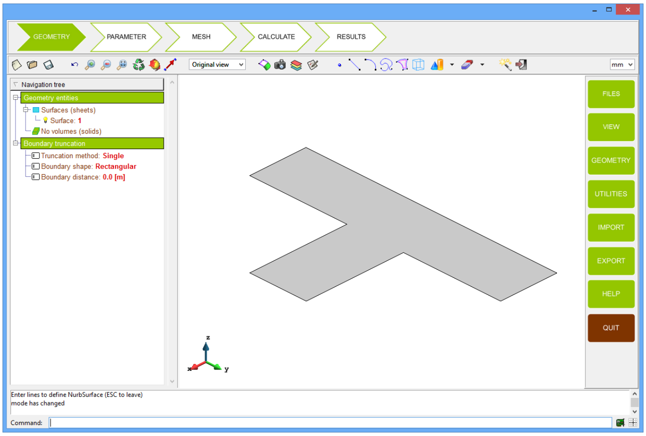
```
Pick LEFTMOUSE to desplace view (ESC to quit).
Pick LEFTMOUSE to desplace view (ESC to quit).
Command:
```

Figure A.9: Lines that conforms the base of the tee junction

6. The next step is to create the `NURBS` surface by selecting the contour of the base. Click on `Geometry` ⇒ `Create` ⇒ `NURBS surface` ⇒ `By contour` menu or click on the icon in the toolbar.

7. Select all the line created in previous steps with the mouse. The identifier of all these lines may also be typed on the command line to proceed with the selection.

8. Press `ESC` key to finish the surface creation process. Figure A.10 shows the base of the tee junction.



Navigation tree
- Geometry entities
  - Surfaces (sheets)
    - Surface: 1
  - No volumes (solids)
- Boundary truncation
  - Truncation method: Single
  - Boundary shape: Rectangular
  - Boundary distance: 0.0 [m]

```
Enter lines to define NurbSurface (ESC to leave)
mode has changed
Command:
```

Figure A.10: Base of the tee junction

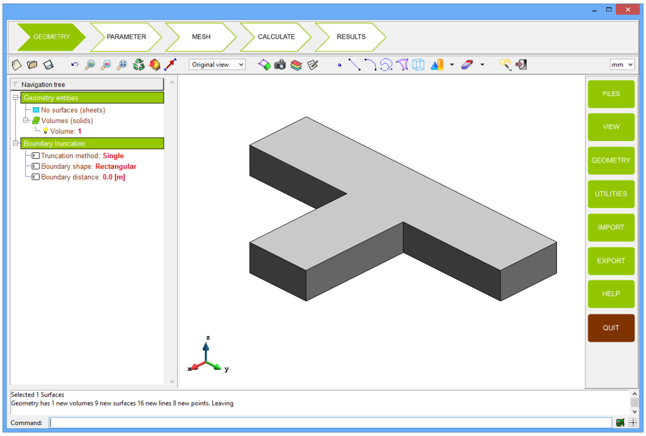
9. To complete the geometry model of the tee junction, an extrude operation is performed. Click on `Utilities` ⇒ `Copy` menu. The copy window will be opened.

10. Select `Surfaces` in the `Entities type` option and `Translation` in the `Transformation` entry. The translation vector goes from (0, 0, 0) to (0, 0, 10.16). Select `Volumes` in the `Do extrude` option.

11. Press the `Select` button and select the base of the tee junction.

12. Press `ESC` key to finish the copy creation process or click on the `Finish` button of the copy window.

13. The creation of the tee junction is completed. Figure A.11 shows the geometry model of the structure.



Geometry entities
- No surfaces (sheets)
- Volumes (solids)
  - Volume: 1

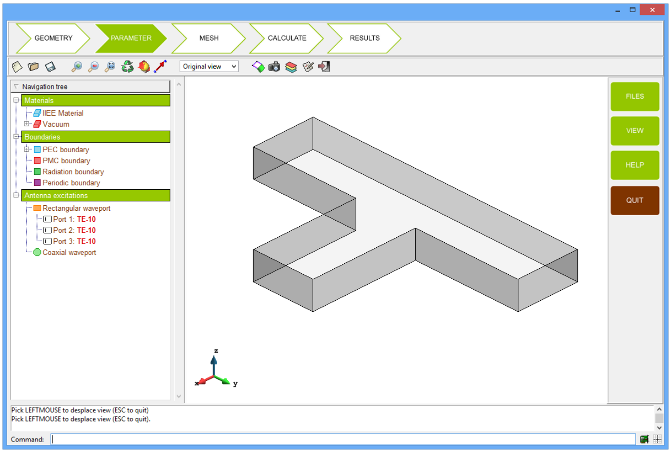
Boundary truncation
- Truncation method: Single
- Boundary shape: Rectangular
- Boundary distance: 0.0 [m]

```
Selected 1 Surfaces
Geometry has 1 new volumes 9 new surfaces 16 new lines 8 new points. Leaving
```

Figure A.11: Geometry of the tee junction complete

A.2 Step 2: Setting up electromagnetic parameters

Once the geometry is completed, the user can start to set up the electromagnetic parameters needed to perform the simulation. Those electromagnetic parameters include specifying not only the parameters shared by all the modules of the suite, such as, materials or excitations, but also particular parameters such as, boundary conditions in the case of the `FEM` module or loadings in the case of the `MoM` module.



```
Pick LEFTMOUSE to desplace view (ESC to quit)
Pick LEFTMOUSE to desplace view (ESC to quit).
```

Figure A.12: Parameter step of the suite for the `FEM` module

Figure A.12 shows the graphical aspect of the suite for this second step. The following sections describe how to manipulate the materials, the boundary conditions and the excitations in the suite.

A.2.1 Materials

The suite provides several tools to manipulate the materials of an electromagnetic problem. It is worth to note that there are two materials created by default in the `FEM` module of the suite: `IIEE material` (the material between the object and the radiation condition) and `Vacuum`. The tools to manipulate the material in the suite are described below.

- **Creating a new material:** To create a new material, click with the right button on the navigation tree item called `Materials`. Figure A.13 shows the result of clicking with the right button on that item.

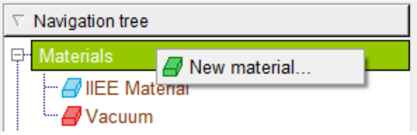
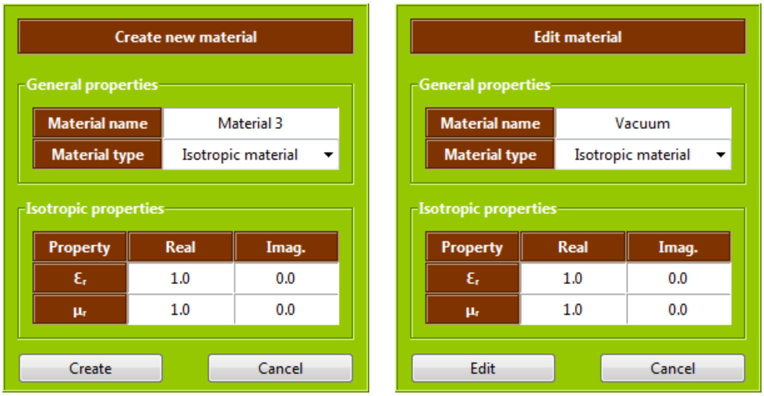


Figure A.13: New material option



Create new material

General properties

Material name	Material 3
Material type	Isotropic material

Isotropic properties

Property	Real	Imag.
ε_r	1.0	0.0
μ_r	1.0	0.0

(a) Creating panel

Edit material

General properties

Material name	Vacuum
Material type	Isotropic material

Isotropic properties

Property	Real	Imag.
ε_r	1.0	0.0
μ_r	1.0	0.0

(b) Editing panel

Figure A.14: Creating/editing material panel

Click on the `New material` option. The creating material panel will appear in the screen (see figure A.14(a)) The first parameter is the name of the material. The second parameter defines the type of the material: isotropic or anisotropic. The electromagnetic properties are defined at the bottom of the panel. To finish the creation process, click on the `Create` button.

- **Editing an existing material:** To edit an existing material, click with the left button of the mouse on the name of any existing material. The editing material panel will appear in the screen (see figure A.14(b)). To confirm the modification of any property of an existing material, click on the `Edit` button. Otherwise, click on `Cancel`.
- **Deleting an existing material:** To delete an existing material, click with the left button on the name of the desired material. Then, click with the right button to deploy the drop-down menu. Select the `Delete` option to finish the deletion process.

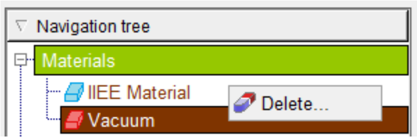


Figure A.15: Delete material option

- **Assigning an existing material:** The materials are assigned to the volume entities of the geometry model. If there are volumes with no material assigned, they will appear under the navigation tree item called `Undefined`. Figure A.16 illustrates an example where there are two volumes with no material assigned.

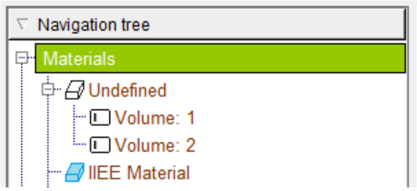


Figure A.16: Volumes with no material assigned

Click with the left button on the desired volumes to perform a selection. Then, click with right button to deploy the drop-down list with the existing material (see figure A.17). Select the material with the left button of the mouse to finish the assignation. It is worth to note that several volumes may be selected at the same time by using the appropriate key modifiers `CRTL` or `SHIFT`.

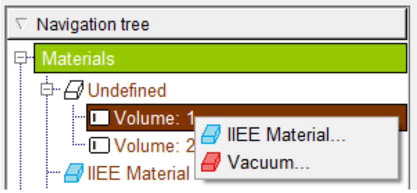


Figure A.17: Assigning materials

- **Unassigning an existing material:** To unassign the material from any volume, select the volume by clicking it with the left button. Now, click with the right button to open the drop-down list (see figure A.18). Select the `Undefined` option with the left button of the mouse to finish the process.

- **Visualizing an existing material:** The suite draws the volumes belonging to an existing material by colors for a better user experience in the manipulation of materials. Click with the left button on any existing material. The volumes belonging to that material will appear colored in the screen. Figure A.19 illustrates an example of the visualization of an existing material.

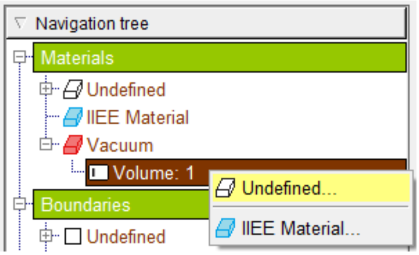
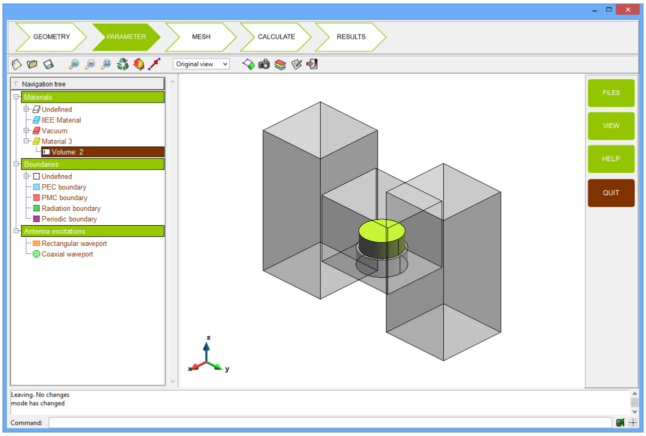


Figure A.18: Unassign materials

A.2.2 Boundary conditions

Four boundary conditions are available to set up the electromagnetic problem. These boundary conditions are Perfect Electric Conductor (`PEC`), Perfect Magnetic Conductor (`PMC`), Radiation condition and Periodic Boundary Condition (`PBC`). The tools to manipulate the boundary conditions in the suite are described below.

- **Assigning boundary conditions:** The boundary conditions are assigned to the surface entities of the geometry model. If there are surfaces with no boundary condition assigned, they will appear under the item called `Undefined`. Figure A.20 illustrates an example where there are two surfaces with no boundary condition assigned.



```
Leaving. No changes
mode has changed
```

Figure A.19: Visualizing an existing material

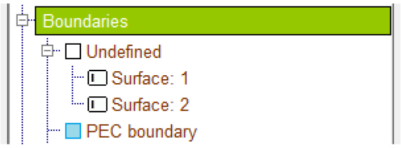


Figure A.20: Surfaces with no boundary condition assigned

To assign any boundary condition, click with the left button on the desired surface to perform the selection. Note that several surfaces may be selected by using the appropriate key modifiers `CRTL` or `SHIFT`. Then, click with the right button to deploy the drop-down list with the boundary conditions available (see figure A.21). Select the boundary condition with the left button to finish the assignation. Note that the particular case of periodic boundary condition requires the selection of two surfaces to assign the boundary condition.

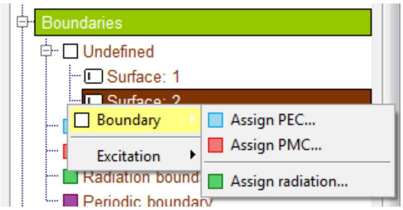


Figure A.21: Assign boundary conditions

- **Unassigning boundary conditions:** To unassign the boundary condition from any surface, click with the left button on the desired surface (`CRTL` or `SHIFT` may be used to select multiple surfaces). Now, click with the right button to open the drop-down list (see figure A.22). Select the `Undefined` option with the left button of the mouse to finish the process.

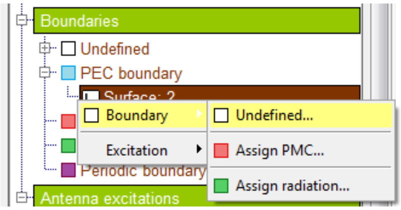
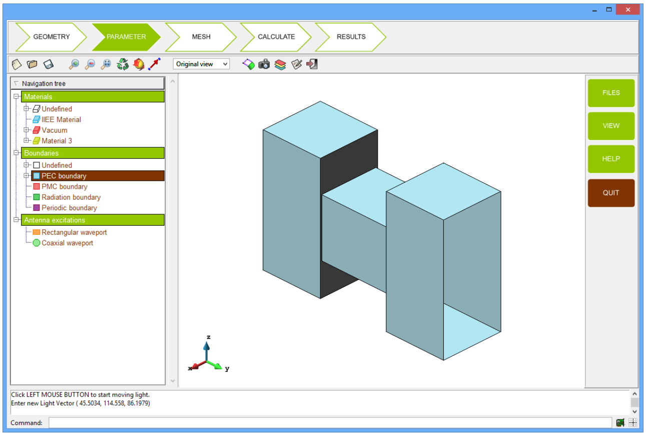


Figure A.22: Unassign boundary conditions



Click LEFT MOUSE BUTTON to start moving light.
Enter new Light Vector (45.5034, 114.558, 86.1979)

Figure A.23: Visualizing `PEC` boundary condition

- **Visualizing boundary conditions:** The suite draws the surfaces with a certain boundary condition assigned by colors for a better user experience in the manipulation of boundary conditions. Click with the left button of the mouse on any boundary conditions. The surfaces which have assigned that boundary condition will be colored in the screen. Figure A.23 illustrates an example of the visualization of boundary conditions.

A.2.3 Excitations

The current version of the suite offers three different sources that users can use to excite their electromagnetic problems: rectangular waveport, coaxial waveport and plane waves. The tools to manipulate these excitations are described below.

- **Creating a rectagular waveport:** This waveport is assigned to rectangular surface entities of the geometry model. If the surfaces are not rectangular, when creating the waveport an error is given. To create the waveport, click with the left button on the desired surface to perform the selection. Then, click with the right button to deploy the drop-down menu with the excitations list (see figure A.24). Select `Excitation` ⇒ `Assign waveport` option with the left button to finish the assignation.

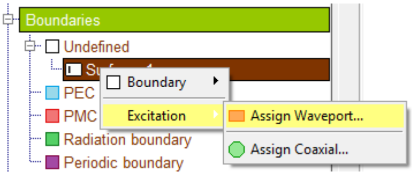


Figure A.24: Assign rectangular waveport

- **Creating a coaxial waveport:** The coaxial waveport is assigned to circular surfaces entities (with an internal hole) of the geometry model. If the surfaces do not have that shape, when creating the waveport an error is given. An example of a surface entity that supports the coaxial waveport assignation is given in figure A.25.

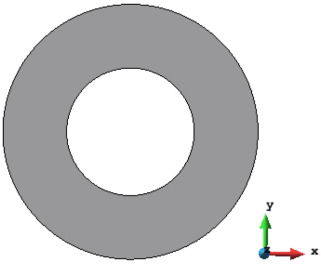


Figure A.25: Example of coaxial waveport

To create a coaxial waveport, select the desired surface by clicking with the left button on it. Then, click with the right button to open the drop-down menu with the excitations list (see figure A.26). Select `Excitation` ⇒ `Assign coaxial` option with the left button to finish the assignation.

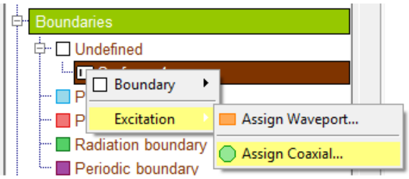
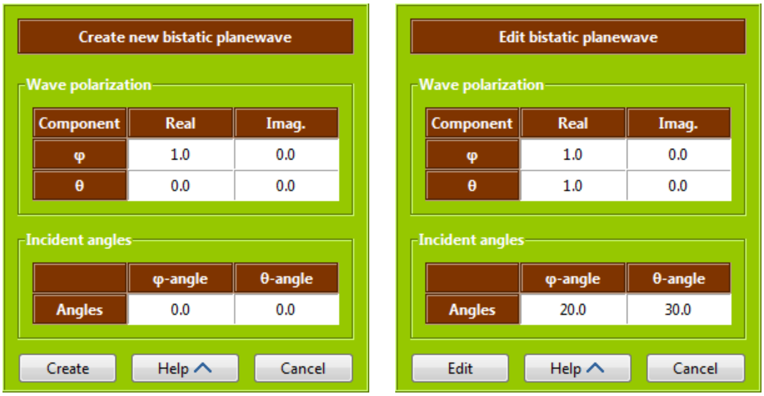


Figure A.26: Assign coaxial waveport



Create new bistatic planewave

Wave polarization

Component	Real	Imag.
φ	1.0	0.0
θ	0.0	0.0

Incident angles

	φ-angle	θ-angle
Angles	0.0	0.0

(a) Creating panel

Edit bistatic planewave

Wave polarization

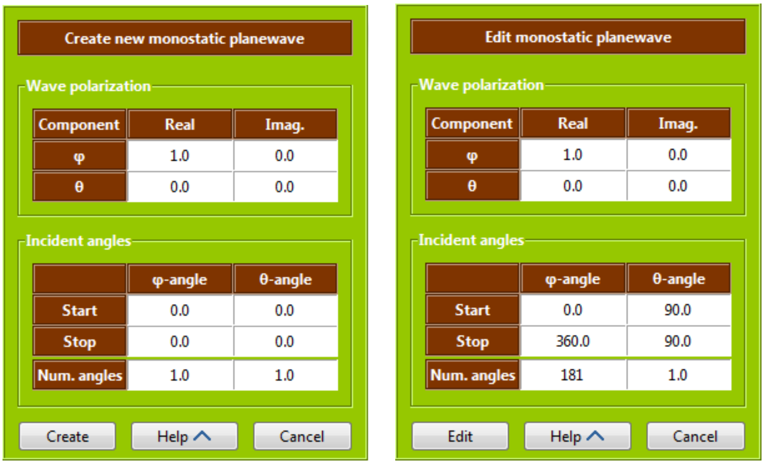
Component	Real	Imag.
φ	1.0	0.0
θ	1.0	0.0

Incident angles

	φ-angle	θ-angle
Angles	20.0	30.0

(b) Editing panel

Figure A.27: Creating/editing bistatic wave panel



Create new monostatic planewave

Wave polarization

Component	Real	Imag.
φ	1.0	0.0
θ	0.0	0.0

Incident angles

	φ-angle	θ-angle
Start	0.0	0.0
Stop	0.0	0.0
Num. angles	1.0	1.0

(a) Creating panel

Edit monostatic planewave

Wave polarization

Component	Real	Imag.
φ	1.0	0.0
θ	0.0	0.0

Incident angles

	φ-angle	θ-angle
Start	0.0	90.0
Stop	360.0	90.0
Num. angles	181	1.0

(b) Editing panel

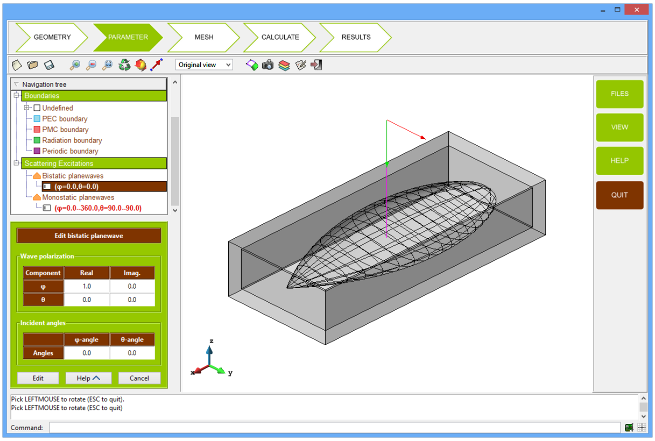
Figure A.28: Creating/editing monostatic wave panel

- **Creating a plane wave:** Two different types of plane wave may be created: bistatic and monostatic waves. To create a new plane wave, click with the left button on the bistatic/monostatic plane wave navigation tree item. The creating panel for bistatic/monostatic waves will be shown in the screen. Figures A.27(a) and A.28(a) show the creating panel for the bistatic/monostatic waves, respectively. The first parameters define the polarization of the wave and the bottom entries of the panel define the incident angle. In the case of monostatic waves, a set of incident angles is required. To finish the creation of the new plane wave, click on the `Create` button.

- **Editing an existing excitation:** To edit an existing excitation, click with the left button on the navigation tree item of that excitation. The editing panel of the corresponding excitation will appear. For example, the editing panels for bistatic/monostatic waves are shown in figure A.27(b) and A.28(b), respectively. To finish the modification click on the `Edit` button.

- **Deleting an existing excitation:** To delete an existing excitation, click with the left button of the mouse on the desired excitation. Then, click with the right button to deploy the drop-down menu. Select the `Delete` option to finish the process.

- **Visualizing an excitation:** To visualize an existing excitation, click with the left button on the navigation tree item of that excitation and it will be shown in the screen. In the case of waveports, the surfaces which have assigned that excitation will be colored. In the case of plane wave, arrows indicating the incident direction and the polarization will be shown. Figure A.29 shows an example of visualizing a bistatic plane wave.



Navigation tree
- Boundaries
  - □ Undefined
  - PEC boundary
  - PMC boundary
  - Radiation boundary
  - Periodic boundary
- Scattering Excitations
  - Bistatic planewaves
    - (φ=0.0,θ=0.0)
  - Monostatic planewaves
    - (φ=0.0--360.0,θ=90.0--90.0)

Edit bistatic planewave

Wave polarization

Component	Real	Imag.
φ	1.0	0.0
θ	0.0	0.0

Incident angles

	φ-angle	θ-angle
Angles	0.0	0.0

Edit | Help | Cancel

```
Pick LEFTMOUSE to rotate (ESC to quit).
Pick LEFTMOUSE to rotate (ESC to quit)
Command:
```

Figure A.29: Visualizing a bistatic plane wave

A.2.4 Example

Following with the tutorial about the simulation of the x-band waveguide tee junction, lets set up the electromagnetic parameters of the problem. The steps to set up the materials, the boundary conditions and the excitations are as follows:

1. Select the `Parameter` step from the top menu of the suite.

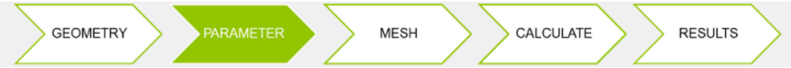
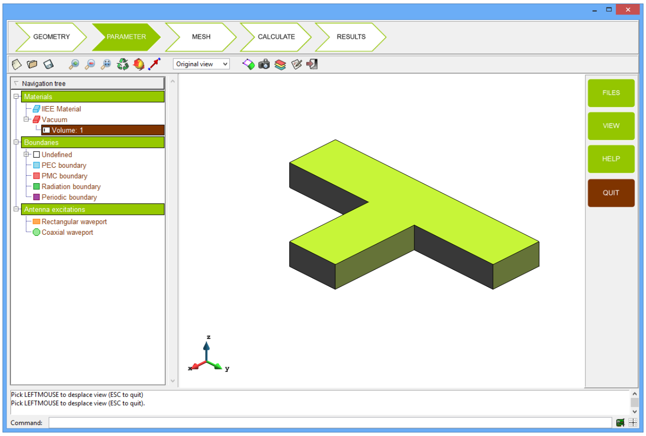


Figure A.30: `Parameter` step selection

2. The tee junction is filled by air. To assign the material, expand the undefined material list by clicking on the corresponding cross of the `Undefined` material item.

3. Click with the left button on the volume 1 to select the volume. Click with the right button to deploy the drop-down menu with the material list. Select `Vacuum` to assign the material. Figure A.31 shows the material assignation of the tee junction.



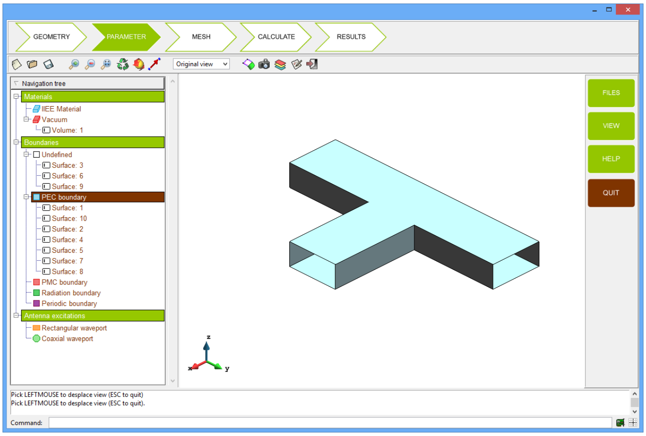
```
Pick LEFTMOUSE to desplace view (ESC to quit)
Pick LEFTMOUSE to desplace view (ESC to quit).
Command:
```

Figure A.31: Tee junction material

4. Now, lets assign the boundary conditions to the structure. Expand the undefined boundary condition list by clicking on the corresponding cross of the `Undefined` boundary condition item.

5. Surfaces 1,2,4,5,7,8 and 10 have a boundary condition of `PEC`. Click with the left button on these surfaces to select them. To perform a continuous selection maintain the modifier key `CRTL` pressed.

6. Click with the right button to deploy the drop-down menu. A list with the boundary condition will appear. Select the `Boundary` ⇒ `PEC` option with the left button of the mouse. Figure A.32 shows the visualization of the `PEC` boundary condition for tee junction.



```
Pick LEFTMOUSE to desplace view (ESC to quit)
Pick LEFTMOUSE to desplace view (ESC to quit).
```

Figure A.32: `PEC` boundary condition for tee junction

7. Surfaces 3, 6 and 9 are the ports of the structure. To assign the first rectangular waveport, click with the left button on the item of surface 3. Click with the right button to deploy the drop-down menu with the excitation list. Select the `Excitation` ⇒ `Assign waveport` option to assign the waveport (see figure A.33).

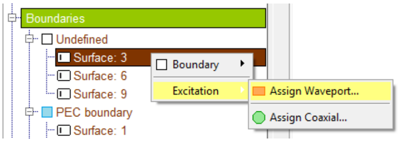
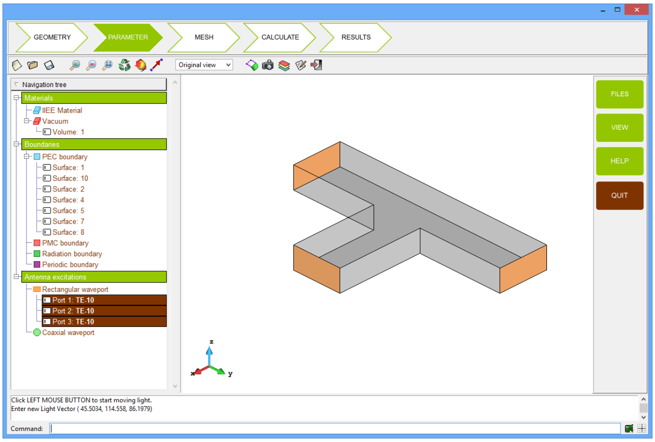


Figure A.33: Rectangular waveport assignation

8. Repeat the previous process with the surfaces 6 and 9.

9. The set up of the electromagnetic parameters of the tee junction is completed. Figure A.34 shows the state of the project at this point.



Navigation tree
- Materials
  - IIEE Material
  - Vacuum
    - Volume: 1
- Boundaries
  - PEC boundary
    - Surface: 1
    - Surface: 10
    - Surface: 2
    - Surface: 4
    - Surface: 5
    - Surface: 7
    - Surface: 8
  - PMC boundary
  - Radiation boundary
  - Periodic boundary
- Antenna excitations
  - Rectangular waveport
    - Port 1: TE-10
    - Port 2: TE-10
    - Port 3: TE-10
  - Coaxial waveport

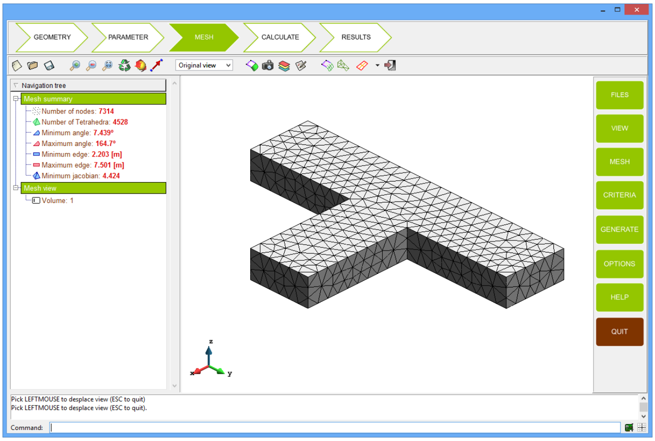
```
Click LEFT MOUSE BUTTON to start moving light.
Enter new Light Vector ( 45.5034, 114.558, 86.1979)
```

Command:

Figure A.34: Electromagnetic parameter set up complete

A.3 Step 3: Meshing models

Generating a mesh is the process by which a mesh is generated from the geometry definition. The meshing process is an indispensable operation before running a simulation. In the suite, all the properties assigned to geometry entities, such as materials, boundary conditions, excitations and so on, will be transferred to the nodes and elements of the mesh. Figure A.35 shows the graphical aspect of the suite in this step.



Navigation tree
- Mesh summary
  - Number of nodes: 7314
  - Number of Tetrahedra: 4528
  - Minimum angle: 7.439°
  - Maximum angle: 164.7°
  - Minimum edge: 2.203 [m]
  - Maximum edge: 7.501 [m]
  - Minimum jacobian: 4.424
- Mesh view
  - Volume: 1

```
Pick LEFTMOUSE to desplace view (ESC to quit)
Pick LEFTMOUSE to desplace view (ESC to quit).
```

Command:

Figure A.35: Graphical aspect of the mesh step of the suite

The suite makes use of the tools provided by GiD for meshing models. These tools gather automatic methods for structures such as Non-Uniform Rational B-Splines (`NURBS`) curves or surfaces. Some of the most important tools are summarized in this section.

A.3.1 Unstructured mesh

The following set of tools generates unstructured meshes where the user can assign different sizes to different entities of the mesh and control the maximum distance between the generated element and the real geometry. These tools are found in the menu `Mesh` ⇒ `Unstructured`. More details about these tools are given below.

- **Assign size on points, lines, surfaces and volumes:** It is possible to assign different sizes to different entities of the mesh. This means that in the vicinity of these entities, the generated elements will be approximately of that size. The default value of this tool is 0.0 which means that the suite automatically will choose the best size for the entities of the mesh. This option is accessible from the menu: `Mesh` ⇒ `Unstructured` ⇒ `Assign sizes on points/lines/surfaces/volumes`. An example of an unstructured mesh is illustrated in figure A.36 where elements with different sizes are appreciated.

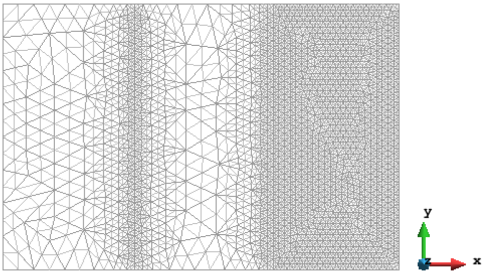
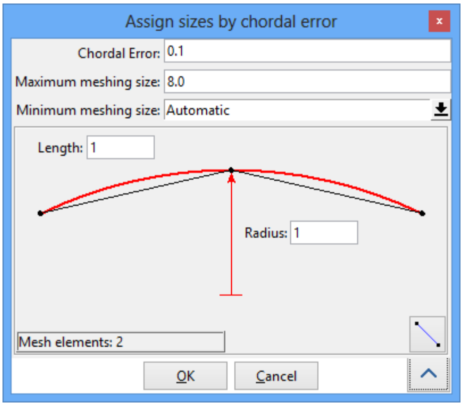


Figure A.36: Example of an unstructured mesh

- **Sizes by chordal error:** This option contains fields for chordal error (the maximum distance between the generated element and the real geometry) and minimum and maximum size limits. The window that controls this option is shown in figure A.37.



Assign sizes by chordal error

Chordal Error: 0.1

Maximum meshing size: 8.0

Minimum meshing size: Automatic

Length: 1

Radius: 1

Mesh elements: 2

Figure A.37: Assign sizes by chordal error window

The suite assigns the corresponding sizes to all the entities to satisfy this condition. It will change the current sizes if the new one is smaller than the one defined previously. This option can be found from the menu: `Mesh` ⇒ `Unstructured` ⇒ `Sizes by chordal error`. Figure A.38 shows an example of this type of mesh.

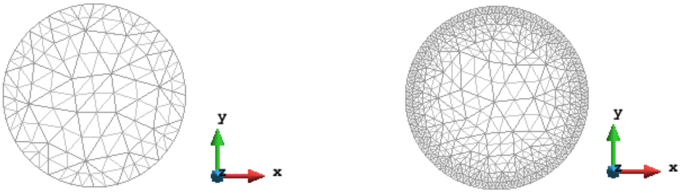


Figure A.38: Unstructured meshes with different chordal error

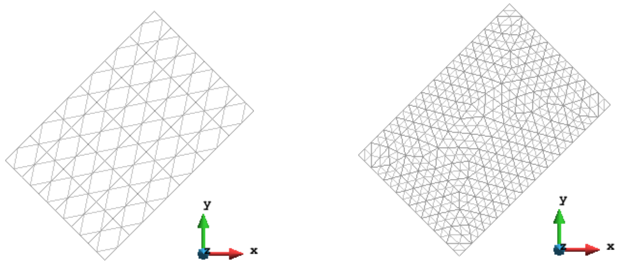
A.3.2 Structured mesh

This set of tools generates structured meshes dividing the geometry by number of cells or by using a given size. These tools are found in the `Mesh` ⇒ `Structured` menu. More details about these tools are given below.

- **Structured mesh by number of cells:** Choose this option to create a structured mesh dividing the lines/edge of the four-sized surface in the number of cell desired

After choosing this option, the window where one sets the number of cells is shown. Once the number of cell is selected, the lines/edges must be selected. Note that for surfaces/volumes, the user needs to select the surface/volume and press `ESC` before setting the number of the cells for its edges. This process can be repeat as many times as necessary until all lines have a new value.

- **Structured mesh by size:** This option creates a structured mesh using the same size for all the elements. After choosing this option, the window where one sets the size of the elements is shown. Once the mesh size is selected, the lines/edges must be selected. Note that the user needs to select the geometry entities and press `ESC` key before setting the size of the elements for its edges. This process can be repeated as many times as necessary until all lines have a new value. Four-sided structured and unstructured mesh for a `NURBS` surface are compared in figure A.39.



(a) Structured mesh

(b) Unstructured mesh

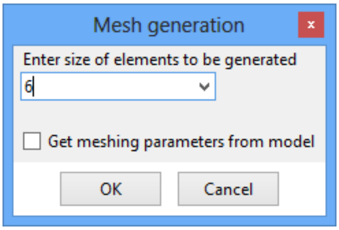
Figure A.39: Meshes of a four-sided `NURBS` surface

A.3.3 Example

Continuing with the tutorial about the simulation of the x-band waveguide tee junction, lets generate the mesh of the model. The steps to generate the mesh are as follows:

1. Select the `Mesh` item from the top menu of the suite to enter in the meshing step.
2. To generate the default mesh provided by the suite just click on the `Generate` option of the right menu.

3. After choosing this option, the window where one sets the size of the elements is shown (see figure A.40).


Mesh generation

Enter size of elements to be generated

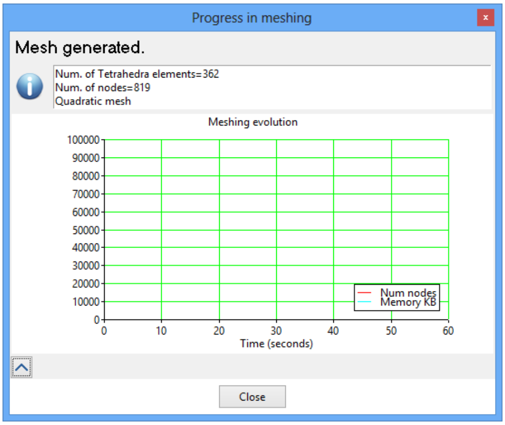
6

□ Get meshing parameters from model

Figure A.40: Mesh generation window

4. Type a number or leave the size by the default. Elements with a size of 8 mm produce a mesh with 362 tetrahedrons. This mesh may be generated by the evaluation version of the suite. Click the `OK` button to generate the mesh.

5. Wait until the `Mesh Info Window` is shown indicating that the meshing process has been finished (see figure A.41). Click on the `Close` button to close the window.

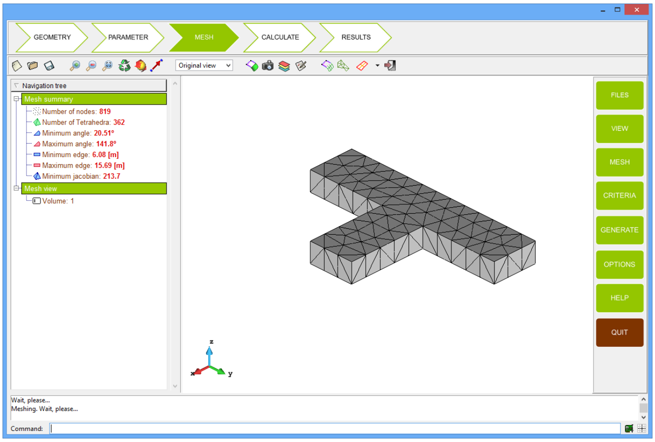

Progress in meshing

Mesh generated.

Num. of Tetrahedra elements=362
Num. of nodes=819
Quadratic mesh

Figure A.41: Mesh generation window

6. The mesh has been generated. Figure A.42 shows the state of the project at this point of the simulation.



Navigation tree

Mesh summary

- Number of nodes: **819**
- Number of Tetrahedra: **362**
- Minimum angle: **20.51°**
- Maximum angle: **141.8°**
- Minimum edge: **6.08 [m]**
- Maximum edge: **15.69 [m]**
- Minimum jacobian: **213.7**

Mesh view

- Volume: 1

```
Wait, please...
Meshing. Wait, please...
```

Command:

Figure A.42: Mesh of the waveguide tee junction

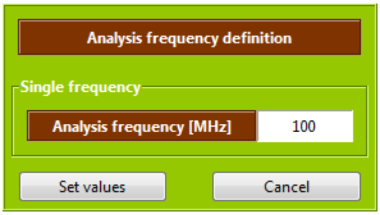
A.4 Step 4: Running a simulation

After the mesh is generated, the problem is ready to be simulated by one of the solver modules of the suite. In this step of the simulation, some parameters are required to be set such as the working frequency and the solver options. These parameters are described in the following subsections.

A.4.1 Working frequency

The suite offers two different modes to set up the working frequency of a simulation. If the user only wants to perform the analysis at one single frequency, the suite provides the `Single frequency` mode. Conversely, if the user wants to perform the analysis in a range of frequencies, the suite provides the `Frequency sweep` mode. The description of both modes is given below.

- **Single frequency:** To set up the desired working frequency using this mode, click with the left button on the navigation tree item called `Analysis frequency`. The definition panel will be shown in the screen (see figure A.43) Enter the desired frequency in the entry box and click on the `Set values` button to confirm the set up.



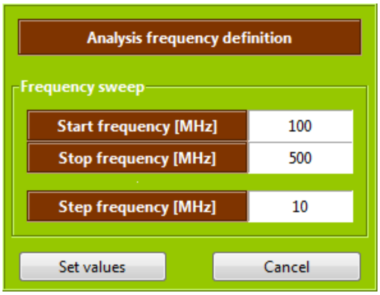
Analysis frequency definition

Single frequency

Analysis frequency [MHz]: 100

Figure A.43: Single frequency definition panel

- **Frequency sweep:** To set up the range of frequencies where the analysis will be carried out, click with the left button on any of the items of the navigation tree under the `Frequency sweep` option (start/stop/step frequency). A panel where to set the frequencies will be shown (see figure A.44). The first two entry boxes are used to set up the start and stop frequencies of the analysis. The last entry box is used to set up the step frequency of the analysis. Finally, click on the `Set values` button to confirm the set up. It is worth to note that a constant step frequency between two consecutive analysis is used in this mode.



Analysis frequency definition

Frequency sweep

Start frequency [MHz]: 100

Stop frequency [MHz]: 500

Step frequency [MHz]: 10

Figure A.44: Frequency sweep definition panel

For instance, if a user wants to perform a simulation from 100 `MHz` to 500 `MHz` with an step of 10 `MHz` between the working frequency of two consecutive analysis just need to type 100, 500 and 10 in the entry boxes, respectively.

It is worth to note that the `Single frequency` mode is selected by default when a user creates a new project. To change the frequency mode, click with the right button on the navigation tree item called `Single frequency`. A drop-down menu will be shown (see figure A.45).

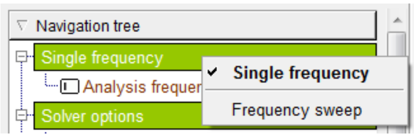


Figure A.45: Drop-down menu to change the frequency mode

Click on the `Frequency sweep` option to change the frequency mode. To stay on the single frequency mode click on the `Single frequency` option. If the frequency sweep mode is the active one, to change the frequency mode click with the right button on the navigation tree item called `Frequency sweep`.

The frequency units may be changed by selecting the desired units in the drop-down menu placed in the top-right corner of the main window. Figure A.46 shows the exact location of this drop-down menu.

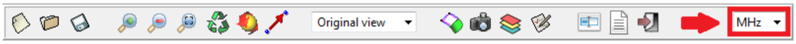


Figure A.46: Frequency units location

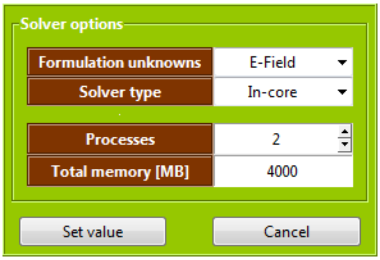
A.4.2 Solver options

The suite provides several options to configure the different electromagnetic solvers. There are some common options to all the solvers such as, the solver type (in-core/out-of-core), the number of processes per simulation or the parallel environment initialization. To set up these options, click with the left button in any of the navigation tree options under the `Solver options` item. A common definition panel will be shown (see figure A.47). Details about the configuration options are given below. In this case, the options of the `FEM` solver of the suite are also described.

- **Solver type:** This option controls the type of the solver employed in the simulation. The solvers available are:

  In-core: The in-core solver only uses the `RAM` memory available in the computer to solve the problem.

  Out-of-core: The out-of-core solver uses the `RAM` memory available in the computer in addition to the harddisk to solve the problem.



Solver options

Formulation unknowns: E-Field

Solver type: In-core

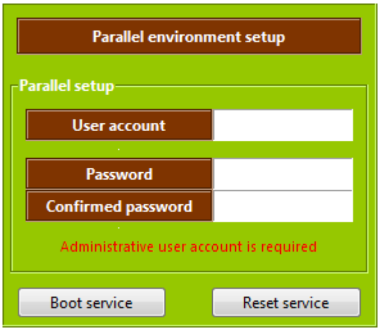
Processes: 2

Total memory [MB]: 4000

Figure A.47: Common definition panel to all the solver options

- **Processes:** This option sets up the number of processes used in a single computer parallel simulation. To set up this option, select the number of desired processes using the mouse and the spin box of the corresponding entry of the panel. Finally, click on the `Set values` button to confirm the selection of the number of processes of the simulation.

- **Parallel environment:** Parallel simulations require the use of specific libraries in order to distribute the problem in several processes. Thus, the parallel environment needs to be booted before running a parallel simulation. To do that, click with the left button to the parallel environment item of the navigation tree. A new panel will be opened as the shown in figure A.48



Parallel environment setup

Parallel setup

User account:

Password:

Confirmed password:

Administrative user account is required

Figure A.48: Parallel environment setup panel

If the status of the environment is `Unbooted`, the user should register the service. Input a user account with administrative rights and the corresponding password. If the account does not have a password, the user has to create one. Click on `Boot service` and the environment setup is completed when the status changes to `Booted`. If the user wants to reboot the environment, click `Reset service` and register it again.

- **Formulation:** This option is a particular option of the `FEM` solver. The user may select two different formulation: `E-field/H-field`. The E-field formulation is selected by default. To set up this option, select the desired formulation in the corresponding drop-down menu.
- **Total memory:** This option is also a particular option of the `FEM` solver. This is the total memory that the solver will use for the solving process. To set up this option, type a positive real number in `MB` in the corresponding entry of the panel.

A.4.3 Serial/Parallel simulation

The suite has two different simulations common to all the modules: serial and parallel simulations. The serial simulation runs only one process at one time on a single computer. The parallel simulation can run multiple processes on multiple cores of a single computer. It is worth to note that the suite supports simulations in `HPC` cluster where multiple processes on several compute nodes are used at the same time. Details about these simulations are given below.

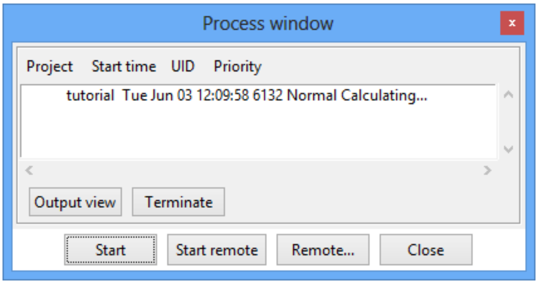
- **Serial:** The user may run the serial simulation by selecting the `Serial` option from the right menu of the suite (see figure A.49). If the amount of available physical memory on the computer is not enough for the simulation, the solver will return an error.

Figure A.49: Serial option from the right menu of the suite

- **Parallel:** Running the parallel simulation is similar to running the serial. To run the parallel simulation, select the `Parallel` option on the right menu of the suite.

Figure A.50: Parallel option from the right menu of the suite

The parallel solver uses the paradigm `MPI` to distribute the simulation into several processes. For that reason, the parallel environment must be booted before run the simulation. The number of processes used in the simulation is given by the solver option called `Processes`.


Process window

Project	Start time	UID	Priority	
tutorial	Tue Jun 03 12:09:58	6132	Normal	Calculating...

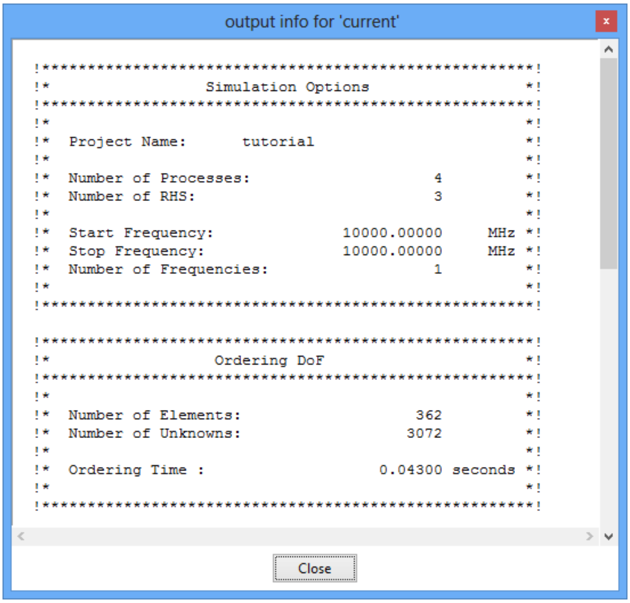
Figure A.51: Process window

Once the simulation starts, the process window appears as shown in figure A.51. The window includes the project name, start time, `UID` and priority of the simulation. To stop the simulation, click the `Terminate` button. The `Close` button closes the process window but does not terminate the simulation. The `Output view` button allows the user to check the detailed process of the simulation. Click the `Output view` button and an information window appears (see figure A.52).

A.4.4 Example

Lets analyze the x-band waveguide tee junction created in the previous steps. The analysis will be performed at 10 `GHz`. The steps to simulate the project are as follows:

1. Select the `Calculate` item from the top menu of the suite to enter in the calculation step.
2. Change the default frequency units from `MHz` to `GHz`.



```
!*******************************************************!
!*                 Simulation Options                  *!
!*******************************************************!
!*                                                     *!
!*  Project Name:      tutorial                        *!
!*                                                     *!
!*  Number of Processes:                    4          *!
!*  Number of RHS:                          3          *!
!*                                                     *!
!*  Start Frequency:              10000.00000      MHz *!
!*  Stop Frequency:               10000.00000      MHz *!
!*  Number of Frequencies:                  1          *!
!*                                                     *!
!*******************************************************!

!*******************************************************!
!*                  Ordering DoF                       *!
!*******************************************************!
!*                                                     *!
!*  Number of Elements:                   362          *!
!*  Number of Unknowns:                  3072          *!
!*                                                     *!
!*  Ordering Time :                    0.04300 seconds *!
!*                                                     *!
!*******************************************************!
```

Figure A.52: Process window

3. Set up the working frequency by clicking with the left button on the navigation tree item called `Analysis frequency`. The definition panel will be shown. Type 10 `GHz` in the entry box and click on the `Set values` button.

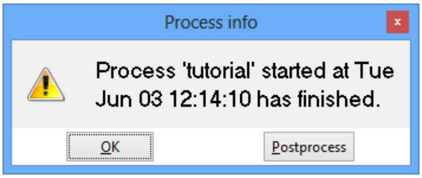
4. Select the number of processes to run the parallel simulation. If a serial simulation is selected, this step can be skipped. Click with the left button on the navigation tree item called `Processes`. Use the mouse and select, for instance, 4 processes in the corresponding entry of the definition panel.

5. Select the maximum memory that will be used by the solver. For example, lets type 3500 `MB`. That value is enough for a simulation with no more than approximately 17000 mesh elements.

6. Save the project before running the simulation.

7. Click on the parallel/serial option from the right menu of the suite.

8. Wait until the `Process window` is shown indicating that the simulation has been finished (see figure A.53). Click on the `Close` button to close the window.

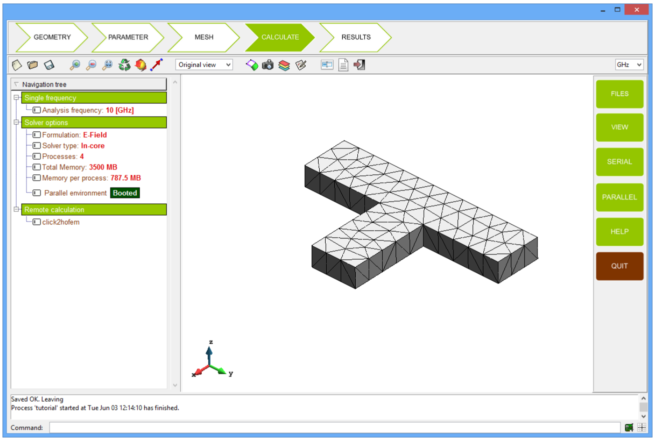


Process info

Process 'tutorial' started at Tue Jun 03 12:14:10 has finished.

Figure A.53: Process window

9. The simulation is finished. Figure A.54 shows the state of the suite at this point of the simulation.



Navigation tree

- Single frequency
  - Analysis frequency: 10 [GHz]
- Solver options
  - Formulation: E-Field
  - Solver type: In-core
  - Processes: 4
  - Total Memory: 3500 MB
  - Memory per process: 787.5 MB
  - Parallel environment Booted
- Remote calculation
  - click2hofem

```
Saved OK. Leaving
Process 'tutorial' started at Tue Jun 03 12:14:10 has finished.
```

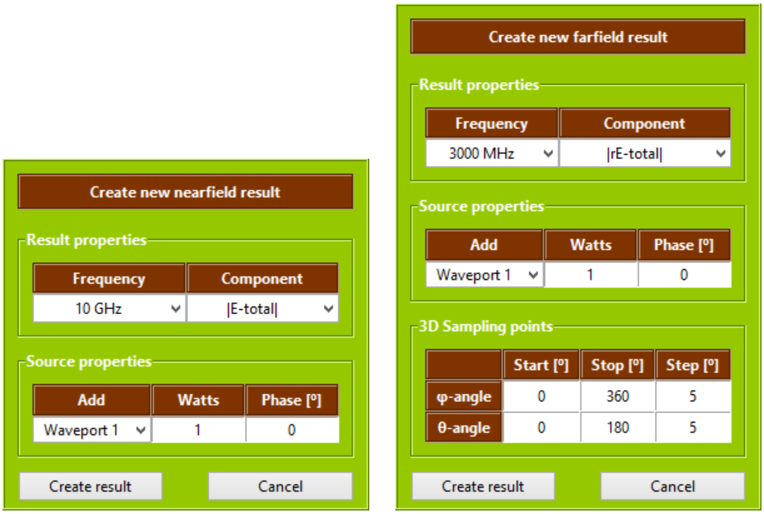
Command:

Figure A.54: State of the suite after the simulation is complete

A.5 Step 5: Visualizing the results

When the simulation of the problem is finished, the suite offers a powerful and user-friendly interface for post-processing. The users may visualize results in 2D/3D, such as near-field, far-field and network parameters. Details about how to create/manipulate/visualize the results of a simulation are described below.

- **Creating a new result:** To create a new result, click with the left button on the corresponding navigation tree item of the desired result. Further information about the results provided by the suite is given below:



Create new nearfield result

Result properties

Frequency	Component
10 GHz	|E-total|

Source properties

Add	Watts	Phase [º]
Waveport 1	1	0

(a) 3D near-field panel

Create new farfield result

Result properties

Frequency	Component
3000 MHz	|rE-total|

Source properties

Add	Watts	Phase [º]
Waveport 1	1	0

3D Sampling points

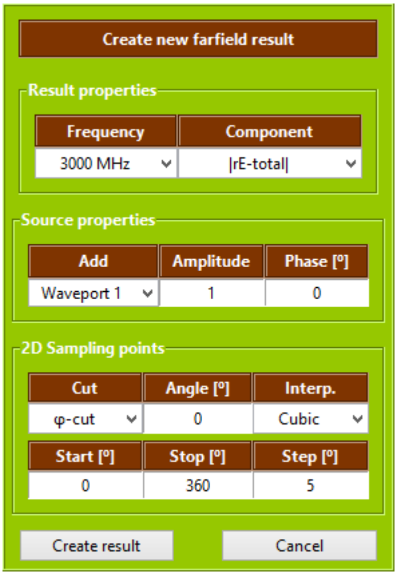
	Start [º]	Stop [º]	Step [º]
φ-angle	0	360	5
θ-angle	0	180	5

(b) 3D far-field panel

Figure A.55: Creating panels for 3D near-field and 3D far-field results

3D near-field result: Click the option `Near-field results` ⇒ `3D result`. A panel where to set all the result properties will be shown (see figure A.55(a)). The first entries are used to select the frequency and the component. The bottom entries of the panel are used to define the excitation properties. One can select the excitation port, the amplitude and the phase. Also, other excitations can be included and added to the existing one. Finally, click on the `Create result` button to confirm the creation process.

3D far-field result: Click the option `Far-field results` ⇒ `3D results`. A panel where to set the result properties will be shown (see figure A.55(b)). The first entries are used to select the frequency and the component. The middle entries of the panel are used to define the excitation properties. As it occurs with the near-field case, one can select the excitation port, the amplitude and the phase, or can add other excitations to the existing one. The bottom entries of the panel are used to define the 3D sampling points where to calculate the far-field. Finally, click on the `Create result` button to confirm the creation process.



Frequency	Component
3000 MHz	|rE-total|

Add	Amplitude	Phase [º]
Waveport 1	1	0

Cut	Angle [º]	Interp.
φ-cut	0	Cubic

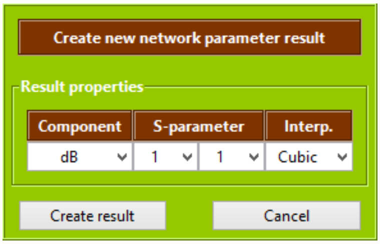
Start [º]	Stop [º]	Step [º]
0	360	5

Figure A.56: Creating panel for 2D near-field

2D far-field result: Click the option `Far-field results` ⇒ `2D results`. The definition panel for this result will be shown (see figure A.56). In this case, the first entries are also used to select the frequency and the component. The middle entries of the panel are used to define the excitation properties. Here, one can also select the excitation port, the amplitude and the phase or can can add other excitations to the existing one. The bottom entries of the panel are used to define 2D cut where to calculate the far-field. Finally, click on the `Create result` button.

S-parameters results: Click the option `Network parameters` ⇒ `S-parameters`. The definition panel for this case is shown in figure A.57. Select the component and the S-parameter. Also, a cubic interpolation may be selected. Finally, click on the `Create result` button to confirm the process.

- **Editing a new result:** To edit an existing result, click with the left button of the mouse on the corresponding result name in the navigation tree. The editing panel with all the result properties will be opened. Click on the `Edit result` button to confirm the changes. The suite will modify the existing result.



Component	S-parameter		Interp.
dB	1	1	Cubic

Figure A.57: Creating panel for S-parameters

- **Deleting an existing result:** To delete an existing result, click with the left button of the mouse on its name to make the selection. Then, click with the right button to open the drop-down menu. Select the `Delete` option to proceed with the deletion process.

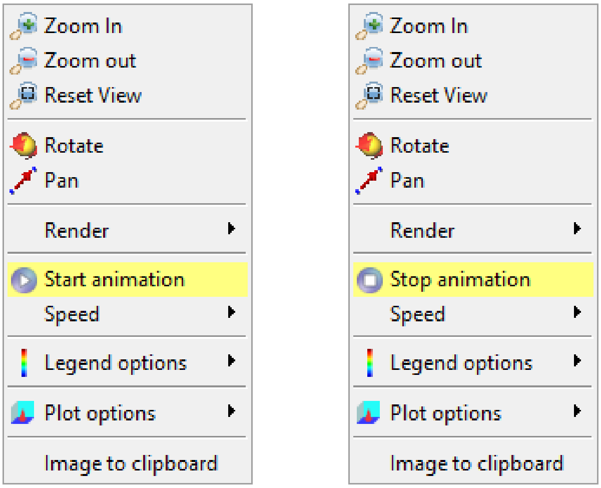
- **Visualizing an existing result:** All the results created by the user appear in the navigation tree under the corresponding root result items. To visualize a result, just click with the left button on any item of the desired result. It will be plotted in the screen.

- **Animating a 3D near-field result:** The suite offers the possibility to animate the 3D near-field results changing its phase from 0° to 360°. The right/left arrow keys increases/decreases the phase of the results, respectively. If the user wants to animate the result continuously, click with the right button of the mouse in the main canvas of the suite. A drop-down menu as the shown in figure A.58(a) appears. Select the option `Start animation`. The animation will start. To stop the animation, click again with the right button on the screen and select the `Stop animation` option from the drop-down menu.

A.5.1 Example

Finally, lets visualize some results for the analysis of the x-band waveguide tee junction. For example, lets create a 3D near-field result and animate it. The steps to perform this visualization are as follows:

1. Select the `Results` item from the top menu of the suite to enter in the result step.



(a) Play animation (b) Stop animation

Figure A.58: Drop-down to animate a 3D near-field result

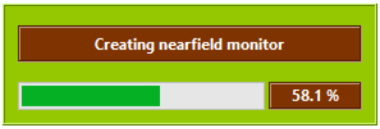
2. Click with the left button of the mouse on `Near-field results` ⇒ `3D result`. The creating panel will be shown in the screen.

3. Select 10 `GHz` as analysis frequency. Select the $\mathtt{E_z}$`-real` component.

4. The next step is to select the appropriate excitation port. In this case, select the waveport 3 in the source properties.

5. Click the `Create result` button to finish the process.

6. Wait until the creation of the result is finished. The progress bar will indicate the status of the process (see figure A.59).

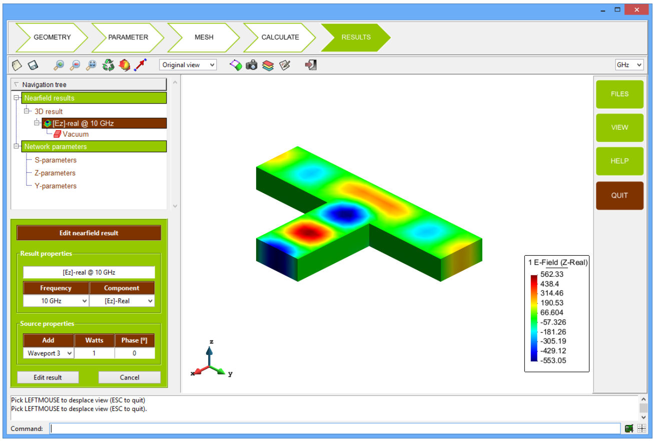


Creating nearfield monitor

58.1 %

Figure A.59: Progress bar

7. The 3D near-field result will appear in the screen (see figure A.60).



[Ez]-real @ 10 GHz

Frequency	Component
10 GHz	[Ez]-Real

Add	Watts	Phase [º]
Waveport 3	1	0

```
Pick LEFTMOUSE to desplace view (ESC to quit)
Pick LEFTMOUSE to desplace view (ESC to quit).
```

Figure A.60: 3D near-field result for the tee junction

8. Use the left/right arrows to animate the results. Also, a continuous animation may be performed by clicking with the right button on the screen. A drop-down menu will be opened showing the `Start animation` option. Click on it.

9. To stop the animation click with the right button on the screen again. A drop-down menu will be opened showing the `Stop animation` option. Click on it to stop the animation.

10. Repeat the previous process to create new results.

APPENDIX B

EWALD REPRESENTATION FOR PERIODIC GREEN'S FUNCTION

This appendix gathers all the expression related to the Ewald representation for periodic Green's function as a summary. Further details about the derivation of the Ewald transformation were given in Section 5.2.1.1.

The periodic Green's functions is represented in two different terms that are combined to compute the numerical value of the function

$$G_p\left(\mathbf{r},\mathbf{r}_s\right) = G_{\mathtt{p1}}\left(\mathbf{r},\mathbf{r}_s\right) + G_{\mathtt{p2}}\left(\mathbf{r},\mathbf{r}_s\right) \tag{B.1}$$

where $G_{\mathtt{p1}}\left(\mathbf{r},\mathbf{r}_s\right)$ is a spectral-domain expression and $G_{\mathtt{p2}}\left(\mathbf{r},\mathbf{r}_s\right)$ is essentially a "modified" spatial-domain portion of the periodic Green's function.

Lets define some variables used in the expressions first. $\mathtt{D_x}$ and $\mathtt{D_y}$ are the unit cell spacing in the x- and y-directions, k_x and k_y are the corresponding propagation vectors with (θ_s, ϕ_s) being the scan angle of the structure,

$$k_x = k_0 \,\mathtt{sin}\,\theta_s \,\mathtt{cos}\,\phi_s \tag{B.2}$$

$$k_y = k_0 \,\mathtt{sin}\,\theta_s \,\mathtt{sin}\,\phi_s \tag{B.3}$$

The coordinate differences between the source point and the observation point are defined as $\xi = x - x'$, $\eta = y - y'$ and $\zeta = z - z'$. Finally, the α_{mn} variable is defined as

$$\alpha_{mn}^2 = \left(\frac{\pi m}{\mathtt{D_x}}\right)^2 + \left(\frac{\pi n}{\mathtt{D_y}}\right)^2 + \left(\frac{\pi m}{\mathtt{D_x}}\right) k_x + \left(\frac{\pi n}{\mathtt{D_y}}\right) k_y + \frac{1}{4}\left(k_x^2 + k_y^2 - k_0^2\right) \quad \text{(B.4)}$$

$G_{\mathtt{p1}}$ term of the Ewald representation

$$G_{\mathtt{p1}}\left(\mathbf{r}, \mathbf{r}_s\right) = \frac{e^{j\left(k_x \xi + k_y \eta\right)}}{8 \mathtt{D_x} \mathtt{D_y}} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{1}{\alpha_{mn}} \times \sum_{\pm} \left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta \mathtt{E}\right) e^{\pm 2\, \alpha_{mn}\, \zeta}\right] e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \quad \text{(B.5)}$$

$G_{\mathtt{p2}}$ term of the Ewald representation

$$G_{\mathtt{p2}}\left(\mathbf{r}, \mathbf{r}_s\right) = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \frac{e^{-j\left(k_x m \mathtt{D_x} + k_y n \mathtt{D_y}\right)}}{R_{mn}} \times \sum_{\pm} \left[e^{\pm j k_0 R_{mn}} \mathtt{erfc}\left(R_{mn} \mathtt{E} \pm \frac{jk}{2\mathtt{E}}\right)\right] \quad \text{(B.6)}$$

The first derivative of the above expression is required in order to compute the field radiated by the `FEM` region. Thereby, the calculation of $\nabla G_{\mathtt{p1}}$ and $\nabla G_{\mathtt{p2}}$ has been performed obtaining the following expressions:

$$\nabla G_{\mathtt{p1}} = \begin{bmatrix} \dfrac{\partial G_{\mathtt{p1}}}{\partial x} \\ \dfrac{\partial G_{\mathtt{p1}}}{\partial y} \\ \dfrac{\partial G_{\mathtt{p1}}}{\partial z} \end{bmatrix} \qquad \nabla G_{\mathtt{p2}} = \begin{bmatrix} \dfrac{\partial G_{\mathtt{p2}}}{\partial x} \\ \dfrac{\partial G_{\mathtt{p2}}}{\partial y} \\ \dfrac{\partial G_{\mathtt{p2}}}{\partial z} \end{bmatrix} \quad \text{(B.7)}$$

Derivative of $G_{\mathtt{p1}}$ with respect to ∂x

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial x} = j\left(k_x + \frac{2\pi m}{\mathtt{D_x}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \\ \times\sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \qquad \text{(B.8)}$$

Derivative of $G_{\mathtt{p1}}$ with respect to ∂y

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial y} = j\left(k_y + \frac{2\pi n}{\mathtt{D_y}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \\ \times\sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \qquad \text{(B.9)}$$

Derivative of $G_{\mathtt{p1}}$ with respect to ∂z

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial z} = \frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}2e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \\ \times\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta} - \mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] \qquad \text{(B.10)}$$

Derivative of $G_{\mathtt{p2}}$ with respect to ∂x

$$\frac{\partial\, G_{\mathtt{p2}}}{\partial x} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left(\xi + m\mathtt{D_x}\right)\frac{e^{-j\left(k_x m\mathtt{D_x} + k_y n\mathtt{D_y}\right)}}{R_{mn}^2}\times\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}}\right. \\ \left. + \sum_{\pm}\left[\frac{-1 \pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}}\right)\right]\right\} \qquad \text{(B.11)}$$

Derivative of G_{p2} with respect to ∂y

$$\frac{\partial\, G_{\mathtt{p2}}}{\partial y} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}(\eta + n\mathtt{D}_{\mathrm{y}})\,\frac{e^{-j\left(k_x m\mathtt{D}_{\mathrm{x}}+k_y n\mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^2}\times\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm}\left[\frac{-1\pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \tag{B.12}$$

Derivative of G_{p2} with respect to ∂z

$$\frac{\partial\, G_{\mathtt{p2}}}{\partial z} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\zeta\,\frac{e^{-j\left(k_x m\mathtt{D}_{\mathrm{x}}+k_y n\mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^2}\times\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm}\left[\frac{-1\pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \tag{B.13}$$

The second derivative of the terms of the Ewald representation for periodic Green's function is also required in order to compute the field radiated by the FEM region. The expressions of $\nabla\nabla\, G_{\mathtt{p1}}$ and $\nabla\nabla\, G_{\mathtt{p2}}$ are given by

$$\nabla G_{\mathtt{p1}} = \begin{bmatrix} \frac{\partial\, G_{\mathtt{p1}}}{\partial x\,\partial x} & \frac{\partial\, G_{\mathtt{p1}}}{\partial x\,\partial y} & \frac{\partial\, G_{\mathtt{p1}}}{\partial x\,\partial z} \\ \frac{\partial\, G_{\mathtt{p1}}}{\partial y\,\partial x} & \frac{\partial\, G_{\mathtt{p1}}}{\partial y\,\partial y} & \frac{\partial\, G_{\mathtt{p1}}}{\partial y\,\partial z} \\ \frac{\partial\, G_{\mathtt{p1}}}{\partial z\,\partial x} & \frac{\partial\, G_{\mathtt{p1}}}{\partial z\,\partial y} & \frac{\partial\, G_{\mathtt{p1}}}{\partial z\,\partial z} \end{bmatrix} \qquad \nabla G_{\mathtt{p2}} = \begin{bmatrix} \frac{\partial\, G_{\mathtt{p2}}}{\partial x\,\partial x} & \frac{\partial\, G_{\mathtt{p2}}}{\partial x\,\partial y} & \frac{\partial\, G_{\mathtt{p2}}}{\partial x\,\partial z} \\ \frac{\partial\, G_{\mathtt{p2}}}{\partial y\,\partial x} & \frac{\partial\, G_{\mathtt{p2}}}{\partial y\,\partial y} & \frac{\partial\, G_{\mathtt{p2}}}{\partial y\,\partial z} \\ \frac{\partial\, G_{\mathtt{p2}}}{\partial z\,\partial x} & \frac{\partial\, G_{\mathtt{p2}}}{\partial z\,\partial y} & \frac{\partial\, G_{\mathtt{p2}}}{\partial z\,\partial z} \end{bmatrix} \tag{B.14}$$

Derivative of G_{p1} with respect to $\partial x\partial x$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial x\partial x} = -\left(k_x + \frac{2\pi m}{\mathtt{D}_{\mathtt{x}}}\right)^2\frac{e^{j\left(k_x\xi+k_y\eta\right)}}{8\mathtt{D}_{\mathtt{x}}\mathtt{D}_{\mathtt{y}}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \times\sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}}\pm\zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D}_{\mathtt{x}}}+\frac{n\eta}{\mathtt{D}_{\mathtt{y}}}\right)} \tag{B.15}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial x \partial y$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial x \partial y} = -\left(k_x + \frac{2\pi m}{\mathtt{D_x}}\right)\left(k_y + \frac{2\pi n}{\mathtt{D_y}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \times \sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \tag{B.16}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial x \partial z$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial x \partial z} = j\left(k_x + \frac{2\pi m}{\mathtt{D_x}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}2e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \times \left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta} - \mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] \tag{B.17}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial y \partial x$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial y \partial x} = -\left(k_x + \frac{2\pi m}{\mathtt{D_x}}\right)\left(k_y + \frac{2\pi n}{\mathtt{D_y}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \times \sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \tag{B.18}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial y \partial y$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial y \partial y} = -\left(k_y + \frac{2\pi n}{\mathtt{D_y}}\right)^2\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D_x}\mathtt{D_y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\frac{1}{\alpha_{mn}} \times \sum_{\pm}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} \pm \zeta\mathtt{E}\right)e^{\pm 2\,\alpha_{mn}\,\zeta}\right]e^{2\pi j\left(\frac{m\xi}{\mathtt{D_x}} + \frac{n\eta}{\mathtt{D_y}}\right)} \tag{B.19}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial y \partial z$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial y \partial z} = j\left(k_y + \frac{2\pi n}{\mathtt{D}_\mathtt{y}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D}_\mathtt{x}\mathtt{D}_\mathtt{y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} 2\,e^{2\pi j\left(\frac{m\xi}{\mathtt{D}_\mathtt{x}} + \frac{n\eta}{\mathtt{D}_\mathtt{y}}\right)}$$
$$\times\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta} - \mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] \tag{B.20}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial z \partial x$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial z \partial x} = j\left(k_x + \frac{2\pi m}{\mathtt{D}_\mathtt{x}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D}_\mathtt{x}\mathtt{D}_\mathtt{y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} 2e^{2\pi j\left(\frac{m\xi}{\mathtt{D}_\mathtt{x}} + \frac{n\eta}{\mathtt{D}_\mathtt{y}}\right)}$$
$$\times\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta} - \mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] \tag{B.21}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial z \partial y$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial z \partial y} = j\left(k_y + \frac{2\pi n}{\mathtt{D}_\mathtt{y}}\right)\frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D}_\mathtt{x}\mathtt{D}_\mathtt{y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} 2\,e^{2\pi j\left(\frac{m\xi}{\mathtt{D}_\mathtt{x}} + \frac{n\eta}{\mathtt{D}_\mathtt{y}}\right)}$$
$$\times\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta} - \mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] \tag{B.22}$$

Derivative of $G_{\mathtt{p1}}$ with respect to $\partial z \partial z$

$$\frac{\partial\, G_{\mathtt{p1}}}{\partial z \partial z} = \frac{e^{j\left(k_x\xi + k_y\eta\right)}}{8\mathtt{D}_\mathtt{x}\mathtt{D}_\mathtt{y}}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} 2\,e^{2\pi j\left(\frac{m\xi}{\mathtt{D}_\mathtt{x}} + \frac{n\eta}{\mathtt{D}_\mathtt{y}}\right)}$$
$$\times\left\{2\,\alpha_{mn}\left[\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} + \zeta\mathtt{E}\right)e^{+2\,\alpha_{mn}\zeta}\right.\right.$$
$$\left.\left. + \,\mathtt{erfc}\left(\frac{\alpha_{mn}}{\mathtt{E}} - \zeta\mathtt{E}\right)e^{-2\,\alpha_{mn}\zeta}\right] - \frac{4\mathtt{E}}{\sqrt{\pi}}e^{-\frac{\alpha_{mn}^2}{\mathtt{E}^2} - \zeta^2\mathtt{E}^2}\right\} \tag{B.23}$$

Derivative of G_{p2} with respect to $\partial x \partial x$

$$\begin{aligned}
&\frac{\partial\, G_{\mathrm{p2}}}{\partial x \partial x} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left[\, R_{mn}^{2} - 2\left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right)^{2} \right] \frac{e^{-j\left(k_x m\mathtt{D}_{\mathtt{x}} + k_y n\mathtt{D}_{\mathtt{y}}\right)}}{R_{mn}^{4}} \times \\
&\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^{2}\mathtt{E}^{2} + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\} \\
&+ \left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right) \frac{e^{-j\left(k_x m\mathtt{D}_{\mathtt{x}} + k_y n\mathtt{D}_{\mathtt{y}}\right)}}{R_{mn}^{2}} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^{2}} + 8\,\mathtt{E}^{3} \right] \frac{\left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right)}{\sqrt{\pi}} e^{-R_{mn}^{2}\mathtt{E}^{2} + \frac{k_0^2}{4\mathtt{E}^2}} + \right. \\
&\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^{2} + 1 \mp jk_0 R_{mn}}{R_{mn}^{3}} \left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right) e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}
\end{aligned} \tag{B.24}$$

Derivative of G_{p2} with respect to $\partial x \partial y$

$$\begin{aligned}
&\frac{\partial\, G_{\mathrm{p2}}}{\partial x \partial y} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} -2\left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right)\left(\eta + n\,\mathtt{D}_{\mathtt{y}}\right) \frac{e^{-j\left(k_x m\mathtt{D}_{\mathtt{x}} + k_y n\mathtt{D}_{\mathtt{y}}\right)}}{R_{mn}^{4}} \times \\
&\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^{2}\mathtt{E}^{2} + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\} \\
&+ \left(\xi + m\,\mathtt{D}_{\mathtt{x}}\right) \frac{e^{-j\left(k_x m\mathtt{D}_{\mathtt{x}} + k_y n\mathtt{D}_{\mathtt{y}}\right)}}{R_{mn}^{2}} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^{2}} + 8\,\mathtt{E}^{3} \right] \frac{\left(\eta + n\,\mathtt{D}_{\mathtt{y}}\right)}{\sqrt{\pi}} e^{-R_{mn}^{2}\mathtt{E}^{2} + \frac{k_0^2}{4\mathtt{E}^2}} + \right. \\
&\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^{2} + 1 \mp jk_0 R_{mn}}{R_{mn}^{3}} \left(\eta + n\,\mathtt{D}_{\mathtt{y}}\right) e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}
\end{aligned} \tag{B.25}$$

Derivative of G_{p2} with respect to $\partial x \partial z$

$$\frac{\partial\, G_{\mathrm{p2}}}{\partial x \partial z} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} -2\,\zeta\,(\xi + m\,\mathtt{D}_{\mathrm{x}})\, \frac{e^{-j\left(k_x m \mathtt{D}_{\mathrm{x}} + k_y n \mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^4} \times$$
$$\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}$$
$$+ (\xi + m\,\mathtt{D}_{\mathrm{x}})\, \frac{e^{-j\left(k_x m \mathtt{D}_{\mathrm{x}} + k_y n \mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^2} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^2} + 8\,\mathtt{E}^3 \right] \frac{\zeta}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \right.$$
$$\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^2 + 1 \mp jk_0 R_{mn}}{R_{mn}^3}\, \zeta\, e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\} \qquad \text{(B.26)}$$

Derivative of G_{p2} with respect to $\partial y \partial x$

$$\frac{\partial\, G_{\mathrm{p2}}}{\partial y \partial x} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} -2\,(\xi + m\,\mathtt{D}_{\mathrm{x}})\,(\eta + n\,\mathtt{D}_{\mathrm{y}})\, \frac{e^{-j\left(k_x m \mathtt{D}_{\mathrm{x}} + k_y n \mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^4} \times$$
$$\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}$$
$$+ (\eta + n\,\mathtt{D}_{\mathrm{y}})\, \frac{e^{-j\left(k_x m \mathtt{D}_{\mathrm{x}} + k_y n \mathtt{D}_{\mathrm{y}}\right)}}{R_{mn}^2} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^2} + 8\,\mathtt{E}^3 \right] \frac{(\xi + m\,\mathtt{D}_{\mathrm{x}})}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \right.$$
$$\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^2 + 1 \mp jk_0 R_{mn}}{R_{mn}^3}\, (\xi + m\,\mathtt{D}_{\mathrm{x}})\, e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}$$
$$\text{(B.27)}$$

Derivative of G_{p2} with respect to $\partial y \partial y$

$$\frac{\partial\, G_{\mathrm{p2}}}{\partial y \partial y} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left[\, R_{mn}^2 - 2\left(\eta + n\,\mathtt{D}_\mathtt{y}\right)^2 \right] \frac{e^{-j\left(k_x m\mathtt{D}_\mathtt{x} + k_y n\mathtt{D}_\mathtt{y}\right)}}{R_{mn}^4} \times$$
$$\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}$$
$$+ \left(\eta + n\,\mathtt{D}_\mathtt{y}\right) \frac{e^{-j\left(k_x m\mathtt{D}_\mathtt{x} + k_y n\mathtt{D}_\mathtt{y}\right)}}{R_{mn}^2} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^2} + 8\,\mathtt{E}^3 \right] \frac{\left(\eta + n\,\mathtt{D}_\mathtt{y}\right)}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \right.$$
$$\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^2 + 1 \mp jk_0 R_{mn}}{R_{mn}^3} \left(\eta + n\,\mathtt{D}_\mathtt{y}\right) e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\} \tag{B.28}$$

Derivative of G_{p2} with respect to $\partial y \partial z$

$$\frac{\partial\, G_{\mathrm{p2}}}{\partial y \partial z} = \frac{1}{8\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} -2\,\zeta\, \left(\eta + n\,\mathtt{D}_\mathtt{y}\right) \frac{e^{-j\left(k_x m\mathtt{D}_\mathtt{x} + k_y n\mathtt{D}_\mathtt{y}\right)}}{R_{mn}^4} \times$$
$$\left\{ \frac{-4\mathtt{E}}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm} \left[\frac{-1 \pm jk_0 R_{mn}}{R_{mn}} e^{\pm jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\}$$
$$+ \left(\eta + n\,\mathtt{D}_\mathtt{y}\right) \frac{e^{-j\left(k_x m\mathtt{D}_\mathtt{x} + k_y n\mathtt{D}_\mathtt{y}\right)}}{R_{mn}^2} \times \left\{ \left[\frac{4\mathtt{E}}{R_{mn}^2} + 8\,\mathtt{E}^3 \right] \frac{\zeta}{\sqrt{\pi}} e^{-R_{mn}^2 \mathtt{E}^2 + \frac{k_0^2}{4\mathtt{E}^2}} + \right.$$
$$\left. \sum_{\pm} \left[\frac{-k_0^2 R_{mn}^2 + 1 \mp jk_0 R_{mn}}{R_{mn}^3}\, \zeta\, e^{\pm\, jk_0 R_{mn}} \mathtt{erfc}\left(R_{mn}\,\mathtt{E} \pm \frac{jk}{2\mathtt{E}} \right) \right] \right\} \tag{B.29}$$

Derivative of G_{p2} with respect to $\partial z\partial x$

$$\begin{gathered}
\frac{\partial\, G_{\mathrm{p2}}}{\partial z\partial x} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} -2\,\zeta\,\left(\xi + m\,\mathtt{D_x}\right)\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^4}\times \\
\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm}\left[\frac{-1\pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \\
+\,\zeta\,\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^2}\times\left\{\left[\frac{4\mathtt{E}}{R_{mn}^2}+8\,\mathtt{E}^3\right]\frac{\left(\xi+m\,\mathtt{D_x}\right)}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}}\right. + \\
\left.\sum_{\pm}\left[\frac{-k_0^2R_{mn}^2+1\mp jk_0R_{mn}}{R_{mn}^3}\left(\xi+m\,\mathtt{D_x}\right)e^{\pm\, jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\,\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\}
\end{gathered} \tag{B.30}$$

Derivative of G_{p2} with respect to $\partial z\partial y$

$$\begin{gathered}
\frac{\partial\, G_{\mathrm{p2}}}{\partial z\partial y} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} -2\,\zeta\,\left(\eta + n\,\mathtt{D_y}\right)\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^4}\times \\
\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm}\left[\frac{-1\pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \\
+\,\zeta\,\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^2}\times\left\{\left[\frac{4\mathtt{E}}{R_{mn}^2}+8\,\mathtt{E}^3\right]\frac{\left(\eta+n\,\mathtt{D_y}\right)}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}}\right. + \\
\left.\sum_{\pm}\left[\frac{-k_0^2R_{mn}^2+1\mp jk_0R_{mn}}{R_{mn}^3}\left(\eta+n\,\mathtt{D_y}\right)e^{\pm\, jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\,\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\}
\end{gathered} \tag{B.31}$$

Derivative of $G_{\mathtt{p2}}$ with respect to $\partial z\partial z$

$$
\begin{multline}
\frac{\partial\, G_{\mathtt{p2}}}{\partial z\partial z} = \frac{1}{8\pi}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left[\,R_{mn}^2 - 2\,\zeta^2\,\right]\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^4}\times \\
\left\{\frac{-4\mathtt{E}}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}} + \sum_{\pm}\left[\frac{-1\pm jk_0R_{mn}}{R_{mn}}e^{\pm jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \\
+\,\zeta\,\frac{e^{-j\left(k_x m\mathtt{D_x}+k_y n\mathtt{D_y}\right)}}{R_{mn}^2}\times\left\{\left[\frac{4\mathtt{E}}{R_{mn}^2}+8\,\mathtt{E}^3\right]\frac{\zeta}{\sqrt{\pi}}e^{-R_{mn}^2\mathtt{E}^2+\frac{k_0^2}{4\mathtt{E}^2}}\,+\right. \\
\left.\sum_{\pm}\left[\frac{-k_0^2R_{mn}^2+1\mp jk_0R_{mn}}{R_{mn}^3}\,\zeta\,e^{\pm\,jk_0R_{mn}}\mathtt{erfc}\left(R_{mn}\,\mathtt{E}\pm\frac{jk}{2\mathtt{E}}\right)\right]\right\} \tag{B.32}
\end{multline}
$$

BIBLIOGRAPHY

[1] R. F. Harrington, *Field Computation by Moment Methods.* IEEE Press, 1993.

[2] K. S. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transaction on Antennas and Propagation*, vol. 14, pp. 302–307, May 1966.

[3] K. L. Shlager and J. B. Schneider, "A selective survey of the finite-difference time-domain literature," *IEEE Antennas and Propagation Magazine*, vol. 37, pp. 39–56, Aug 1995.

[4] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Norwood, MA: Artech House Publishers, Inc., 2005.

[5] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers.* Cambridge University Press, 1983, 1991, 1996.

[6] J. M. Jin, *The Finite Element Method in Electromagnetics.* John Wiley & Sons, Inc., 1993.

[7] M. Salazar-Palma, T. K. Sarkar, L. E. García-Castillo, T. Roy, and A. R. Djordjevic, *Iterative and Self-Adaptive Finite-Elements in Electromagnetic Modeling.* Norwood, MA: Artech House Publishers, Inc., 1998.

[8] "HFSS." http://www.ansys.com/.

[9] "CST STUDIO SUITE." https://www.cst.com/.

[10] B. Krietenstein, R. Schuhmann, P. Thoma, and T. Weiland, "The perfect boundary approximation technique facing the challenge of high precision field computation," in *LINAC XIX International Linear Accelerator Conference*, (Chicago, USA), pp. 860–862, 1998.

[11] T. Weiland, "Finite integration method and discrete electromagnetism," in *Computational Electromagnetics*, vol. 28 of *Lecture Notes in Computational Science and Engineering*, pp. 183–198, Springer Berlin Heidelberg, 2003.

[12] T. Weiland, "Time domain electromagnetic field computation with finite difference methods," *Journal of Numerical Modelling*, vol. 9, pp. 295–319, 1996.

[13] "FEKO." http://www.feko.info/.

[14] "COMSOL." http://www.comsol.com/.

[15] "XFdtd." http://www.remcom.com/xf7/.

[16] "XGtd." http://www.remcom.com/xgtd/.

[17] "Advanced Design System." http://www.agilent.com/find/ads.

[18] "Momentum." http://www.agilent.com/find/eesof-momentum.

[19] "IMEC." http://www2.imec.be/.

[20] "GEMS." http://www.2comu.com/.

[21] "Efield." http://www.efieldsolutions.com/.

[22] "WIPL-D." http://www.wipl-d.com/.

[23] "GRASP." http://www.ticra.com/products/software/grasp.

[24] "TICRA." http://www.ticra.com/.

[25] "CHAMP." http://www.ticra.com/products/software/champ.

[26] "Mentor Graphics." http://www.mentor.com/.

[27] "HyperLynx." http://www.mentor.com/pcb/hyperlynx/.

[28] "NewFasant." http://www.fasant.com/.

[29] "CEMWorks." https://www.cemworks.com/.

[30] "μWave Wizard." http://www.mician.com/content/products/.

[31] "Tech-X Corporation." http://www.txcorp.com/.

[32] "VsimEM." http://www.txcorp.com/home/vsim/vsim-em/.

[33] "openEMS." http://openems.de/start/index.php.

[34] "LC." `http://lc.cray.com/doc/intro.html`.

[35] "ERMES:." `https://web.cimne.upc.edu/users/rotin/`.

[36] Y. Zhang and T. K. Sarkar, *Parallel Solution of Integral Equation Based EM Problems in the Frequency Domain.* Wiley-IEEE Press, July 2009.

[37] R. F. Harrington, "Boundary integral formulations for homogenous material bodies," *Journal of Electromagnetic Waves and Applications*, vol. 3, no. 1, pp. 1–15, 1989.

[38] S. M. Rao, C. C. Cha, R. L. Cravey, and D. L. Wilkes, "Electromagnetic scattering from arbitrary shaped conducting bodies coated with lossy material of arbitrary thickness," *IEEE Transaction on Antennas and Propagation*, vol. 3, no. 1, pp. 1–15, 1989.

[39] "Message Passing Interface Forum." `http://www.mpi-forum.org/`.

[40] "ScaLAPACK project." `http://www.netlib.org/scalapack/`.

[41] L. E. García-Castillo and M. Salazar-Palma, "Second-order Nédélec tetrahedral element for computational electromagnetics," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (John Wiley & Sons, Inc.)*, vol. 13, pp. 261–287, March-June 2000.

[42] L. E. García-Castillo, A. J. Ruiz-Genovés, I. Gómez-Revuelto, M. Salazar-Palma, and T. K. Sarkar, "Third-order Nédélec curl-conforming finite element," *IEEE Transactions on Magnetics*, vol. 38, pp. 2370–2372, Sept. 2002.

[43] J. C. Nédélec, "Mixed finite elements in R^3," *Numerische Mathematik*, vol. 35, pp. 315–341, 1980.

[44] K. E. Jordan, "An efficient numerical evaluation of the green's function for the helmholtz operator on periodic structures," *Journal of computational physics*, no. 63, pp. 222–235, 1986.

[45] D. R. W. Thomas F. Eibert, John L. Volakis and D. R. Jackson, "Hybrid FE/BI modeling of 3-D doubly periodic structures utilizing triangular prismatic elements and an MPIE formulation accelerated by the ewald transformation," *IEEE Transaction on Antennas and Propagation*, pp. 843–850, may 1999.

[46] "GiD: The personal pre and postprocessor." International Center for Numerical Methods in Engineering, `http://www.gidhome.com/`.

[47] M. J. D. Powell, "The newuoa software for unconstrained optimization without derivatives," *Nonconvex Optimization and Its Applications*, no. 83, 2006.

[48] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. Journal*, no. 7:308-13, 1965.

[49] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[50] “MUMPS Solver.” `http://www.enseeiht.fr/lima/apo/MUMPS/`.

[51] “HSL (2002). A collection of fortran codes for large scale scientific computation.” `http://www.numerical.rl.ac.uk/hsl`.

[52] “PARDISO.” `http://www.pardiso-project.org/`.

[53] R. Fernández-Recio, L. E. Garcia-Castillo, I. Gómez-Revuelto, and M. Salazar-Palma, “Convergence study of a non-standard Schwarz domain decomposition method for finite element mesh truncation in electromagnetics,” *Progress In Electromagnetics Research (PIER)*, vol. 120, pp. 439–457, 2011.

[54] P. A. Raviart and J. M. Thomas, “A mixed finite element method for 2nd order elliptic problems,” in *Mathematical Aspects of Finite Element Methods* (A. Dold and B. Eckmann, eds.), pp. 292–315, Springer Verlag, 1977. Lecture Notes Math. 606.

[55] I. Gomez-Revuelto, L. E. Garcia-Castillo, D. Pardo, J. Kurtz, and M. Salazar-Palma, “A three-dimensional self-adaptive *hp* finite element method for the characterization of waveguide discontinuities,” *Computer Methods in Applied Mechanics and Engineering*, no. 249–252, pp. 62–74, 2012. Special issue on “Higher Order Finite Element and Isogeometric Methods”.

[56] R.-M. B. Garrido, . S. L. Romano, M. S. Palma, A. O. Navarro, and I. H. Carpintero, “Design, construction and experimental characterization of a broadband highly selective filter in waveguide technology in ka band,” in *IEEE MTT-S International Microwave Symposium Digest*, (San Francisco, California, USA), pp. 250–253, Institute of Electrical and Electronics Engineer (IEEE), June 2006.

[57] F. Alessandri, M. Chiodetti, A. G. D. Maiarelli, G. Martirano, D. Schmitt, L. Vanni, and F. Vitulli, “The electric-field integral-equation method for the analysis and design of a class of rectangular cavity filters loaded by dielectric and metallic cylindrical pucks,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 1790–1797, Aug. 2004.

[58] A. C. Woo, H. T. G. Wang, M. J. Schuh, and M. L. Sanders, “Benchmark radar targets for the validation of computational electromagnetics programs,” *IEEE Antennas and Propagation Magazine*, vol. 35, no. 1, pp. 84–89, 1993.

[59] S. Rao, D. Wilton, and A. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *Antennas and Propagation, IEEE Transactions on*, vol. 30, pp. 409–418, May 1982.

[60] Y. Zhang, M. Taylor, T. Sarkar, H. Moon, and M. Yuan, "Solving large complex problems using a higher-order basis: parallel in-core and out-of-core integral-equation solvers," *Antennas and Propagation Magazine, IEEE*, vol. 50, pp. 13–30, Aug 2008.

[61] V. G. N. Amitay and C. P. Wu, *Theory and Analysis of Phased Array Antennas*. New York: Wiley, 1972.

[62] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover, 1965.

[63] D. Sanks, "Non-linear transformations of divergent and slowly-convergent sequences," *Journal of Mathematical Physics*, no. 34, pp. 1–42, 1955.

[64] P. P. Ewald, "Dispersion und doppelbrechung von Elecktronengittern (Kristallen)." Dissertation, München, 1912, also *Ann. Phys.* 49, p.1, 1916.

[65] P. P. Ewald, "Die Berechnung optischer und elektrostatischer Gitterpotentiale," *Ann. Phys.*, no. 64, pp. 253–287, 1921.

[66] N. Kinayman and M. I. Aksun, "Comparative study of acceleration techniques for integral and series in electromagnetic problems," *Radio Science*, no. 30, pp. 1713–1722, 1995.

[67] J.-M. Jin, Z. Lou, Y.-J. Li, N. Riley, and D. Riley, "Finite element analysis of complex antennas and arrays," *IEEE Transaction on Antennas and Propagation*, vol. 56, pp. 2222–2240, Aug 2008.

[68] A. G. Lamperez, T. K. Sarkar, and M. S. Palma, "Filter model generation from scattering parameters using the cauchy method," in *Microwave Conference, 2002. 32nd European*, pp. 1–4, Sept 2002.