UNIVERSIDAD DE GRANADA

Tesis Doctoral

A subgridding method based on grid orthogonalization in Finite-Differences Time-Domain

Memoria presentada por

Antonio Jesús Martín Valverde

para optar al grado de Doctor por la Universidad de Granada

realizada en la
Facultad de Ciencias
Departamento de Electromagnetismo y Física de la Materia
Grupo de Electromagnetismo de Granada
en el programa de doctorado de
Física y Matemáticas (FisyMat)

Bajo la dirección de los Doctores
Salvador González García y Miguel David Ruiz-Cabello Núñez

Editor: Universidad de Granada. Tesis Doctorales
Autor: Antonio Jesús Martín Valverde
ISBN: 978-84-1195-570-6
URI: https://hdl.handle.net/10481/97452

A mis padres y mi hermana.

A mis sobrinos.

A Juanma.

Agradecimientos

Aquí comienza el documento que da forma al trabajo de cinco años. Tal vez lleve mi nombre en la portada, pero yo solo nunca habría podido realizar esta labor.

En primer lugar, debo agradecer a Juanma, mi pareja, sin cuyo apoyo en los momentos más duros y estresantes no creo que hubiera podido seguir adelante. Se suele decir que el estrés te quita años de vida, pero tu sonrisa me los da cada día.

En segundo lugar, debo agradecer a mi familia por ser un apoyo persistente en el tiempo y aportarme un lugar al que siempre poder regresar, tanto en sentido geográfico como emocional. Mención especial a mi sobrina Mía, que llegó durante este periodo y tantas veces me ayudó a reir.

También a mis directores de tesis, Salva y Miguel. A Salva, por descubrirme el electromagnetismo, presentarme al grupo de investigación y prestarme su ayuda siempre que lo he necesitado. A Miguel, por trabajar conmigo codo con codo durante todos estos años y ayudarme a encontrar la motivación necesaria para aprender. A ambos, en definitiva, porque sin su mentorización no habría podido llegar tan lejos.

A la Ohio State University y al profesor Fernando Teixeira por acogerme bajo su supervisión durante tres meses en los que pude expandir mis horizontes como investigador.

A todos los miembros del GEG que de una forma u otra me han mentorizado: Rafael, Amelia, Mario y Luis. Del mismo modo, a mis compañeros de laboratorio: Alberto, Alejandro, Kenan, Farah e Isaac.

A todos,
Muchas gracias.

Este trabajo ha sido posible, en parte, gracias a:

La Universidad de Granada mediante el contrato FPU del Plan del Vicerrectorado de Investigación y Transferencia (programa 6).

Airbus Defence and Space a través del proyecto “Support to Alhambra-LFT Project: Software development and validation”. Contrato OTRI número 4023.

Ministerio de Economía y Competitividad a través del proyecto: “Evaluación Numérica y Experimental de Efectos Electromagnéticos Ambientales En Aviones No Tripulados”. Ref. TEC2016-79214-C3-3-R.

Ministerio de Ciencia e Innovación a través del proyecto: “Efectos Ambientales Electromagnéticos (E3) en Fuselajes Inteligentes y Nuevas Tecnologías de Ensamblado para Vehículos Aéreos no Tripulados (UAV)”. Ref. PID2019-106120RB-C33.

Huawei a través del proyecto: “Research & Development Agreement with Huawei Device Co. Ltd.”. Contrato UGR-OTRI número 4776.

La Junta de Andalucía a través del proyecto: “COMPutational tools for the analysis of Environmental Electromagnetic Effects in novel vehicle technologies (COMP-EEE)”. Ref. B-TIC.700UGR20.

El programa Horizon Europe Framework Programme (HORIZON) a través del proyecto: “Hybrid Electric regional Aircraft distribution Technologies”. Call: CLEAN-AVIATION-2022-01-HER-03, contrato número 101101961.

Ministerio de Ciencia e Innovación a través del proyecto: “Métodos numéricos avanzados para el análisis de materiales eléctricos y magnáticos en aplicaciones aeroespaciales”. Ref. PID2022-137495OB-C31.

Airbus Defense and Space a través del proyecto “Gondola Project. Ref. TMC-CD-2110004”. Contrato UGR-OTRI número 5261.

El tiempo de computación provisto por el Servicio de Supercomputación de la Universidad de Granada (https://supercomputacion.ugr.es).

Resumen

Esta tesis está enmarcada dentro del ámbito del electromagnetismo computacional. Al igual que sus homólogos numéricos en otras áreas de la física, el electromagnetismo computacional ha venido cobrando cada vez mayor importancia en las últimas décadas, pues permite obtener soluciones a problemas complejos que serían inabarcables analíticamente. Además, el uso de métodos numéricos es particularmente interesante en etapas de diseño en procesos industriales, pues permite recrear entornos sintéticamente, evitando así la construcción de prototipos físicos y, por tanto, ahorrando tiempo y costes. Dentro de los métodos numéricos en electromagnetismo, el algoritmo de las diferencias finitas en el dominio del tiempo (FDTD en inglés) es conocido por su sencillez y eficiencia. Esta técnica require de una discretización completa y estructurada del espacio computacional en celdas ortoédricas, lo que le confiere una particular robustez y eficiencia computacional al ser fácilmente paralelizable. No obstante, es esta misma naturaleza estructurada la que le otorga algunas de sus principales limitaciones. En primer lugar, las celdas no permiten capturar adecuadamente la curvatura de aquellos materiales que no estén orientados según ejes cartesianos. En segundo lugar, nos encontramos los denominados problemas multiescala: la presencia de detalles pequeños en una simulación nos obliga a utilizar una malla suficientemente fina que nos permita resolverlos, obligando con ello a malgastar recursos computacionales en otras partes de la simulación que no contengan información geométrica relevante. El primer problema se puede resolver mediante la utilización de mallas conformes, pues estas otorgan una mayor flexibilidad en la geometría de los materiales. Por otro lado, existen modelos subcelda que permiten resolver problemas específicos en casos multiescala en FDTD, sin embargo, no existe ninguna solución universal. En este contexto, los métodos de submallado (subgridding en inglés) constituyen un prometedor campo de estudio al tratarse de una familia de técnicas que permiten refinar partes específicas de las simulaciones en celdas más pequeñas, dividiendo por tanto el dominio computacional en distintas regiones con diferentes niveles de discretización espacial. Específicamente, esta tesis estudia un método originalmente publicado en 1997, y posteriormente denominado en la literatura “Orthogonalized Integral-based Subgridding” (OI-SG). Este trabajo está centrado en la derivación de este método, su implementación para casos de simulación arbitrarios, el estudio de sus propiedades numéricas, su precisión y eficiencia en casos numéricos reales, y la identificación de los puntos fuertes y desventajas que ofrece.

Partiendo de las ecuaciones de Ampère y Faraday en su forma integral, y del formalismo de la malla principal-dual en FDTD, el método OI-SG obtiene una formulación explícita que permite conectar de forma natural las regiones finas y gruesas. Opcionalmente, se propone una metodología de paso temporal local basado en diferencias finitas no centradas que permite proveer de una diferente discretización temporal en ambas regiones. El refinamiento se puede aplicar de manera recursiva, llevando con ello a distintos niveles de submallado anidados. Adicionalmente, este método propone una deformación de las celdas entre los distintos niveles de submallado, logrando con ello la ortogonalización de ciertas componentes electromagnéticas discretas. Esta deformación es definida a través de un parámetro de ortogonalización, y en este trabajo se demuestra que existe un valor óptimo que maximiza la estabilidad del algoritmo.

La estabilidad del método OI-SG es analizada mediante tres metodologías distintas. Estas metodologías buscan identificar, en función del parámetro de ortogonalización, cuál es el paso temporal más alto que se puede utilizar sin que el método sea inestable. Esto se hace identificando el valor estable más alto del denominado número de Courant-Friedrichs-Lewy (CFLN), que se trata de un parámetro adimensional proporcional al paso temporal clásicamente utilizado en FDTD y sus submétodos. En primer lugar, se obtiene una aproximación analítica de la condición necesaria para lograr estabilidad. Si bien no es rigurosa, su validez es verificada mediante las siguientes metodologías empleadas para la implementación con paso temporal local. Además, esta expresión analítica nos aporta información acerca de cuáles son las componentes discretas más críticas en la estabilidad, y esto es más adelante utilizado para desarrollar otra metodología para aumentar el valor del CFLN. En segundo lugar, se utiliza un método heurístico consistente en comprobar la conservación de la energía en una simulación tras un gran número de iteraciones, típicamente del orden de millones. En tercer lugar, se utiliza una metodología basada en el análisis espectral del operador de evolución temporal que permite hallar inestabilidades tardías en sistemas diagonalizables. Adicionalmente, se proporciona una variante del método espectral válida para sistemas no diagonalizables. Mediante las metodologías descritas, el valor óptimo del parámetro de ortogonalización y el CFLN son obtenidas para la implementación tanto con paso temporal global como local.

Para enfocar el problema de la dispersión numérica, en primer lugar, se analiza un problema de submallado en un escenario 1D. La expresión que rige la reflexión numérica en función de la frecuencia se obtiene analíticamente asumiendo un paso temporal global. Por otro lado, se demuestra que no existe solución analítica para la reflexión numérica con ondas

sinusoidales en el caso con paso temporal local, pero esta se aproxima mediante el comportamiento asintótico para altos refinamientos. En ambos casos, se obtienen reflexiones muy similares con una convergencia de segundo orden. En el caso 3D, las reflexiones son estudiadas numéricamente mediante simulaciones con incidencia normal, y los resultados son comparados con la reflexión numérica en 1D. Se muestra que, en el caso 3D, la convergencia cae a primer orden. La dispersión y la convergencia son estudiadas numéricamente mediante un caso canónico: una caja vacía con condiciones periódicas que contiene una región submallada. La matriz de evolución es obtenida para distintos refinamientos y, en cada caso, se calculan los autoestados y autovalores correspondientes. Se identifica el autoestado que mejor se ajusta al modo armónico fundamental en una de las direcciones de la caja y, a partir del autovalor correspondiente, se obtiene la frecuencia numérica. Dicha frecuencia es comparada con el valor analítico y se obtiene que la convergencia cae de segundo a primer orden al aumentar el refinamiento. El coste computacional del cálculo de los autoestados crece significativamente al aumentar el refinamiento y, por ende, las conclusiones obtenidas de este análisis están limitadas al rango que ha podido ser computado.

Diversas modificaciones para el método OI-SG son propuestas. En primer lugar, a partir de la información obtenida mediante la expresión analítica para la estabilidad, se desarrolla una técnica de celdas localmente agrandadas (LECT en inglés). Mediante el método espectral, se demuestra que esta metodología permite aumentar el valor del CFLN. En segundo lugar, se expone un método alternativo para lograr un paso temporal local sin recurrir a diferencias finitas no centradas, mediante el uso de interpolaciones y extrapolaciones. Se encuentra que este esquema es incondicionalmente inestable.

Con intención de lograr una implementación del método OI-SG para casos de simulación arbitrarios, se propone e implementa una metodología de mallado adaptativo. Esta técnica permite lograr simulaciones cuyas zonas refinadas se generan automáticamente en base a los objetos materiales presentes en el problema. Además, dentro de esta metodología se propone la posibilidad de dejar un espaciado opcional entre los objetos y la frontera del submallado. Se demuestra que este espaciado tiene un impacto positivo sobre la precisión numérica. Se propone también una implementación de paralelización híbrida OMP-MPI que permite lograr una mayor eficiencia en la ejecución.

Finalmente, el método OI-SG es puesto a prueba en diversos casos de simulación en escenarios variados para comprobar su desempeño. En primer lugar, se propone un caso de simulación con una zona vacía submallada con forma esférica para evaluar las reflexiones producidas por las fronteras numéricas. En segundo lugar, se evalúa la transmisión y

la reflexión en una superficie selectiva en frecuencia (FSS en inglés), así como el error en la transmisión de una metasuperficie con índice de refracción negativo. En tercer lugar, se evalúa la detectabilidad mediante la sección radar en una esfera metálica y dos casos de baja observabilidad: la ojiva denominada NASA almond y el objeto de baja detectabilidad FLAMME. En cuarto lugar, se evalúan diversos casos de compatibilidad electromagnética que evalúan el blindaje proporcionado por diversas estructuras ante incidencia de campos externos: una esfera conductiva con espesor, una caja metálica con dos ranuras, y un modelo modificado del avión EV55 en el que se estudia la generación de corrientes en el fuselaje y en cables contenidos en el interior de la cabina. En los casos de la NASA almond y el avión EV55, el submallado es además combinado con mallas conformes. En primer lugar, se observa que el uso del submallado permite aumentar la eficiencia en tiempo de cómputo y, especialmente, en el uso de memoria, al reducir drásticamente el número de campos electromagnéticos discretos presentes en una simulación. En todos los casos comprobados, se observa que dejar un espaciado entre los objetos y la frontera de nivel más fina aumenta significativamente la precisión numérica, aunque en algunos casos esto puede reducir drásticamente la eficiencia del método. Partiendo de una malla gruesa, se comprueba que el refinamiento mediante el submallado aumenta la precisión en términos generales. También se observa que el uso de LECT, si bien aumenta la estabilidad, disminuye significativamente la precisión, siendo por tanto poco apto en la mayor parte de escenarios. En los casos ejecutados en combinación con mallas conformes, se observa que la eficiencia puede aumentar significativamente más al utilizar refinamientos más gruesos, sin comprometer con ello la precisión. En definitiva, se muestra el rendimiento del método OI-SG, tanto en términos de eficiencia como de precisión, y con ello se comprueba su viabilidad como método numérico fiable.

Abstract

This dissertation is embedded within the field of computational electromagnetics (CEM). Similar to its numerical counterparts in other areas of physics, the importance of CEM methods has increased in recent decades, as it allows us to obtain solutions to complex problems that would be impossible to solve analytically. Furthermore, the use of numerical methods is particularly interesting in design stages in industrial processes, as it allows environments to be recreated synthetically, thus avoiding the construction of physical prototypes and, therefore, saving time and costs. Among numerical methods in electromagnetism, the finite-difference time-domain (FDTD) algorithm is known for its simplicity and efficiency. This technique requires a complete and structured discretization of the computational space in cells shaped as rectangular cuboids, thus providing particular robustness and computational efficiency as it is easily parallelizable. However, this same structured nature also results in some of the main limitations of the method. First, the cells do not adequately capture the curvature of materials that are not oriented along Cartesian axes. Second, we find the so-called multiscale problems: the presence of small details in a large simulation enforces a sufficiently fine mesh to capture them, thereby forcing us to waste computational resources in other parts of the simulation that do not contain relevant geometric information. The first problem can be solved by using conformal meshes, as these provide greater flexibility in the geometry of the materials. On the other hand, subcell models have been proposed in the literature to solve specific problems in multiscale cases in FDTD. However, a universal solution has not been yet found. In this context, subgridding methods constitute a promising field as they are a family of techniques that allow the refinement of specific parts of the simulations into smaller cells, therefore dividing the computational domain into different regions with different levels of spatial discretization. Specifically, this thesis studies a method originally published in 1997 and later denoted in the literature as the orthogonalized integral-based subgridding (OI-SG) method. This work is focused on the derivation of this method, its implementation for arbitrary simulation cases, the study of its numerical properties, its precision and efficiency in real numerical cases, and the identification of its strong and weak points.

Starting from the Ampère and Faraday equations in their integral form, and using the formalism of the main-dual grids in FDTD, the OI-SG method obtains an explicit formulation that allows the fine and coarse regions to be connected naturally. Optionally, a local

time stepping methodology based on non-centered finite differences is proposed, thus allowing us to use a different temporal discretization in both regions. Refinement can be applied recursively, thus leading to nested subgridding regions. Additionally, this method proposes a deformation of the cells between the different subgridding levels, thereby achieving the orthogonalization of certain discrete electromagnetic components. This deformation is defined by an orthogonalization parameter, and it is shown that an optimal value exists such that the stability of the algorithm is maximized.

The stability of the OI-SG method is analyzed using three different methodologies. These methodologies seek to identify the highest stable time step that can be used based on the orthogonalization parameter. This is done by identifying the highest stable value of the so-called Courant-Friedrichs-Lewy number (CFLN), which is a dimensionless parameter proportional to the time step and classically used in FDTD and its submethods. First, an analytical approximation of the required condition to achieve stability is obtained. Although it is not rigorous, its validity is verified through the following methodologies for the executions with local time stepping. Furthermore, this analytical expression provides us with information about which discrete components are most critical to stability, and this is later used to develop another methodology to increase the value of the CFLN. Second, a heuristic method is used consisting of checking the conservation of energy in a simulation after a large number of iterations, typically on the order of millions. Third, a methodology based on the spectral analysis of the evolution operator is used, allowing us to find late instabilities in diagonalizable systems. Additionally, a variant of the spectral method valid for non-diagonalizable systems is provided. Through the described methodologies, the optimal values of the orthogonalization parameter and CFLN are obtained for the implementation with both global and local time stepping.

As an approach to the problem of numerical dispersion, first, a 1D subgridding problem is analyzed. The expression that governs the numerical reflection as a function of frequency is obtained analytically assuming a global time stepping. On the other hand, it is shown that there is no analytical solution for the numerical reflection with sinusoidal waves when local time stepping is employed, but it can approximated by the asymptotic behavior for high refinements. In both cases, very similar reflections are obtained with second-order convergence. In the 3D case, the reflections are studied numerically through simulations with normal incidence, and the results are compared with the numerical reflection in 1D. It is shown that, in the 3D case, the convergence decays to first order. Dispersion and convergence are studied numerically using a canonical case: an empty box with periodic

conditions containing a subgridded region. The evolution matrix is obtained for different refinements and, in each case, the corresponding eigenstates and eigenvalues are calculated. The eigenstate that best fits the fundamental harmonic mode in one of the directions of the box is identified and, from the corresponding eigenvalue, the numerical frequency is obtained. This frequency is compared with the analytical value, and we find that the convergence falls from second to first order as the refinement increases. The computational cost of calculating the eigenstates grows significantly with increasing refinement and, therefore, the conclusions obtained from this analysis are limited to the range that could be computed.

Various modifications of the OI-SG method are proposed. First, from the information obtained through the analytical expression for stability, a locally enlarged cell technique (LECT) is developed. Using the spectral method, this methodology is demonstrated to allow larger stable values of CFLN. Second, an alternative method is presented to achieve a local time stepping without resorting to non-centered finite differences through the use of interpolations and extrapolations. This scheme is found to be unconditionally unstable.

To achieve an implementation of the OI-SG method for arbitrary simulation cases, an adaptive mesh refinement methodology is proposed and implemented. This technique allows us to obtain simulations whose refined regions are automatically generated based on the material objects present in the problem. Furthermore, within this methodology, the possibility of leaving an optional buffer spacing between the objects and the subgridding boundary is proposed. This spacing is shown to have a positive impact on numerical precision. An implementation of a hybrid OMP-MPI parallelization scheme is also proposed that allows greater efficiency in execution time.

Finally, the OI-SG method is tested in simulation cases in various scenarios to verify its performance. First, a simulation case is proposed with an empty sphere-shaped subgridded region to evaluate the reflections produced by the numerical boundaries. Second, the transmission and reflection in a frequency-selective surface (FSS) are evaluated as well as the error in the transmission of an all-angle negative refraction metasurface. Third, the radar cross section is evaluated in a perfect electric conductor (PEC) sphere and two low observability cases: the NASA almond ogive and the FLAMME stealth object. Fourth, various electromagnetic compatibility (EMC) are studied: the shielding effectiveness (SE) of a conductive spherical shell and a metal box with two slots is measured, and a modified model of the EV55 aircraft is used to study the generation of currents in the fuselage and cables present inside the cabin. In the cases of the NASA almond and the EV55 aircraft, subgridding is also combined with conformal meshes. First, we observe that the usage of

subgridding increases the efficiency in terms of computing time and, especially, in memory usage by drastically reducing the number of discrete electromagnetic fields present in a simulation. In all tested cases, it is observed that leaving a finer spacing between the objects and the level boundary significantly increases the numerical precision, although in some cases this can drastically reduce the efficiency of the method. Starting from a coarse mesh, it is verified that refinement through subgridding increases the precision in general terms. It is also observed that the use of LECT, despite enhancing the stability, significantly decreases precision, therefore being unsuitable for most scenarios. In cases executed in combination with conformal meshes, it is observed that efficiency can be further increased significantly by using coarser refinements without compromising accuracy. In short, the performance of the OI-SG method is shown, both in terms of efficiency and precision, and thus its viability as a reliable numerical method is verified.

Contents

CHAPTER 1

Introduction

The present dissertation is framed within the field of the computational electromagnetic (CEM) methods, focused specifically on subgridding (SG) techniques for the finite-difference time-domain (FDTD) method. The research conducted over five years is thoroughly documented and presented in this manuscript. This introductory chapter aims to provide a brief overview of the historical motivation that has led to this work. First, from an overview of the general electromagnetics and the necessity of CEM methods, this chapter delves into the specifics of the FDTD, subcell methods, and subgridding algorithms. Finally, it covers the aims of this work, which are focused on the study and application of the orthogonalized integral-based subgridding (OI-SG) method.

1.1 Overview of CEM methods

The equations of electromagnetism, also known as Maxwell's equations, were first published in 1865 [1]. They govern the behavior of electromagnetic phenomena, and since their original publication, they have allowed for great advances in science and industry. Maxwell's equations can be formulated as a set of four expressions that relate electrical charges, currents, and EM fields. The study of all the associated phenomena that emerge from these physical laws has been a key point in the development of human society for the last centuries, allowing for technologies such as circuitry and electromagnetic compatibility (EMC) in air vehicles, antennas in telecommunications, radars, novel materials development, etc. Over time, these technologies have become more complex, thus requiring more advanced methodologies to assist in their design. An analytical approach can be taken to predict the electromagnetic response of some designs, but the range of scenarios where such solutions can be found is very limited. On the other hand, experimental setups can be built and measured directly. However, in some design stages, especially when seeking to optimize a series of parameters, constant rework is required. This is impractical when working with physical prototypes as can easily lead to prohibitive costs. Furthermore, experimental setups involving complete objects, such as vehicles, aircraft, and ships, necessitate large facilities, which may be unfeasible.

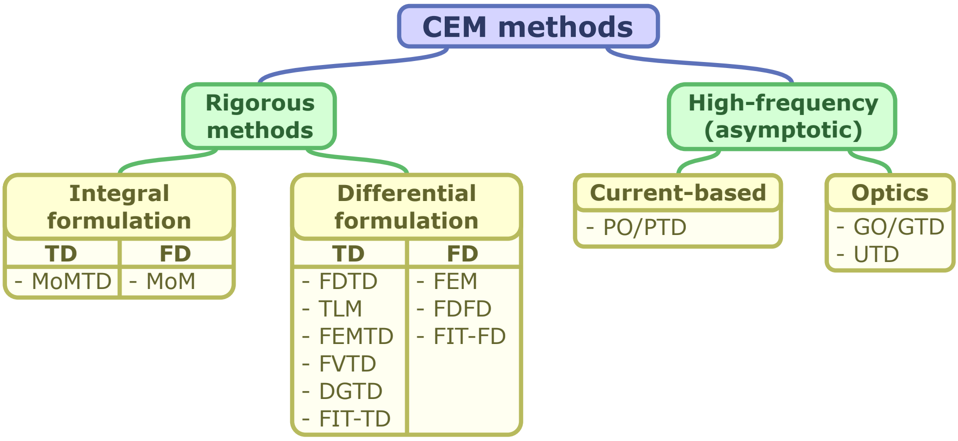

Figure 1.1: General classification of CEM methods.

Since the 1960s, with the advances of computation power, CEM entered the scene and began an exponential growth that continues to this day. This approach consists of computing synthetic scenarios, namely *simulations*, via numerical approximations of Maxwell's equations. This allows us to simulate any possible environment as long as we can derive a numerical approximation that can be computed with the present computer resources. One possible classification of the CEM methods most commonly found in the literature is provided in fig. 1.1. They are divided into full-wave (rigorous) methods and high frequency (HF) methods. Rigorous methods are based on a discretization of Maxwell's equations in their entirety and are classified depending on whether they discretize the entire computational domain (differential formulation) or disconnected regions (integral formulation), and whether they work in time-domain (TD) or frequency-domain (FD). An illustration of different types of discretization and examples of the methods that use them is provided in fig. 1.2. On the other hand, asymptotic methods are based on approximations in the high-frequency limit. A brief description of some of the main methods is provided below:

- **HF methods**: This family of methods is characterized by considering the asymptotic behavior at high frequencies, i.e. the limit at which the object sizes are much greater relative to the wavelength [2]. They typically have much less associated computational cost as we converge to high-frequency scenarios, at the expense of being only valid within this scope. Ray-optic methods are based on the laws of reflection and refraction supported by ray-tracing methods. In this regard, geometrical optics (GO) are very efficient since they only need to keep track of the light rays, but they don't properly reproduce phenomena such as interference or diffraction. Geometric theory of diffraction (GTD) [3] was designed for this purpose, incorporating wave information into GO. Similarly, uniform theory of diffraction (UTD) extends GTD by incorporating near-field phenomena [4–6]. Closer to full-wave methods, we find physical optics (PO) and physical theory of diffraction (PTD), which incorporate tangential surface currents to the calculus when rays impact a surface to GO and GTD [2, ch. 7].

- **Method of moments (MoM)**: Also known as the method of weighted residuals, it is one of the most applied ones in areas related to scattering and radiation [7, 8]. It operates in the frequency domain and requires a discretization of the objects present in the simulation, but not the free space. Each discretized element is then typically assigned a surface current consisting of an unknown linear combination of functions from a given basis, and interactions between elements are calculated using Green's functions. By minimizing the residual error, the resulting configuration forms a linear system of algebraic equations that must then be solved. Even though this is a highly accurate method, it requires large matrix inversions in order to solve the system, thus making it computationally intensive. MoM-TD is variation of MoM in time-domain, although this formulation is less common [9, 10].

- **Finite element method (FEM)**: Similar to MoM, FEM operates in the frequency domain and attempts to obtain a solution by minimizing the residual error [8]. Contrary to MoM, this method discretizes the whole computational domain into tetrahedrons, including free space, and therefore requires the usage of boundary conditions (BC) in the computational limits, typically consisting of perfect electric conductor (PEC) or absorbing boundary conditions (ABCs). By applying a variational formulation, a system of linear algebraic equations is obtained, which must be then solved. One of the main advantages of FEM is that, similarly to MoM, it adapts naturally to objects curvatures and small details, thus providing good accuracy. Additionally, unlike MoM, it adapts properly to bulk materials with continuous transitions or anisotropic behavior, and the complexity also scales slower than MoM. However, the discretization of the whole computational domain can lead to high memory usage depending on the case.

- **Finite-difference time-domain (FDTD)**: Similar to FEM, FDTD requires a discretization of the whole computational domain. However, FDTD follows a lattice structure, thus dividing the domain into a set of rectangular cuboids. More details about this method are provided in the next section and a technical discussion in section 2.1.

- **Other methods**: Finite-difference frequency-domain (FDFD) is the frequency-domain counterpart of FDTD. Similarly to MoM and FEM, it requires solving a system of algebraic equations. [11] Finite volume time domain (FVTD) is similar to FEM-TD but uses volumes as elements instead of surfaces and averages the magnitudes within them [12]. Discontinuous galerkin time domain (DGTD) is similar to FEM, but allowing for discontinuous function basis [13] Transmission-line-matrix (TLM) meshes the entire computational domain and models the EM field by an analogy with transmission lines [14]. Finite integration technique (FIT) is similar to FDTD but uses the integral formulation of Maxwell's equations [15, 16]. It still requires a discretization of the whole computational domain, but allows certain distortions of the grid, thus allowing it to adapt better to the problem geometry.

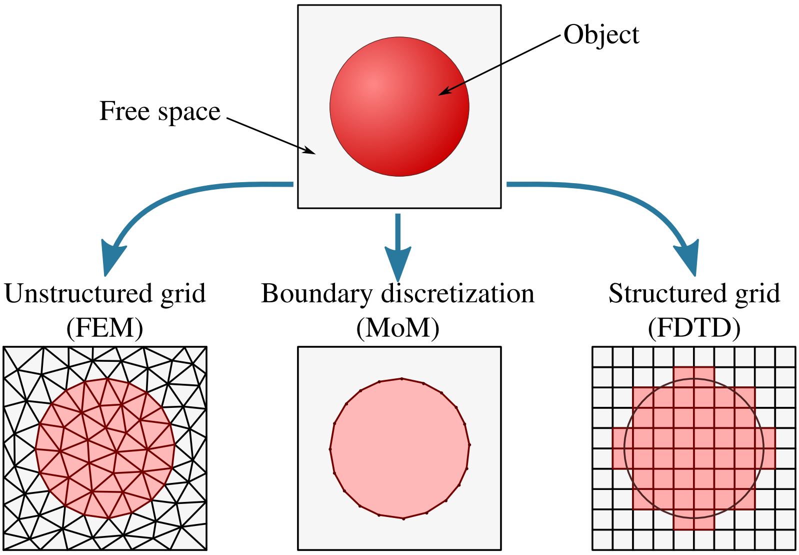

Figure 1.2: Examples of typical CEM discretizations and methods that use them.

No CEM method is suited for all tasks, each one providing trade-offs, thus requiring the user to weigh them and decide which one works better for their purpose. Particularly, TD methods are especially useful in cases where we require to determine the causality after a certain impulse containing a broad frequency band, or if the simulation contains non-linear materials. In this context, the present work focuses on the well-known FDTD method.

1.2 FDTD and limitations

FDTD method, also known as Yee's method, is a central piece of this work. A technical discussion is provided in section 2.1, however, an introduction to this method and its limitations is provided in what follows. Additionally, for a full comprehension of the method and many of its variations and submethods, the reader is encouraged to refer to [17] or [18].

FDTD was originally formulated by Yee in 1966 [19]. It requires a full discretization of the computational domain into a set of two staggered grids: one representing the electric field and the other one representing the magnetic field. The minimal voxels provided by these grids are called *cells*. These fields are configured in such a way that electric field components rotate around magnetic ones and vice versa. Additionally, a singular characteristic of this method is that discrete electric and magnetic field components are separated from each other in one-half time step. This configuration provides a natural discretization of Faraday's and Ampere's laws, precisely because said laws describe how time variations of electric and magnetic fields are related to rotations of one another. Then, by applying finite-difference to both spatial and time derivatives, we get to a simple and explicit time-stepping scheme. In the resulting algorithm, discrete electric and magnetic field components exist in different time instants, separated by half a time step. This is typically called a leapfrog time-stepping scheme.

The lattice meshing of the computational domain that FDTD utilizes can be, in principle, adapted to use a different space step in each Cartesian axis, and even accommodate different space steps in the same direction, which is referred to as a non-uniform grid. However, when a uniform grid is used in FDTD, finite differences are centered and therefore the convergence rate is of order $O(h^2)$ instead of $O(h)$ as in non-centered finite differences, where h is the characteristic length of the discretization [17]. It is worth mentioning that other higher-order FDTD formulations can be found in the literature [20, 21]. In this sense, the conventional FDTD is typically distinguished as FDTD (2,2). For the present work, unless otherwise indicated, this last one is simply referred to as FDTD.

One of the main advantages of FDTD is that it is arguably one of the simplest CEM methods to implement. Additionally, its numerical properties and stability are very well-known [17, 18, 22], it behaves properly with bulk materials, many BCs have been designed to simulate different environments and its structured-grid nature makes it an exceptionally efficient method to parallelize. This makes FDTD a very powerful tool despite its simplicity, and for this reason, it has been widely used over the past decades in both scientific literature and industrial applications. A recent review of FDTD and its applications can be found in [23].

The lattice structure of FDTD, however, comes at a cost. It requires meshing the entire computational domain following an, in principle, non-adaptative grid. Two main problems arise for this reason:

- The curvature of arbitrarily oriented materials cannot be properly captured by the FDTD grid. Particularly, any surface with a non-cartesian orientation results in a staircase mesh. This means that the distance between any two points in a grid is provided by the Manhattan distance regardless of the discretization length, therefore not converging to the real Euclidean distance.

- Simulation cases that combine large structures with very small objects (known as *multiscale problems*) require a fine enough discretization to capture the finest details. Due to the structured nature of the FDTD grid, this fine discretization extends to the rest of the computational domain. Consequently, computational resources are wasted on parts of the simulation that do not contain relevant geometric information, such as large volumes of free space.

The first problem mentioned can be solved by applying conformal methods. This is a family of techniques that allow to define material surfaces that may traverse through an FDTD cell (illustrated in fig. 1.3). Therefore, they adapt exceptionally well to large object curvatures. Some examples of conformal methods can be found in [24–30].

The problem related to multiscale scenarios can be partially overcome by the usage of non-uniform grids. However, as mentioned previously, this makes the convergence order of the finite-differences drop to $O(h)$, also inducing numerical reflections (typically called *spurious reflections*) and, more importantly, the non-uniformity of the grid necessarily extends along the orthogonal Cartesian axes (see fig. 1.4). Non-uniform grids also run into the Courant-Friedrichs-Lewy (CFL) criterion [22], which restricts the maximum time step allowed to ensure numerical stability through the Courant-Friedrichs-Lewy number (CFLN). This criterion fixes the maximum time step as a function of the minimum space step in each Cartesian axis, and therefore in a non-uniform grid, the overall time step of the simulation is restricted by the refined region in the simulation.

Another solution for the multiscale problems is the usage of *subcell methods*. This term can be somewhat ambiguous, but it often refers to any technique that modifies or substitutes the usual FDTD algorithm within a cell or a set of cells to simulate some behavior. One of the simplest approaches is the family of effective material techniques, which ponderates the physical properties of the closest materials to any given discrete field [31, 32]. Another example is the family of thin wire models, which permit the modeling of wires thinner than an FDTD cell [33, 34], and, on occasions, with arbitrary orientations [35]. As a last example, thin panel models, similar to the thin wire models, allow the simulation of panels thinner than a cell, in some cases with a layered structure [36–38].

In due context, all the mentioned subcell methods provide very good results, thus appreciably improving the utility of FDTD. However, they allow us to solve only specific scenarios, and thus a universal solution for the multiscale problem in FDTD requires diving into the so-called *subgridding* methods. In essence, all SG methods are based on the idea of refining a set of FDTD cells into two or more cells in each Cartesian axis (illustrated in fig. 1.4). In general, SG methods are very versatile, since they typically divide the computational domain into two regions: one made of non-refined (coarse) cells, and another one made of refined (fine) cells. This means that the original FDTD algorithm may be applied in the inner part of each region, and therefore any bulk material and subcell method can be potentially used within them. Additionally, the refinement process may be applied recursively, thus nested regions with different spatial refinement, typically referred to as *subgridding levels*. Furthermore, being separated regions (aside from the boundaries between them), a different time step can be used at each level. This technique is also known as local time-stepping (LTS), and it allows to keep a constant relation between the time step and the space step, therefore meeting the CFL criterion without limiting the time step in the entire computational domain, unlike non-uniform grids. On the other hand, global time-stepping (GTS) is defined as the scheme that keeps the same time discretization across all the regions.

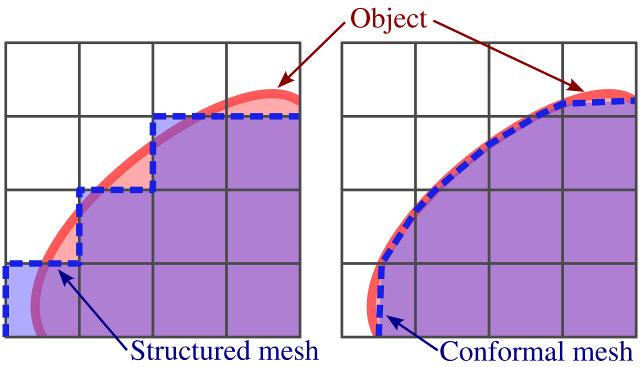

Figure 1.3: Illustration of the geometry problem solved by a conformal mesh.

The utility of subgridding methods may be described from two points of views. First, if we start from the finest mesh and grow the grid size on the empty parts of the simulation, we are interested in increasing efficiency without compromising accuracy. Second, if we start

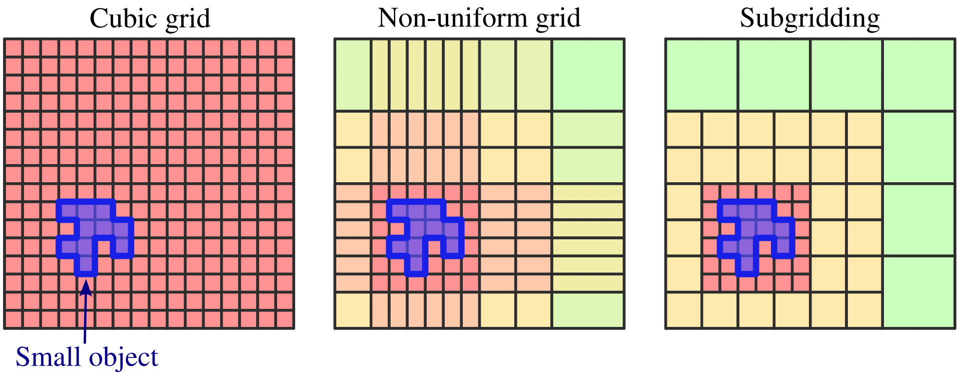

Figure 1.4: Illustration of the multiscale problem solved by a non-uniform grid and subgridding. The colors red, orange, and green are used to represent different levels of discretization.

from the coarsest level and reduce the mesh size in the most critical parts of the computational domain, we aim to increase the accuracy without dealing with the computational cost of a fully fine grid. In this sense, the idea of subgridding is exceptionally powerful and versatile, being a universal solution for many of the greatest weakest points in FDTD that can be combined with any other method. It is particularly interesting in multiscale problems, such as cases involving metasurfaces that typically have small repeating patterns [39–41], or electromagnetic compatibility (EMC) problems [42–44]. However, a non-trivial issue that needs to be addressed when dealing with SG methods is the algorithm employed to connect the different subgridding levels. This point is not negligible, since many different approaches have been designed in the literature for this purpose, many of them having minor and major drawbacks, as is exposed in section 2.3. Subgridding algorithms typically present instabilities, accuracy issues, and spurious reflections. This is particularly problematic, for example, in low observability (LO) cases, when measuring the EM reflection of an object that is expected to be very low, and thus the spurious reflections due to subgridding can fake the result. In some occasions, more advanced techniques are employed in SG methods at the cost of increasing the complexity and decreasing the computational efficiency. However, at this point, there is not one subgridding scheme that can be universally accepted as the best one in any of the use cases, and many authors still publish advances to this day.

1.3 Aims of the thesis

Electromagnetic designs become more complex as time passes, and thus more advanced CEM methods are required. As stated previously, FDTD is one of the most widely used CEM methods due to its simplicity, robustness and efficiency, and it has been so for almost

60 years at the moment of writing these lines. However, to continue being competitive, variations of the method that may overcome its weaknesses must be developed and studied properly. In this sense, subgridding algorithms entail a promising field due to their versatility. The hypothetical existence of a subgridding scheme with perfect characteristics would potentially make FDTD one of the most powerful CEM methods.

This work focuses on the orthogonalized integral-based subgridding (OI-SG) method, a general-purpose 3D SG method compatible with LTS. This method is based on the original work of Ritter and Arndt in 1997 [45]. It constitutes an interesting approach as its equations can be viewed as a generalization of the FDTD, thus barely increasing its complexity. This is the starting point of this dissertation. The aim of this work can be summarized in:

- Present the OI-SG method from a technical perspective.
- Properly study the numerical properties: stability, dispersion relation, convergence, accuracy, and efficiency.
- Propose and study modifications of the method in order to enhance its behavior.
- Study the method in real simulation cases, validating it against other methods and trying to find its weaknesses and strong points. Additionally, study the combination of the OI-SG method with other subcell methods within these simulation cases.

The study of the OI-SG method so far has led to the following original publications: [29, 30, 46–48].

The dissertation is structured as follows. In chapter 2, the background to this work, including the FDTD and the state-of-the-art in subgridding methods, is presented. Chapter 3 consists of a technical explanation and discussion of the OI-SG method, along with the study of some of its modifications and numerical properties. In chapter 4, the OI-SG method is tested against some canonical cases as well as potential industrial applications. Finally, in chapter 5, the conclusions drawn out of this work are presented, along with some future projections aimed at continuing the work in the following years.

CHAPTER 2

Background

This chapter focuses on the background that leads to this work. The following contents are provided:

- A technical discussion of the FDTD method and its limitations, including all the necessary basics required for any reader to understand this dissertation.
- A technical discussion of the subcell methods that are later utilized in combination with the OI-SG method.
- A thorough review of the subgridding methods, their different characteristics and the state-of-the-art in the literature.

2.1 The FDTD method

In this section, an overview of the FDTD method is provided, as well as the associated numerical characteristics, such as stability, dispersion relation, and convergence. Additionally, this chapter delves into the technical limitations of the standard FDTD related to the grid structure that lead to the necessity of extensions of the FDTD such as the subcell methods or subgridding.

2.1.1 Method description

FDTD method is typically derived from Maxwell's equations in their differential form. A detailed description of this procedure can be found in [17, Ch. 3] and [18]. However, for this work, the integral form is used, as it is later required for the derivation of the OI-SG method.

First, let us properly define the spatial and temporal discretizations along with the notation that will be used. Let us suppose an orthogonal 3D grid whose vertices in space are characterized by the 3-tuples indices (i, j, k), where i, j and k are integer values that range from 0 to N_x, N_y and N_z, respectively. The set of indices of all vertices in the grid can be viewed just as the Cartesian product of the 1D indices in each direction: $\{0, ..., N_x\} \times \{0, ..., N_y\} \times \{0, ..., Nz\}$. We denote the position of the vertex (i, j, k) as (x_i, y_j, z_k), where x_i, y_j, and z_k are just the 1D physical position of the vertex in their respective Cartesian axis. The voxels spawning from any point (x_i, y_j, z_k) to $(x_{i+1}, y_{j+1}, z_{k+1})$ are denoted *cells*, and its faces are denoted *surfels*. The line that joins two adjacent points in a given Cartesian direction is denoted *edge*. In general, two different edges may have different lengths, therefore we denote

$$\Delta x_{i+\frac{1}{2}} = x_{i+1} - x_i, \tag{2.1}$$

$$\Delta y_{j+\frac{1}{2}} = y_{j+1} - y_j, \tag{2.2}$$

$$\Delta z_{k+\frac{1}{2}} = z_{k+1} - z_k. \tag{2.3}$$

Within this grid, we also need to refer to physical positions that are not located at vertices, but at the center of edges, surfels, and cells. Thus, we generalize the 3-tuples indices as (I, J, K), where I, J, K can be integers or semi-integers. In this sense, if a 1D index I is semi-integer, we can write it as $i + \frac{1}{2}$ and it corresponds to the 1D physical position

$$x_{i+\frac{1}{2}} = x_i + \frac{\Delta x_{i+\frac{1}{2}}}{2}. \tag{2.4}$$

If all values I, J and K are integers we are referring to a grid vertex, if one of them is semi-integer we are referring to the center of an edge, if two of them are semi-integer we are referring to the center of a surfel, and if all three are semi-integers we are referring to the center of a cell. An illustration of this kind of grid is depicted in fig. 2.1. We also define the following space steps:

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \tag{2.5}$$

$$\Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \tag{2.6}$$

$$\Delta z_k = z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}. \tag{2.7}$$

Similarly, the time is discretized using time steps of length Δt. Given a physical magnitude

U, we denote $U|_{I,J,K}$ to U evaluated at the discrete position defined by the index (I,J,K), and $U|^{N}_{I,J,K}$ refers to $U|_{I,J,K}$ evaluated at a time instant $t = N\Delta t$, where N can be integer or semi-integer. Throughout this work, the following distinction is applied: lowercase indices i, j, k, and n are always necessarily integer; uppercase indices I, J, K, and N might be integer or fractional depending on the context.

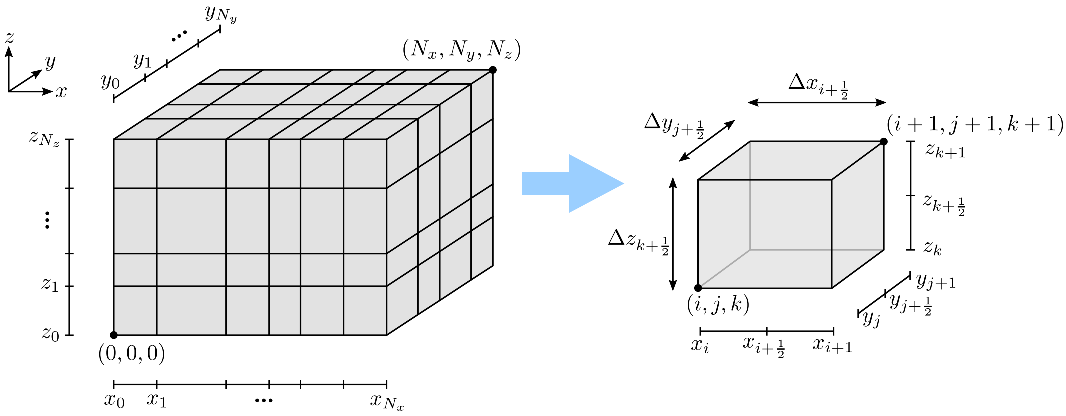

Figure 2.1: Illustration of the orthogonal grid used in FDTD and a cell with a diagonal spawning from (i,j,k) to $(i+1,j+1,k+1)$.

Now, let us take the rectangular surfel defined by a diagonal spawning from the index $(0,0,0)$ to $(1,1,0)$. We denote this surface as S. Let us also place a discrete magnetic field component H_z in the center of the surfel and four discrete electric components in the edges, each one oriented in the same direction as their respective edge. The resulting configuration is depicted in fig. 2.2.

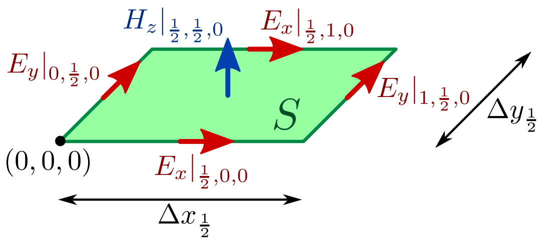

Figure 2.2: Square surfel used as integration surface to derive the FDTD update equation for a discrete field component H_z.

Let us now consider Faraday's Law with magnetic currents in its integral form applied to the surface S:

$$\oint_{\partial S} \vec{E} \cdot d\vec{l} = -\frac{d}{dt}\int_S \mu \vec{H} \cdot d\vec{s} - \int_S \sigma_m \vec{H} \cdot d\vec{s} - \int_S \vec{J}_m \cdot d\vec{s}, \tag{2.8}$$

where $\vec{E}$ is the electric field, $\vec{H}$ is the magnetic field, $\vec{J}_m$ is the free magnetic current density, μ is the magnetic permeability, σ_m is the magnetic conductivity, and ∂S denotes the boundary of S.

The time derivative of eq. (2.8) is approximated by utilizing the centered finite-difference technique. Thus, the approximate time derivative of a generic component $U|^N_{I,J,K}$ is written as

$$\frac{d}{dt} U|^N_{I,J,K} \simeq \frac{U|^{N+\frac{1}{2}}_{I,J,K} - U|^{N-\frac{1}{2}}_{I,J,K}}{\Delta t}. \tag{2.9}$$

Similarly, the integrals of eq. (2.8) can be numerically approximated by considering the value of H_z constant through the entire surfel, as well as the discrete electric field components through their respective edges. Applying both time and spatial approximations to eq. (2.8) evaluated at the time instant $t = n\Delta t$, we obtain the following expression:

$$\begin{aligned} &\frac{E_x|^n_{\frac{1}{2},0,0} - E_x|^n_{\frac{1}{2},1,0}}{\Delta y_{\frac{1}{2}}} + \frac{E_y|^n_{1,\frac{1}{2},0} - E_y|^n_{0,\frac{1}{2},0}}{\Delta x_{\frac{1}{2}}} = \\ &- \mu|_{\frac{1}{2},\frac{1}{2},0} \frac{H_z|^{n+\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0} - H_z|^{n-\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0}}{\Delta t} - \sigma_m|_{\frac{1}{2},\frac{1}{2},0} H_z|^n_{\frac{1}{2},\frac{1}{2},0} - J_{m,z}|^n_{\frac{1}{2},\frac{1}{2},0}. \end{aligned} \tag{2.10}$$

As a final step, we need to interpolate the discrete magnetic field component evaluated in $t = n\Delta t$ in order to have only two distinct discrete times for the magnetic components. This interpolation is written as

$$H_z|^n_{\frac{1}{2},\frac{1}{2},0} \simeq \frac{H_z|^{n+\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0} + H_z|^{n-\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0}}{2}. \tag{2.11}$$

Substituting eq. (2.11) in the discrete magnetic field associated with the magnetic conductivity in eq. (2.10), and performing some algebraic operations, we finally get to:

$$
\begin{aligned}
H_z\big|^{n+\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0} = {} & \frac{1-\dfrac{\sigma_m\big|_{\frac{1}{2},\frac{1}{2},0}\Delta t}{2\mu\big|_{\frac{1}{2},\frac{1}{2},0}}}{1+\dfrac{\sigma_m\big|_{\frac{1}{2},\frac{1}{2},0}\Delta t}{2\mu\big|_{\frac{1}{2},\frac{1}{2},0}}} H_z\big|^{n-\frac{1}{2}}_{\frac{1}{2},\frac{1}{2},0} \\
& + \frac{\Delta t}{\mu\big|_{\frac{1}{2},\frac{1}{2},0}\left(1+\dfrac{\sigma_m\big|_{\frac{1}{2},\frac{1}{2},0}\Delta t}{2\mu\big|_{\frac{1}{2},\frac{1}{2},0}}\right)} \left(\begin{array}{l} +\dfrac{E_x\big|^n_{\frac{1}{2},1,0}-E_x\big|^n_{\frac{1}{2},0,0}}{\Delta y_{\frac{1}{2}}} \\ -\dfrac{E_y\big|^n_{1,\frac{1}{2},0}-E_y\big|^n_{0,\frac{1}{2},0}}{\Delta x_{\frac{1}{2}}} \\ -J_{m,z}\big|^n_{\frac{1}{2},\frac{1}{2},0} \end{array} \right).
\end{aligned}
\tag{2.12}
$$

The expression obtained in eq. (2.12) is the FDTD update equation of the discrete field component $H_z\big|_{\frac{1}{2},\frac{1}{2},0}$. It is an explicit expression that allows us to calculate the next discrete value of the magnetic field at the time $t = (n+\frac{1}{2})\Delta t$, assuming that we know the previous value at the time instant $t = (n-\frac{1}{2})\Delta t$ and the surrounding electric fields at $t = n\Delta t$. This scheme places the discrete electric and magnetic field components not only in different locations in space but also in different discrete time instants. Discrete electric field components exist in times that are integer multiples of Δt, whereas magnetic ones exist in semi-integer multiples. This scheme allows us to update the discrete electric and magnetic field alternating between them, this is known as a *leapfrog* update algorithm.

An important aspect of eq. (2.12) is that an analogous expression can be derived for any field component H placed at the center of a surfel and surrounded by discrete electric components placed at the edges of said surfel. This is a consequence of the curl nature of Faraday's Law. Analogously, the update scheme can be derived for any electric component surrounded by magnetic ones using Ampère's Law. The last task that we must address to fully complete the FDTD derivation is to design a configuration within the grid for which all discrete field components meet the condition of being surrounded by discrete components of the other electromagnetic field. This can be addressed by utilizing the main-dual grid formalism. Let us consider the previously defined grid: we call it the *main grid*. Now, let us define a secondary grid whose vertices are placed in the center of each cell of the main grid, this is, in the indices $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$, with i, j and k integer values. We call this one the *dual* grid. The physical limits of the computational domain are defined by the main grid cells, however, it is worth noting that other implementations can consider the limits of the domain defined by the dual grid cells. Now, let us place the discrete electric field components at the center of each edge of the main grid, with each field component oriented in the direction of the corresponding edge. Analogously, let us place the discrete magnetic components in the center of the edges of the dual grid, again oriented towards its direction. This configuration is depicted in fig. 2.3.

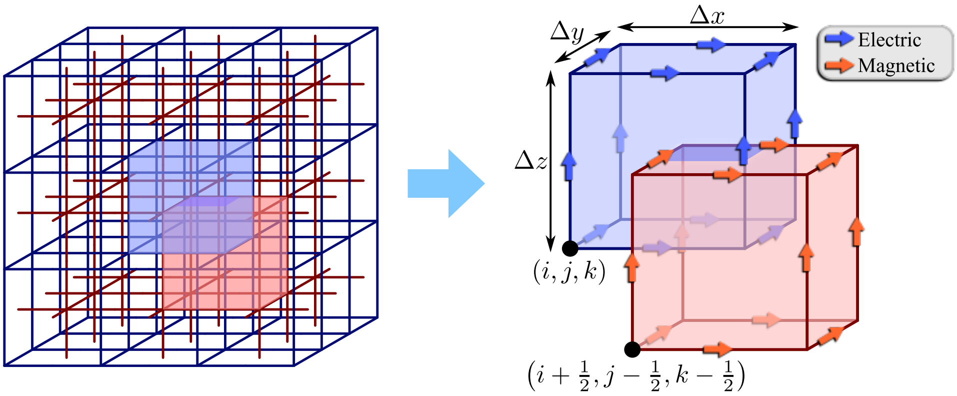

Figure 2.3: Illustration of the main-dual grid configuration used to derive the FDTD equations.

It is easy to see that this configuration provides exactly the conditions required to derive the update equations for every discrete field component in the same way that we did in eq. (2.12). Every discrete electric component is placed in the center of a surfel whose edges contain discrete magnetic components, and vice-versa. Now, we just need to consider Ampère's Law,

$$\oint_{\partial S} \vec{H} \cdot d\vec{l} = \frac{d}{dt} \int_S \varepsilon \vec{E} \cdot d\vec{s} + \int_S \sigma \vec{E} \cdot d\vec{s} + \int_S \vec{J} \cdot d\vec{s}, \tag{2.13}$$

where ε is the electric permittivity, and $\vec{J}$ is the electric current density. By applying an analogous procedure as we did for eq. (2.12), using Faraday's Law in the main grid's surfels, and Ampère's Law dual grid's surfels, we obtain the equations that constitute the FDTD method:

$$E_x\Big|_{i+\frac{1}{2},j,k}^{n+1} = \frac{1-\dfrac{\sigma|_{i+\frac{1}{2},j,k}\Delta t}{2\varepsilon|_{i+\frac{1}{2},j,k}}}{1+\dfrac{\sigma|_{i+\frac{1}{2},j,k}\Delta t}{2\varepsilon|_{i+\frac{1}{2},j,k}}}\,E_x\Big|_{i+\frac{1}{2},j,k}^{n} + \frac{\Delta t}{\varepsilon|_{i+\frac{1}{2},j,k}\left(1+\dfrac{\sigma|_{i+\frac{1}{2},j,k}\Delta t}{2\varepsilon|_{i+\frac{1}{2},j,k}}\right)}\left(\begin{array}{l}\dfrac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta y_j} \\ -\dfrac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z_k} \\ - J_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}\end{array}\right), \tag{2.14}$$

$$E_y\Big|_{i,j+\frac{1}{2},k}^{n+1} = \frac{1-\dfrac{\sigma|_{i,j+\frac{1}{2},k}\Delta t}{2\varepsilon|_{i,j+\frac{1}{2},k}}}{1+\dfrac{\sigma|_{i,j+\frac{1}{2},k}\Delta t}{2\varepsilon|_{i,j+\frac{1}{2},k}}}\,E_y\Big|_{i,j+\frac{1}{2},k}^{n} + \frac{\Delta t}{\varepsilon|_{i,j+\frac{1}{2},k}\left(1+\dfrac{\sigma|_{i,j+\frac{1}{2},k}\Delta t}{2\varepsilon|_{i,j+\frac{1}{2},k}}\right)}\left(\begin{array}{l}\dfrac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z_k} \\ -\dfrac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta x_i} \\ - J_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}}\end{array}\right), \tag{2.15}$$

$$E_z\big|^{n+1}_{i,j,k+\frac{1}{2}} = \frac{1-\dfrac{\sigma\big|_{i,j,k+\frac{1}{2}}\Delta t}{2\,\varepsilon\big|_{i,j,k+\frac{1}{2}}}}{1+\dfrac{\sigma\big|_{i,j,k+\frac{1}{2}}\Delta t}{2\,\varepsilon\big|_{i,j,k+\frac{1}{2}}}}\,E_z\big|^{n}_{i,j,k+\frac{1}{2}} + \frac{\Delta t}{\varepsilon\big|_{i,j,k+\frac{1}{2}}\left(1+\dfrac{\sigma\big|_{i,j,k+\frac{1}{2}}\Delta t}{2\,\varepsilon\big|_{i,j,k+\frac{1}{2}}}\right)}\left(\begin{array}{l} \dfrac{H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}-H_y\big|^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}}}{\Delta x_i} \\ -\dfrac{H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}-H_x\big|^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k+\frac{1}{2}}}{\Delta y_j} \\ -\,J_z\big|^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} \end{array}\right), \tag{2.16}$$

$$H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = \frac{1-\dfrac{\sigma_m\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}}}{1+\dfrac{\sigma_m\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}}}\,H_x\big|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta t}{\mu\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}\left(1+\dfrac{\sigma_m\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i,j+\frac{1}{2},k+\frac{1}{2}}}\right)}\left(\begin{array}{l} \dfrac{E_y\big|^{n}_{i,j+\frac{1}{2},k+1}-E_y\big|^{n}_{i,j+\frac{1}{2},k}}{\Delta z_{k+\frac{1}{2}}} \\ -\dfrac{E_z\big|^{n}_{i,j+1,k+\frac{1}{2}}-E_z\big|^{n}_{i,j,k+\frac{1}{2}}}{\Delta y_{j+\frac{1}{2}}} \\ -\,J_{m,x}\big|^{n}_{i,j+\frac{1}{2},k+\frac{1}{2}} \end{array}\right), \tag{2.17}$$

$$H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{1-\dfrac{\sigma_m\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}}}{1+\dfrac{\sigma_m\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}}}\,H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} + \frac{\Delta t}{\mu\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}\left(1+\dfrac{\sigma_m\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}\Delta t}{2\,\mu\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}}\right)}\left(\begin{array}{l} \dfrac{E_z\big|^{n}_{i+1,j,k+\frac{1}{2}}-E_z\big|^{n}_{i,j,k+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}} \\ -\dfrac{E_x\big|^{n}_{i+\frac{1}{2},j,k+1}-E_x\big|^{n}_{i+\frac{1}{2},j,k}}{\Delta z_{k+\frac{1}{2}}} \\ -\,J_{m,y}\big|^{n}_{i+\frac{1}{2},j,k+\frac{1}{2}} \end{array}\right), \tag{2.18}$$

$$H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{1-\dfrac{\sigma_m|_{i+\frac{1}{2},j+\frac{1}{2},k}\Delta t}{2\mu|_{i+\frac{1}{2},j+\frac{1}{2},k}}}{1+\dfrac{\sigma_m|_{i+\frac{1}{2},j+\frac{1}{2},k}\Delta t}{2\mu|_{i+\frac{1}{2},j+\frac{1}{2},k}}} H_z\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} + \frac{\Delta t}{\mu|_{i+\frac{1}{2},j+\frac{1}{2},k}\left(1+\dfrac{\sigma_m|_{i+\frac{1}{2},j+\frac{1}{2},k}\Delta t}{2\mu|_{i+\frac{1}{2},j+\frac{1}{2},k}}\right)} \begin{pmatrix} \dfrac{E_x|^n_{i+\frac{1}{2},j+1,k} - E_x|^n_{i+\frac{1}{2},j,k}}{\Delta y_{j+\frac{1}{2}}} \\ -\dfrac{E_y|^n_{i+1,j+\frac{1}{2},k} - E_y|^n_{i,j+\frac{1}{2},k}}{\Delta x_{i+\frac{1}{2}}} \\ - J_{m,z}|^n_{i+\frac{1}{2},j+\frac{1}{2},k} \end{pmatrix}. \tag{2.19}$$

The equations derived in eqs. (2.14)–(2.19) constitute the general form of the FDTD method in a non-uniform grid with arbitrary bulk isotropic materials and free currents with magnetic monopoles. If we consider the special case of a cubic grid, i.e. $\Delta x_I = \Delta y_J = \Delta z_K \equiv \Delta$, and a computational domain without sources and free-space only, these equations simplify to:

$$E_x\big|^{n+1}_{i+\frac{1}{2},j,k} = E_x\big|^{n}_{i+\frac{1}{2},j,k} + \frac{\Delta t}{\varepsilon_0\,\Delta} \begin{pmatrix} H_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k} \\ - H_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} + H_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}} \end{pmatrix}, \tag{2.20}$$

$$E_y\big|^{n+1}_{i,j+\frac{1}{2},k} = E_y\big|^{n}_{i,j+\frac{1}{2},k} + \frac{\Delta t}{\varepsilon_0\,\Delta} \begin{pmatrix} H_x|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} - H_x|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k-\frac{1}{2}} \\ - H_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} + H_z|^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2},k} \end{pmatrix}, \tag{2.21}$$

$$E_z\big|^{n+1}_{i,j,k+\frac{1}{2}} = E_z\big|^{n}_{i,j,k+\frac{1}{2}} + \frac{\Delta t}{\varepsilon_0\,\Delta} \begin{pmatrix} H_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H_y|^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}} \\ - H_x|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} + H_x|^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k+\frac{1}{2}} \end{pmatrix}, \tag{2.22}$$

$$H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = H_x\big|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta t}{\mu_0\,\Delta} \begin{pmatrix} E_y|^{n}_{i,j+\frac{1}{2},k+1} - E_y|^{n}_{i,j+\frac{1}{2},k} \\ - E_z|^{n}_{i,j+1,k+\frac{1}{2}} + E_z|^{n}_{i,j,k+\frac{1}{2}} \end{pmatrix}, \tag{2.23}$$

$$H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} = H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} + \frac{\Delta t}{\mu_0\,\Delta}\begin{pmatrix} E_z\big|^{n}_{i+1,j,k+\frac{1}{2}} - E_z\big|^{n}_{i,j,k+\frac{1}{2}} \\ - E_x\big|^{n}_{i+\frac{1}{2},j,k+1} + E_x\big|^{n}_{i+\frac{1}{2},j,k} \end{pmatrix}, \tag{2.24}$$

$$H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} = H_z\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} + \frac{\Delta t}{\mu_0\,\Delta}\begin{pmatrix} E_x\big|^{n}_{i+\frac{1}{2},j+1,k} - E_x\big|^{n}_{i+\frac{1}{2},j,k} \\ - E_y\big|^{n}_{i+1,j+\frac{1}{2},k} + E_y\big|^{n}_{i,j+\frac{1}{2},k} \end{pmatrix}. \tag{2.25}$$

The simplified equations derived in eqs. (2.20)–(2.25) are commonly used in FDTD because the cubic grid reduces the numerical anisotropy. Additionally, especially in radiating problems, most of the computational domain is typically empty, therefore these are the equations employed in such cases. In chapter 3, these equations are relevant since the derivation of the OI-SG method relies on a very similar, but more general, approach as the one used here.

2.1.2 Boundary conditions

It is easy to notice that the update of every discrete field component in the FDTD method requires four surrounding components. Yet, this condition cannot be met for the discrete electric fields tangential to the boundaries of the computational domain. For example, let us consider the discrete electric component $E_x|_{\frac{1}{2},j,0}$ for any j. By looking at eq. (2.14) or eq. (2.20), it is clear that we require the value of $H_y|_{\frac{1}{2},j,-\frac{1}{2}}$. However, this discrete field component does not exist, since it is outside of the limits of the main grid. For this reason, the FDTD method requires utilizing boundary conditions.

Many different boundary conditions have been designed for a variety of purposes. The most common ones are perfect electric conductor (PEC), perfect magnetic conductor (PMC), periodic, and the family of absorbing boundary conditions (ABCs). Some of these boundary conditions are used in the numerical simulations presented throughout this work. Thus, a brief explanation of them is presented in this section.

Perfect Electric Conductor (PEC)

In FDTD implementations where the computational domain limits are defined by the main (electric) grid, PEC boundary condition constitutes the simplest one that can be implemented. It consists of simulating a PEC surface (i.e. a surface with infinite electric conductivity) in the boundaries of the computational domain. A well-known fact about PEC objects is that the electric field tangential to the surface is necessarily zero. Therefore, the implementation of this boundary condition reduces to keeping all the electric fields tangential to the computational boundaries zero-valued:

$$\begin{aligned}
&E_x|^n_{i+\frac{1}{2},j,0} = E_x|^n_{i+\frac{1}{2},j,N_z} = E_x|^n_{i+\frac{1}{2},0,k} = E_x|^n_{i+\frac{1}{2},N_y,k} = \\
&E_y|^n_{0,j+\frac{1}{2},k} = E_y|^n_{N_x,j+\frac{1}{2},k} = E_y|^n_{i,j+\frac{1}{2},0} = E_y|^n_{i,j+\frac{1}{2},N_z} = \\
&E_z|^n_{i,0,k+\frac{1}{2}} = E_z|^n_{i,N_y,k+\frac{1}{2}} = E_z|^n_{0,j,k+\frac{1}{2}} = E_z|^n_{N_x,j,k+\frac{1}{2}} = 0 \\
&\forall i \in \{0,...,N_x\}, j \in \{0,...,N_y\}, k \in \{0,...,N_z\}, n \in \mathbb{N}.
\end{aligned} \tag{2.26}$$

Perfect Magnetic Conductor (PMC)

Analogously to a PEC object, a PMC object has infinite magnetic conductivity. In this case, it is the magnetic tangential field that must be zero. However, unlike what happened with the electric field, the configuration that we have defined for the discrete field component positions does not provide any discrete magnetic component tangential to the computational boundaries. To simulate this, let us consider the discrete magnetic components that are closest and parallel to the computational boundaries. For each one of them, we may assume the existence of another magnetic component outside the computational domain, at the same distance, with the same orientation, but opposite sign. If we now interpolate the parallel discrete magnetic components in the simulation boundary, since they have opposite signs, the result would be zero, hence this configuration simulates a PMC surface. Now, we only need to update the discrete tangential electric components in the simulation boundaries accordingly. For example, assuming a cubic grid with a space step of Δ, the update of an arbitrary discrete electric component E_x located at the Z-lower bound of the domain would be written as

$$\begin{aligned}
E_x|^{n+1}_{i+\frac{1}{2},j,0} = {} & E_x|^n_{i+\frac{1}{2},j,0} \\
& + \frac{\Delta t}{\varepsilon_0 \Delta}\left(H_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},0} - H_z|^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},0} - 2\, H_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},0,\frac{1}{2}} \right).
\end{aligned} \tag{2.27}$$

Analogous equations can be obtained for all the discrete tangential electric components within the computational boundaries.

As a last note, when a discrete tangential electric component is located at one edge of the computational domain, i.e. it belongs to two boundary faces simultaneously, it must be updated with two duplicated discrete magnetic components. As an example, the update of an arbitrary discrete electric component E_x located at the Z-lower and Y-lower bound of the

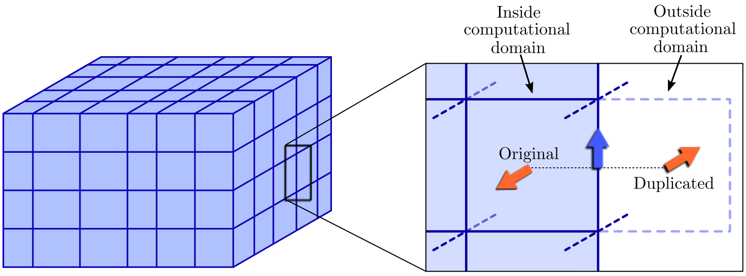

Figure 2.4: Scheme of the magnetic field duplication performed in PMC boundary conditions.

domain would be written as

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},0,0} &= E_x\big|^{n}_{i+\frac{1}{2},0,0} \\ &+ \frac{\Delta t}{\varepsilon_0\,\Delta}\left(2\;H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},\frac{1}{2},0} - 2\;H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},0,\frac{1}{2}}\right). \end{aligned} \tag{2.28}$$

Periodic conditions

Periodic boundary conditions must be applied in pairs of lower-upper bound surfaces of the computational domain, and they simulate an infinite repetition of the simulation in the direction normal to said bounds. This approach creates an identification between said bound surfaces, and thus discrete tangential electric components in a lower bound make use of the discrete parallel magnetic components placed closest to the upper bound. Analogously, discrete tangential electric components situated in the upper bound make use of the parallel magnetic ones situated closest to the lower bound. For example, the update of an arbitrary discrete electric component E_x located at the Z-lower bound of the domain would be written as

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},j,0} &= E_x\big|^{n}_{i+\frac{1}{2},j,0} \\ &+ \frac{\Delta t}{\varepsilon_0\,\Delta}\left(H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},0} - H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},0} - H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,\frac{1}{2}} + H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,N_z-\frac{1}{2}}\right). \end{aligned} \tag{2.29}$$

Similarly, if a field component E_x is located at an edge of the computational domain defined by the Y-lower and Z-lower bounds, and both faces implement periodic conditions, the update would be:

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},0,0} &= E_x\big|^{n}_{i+\frac{1}{2},0,0} \\ &+ \frac{\Delta t}{\varepsilon_0\,\Delta}\left(H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},\frac{1}{2},0} - H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},N_y-\frac{1}{2},0} - H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},0,\frac{1}{2}} + H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},0,N_z-\frac{1}{2}}\right). \end{aligned} \tag{2.30}$$

Convolutional Perfectly Matched Layer (CPML)

PML boundary conditions belong to the family of ABCs. These algorithms aim to simulate an infinite space extending beyond the computational domain, which is necessary for many problems. Multiple variants of PML can be found in the literature, but the convolutional perfectly matched layer (CPML) [49] are known for providing an exceptionally high absorption, thus making them suitable for most problems. The main idea behind the CPML method is to extend the computational domain and fill the extended region with a reflectionless lossy material, such that any incident wavefront with any incidence angle enters this material and loses all its energy before reaching the actual boundary of the computational domain. Such material is, of course, non-physical, but it can be simulated in various ways. The CPML approaches this by operating in a non-orthogonal coordinate system in FD referred to as *stretched-coordinates*. This coordinate system translates into a convolutional formulation of Maxwell's curl equations in TD. In turn, this convolution translates into electric and magnetic density currents that, in every discrete time instant, are recalculated based on the previous values of the discrete electric and magnetic fields, respectively. Additionally, it implements a lossy material with electric and magnetic conductivity, whose properties grow polynomically from the beginning of the region until reaching the computational boundary. A fully detailed explanation can be found in [17, Ch. 7]. Nonetheless, a formulation is presented here for completitude.

As an example, let us write down the update of the discrete field components E_x in the upper-Y CPML region, assuming it is embedded in a free-space cubic grid:

$$\begin{aligned} E_x\big|_{i+\frac{1}{2},j,k}^{n+1} = {} & E_x\big|_{i+\frac{1}{2},j,k}^{n} \\ & + \frac{\Delta t}{\varepsilon_0\,\Delta}\left(\begin{aligned} & H_z\big|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z\big|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} - H_y\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + H_y\big|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \\ & + \Psi_{E_{x,y}}\big|_{i+\frac{1}{2},j,k}^{n} \end{aligned} \right), \end{aligned} \tag{2.31}$$

where $\Psi_{E_{x,y}}\big|_{i+\frac{1}{2},j,k}^{n}$ is the electric current associated to the field E_x in the upper- and lower-Y bounds. It is calculated as:

$$\Psi_{E_{x,y}}\big|_{i+\frac{1}{2},j,k}^{n} = b_{y_j}\,\Psi_{E_{x,y}}\big|_{i+\frac{1}{2},j,k}^{n-1} + c_{y_j}\left(\frac{H_z\big|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z\big|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta} \right). \tag{2.32}$$

The parameters b_{y_j} and c_{y_j} are calculated, respectively, as:

$$b_{y_j} = e^{-\left(\frac{\sigma_{y\,j}}{\varepsilon_0 \kappa_{y\,j}} + \frac{\alpha_{y\,j}}{\varepsilon_0}\right)\Delta t}, \tag{2.33}$$

$$c_{y_j} = \frac{\sigma_{y_j}}{\sigma_{y_j}\kappa_{y_j} + \kappa_{y_j}^2 \alpha_{y_j}} \left[e^{-\left(\frac{\sigma_{y\,j}}{\varepsilon_0 \kappa_{y\,j}} + \frac{\alpha_{y\,j}}{\varepsilon_0}\right)\Delta t} - 1 \right]. \tag{2.34}$$

Here, the parameters σ_{y_j}, α_{y_j}, and κ_{y_j} are numerical parameters that grow polynomically from the beginning of the CPML until reaching the computational boundary,

$$\sigma_{y_j} = \sigma_{\text{max}} \left(\frac{j - j_{\text{begin}}}{j_{\text{end}} - j_{\text{begin}}} \right)^m, \tag{2.35}$$

$$\alpha_{y_j} = \alpha_{\text{max}} \left(\frac{j - j_{\text{begin}}}{j_{\text{end}} - j_{\text{begin}}} \right)^m, \tag{2.36}$$

$$\kappa_{y_j} = 1 + (\kappa_{\text{max}} - 1) \left(\frac{j - j_{\text{begin}}}{j_{\text{end}} - j_{\text{begin}}} \right)^m, \tag{2.37}$$

where j_{begin} and j_{end} are, respectively, the first and last indices of the CPML region, m is the polynomial order, and σ_{max}, α_{max}, and κ_{max} are numerical parameters that must be chosen accordingly to minimize the numerical reflection of the CPML region. We also define the number of layers of the CPML region as $N_{\text{layers}} = j_{\text{end}} - j_{\text{begin}}$, and the physical depth $L = N_{\text{layers}}\Delta$. Typically, the value of σ_{max} is calculated from the desired reflection coefficient for normal incident wavefronts, $R(0)$,

$$\sigma_{\text{max}} = -\frac{(m+1)\ln[R(0)]}{2\eta_0 L}, \tag{2.38}$$

where η_0 is the vacuum impedance,

$$\eta_0 = \sqrt{\frac{\mu_0}{\varepsilon_0}}. \tag{2.39}$$

An example of values frequently used throughout this work are $N_{\text{layers}} = 10$, $m = 2$, $R(0) = 10^{-7}$, $\alpha_{\text{max}} = 0$, $\kappa_{\text{max}} = 1$.

Analogous expression can be obtained for every field component normal to the desired bound. Additionally, the convolutional currents must be added when two CPML regions converge in an edge or corner region.

2.1.3 Plane wave source

Plane wave source techniques simulate the propagation of plane waves with arbitrary directions within a particular region of the computational space. A fully detailed explanation of different kinds of plane wave sources can be found in [17, Ch. 5]. In this section, the kind used for this work will be presented.

A plane wave source is defined by four things: the profile, the propagation direction, the wave polarization, and the region. The profile is the time function $f(t)$ that defines the shape of the wavefront. By convention, we assume that it begins at $t = 0$, i.e. $f(t) = 0$ if $t < 0$. The direction of propagation is marked by a unit vector $\hat{k}$. The wave polarization is defined by two unit vectors $\hat{p}_e$ and $\hat{p}_h$, where $\hat{p}_e$ defines the polarization of the electric field and $\hat{p}_h$ defines the polarization of the magnetic field. It is required that $\hat{k}$ and $\hat{p}_e$ are orthogonal, and $\hat{p}_h = \hat{k} \times \hat{p}_e$. The region is a computational box that bounds the wave propagation. It is defined by two main-grid vertices (i_0, j_0, k_1) and (i_1, j_1, k_1). A 2D cross-section is illustrated in fig. 2.5. Within the plane wave region, the total electromagnetic field can be split into two: the propagating plane wave, and the rest of the fields, which include scattered fields and other fields induced by other sources. This, for an arbitrary component E_x can be written as

$$\begin{aligned} &E_x|_{i+\frac{1}{2},j,k} = E_{x,\mathrm{PW}}|_{i+\frac{1}{2},j,k} + E_{x,\mathrm{rest}}|_{i+\frac{1}{2},j,k} \equiv E_{x,\mathrm{tot}}|_{i+\frac{1}{2},j,k} \\ &\text{if} \quad i+\frac{1}{2} \in [i_0, i_1], j \in [j_0, j_1], k \in [k_0, k_1], \end{aligned} \tag{2.40}$$

where $E_{x,\mathrm{PW}}$ is the contribution due to the plane wave, $E_{x,\mathrm{rest}}$ is the rest of the propagating fields, and $E_{x,\mathrm{tot}}$ is the summation of both. Similarly, outside this region, we do not have the plane wave field:

$$\begin{aligned} &E_x|_{i+\frac{1}{2},j,k} = E_{x,\mathrm{rest}}|_{i+\frac{1}{2},j,k} \\ &\text{if} \quad i+\frac{1}{2} \notin [i_0, i_1] \text{ or} j \notin [j_0, j_1] \text{ or} k \notin [k_0, k_1]. \end{aligned} \tag{2.41}$$

Similar expressions can be obtained for the other electromagnetic components.

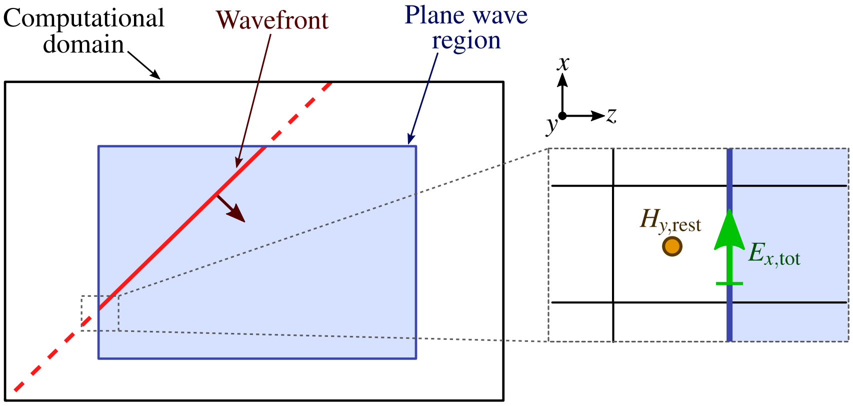

Figure 2.5: Scheme of the plane wave source in FDTD. The wavefront only propagates within the designated region.

The plane wave field can be written as

$$\begin{aligned} E_{\alpha,\text{PW}}\big|^n_{I,J,K} &= (\hat{\alpha}\cdot\hat{p}_e)\, f(n\Delta t - t_0|_{I,J,K}), \\ H_{\alpha,\text{PW}}\big|^n_{I,J,K} &= (\hat{\alpha}\cdot\hat{p}_h)\, f(n\Delta t - t_0|_{I,J,K}), \end{aligned} \tag{2.42}$$

where $\alpha \in \{x,y,z\}$, $\hat{\alpha}$ is the α-directed unit vector, and $t_0|_{I,J,K}$ is the delay of the wavefront at the location $\vec{r}|_{I,J,K} = (I\Delta x, J\Delta y, K\Delta z)$. This delay can be easily calculated as

$$t_0|_{I,J,K} = \frac{\left(\vec{r}|_{I,J,K} - \vec{r}_0\right)\cdot\hat{k}}{c_0}, \tag{2.43}$$

where $\vec{r}_0$ is a point indicating the plane where the wavefront begins at the beginning of the simulation, and must be chosen appropriately in such a way that this plane does not intersect with the propagation region but the wavefront is directed towards it.

Let us now take a closer look at one of the boundaries of the plane wave region. For example, let us consider the component E_x and H_y at the lower-z boundary, i.e. $k = k_0$. Here, we may see that $E_x|_{i+\frac{1}{2},j,k_0}$ should contain both the plane wave and the rest of the field contributions, whereas $H_y|_{i+\frac{1}{2},j,k_0-\frac{1}{2}}$ should not contain the plane wave because it is outside the propagation region. If we apply the FDTD update eqs. (2.20) and (2.24) to these fields, we find:

$$E_x\big|^{n+1}_{i+\frac{1}{2},j,k_0} = E_x\big|^{n}_{i+\frac{1}{2},j,k_0} + \frac{\Delta t}{\varepsilon_0\Delta} H_{y,\text{tot}}\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} + \ldots, \tag{2.44}$$

$$H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} = H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} - \frac{\Delta t}{\mu_0\,\Delta}\,E_{x,\mathrm{rest}}\big|^{n}_{i+\frac{1}{2},j,k_0} + \dots. \tag{2.45}$$

It is clear that FDTD cannot be applied like this, because neither $H_{y,\mathrm{tot}}\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}}$ nor $E_{x,\mathrm{rest}}\big|^{n}_{i+\frac{1}{2},j,k_0}$ are stored in memory. However, if we split $H_{y,\mathrm{tot}} = H_{y,\mathrm{PW}} + H_{y,\mathrm{rest}}$ and $E_{x,\mathrm{rest}} = E_{x,\mathrm{tot}} - E_{x,\mathrm{PW}}$, we obtain an expression that can be achieved by applying the usual FDTD with an additional source in each expression:

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},j,k_0} = E_x\big|^{n}_{i+\frac{1}{2},j,k_0} &+ \frac{\Delta t}{\varepsilon_0\,\Delta}\,H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} + \dots \\ &+ \frac{\Delta t}{\varepsilon_0\,\Delta}\,H_{y,\mathrm{PW}}\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}}, \end{aligned} \tag{2.46}$$

$$\begin{aligned} H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} = H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k_0-\frac{1}{2}} &- \frac{\Delta t}{\mu_0\,\Delta}\,E_x\big|^{n}_{i+\frac{1}{2},j,k_0} + \dots \\ &+ \frac{\Delta t}{\mu_0\,\Delta}\,E_{x,\mathrm{PW}}\big|^{n}_{i+\frac{1}{2},j,k_0}. \end{aligned} \tag{2.47}$$

By following an analogous procedure, similar expressions can be obtained for $E_y\big|_{i,j+\frac{1}{2},k_0}$ and $H_x\big|_{i,j+\frac{1}{2},k-\frac{1}{2}}$ in the lower-z boundary, and for the tangential fields in the rest of the region boundaries.

Boundary truncation

An inherent limitation of the plane wave source is the fact that it is necessarily confined to a region within the computational domain. This is particularly problematic in conditions of normal incidence on a surface that reaches the computational limits. In principle, the boundaries of the plane wave region cannot reach the computational limits because they require modifying magnetic fields half a cell outside the region. However, in some conditions, it is possible to simulate a plane wave region that extends until the computational limits on some axes.

Particularly, let us assume a plane wave source that is propagating towards a purely Cartesian direction $\hat{k}$ and has a purely Cartesian electric and magnetic polarizations $\hat{p}_e$ and $\hat{p}_h$. This means that the electric fields directed towards $\hat{k}$ or $\hat{p}_h$ are all zero, and the same happens with the magnetic fields directed towards $\hat{k}$ or $\hat{p}_e$. Thus, if we apply PEC boundary conditions to the boundaries orthogonal to $\hat{p}_e$ and PMC to the boundaries orthogonal to $\hat{p}_h$, we may extend the plane wave region in these two axes and truncate it at the computational limit, providing exactly the required conditions for the wave propagation without any distortions. This is depicted in fig. 2.6.

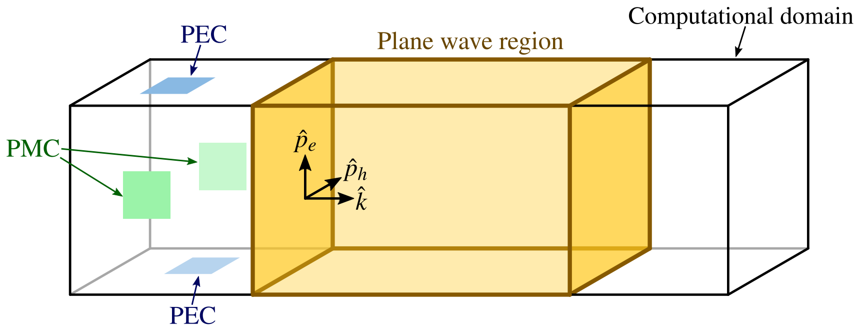

Figure 2.6: Scheme of the PEC-PMC plane wave truncation conditions.

Another possibility is to apply periodic boundary conditions in said boundaries, but they are usually more difficult to implement and can be slightly less efficient. The choice between periodic and PEC-PMC truncation conditions will mostly depend on the requirements of each case.

2.1.4 Dispersion and stability

In electromagnetics and many other fields in physics, the behavior of magnitudes is typically decomposed into different frequencies via the Fourier transform. This provides a linear bijection from TD to FD, and vice-versa. Additionally, this mechanism transforms real differential equations in TD to complex arithmetic equations in FD. A direct consequence is that the behavior of different frequencies is completely independent of each other: you must solve a different system of arithmetic equations for each different frequency, whereas in TD the values in one time instant depend on the immediate previous ones. In the context of propagating waves in 3D space, the wavefront of a magnitude U corresponding to a given angular frequency ω is

$$U(\vec{r},t) = U_0\, e^{\mathbb{i}(\omega t - k_x x - k_y y - k_z z + \phi_0)}, \tag{2.48}$$

where $\vec{r} = (x,y,z)$ is the position in space, t is the time, U_0 is the wave amplitude, $\mathbb{i}$ is the imaginary unit, ϕ_0 is the initial phase at $\vec{r} = 0$, and $\vec{k} = (k_x, k_y, k_z)$ is the wavenumber vector, which is oriented towards the direction of propagation and whose magnitude k is related to the wavelength λ as

$$k = \frac{2\pi}{\lambda}. \tag{2.49}$$

The wavefront written in eq. (2.48) is typically called a *monochromatic* wavefront because it contains only one frequency. In electromagnetics, U is typically the electric and magnetic field components. It is worth noting that a real monochromatic sinusoidal wave can be obtained by the composition (summation) of two complex-conjugate monochromatic wavefronts. Nonetheless, it is more common to work with monochromatic wavefronts in the complex-exponential form for simplicity.

The traveling speed of a monochromatic wavefront is typically called *phase velocity* and is denoted as v_{phase}. It can be calculated as

$$v_{\text{phase}} = \frac{\omega}{k}. \tag{2.50}$$

When referring to electromagnetic waves propagating in free space, this speed is analytically the vacuum speed of light c_0, which can be calculated from the vacuum electric permittivity ε_0 and μ_0 as

$$c_0 = \frac{1}{\sqrt{\varepsilon_0\,\mu_0}}. \tag{2.51}$$

When propagating inside a medium, v_{phase} is usually different from c_0 and might even depend on the frequency if the material is electromagnetically dispersive. In this context, eq. (2.50) is often referred to as the *dispersion relation*.

Ideally, CEM methods should preserve the analytical dispersion relation for any given medium, and particularly they should do so in free space. However, it can be proven that this is not the case in FDTD. Let us consider a uniform grid with arbitrary space steps in each Cartesian direction: Δ_x, Δ_y, and Δ_z. Now, let us assume a monochromatic wave solution for the system, which can be written as

$$E_x\big|_{i,j,k}^{n} = E_{x,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}, \tag{2.52}$$

$$E_y\big|_{i,j,k}^{n} = E_{y,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}, \tag{2.53}$$

$$E_z\big|_{i,j,k}^{n} = E_{z,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}, \tag{2.54}$$

$$H_x\big|_{i,j,k}^{n} = H_{x,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}, \tag{2.55}$$

$$H_y\big|_{i,j,k}^{n} = H_{y,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}, \tag{2.56}$$

$$H_z\big|_{i,j,k}^{n} = H_{z,0}\, e^{\mathbb{i}(\omega n \Delta t - k_x i \Delta x - k_y j \Delta y - k_z k \Delta z)}. \tag{2.57}$$

For simplicity, we have assumed $\phi_0 = 0$ and $(x_0, y_0, z_0) = (0,0,0)$. Now, by plugging eqs. (2.52)–(2.57) into the FDTD equations in free space without sources, we obtain:

$$\frac{\varepsilon_0}{\Delta t} E_{x,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta z} H_{y,0} \sin\left(\frac{k_z \Delta z}{2}\right) - \frac{1}{\Delta y} H_{z,0} \sin\left(\frac{k_y \Delta y}{2}\right), \tag{2.58}$$

$$\frac{\varepsilon_0}{\Delta t} E_{y,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta x} H_{z,0} \sin\left(\frac{k_x \Delta x}{2}\right) - \frac{1}{\Delta z} H_{x,0} \sin\left(\frac{k_z \Delta z}{2}\right), \tag{2.59}$$

$$\frac{\varepsilon_0}{\Delta t} E_{z,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta y} H_{x,0} \sin\left(\frac{k_y \Delta y}{2}\right) - \frac{1}{\Delta x} H_{y,0} \sin\left(\frac{k_x \Delta x}{2}\right), \tag{2.60}$$

$$\frac{\mu_0}{\Delta t} H_{x,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta y} E_{z,0} \sin\left(\frac{k_y \Delta y}{2}\right) - \frac{1}{\Delta z} E_{y,0} \sin\left(\frac{k_z \Delta z}{2}\right), \tag{2.61}$$

$$\frac{\mu_0}{\Delta t} H_{y,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta z} E_{x,0} \sin\left(\frac{k_z \Delta z}{2}\right) - \frac{1}{\Delta x} E_{z,0} \sin\left(\frac{k_x \Delta x}{2}\right), \tag{2.62}$$

$$\frac{\mu_0}{\Delta t} H_{z,0} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{1}{\Delta x} E_{y,0} \sin\left(\frac{k_x \Delta x}{2}\right) - \frac{1}{\Delta y} E_{x,0} \sin\left(\frac{k_y \Delta y}{2}\right). \tag{2.63}$$

Eqs. (2.58)–(2.63) constitute a linear system with six equations and six variables that do not depend on the position in space. Non-trivial solutions exist only if these equations are linearly dependent, i.e. the determinant of the matrix that represents the system is zero. By forcing this condition, we obtain the following expression:

$$\begin{aligned}&\left[\frac{1}{c_0 \Delta t} \sin\left(\frac{\omega \Delta t}{2}\right)\right]^2 = \\ &\left[\frac{1}{\Delta x} \sin\left(\frac{k_x \Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta y} \sin\left(\frac{k_y \Delta y}{2}\right)\right]^2 + \left[\frac{1}{\Delta z} \sin\left(\frac{k_z \Delta z}{2}\right)\right]^2.\end{aligned} \tag{2.64}$$

The expression derived in eq. (2.64) is the FDTD dispersion relation in free space. It is clear that there is not a linear dependency between ω and k, and therefore a different phase velocity exists, not only for each different frequency, but also for each propagation direction. Another important point related to eq. (2.64) is the Nyquist-Shannon sampling theorem. Assuming that we want to discretize a function whose maximum frequency is F_s, then the sampling frequency f_s must fulfill $f_s > 2F_s$. This, translated into the temporal and spatial discretization of a wavefront, implies that

$$\frac{\omega \Delta t}{2} < \frac{\pi}{2}, \tag{2.65}$$

$$\frac{k_\alpha \Delta \alpha}{2} < \frac{\pi}{2}, \quad \alpha \in \{x, y, z\}, \tag{2.66}$$

therefore the only relevant range of the arguments of all the sines in eq. (2.64) is $[0, \frac{\pi}{2}]$.

It is interesting to note that a stability condition can be additionally derived from eq. (2.64).

The value of ω can be calculated as a function of the wavenumber vector $\vec{k}$. However, ω is inside the argument of a sine function, and therefore we may wonder what would happen if the configuration is such that the sine is necessarily equal to a number greater than one. It is easy to see that the worst-case scenario would happen when the sines at the right-hand side of eq. (2.64) are all one. In this case, we would obtain

$$\sin\left(\frac{\omega\,\Delta t}{2}\right) = c_0\,\Delta t\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}}. \tag{2.67}$$

If the value of the right-hand side of this expression happened to be greater than one, then ω would necessarily be complex. This is depicted in fig. 2.7 If this was the case, we could write $\omega = \omega_r + \mathbb{i}\,\omega_i$, where ω_r and ω_i are, respectively, the real and imaginary parts, both being real. It is straightforward to prove that

$$\omega_r = \frac{\pi}{\Delta t}, \tag{2.68}$$

$$\omega_i = \pm\frac{2}{\Delta t}\operatorname{arccosh}\left(c_0\,\Delta t\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}}\right). \tag{2.69}$$

If we plug this into, for example, eq. (2.52), we get

$$E_x\big|_{i,j,k}^{n} = E_{x,0}\,e^{\mathbb{i}(\omega_r\,n\,\Delta t - k_x\,i\,\Delta x - k_y\,j\,\Delta y - k_z\,k\,\Delta z)}e^{-\omega_i\,n\,\Delta t}. \tag{2.70}$$

Now, it is trivial to see that a negative value of ω_i necessarily implies an exponential growth, and, as shown in eq. (2.69), the system does admit such a solution. Therefore, to prevent it, it is a necessary condition that the right-hand side of eq. (2.67) is less than one. This is known as the Courant-Friedrichs-Lewy (CFL) condition [22], and is typically written by defining and constraining the CFL number (CFLN):

$$\text{CFLN} := c_0\,\Delta t\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}} \leq 1. \tag{2.71}$$

In order to better visualize the dispersion relation in FDTD, let us consider a cubic grid ($\Delta x = \Delta y = \Delta z = \Delta$) and a wave propagating towards the direction $\hat{h} = (1,1,1)/\sqrt{3}$. This implies that $k_x = k_y = k_z = k/\sqrt{3}$. CFLN is now written as

$$\text{CFLN} = \frac{\sqrt{3}\,c_0\,\Delta t}{\Delta} \leq 1, \tag{2.72}$$

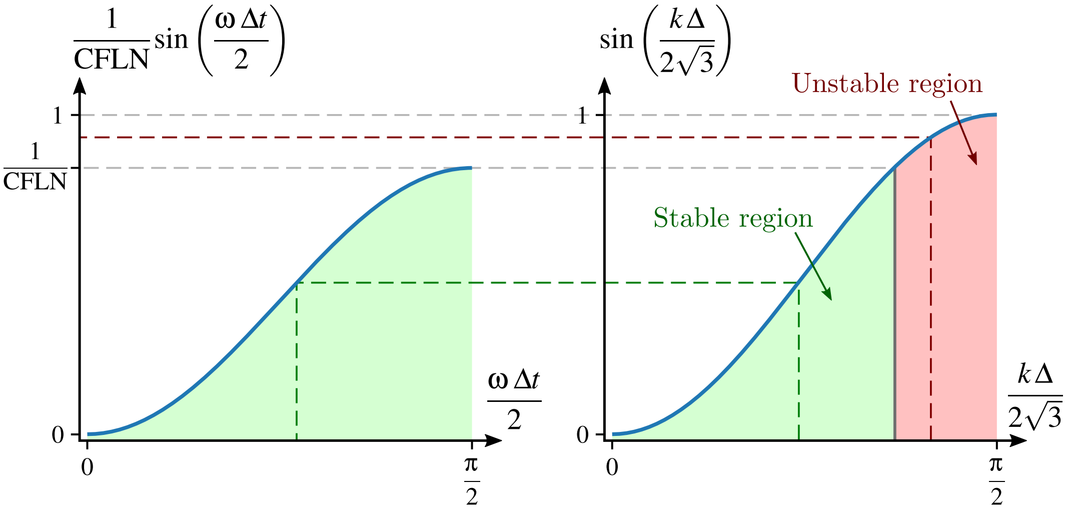

Figure 2.7: Graphic showing the unstable modes when CFLN > 1 for a propagation direction $\hat{k} = (1,1,1)/\sqrt{3}$. For every wavenumber k belonging to the green area in the right graphic, we can find a real value ω that satisfies the dispersion relation. For wavenumbers in the red area, no such solution exists, thus ω is necessarily complex.

and the dispersion relation in eq. (2.64) can be re-written as

$$\sin^2\left(\frac{\omega\Delta t}{2}\right) = \mathrm{CFLN}^2 \sin^2\left(\frac{k\Delta}{2\sqrt{3}}\right). \tag{2.73}$$

As stated previously, if CFLN > 1 then the system admits unstable solutions. This is depicted in fig. 2.4. If we now calculate the phase velocity from eq. (2.73), we obtain:

$$v_{\text{phase}} = \frac{\omega}{k} = c_0 \frac{\sqrt{3}\,\mathrm{PPW}}{\pi\,\mathrm{CFLN}} \arcsin\left[\mathrm{CFLN}\sin\left(\frac{\pi}{\sqrt{3}\,\mathrm{PPW}}\right)\right], \tag{2.74}$$

where PPW $= \lambda/\Delta$ is known as the *Points Per Wavelength*, and indicates the spatial resolution of a given wavelength. In fig. 2.8, the v_{phase} is plotted as a function of PPW for different values of CFLN. It is clear that the phase velocity converges asymptotically to c_0 as PPW grows, independently of CFLN. This, in other words, means that the dispersion relation converges to the analytical expression as we refine the grid. If we had considered a different propagation direction instead, such as $\hat{k} = (1,0,0)$, eq. (2.74) would transform into

$$v_{\text{phase}} = \frac{\omega}{k} = c_0 \frac{\mathrm{PPW}}{\pi\,\mathrm{CFLN}} \arcsin\left[\mathrm{CFLN}\sin\left(\frac{\pi}{\mathrm{PPW}}\right)\right], \tag{2.75}$$

which also converges to the analytical expression when PPW $\to +\infty$.

As a final note, when we consider a non-uniform grid, the procedure presented in this

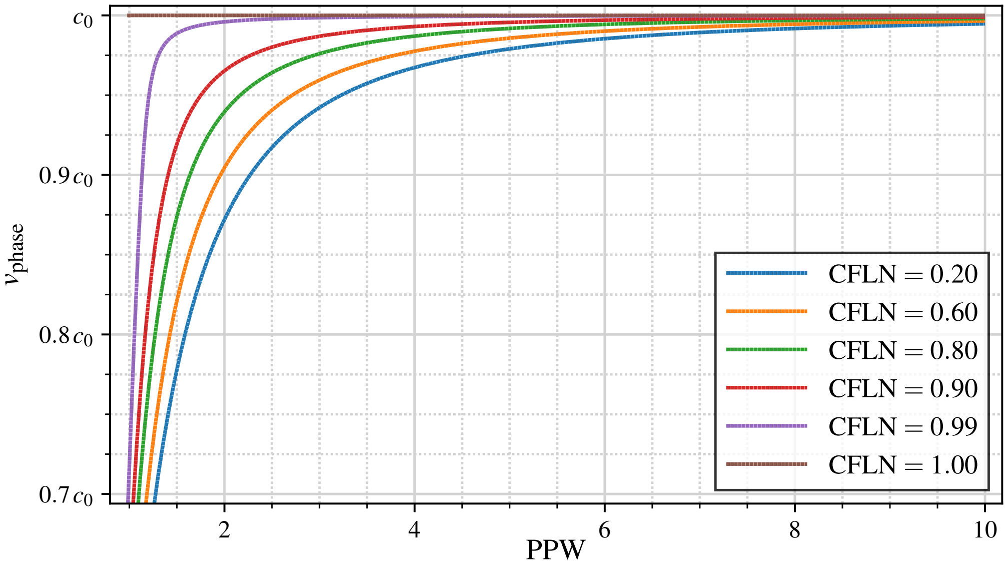

Figure 2.8: Plot of v_{phase} as a function of PPW for different values of CFLN in a cubic grid, assuming a wavefront propagating towards the direction $\hat{k} = (1,1,1)/\sqrt{3}$.

section gets much more complicated. The fields cannot be written simply as a wavefront propagating in one direction but as the composition of multiple incident, reflected, and transmitted waves. A small grasp on this issue is provided in section 3.5.1. In general, a variant of the CFL criterion is used in this case:

$$\mathrm{CFLN} = c_0\,\Delta t\,\sqrt{\max_i\left\{\frac{1}{\Delta x^2_{i+\frac{1}{2}}}\right\} + \max_j\left\{\frac{1}{\Delta y^2_{j+\frac{1}{2}}}\right\} + \max_k\left\{\frac{1}{\Delta z^2_{k+\frac{1}{2}}}\right\}} \leq 1. \tag{2.76}$$

2.1.5 Convergence and Lax Equivalence Theorem

In the context of finite-difference numerical methods, we say that an approximation is *consistent* if the truncation error of the discrete operators approaches zero when the refinement length goes to zero:

$$\lim_{h\to 0}\{A_h\} = A, \tag{2.77}$$

where h is the discretization length that characterizes the refinement, A is an analytical operator and A_h is the discrete approximation of A. Depending on the context, h can refer to the time or space steps, or all of them simultaneously, depending on the dimensions involved in the operator A. For example, for the discrete time derivative (eq. (2.9)) this would be written as

$$\lim_{\Delta t\to 0}\left\{\frac{U|_{I,J,K}^{N+\frac{1}{2}}-U|_{I,J,K}^{N-\frac{1}{2}}}{\Delta t}\right\}=\left.\frac{dU|_{I,J,K}(t)}{dt}\right|_{t=N\Delta t}. \tag{2.78}$$

Similarly, the finite-integration technique applied in Ampère's and Faraday's equations also converges to the rotational operator:

$$\lim_{(\Delta x,\Delta y)\to(0,0)}\left\{\begin{array}{c}\dfrac{U_x|_{I,J+\frac{1}{2},K}^{N}-U_x|_{I,J-\frac{1}{2},K}^{N}}{\Delta y}\\ -\dfrac{U_y|_{I+\frac{1}{2},J,K}^{N}-U_y|_{I-\frac{1}{2},J,K}^{N}}{\Delta x}\end{array}\right\}=\left(\nabla\times U|_K\right)_z|_{(x,y)=(I\Delta x,J\Delta y)}. \tag{2.79}$$

We say that a finite-difference numerical approximation is *convergent* if the solution for any given problem converges to the analytical one when the refinement length approaches zero. In the context of FDTD, let us imagine a certain simulation setup that we let evolve for a given amount of iterations. The numerical solution to the problem is the set of all discrete field components throughout all the iterations: $U|_{I,J,K}^{N}$, where U can be any electromagnetic component. Here, convergence implies that

$$\lim_{(\Delta x,\Delta y,\Delta z,\Delta t)\to(0,0,0,0)}\left\{U|_{I,J,K}^{N}\right\}=U_{\text{analytical}}(I\Delta x,J\Delta y,K\Delta z,N\Delta t) \tag{2.80}$$

for any given I, J, K, and N, where $U_{\text{analytical}}$ is the analytical solution of the problem.

FDTD is consistent by construction, as all discrete differential operators converge to the analytical ones. However, this does not necessarily imply convergence. In this regard, we may refer to Lax Equivalence Theorem [50], which states that a consistent finite-difference system is convergent if and only if it is stable. Stability can often refer to different things, but throughout this work, we refer to bounded-input bounded-output (BIBO) stability. This means that, for any given discretization length, no bounded initial state diverges in time. In mathematical terms,

$$\forall U_0\,/\,\|U_0\|<+\infty,\quad \exists B\geq 0\,/\,\|U|^n\|<B\quad \forall n>0, \tag{2.81}$$

where U_0 is the initial state of the system, $U|^n$ is the state after the n-th iteration, and $\|\cdot\|$ is any norm operator, typically the energy. One could argue that this definition of stability is not necessarily appropriate in all contexts. For example, we may think of a system whose analytical solution grows indefinitely. In these cases, a better stability definition would be:

$$\forall U_0 \,/\, \|U_0\| < +\infty, \quad \exists B \geq 0 \,/\, \|U|^n - U|^n_{\text{analytical}}\| < B \quad \forall n > 0, \tag{2.82}$$

where $U|^n_{\text{analytical}}$ is the analytical solution evaluated at the same time instants as the state components in $U|^n$. The definition presented in eq. (2.82) measures the divergence from the analytical solution. However, none of the cases presented throughout this work admit an ever-growing analytical solution, as they are all physical cases with bounded energy always bounded, and thus the analytical solution cannot diverge. In these cases, we can prove that eq. (2.81) implies eq. (2.82) by the triangle inequality:

$$\begin{aligned} &\|U|^n\| < B_1, \quad \|U|^n_{\text{analytical}}\| < B_2 \quad \forall n > 0 \\ &\Rightarrow \|U|^n - U|^n_{\text{analytical}}\| \leq \|U|^n\| + \|U|^n_{\text{analytical}}\| < B_1 + B_2 \quad \forall n > 0. \end{aligned} \tag{2.83}$$

It is possible to prove that the dispersion relation converges to the analytical expression when the discretization parameters Δx, Δy, Δz, and Δt approach zero. By taking the asymptotic limits:

$$\Delta t \to 0 \Rightarrow \sin\left(\frac{\omega \Delta t}{2}\right) \to \frac{\omega \Delta t}{2}, \tag{2.84}$$

$$\Delta\alpha \to 0 \Rightarrow \sin\left(\frac{k_\alpha \Delta\alpha}{2}\right) \to \frac{k_\alpha \Delta\alpha}{2}, \quad \alpha \in \{x, y, z\}, \tag{2.85}$$

and substituting them into eq. (2.64), we obtain

$$\omega^2 = c_0^2 \left(k_x^2 + k_y^2 + k_z^2\right) \Rightarrow \frac{\omega}{k} = c_0. \tag{2.86}$$

Now, we may wonder how fast the dispersion relation converges to the analytical expression. In order to do so, we must define a discretization length parameter that characterizes the refinement h. For a fixed CFLN value in a cubic grid, Δt is necessarily proportional to the space step Δ, so we may define

$$h = \frac{\Delta}{\lambda} = \frac{1}{\text{PPW}}, \tag{2.87}$$

which indicates how well-refined the grid is compared to the wavelength. Now, let us take as an example the case of a cubic grid with a propagation direction $\hat{k} = (1, 1, 1)/\sqrt{3}$. We may write the relative error of the phase velocity v_{phase} as

$$\left|\frac{v_{\text{phase}} - c_0}{c_0}\right| = 1 - \frac{\sqrt{3}}{\pi\,\text{CFLN}}\frac{1}{h}\arcsin\left[\text{CFLN}\sin\left(\frac{\pi}{\sqrt{3}}h\right)\right]. \tag{2.88}$$

The order of convergence can be obtained from the Taylor expansion series around $h = 0$, which can be written as

$$\left|\frac{v_{\text{phase}} - c_0}{c_0}\right| = \frac{\pi^2\left(1 - \text{CFLN}^2\right)}{18}h^2 + O(h^4). \tag{2.89}$$

In this case, we see that the dispersion relation converges with order $O(h^2)$, which is shown in fig. 2.9 by plotting the error in log-log scale.

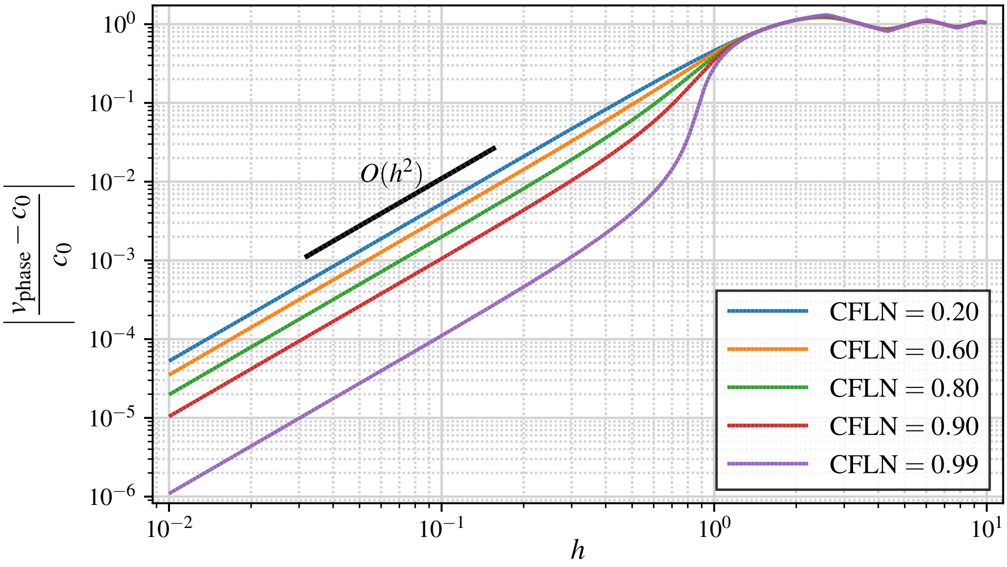

Figure 2.9: Plot of the relative error of v_{phase} as a function of PPW for different values of CFLN in a cubic grid, assuming a wavefront propagating towards the direction $\hat{k} = (1,1,1)/\sqrt{3}$.

2.1.6 Limitations

So far, the FDTD method has been presented along with some of its most relevant characteristics. It is clear that FDTD is a simple and efficient method, whose numerical properties are well-known, and convergences to the analytical solution with order $O(h^2)$. Additionally, the structured nature makes it very simple to implement and parallelize, and thus it is a relevant method almost 60 years after the original formulation [19]. However, a discussion about its limitations is also necessary.

The CFL criterion limits the time step that we can use in a simulation case by the smallest space steps in each direction, as shown in eq. (2.76). When we need to set a high refinement in a specific region of the simulation, this forces the time step in the entire computational domain, and therefore we require more iterations to reach the same physical time compared to a case without such refinement. By looking at eq. (2.64) and fig. 2.7, it is clear that the modes with the shortest wavelengths are responsible for this. In this sense, works involving spatial filtering (SF) have been present in recent years [51–55], allowing in some cases for arbitrarily large CFLN values.

Another major limitation of the FDTD method is its staircase approach. Due to the structured nature of the grid, curvatures that present orientations in non-Cartesian directions cannot be correctly captured within a simulation. Even when a high refinement is used, the lengths and surfaces of object faces never converge to the analytical ones because the staircase enforces Manhattan distance, which in some scenarios may have a relevant impact. This is illustrated in fig. 2.10. A general solution for this is the conformal mesh, which allows surfaces to cross through FDTD cells [24–26].

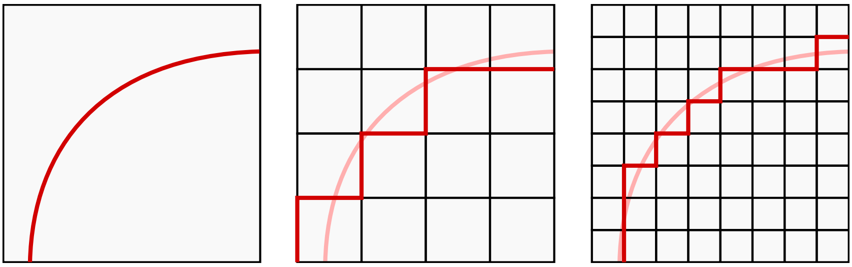

Figure 2.10: Illustration of a curved surface (red line) that cannot be correctly captured by the FDTD grid. Additionally, the length never converges because the structured grid enforces Manhattan distance.

As a last example, multiscale simulations entail another problem in FDTD due to its structured nature. When significantly small geometries need refinement in an FDTD grid, a fine enough space step is required. In some cases, this is not feasible due to either memory or CPU time constraints. We have the possibility of employing non-uniform grids in order to limit the refinement region as much as possible, but this has three major drawbacks. First, a non-uniform space step in one Cartesian axis necessarily extends in the other two. Second, non-uniform grids are not generally desirable because they induce more dispersion, the convergence order decays to $O(h)$ and they present numerical reflections [17]. Last, we encounter yet again the CFL criterion, which forces a short time step limited by the most refined region. Several subcell methods have been designed in the past for specific scenarios, such as thin-wire models [33, 34], thin-pannel models [36–38], and thin-slot models [30, 56]. A general solution for this problem requires delving into subgridding approaches, which allow the subdivision of cells into smaller ones.

2.2 Subcell methods in FDTD

Throughout this work, some numerical simulations are performed involving the OI-SG algorithm in combination with subcell techniques. Particularly, the methods implemented for this work were studied in previous PhD dissertation [57, 58], and therefore a detailed analysis can be found there. Nonetheless, a brief explanation is provided in this section.

2.2.1 Conformal methods

Within the FDTD framework, conformal meshes allow us to define surfaces with non-Cartesian orientations, and thus they can capture object curvatures much better than the usual FDTD grid. They simulate material surfaces traversing across FDTD cells, as depicted in fig. 2.11. The cell surfels affected by the traversing surfaces need to be split into two or more regions, depending on how many surfaces are traversing (the example shown in fig. 2.11 uses only one surface). The discrete magnetic components contained within said surfels need to be split as well, giving one to each region. Similarly, the grid edges affected by the surface and their corresponding discrete electric components need to be split as well. Note that the intersections of the material surfaces with the grid surfels must result in straight lines, but these lines do not need to be co-planar within a cell. These lines create new edges in the original grid surfels, namely *conformal edges*, and thus new discrete electric components appear contained within them. We may consider only one discrete electric component for each conformal edge, however, this technique in general permits considering two co-located electric components, one for each side. An example is shown in fig. 2.11 denoting them as $E_{\text{conf}}|_0$ and $E_{\text{conf}}|_1$.

Once this scheme is set, we need only to adapt the FDTD update equations to solve the field values in each iteration. Split electric fields located at the edges update normally, using the original FDTD update expressions from eqs. (2.14)–(2.19), as if they were located at the center of their original edge. The only relevant consideration is that they must use the split magnetic field corresponding to their region. For example, in fig. 2.11, the discrete component $E_z|_0$ must use $H_x|_0$, and $E_z|_1$ must use $H_x|_1$; similarly, $E_y|_{0,0}$ and $E_y|_{1,0}$ must both

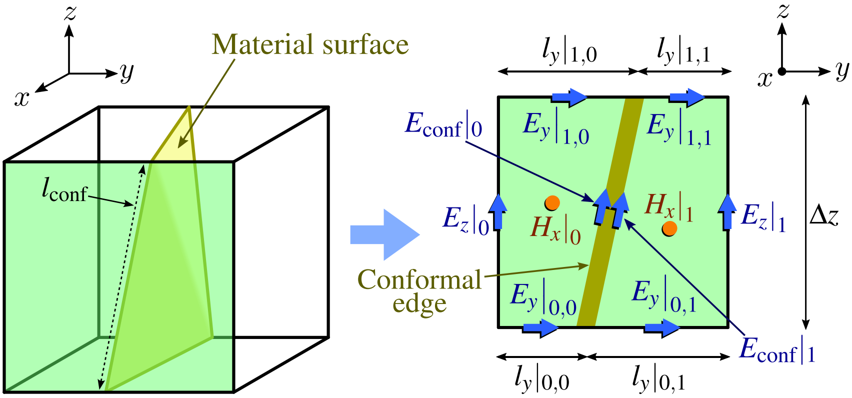

Figure 2.11: Example of the conformal PEC approach. The faces of the affected cells get split into two or more regions, increasing the amount of discrete field components.

use $H_x|_0$, whereas $E_y|_{0,1}$ and $E_y|_{1,1}$ must use $H_x|_1$. For the split magnetic fields, a simple finite integration is applied. For the example in fig. 2.11, we obtain

$$H_x|_0^{n+\frac{1}{2}} = H_x|_0^{n-\frac{1}{2}} + \frac{\Delta t}{\mu\, S_0}\left(\Delta z\, E_z|_0^n - l_y|_{0,0}\, E_y|_{0,0}^n + l_y|_{1,0}\, E_y|_{1,0}^n - l_{\text{conf}}\, E_{\text{conf}}|_0^n\right), \tag{2.90}$$

$$H_x|_1^{n+\frac{1}{2}} = H_x|_1^{n-\frac{1}{2}} + \frac{\Delta t}{\mu S_1}\left(-\Delta z\, E_z|_1^n - l_y|_{0,1}\, E_y|_{0,1}^n + l_y|_{1,1}\, E_y|_{1,1}^n + l_{\text{conf}}\, E_{\text{conf}}|_1^n\right), \tag{2.91}$$

where l_{conf} is the length of the conformal edge, and S_0 and S_1 are the integration surfaces, which can be calculated with the trapeze formula:

$$S_0 = \Delta_z \frac{l_y|_{0,0} + l_y|_{1,0}}{2}, \tag{2.92}$$

$$S_1 = \Delta_z \frac{l_y|_{0,1} + l_y|_{1,1}}{2}. \tag{2.93}$$

As a final note, there is no general rule to calculate the values of $E_{\text{conf}}|_0$ and $E_{\text{conf}}|_1$, as they depend on the material that is being simulated with the conformal mesh. A very common example is a material consisting of a PEC surface, which is simulated by just enforcing $E_{\text{conf}}|_0^n = E_{\text{conf}}|_1^n = 0 \forall n$. In other scenarios, more complex techniques can be applied to simulate different behaviors.

2.2.2 SGBC thin panel

Subgridding boundary conditions (SGBC) thin panel model [37] is a model that hybridizes Crank Nicolson (CN) and FDTD to simulate lossy thin panels. The conformal extension [28] provides a methodology to apply them to conformal meshes. Both schemes are depicted in fig. 2.12. Despite having the word *subgridding* on its name, SGBC is not related to the family of subgridding methods in FDTD as has been defined in this work.

The basic idea behind SGBC in both structured and conformal meshes is nearly identical. The intersection between the thin panel surface and the grid affects certain edges. The discrete electric fields belonging to these edges split into two co-located components. In fig. 2.12, they are named $E_\mathrm{c}|_0$ and $E_\mathrm{c}|_1$. Each one corresponds to one side of the thin panel. The main difference between structured and conformal meshes is that, in a structured mesh, the split discrete electric components belong to a natural FDTD grid edge, whereas in a conformal mesh, they belong to a conformal edge. In both cases, each discrete electric component is a neighbor to a discrete magnetic component, $H_\mathrm{c}|_0$ and $H_\mathrm{c}|_1$, respectively. In a conformal mesh, both discrete magnetic components belong to the same cell face.

Between $E_\mathrm{c}|_0$ and $E_\mathrm{c}|_1$, a fine 1D grid with N layers is placed containing a series of discrete electric and magnetic components. In fig. 2.12, they are named $E_\mathrm{f}|_i$ and $H_\mathrm{f}|_i + \frac{1}{2}$, and the following identities are true:

$$\begin{aligned} E_\mathrm{f}|_1 &\equiv E_\mathrm{c}|_0, && (2.94)\\ E_\mathrm{f}|_{N+1} &\equiv E_\mathrm{c}|_1, && (2.95)\\ H_\mathrm{f}|_{\frac{1}{2}} &\equiv H_\mathrm{c}|_0, && (2.96)\\ H_\mathrm{f}|\,|_{N+\frac{3}{2}} &\equiv H_\mathrm{c}|_1. && (2.97) \end{aligned}$$

Contrary to usual FDTD, the discrete magnetic and electric field components belonging to the fine 1D grid are contemporary, and they are calculated at each iteration using a CN implicit algorithm. Let us write the 1D FDTD equation for any given discrete fine electric component $E_\mathrm{f}|_i$:

$$E_\mathrm{f}|_i^{n+1} = C_{a,i}\, E_\mathrm{f}|_i^n + C_{b,i} \left(H_\mathrm{f}|_{i-\frac{1}{2}}^{n+\frac{1}{2}} - H_\mathrm{f}|_{i+\frac{1}{2}}^{n+\frac{1}{2}} \right), \tag{2.98}$$

where $C_{a,i}$ and $C_{b,i}$ are constants that depend on the media and the spatiotemporal steps, analogously to those found in eqs. (2.14)–(2.19). By performing a time interpolation on the discrete magnetic components, eq. (2.98) transforms into

$$E_\mathrm{f}|_i^{n+1} = C_{a,i}\, E_\mathrm{f}|_i^n + \frac{C_{b,i}}{2} \left(H_\mathrm{f}|_{i-\frac{1}{2}}^{n+1} - H_\mathrm{f}|_{i+\frac{1}{2}}^{n+1} + H_\mathrm{f}|_{i-\frac{1}{2}}^{n} - H_\mathrm{f}|_{i+\frac{1}{2}}^{n} \right). \tag{2.99}$$

Analogously, we may obtain a similar expression for $E_{\text{f}}|_{i+\frac{1}{2}}$:

$$H_{\text{f}}|_{i+\frac{1}{2}}^{n+1} = D_{a,i}\, H_{\text{f}}|_{i+\frac{1}{2}}^{n} + \frac{D_{b,i}}{2}\left(E_{\text{f}}|_{i}^{n+1} - E_{\text{f}}|_{i+1}^{n+1} + E_{\text{f}}|_{i}^{n} - E_{\text{f}}|_{i+1}^{n} \right), \tag{2.100}$$

where $D_{a,i}$ and $D_{b,i}$ are the magnetic analogous of $C_{a,i}$ and $C_{b,i}$. By substituting eq. (2.100) into eq. (2.99), we get an expression with the following shape:

$$a_i\, E_{\text{f}}|_{i}^{n+1} + b_i\, E_{\text{f}}|_{i-1}^{n+1} + c_i\, E_{\text{f}}|_{i+1}^{n+1} = d_i, \tag{2.101}$$

where a_i, b_i and c_i are constants, and d_i depends only on the fields in the previous time instant $t = n\Delta t$.

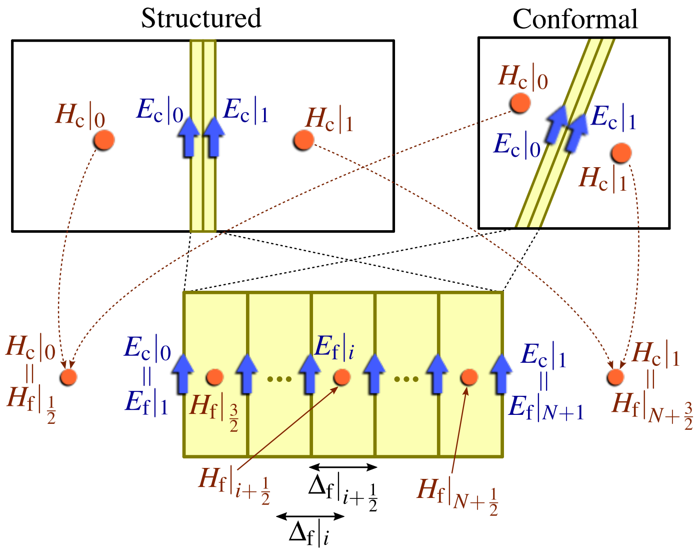
Structured
Conformal
$H_{\text{c}}|_0$
$E_{\text{c}}|_0$
$E_{\text{c}}|_1$
$H_{\text{c}}|_1$
$H_{\text{c}}|_0$
$E_{\text{c}}|_1$
$E_{\text{c}}|_0$
$H_{\text{c}}|_1$
$H_{\text{c}}|_0 \,||\, H_{\text{f}}|_{\frac{1}{2}}$
$E_{\text{c}}|_0 \,||\, E_{\text{f}}|_1$
$H_{\text{f}}|_{\frac{3}{2}}$
$E_{\text{f}}|_i$
$E_{\text{c}}|_1 \,||\, E_{\text{f}}|_{N+1}$
$H_{\text{c}}|_1 \,||\, H_{\text{f}}|_{N+\frac{3}{2}}$
$H_{\text{f}}|_{i+\frac{1}{2}}$
$\Delta_{\text{f}}|_{i+\frac{1}{2}}$
$H_{\text{f}}|_{N+\frac{1}{2}}$
$\Delta_{\text{f}}|_i$

Figure 2.12: Scheme of the SGBC Thin Panel model in both structured and conformal meshes. In the intersection between the thin panel surface and the cell face, two co-located discrete electric components are placed. The inner part of the thin panel contains N 1D FDTD cells whose values are calculated in every iteration via a 1D CN algorithm.

The procedure described above is only valid for $i \neq 1$ and $i \neq N+1$. The electric fields $E_{\text{f}}|_1$ and $E_{\text{f}}|_{N+1}$ depend on $H_{\text{c}}|_0$ and $H_{\text{c}}|_1$, respectively, which belong to the coarse grid, and thus are not contemporary to the discrete electric components. However, they can be easily calculated by performing the previous temporal interpolation only on the fine magnetic components:

$$E_{\mathrm{f}}\big|_{1}^{n+1} = C_{a,1}\, E_{\mathrm{f}}\big|_{1}^{n} + C_{b,1}\left(H_{\mathrm{f}}\big|_{\frac{1}{2}}^{n+\frac{1}{2}} - \frac{H_{\mathrm{f}}\big|_{\frac{3}{2}}^{n+1} + H_{\mathrm{f}}\big|_{\frac{3}{2}}^{n}}{2} \right), \tag{2.102}$$

$$E_{\mathrm{f}}\big|_{N+1}^{n+1} = C_{a,N+1}\, E_{\mathrm{f}}\big|_{N+1}^{n} + C_{b,N+1}\left(\frac{H_{\mathrm{f}}\big|_{N+\frac{1}{2}}^{n+1} + H_{\mathrm{f}}\big|_{N+\frac{1}{2}}^{n}}{2} - H_{\mathrm{f}}\big|_{N+\frac{3}{2}}^{n+\frac{1}{2}} \right). \tag{2.103}$$

Now, by substituting eq. (2.100) into eqs. (2.102) and (2.103), we obtain:

$$a_1\, E_{\mathrm{f}}\big|_{1}^{n+1} + c_1\, E_{\mathrm{f}}\big|_{2}^{n+1} = d_1, \tag{2.104}$$

$$a_{N+1}\, E_{\mathrm{f}}\big|_{N+1}^{n+1} + b_{N+1}\, E_{\mathrm{f}}\big|_{N}^{n+1} = d_i. \tag{2.105}$$

Eq. (2.101), along with eqs. (2.104) and (2.105), form a tridiagonal system that, if solved, allows us to obtain the values of all the discrete fine electric components at each iteration based on the values in the previous ones. Additionally, we can now directly apply eq. (2.100) in order to obtain the discrete fine magnetic values, thus completing the procedure. The values $H_{\mathrm{c}}|_0$ and $H_{\mathrm{c}}|_1$ can be calculated via the original FDTD equations if they are in a structured mesh, using $E_{\mathrm{c}}|_0$ and $E_{\mathrm{c}}|_1$, respectively. If they are affected by a conformal mesh, they can be simply calculated using eqs. (2.90) and (2.91).

2.3 Review of subgridding methods in FDTD

The word *subgridding* is a particularly generic term within the context of CEM methods and FDTD, and can often refer to different things. As an example, the method explained in section 2.2.2, subgridding boundary conditions (SGBC), uses this word. However, throughout this work, subgridding refers to the family of FDTD methods that consist, in essence, of any technique that can locally divide the computational domain into regions with different space steps. In this section, a thorough review of subgridding methods in FDTD, their main characteristics and considerations, as well as the state-of-the-art, are presented.

As a prior note, throughout this work the terms *subgridding method*, *subgridding algorithm*, *subgridding scheme* and *subgridding technique* are used interchangeably: they refer to the aspects that characterize the method in question, which may include the spatial or temporal configuration, the associated equations, or any other considerations related to it.

2.3.1 Basics of subgridding methods

In general, subgridding methods found in the literature can be very different from one another. As a general rule, we may say that all subgridding methods divide the computational domain into, at least, two regions: a coarse one and a fine (or refined) one. Each region must have a different space step or we would not be able to call it subgridding, but they don't need to have different time steps. These regions are often referred to as *refienement levels* or just *levels*. Throughout this work, the region that lies between two different levels is called *subgridding boundary*. In some schemes, two adjacent levels do not have any overlap, and therefore the subgridding boundary is just a surface, however, other subgridding schemes do have some overlap and therefore this boundary has a volume. Another important part of all subgridding schemes is that they must provide some kind of connection algorithm to communicate both regions with one another in the subgridding boundary. This is often referred to as *interpolation/extrapolation rule*. At this point, similitudes between most subgridding schemes end, and each one can be completely different. Nonetheless, some general classification can be made based on some typically present characteristics:

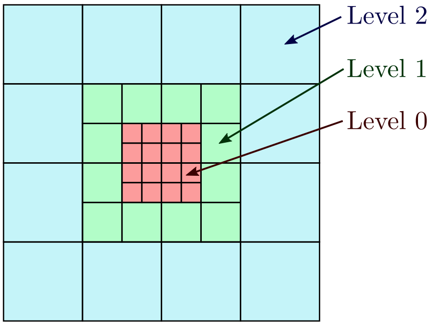

Figure 2.13: Illustration of a recursive 2-dimensional subgridding with a refinement ratio $r = 2$.

- **Dimension**: Most subgridding schemes found in the literature are 3-dimensional or can be expanded to 3D, but in some cases we may find approaches that are only applied to 2-dimensional simulations [59–71].
- **Refinement ratio**: The refinement ratio r of a subgridding scheme is the ratio between coarse and fine cell sizes. Some subgridding schemes are flexible in this regard, allowing for any arbitrary ratio [54, 71–79], and in some occasions only odd ratios [76, 80, 81]. The most common values to find in the literature are $r = 2$ [45, 62, 66, 69, 70, 82–89], $r = 3$ [55, 63, 67, 74, 80, 90–106], and $r = 4$ [59, 64, 72, 107–110] Less commonly, we see simulations with a larger ratio: $r = 5$ [68, 111–115], $r = 6$

[72], $r = 7$ [114, 116, 117], $r = 8$ [64, 118], $r = 9$ [81, 119], $r = 11$ [114, 120], $r = 15$ [114, 121]. In some rare occasions, we may find schemes using fractional values [122–124] or even irrational values [65, 125–127]. In general, we find that odd ratios are more common than even ratios, mostly because odd ratios allow for the co-location of some field components from both fine and coarse grids, thus not requiring any fine-to-coarse interpolation [83, 97, 117, 128, 129].

- **Time refinement**: Most subgridding algorithms can naturally implement global time-stepping (GTS), which simply means using the same time step in every refinement level. Nonetheless, many authors in the literature provide a methodology to use local time-stepping (LTS), i.e. they allow different time steps in the fine and coarse meshes [66, 78, 88, 94, 102, 130]. Other authors, in contrast, do not provide such a scheme [71, 72, 74, 77, 81, 101]. The interest in LTS is mostly due to the CFL criterion (see eq. (2.71)), which roughly states that the maximum stable time step is proportional to the minimum space step, and thus the time step in the whole simulation is constrained by the fine region. However, LTS implies designing some kind of time interpolation scheme, and it is known that time interpolations generally worsen stability or make it more difficult to analyze [95, 106]. For this reason, some authors purposely avoid implementing LTS and they rather apply other techniques such as spatial filtering (SF) to allow the usage of a time step only constrained by the CFL criterion in the coarse mesh [54, 76, 106].

- **Recursivity**: Most subgridding algorithms apply the same update scheme in the inner part of both fine and coarse regions, i.e. the classical FDTD update equations. This means that the refined region, aside from the subgridding boundaries, acts like a completely normal FDTD region, and therefore we may further refine some of their cells. In these cases, we may say that the subgridding scheme is recursive and therefore it allows for multiple nested refinement levels [45, 62, 107, 131–133]. An illustration is presented in fig. 2.13. However, some other methods do not allow such recursion, usually because they apply a different update algorithm within the fine region, such as ADI-FDTD or higher-order methods [64, 66, 74, 76, 77].

- **Explicitness**: The classical FDTD update scheme consists of a linear explicit algorithm, i.e. the values on the next time step are directly calculated from the values in the previous one by a linear operator. This can be represented as

$$U|^{n+1} = AU|^{n}, \tag{2.106}$$

where $U|^n$ is the system state at the discrete time $n\Delta t$ and A is the linear operator known as the update operator. However, some TD methods are implicit, which means that the next values cannot be directly calculated from the previous ones:

$$BU|^{n+1} = CU|^{n}, \tag{2.107}$$

where B and C are linear operators. In these cases, an explicit operator could be obtained from inverting B, but matrices inversions are especially expensive, and therefore these methods resort to other algorithms to solve the system in each time step separately. Noticeably, implicit methods are much less efficient than explicit ones, and this is one of the reasons why FDTD is particularly efficient. In this regard, most subgridding algorithms are explicit since they just consist of an interpolation scheme to calculate the required field components. However, on some occasions, we may find subgridding methods that use implicit methods in the refined region [64, 68, 74, 77, 113, 134–136].

- **Material traverse**: In their most simple form, subgridding methods are formulated in free space due to the simplicity associated with it. In most cases, this is what one can find in the literature, and the presence of materials traversing the subgridding boundary is rare. Nonetheless, some authors have offered subgridding schemes that allow for it [62, 72, 81, 105, 135, 137, 138]. In general, it has been found that material traverse is typically a source of significant errors [100]

- **Stability**: One of the main concerns in subgridding schemes is the stability and, more specifically, the presence of late-time instabilities. This kind of instability consists of modes that grow exponentially but do not become visually appreciable until a large portion of the simulation time has run, typically over 10^5 iterations. Subgridding methods are particularly problematic in this regard because late-time instabilities are one of the most common issues that appear within them [78, 102, 114, 126, 127, 139]. A well-known case is the HSG method [140], which is known to be unstable despite its accuracy, and filters are required to delay the appearance of instabilities [141].

- **Accuracy and spurious reflections**: Usually, the main characteristic that distinguishes subgridding methods from one another is the treatment that they do in the subgridding boundary. However, this is not a trivial task, and in most cases, the algorithm in the subgridding boundary produces numerical reflections, also known as *spurious reflections*, and other numerical artifacts that worsen the accuracy of the method [59, 67, 86, 100, 115]. On occasions, some different methodologies are employed to prevent these reflections, such as higher-order methods [142], domain overriding [143] or methods hybridations [73].

- **Efficiency and parallelization**: FDTD is known for being exceptionally simple to parallelize due to its structured nature, thus making it particularly efficient. However, subgridding methods break this grid structure and therefore may potentially be less efficient than the classical FDTD on a per-processed-field basis. This means that the CPU takes longer to process each discrete field component on average. However, the presence of subgridding on itself implies the existence of less discrete fields to process compared to a simulation with only the fine grid, and even further, if LTS is applied, each discrete coarse field component must be processed fewer times compared to the fine ones. This effect gets even more pronounced when multiple subgridding levels exist within a simulation, and thus subgridding schemes can potentially be much more efficient than a fine-only grid. Actually, this is the main reason why subgridding methods exist in the first place: to obtain accurate results without requiring an expensive refined grid in the whole computational domain. For this reason, efficiency is one key point and a requisite for any subgridding method, and thus efficiency charts are often found in the literature [68, 70, 76, 77, 122, 144–147]. In some cases, we may find authors explaining parallelized implementations [62, 132, 145, 148, 149]

- **Additional characteristics**: Less commonly, we may find other characteristics that define specific subgridding methods. These include domain decomposition [150–152], domain overriding [118, 138, 143, 153], higher-order methods [66, 75], rotated subgrids [65, 126], non-uniform subgrids [127], dynamically adaptative meshes [147], among others.

2.3.2 Studies of stability

Stability is a key point within subgridding methods in FDTD, mainly because typically they worsen the CFL stability condition and, in many occasions, they induce late-time instabilities [78, 102, 114, 126, 127, 139].

Even when a particular scheme is stable in late-time iterations, the CFLN is more constrained than in the classical FDTD. Let us suppose a simulation case with a cubic grid and two regions: fine and coarse. The space steps in the coarse and fine regions are, respectively, Δ_c and Δ_f. Assuming a refinement ratio of r, then we have that $\Delta_c = r\Delta_f$. Similarly, the respective time steps are Δt_c and Δt_f. According to the CFL criterion (eq. (2.71)), the time step is constrained in both regions by the space step. Thus, we may define CFLN$_c$ and CFLN$_f$ as the CFL number in the coarse and fine regions, respectively,

$$\mathrm{CFLN}_{\{c,f\}} = \frac{\sqrt{3}\,c_0\,\Delta t_{\{c,f\}}}{\Delta_{\{c,f\}}}. \tag{2.108}$$

First, let us consider a simulation case with GTS, which means that $\Delta t_c = \Delta t_f \equiv \Delta t$. In this scenario, it is trivial to prove that $\mathrm{CFLN}_c = \mathrm{CFLN}_f/r$, and, since both values of CFLN must be less than 1 to ensure stability, this forces $\mathrm{CFLN}_c \leq 1/r$. This creates a great constraint on the coarse time step. Additionally, if we consider a simulation with $N+1$ subgridding levels, we have that $\mathrm{CFLN}_c \leq 1/r^N$, which makes the problem much worse.

Many subgridding algorithms, to avoid the mentioned problem, aim to provide some kind of time interpolation algorithm to allow a different time step in each level, i.e. LTS. If this is the case, then we can keep the same proportion between the space and time steps in each level, and therefore the value of CFLN remains constant, thus allowing a value of $\mathrm{CFLN} = 1$ in each level. However, it is known that typically subgridding schemes do not allow such value because they are more unstable than the classical FDTD method [46, 112, 118], and this gets worse with time interpolations [106]. Additionally, some subgridding methods have late-time instabilities that cannot be overcome by any value of CFLN. Such is the case of the well-known HSG [140, 141, 154, 155], which has late-time instabilities that are typically delayed by filters [141].

On many occasions, a strict analysis of the stability is very difficult or not feasible, and this forces many authors to prove the stability of their proposed methods by performing simulations for a large number of iterations, typically between 10^5 and 10^7 iterations. Examples can be found in [66, 70, 76, 77, 85, 96, 122, 156]. Nonetheless, in the literature, some interesting works can be found in which the stability is tested more rigorously. A classical example of stability analysis in FDTD is the work of Thoma and Weiland [84], in which instabilities are classified into static (accumulation of electric or magnetic charges) and dynamic (increases in energy derived from the discretization of spatial and time derivatives). These instabilities tend to occur with changes in topology due to the subgridding boundaries. Two interesting works in this regard can be found in [94, 157], where the authors state that reciprocity in the update of the discrete field components enhances stability. In the leapfrog context, reciprocity means that, for any discrete magnetic component required to update a discrete electric component, the latter is also required to update the first. Similarly, in [95] it can be seen that the usage of LTS makes the analysis more difficult and a complete proof of stability cannot be completely provided.

Frequently, we may find authors that provide a stability analysis based on the spectral radius of the evolution operator. If we consider a linear update algorithm and the operator A that represents one iteration, the spectral radius $\rho(A)$ is the maximum magnitude of its eigenvalues. When A is diagonalizable it is easy to prove that the algorithm is stable if and only if $\rho(A) \leq 1$ [158]. In this sense, stable means non-divergent. An equivalent analysis can be made of the discrete curl-curl operator M in non-orthogonal FDTD schemes, which includes some subgridding algorithms. The operator M must be positive definite and have distinct real eigenvalues [159, 160]. An interesting work in this regard is the one performed by Yan and Yiao [160–162], in which they derive symmetric positive semi-definite matrices by filtering out unstable modes in a subgridding scheme, achieving unconditional stability. In [163] it was extended to unsymmetric matrices. In [130], Zeng and Jiao develop a similar methodology with matrices involving LTS. Other spectral analyses can be found in [47, 68, 133].

In some cases, we may find authors that hybridize unconditionally stable CEM methods with FDTD to derive stable subgridding schemes. A notable case is the ADI-FDTD, which is frequently found in subgridding schemes [64, 68, 74, 77, 113, 134–136]. Another example is the Crank Nicolson (CN) method [164].

Some authors use filters to erase high-frequencies unstable modes. This is already justified in section 2.1.6 by looking at the dispersion relation of FDTD (eq. (2.64)). Additionally, some authors justify this by a spectral analysis [162, 163] or by looking at the dispersion relation of FDTD (eq. (2.64)). The greater the wavenumbers k are, the more likely ω requires to be complex. These spatial modes correspond to the lowest possible wavelength that the grid can solve while still meeting Nyquist's sampling theorem, and thus it corresponds to the largest frequencies. The work presented by Gaffar and Jiao [52, 53] indeed shows that the highest frequencies are the source of instabilities in FDTD and thus they can be filtered out to enhance stability, even for values CFLN > 1. It has also been shown that this same rule can be applied in many cases with subgridding. An interesting example is the work presented by Wei [55], Xu and Xie [106], and Xie et al. [75], where spatial filtering (SF) is used to filter out the unstable modes and thus allow the usage of a time step constrained only by the CFL criterion in the coarse grid.

Another interesting work related to stability is performed by Bekmambetova et al., first in [72, 80] for a 2D subgridding, and later generalized to 3D in [81]. In their work, a methodology based on the theory of dissipative systems is derived to separate the analysis of the stability of the different components in an FDTD simulation, thus providing stability conditions for the subgridding scheme.

2.3.3 Studies of the accuracy

A large variety of connection algorithms are developed to communicate fine and coarse regions in the subgridding boundaries. One way or another, each one of them modifies the classical FDTD algorithm, and in the process, they typically worsen the accuracy if we compare it to a simulation containing a fine-only grid, especially since they usually create spurious reflections [59, 86, 165]. Nonetheless, it is expected that subgridding algorithms provide better accuracy than a coarse-only simulation, otherwise they would not be useful. For this reason, it is exceptionally common to find numerical validations of subgridding algorithms when they are presented in the literature. These validations typically consist of evaluating the numerical error in a few concrete examples, and a more rigorous and systematic approach is less frequent. Nonetheless, there are some good examples of accuracy and error analysis in the literature.

An interesting example of error analysis was presented by Celuch-Marcysiak and Rudniki in [86]. In their work, they present a systematic approach to predict numerical reflections both in non-uniform grids and subgridding boundaries. By assuming incident, transmitted, and reflected plane waves with arbitrary wave numbers, frequencies, and incidence angles, and plugging the solution into the FDTD equations, they derive the expected reflection coefficient.

In [67], Le et al. analyze the numerical errors obtained in 1D and 2D subgridding schemes with a refinement ratio of 3. The authors suggest that there is a direct correlation between the subgrid electrical size and the cumulative errors caused by propagated waves, implying that the highest errors are produced by short wavelengths, or high frequencies, compared to the grid resolution. Related to this, in [142], authors utilize higher-order methods to improve accuracy even with low-resolution grids.

Other works worth mentioning include the one performed by Okoniewski et al. [165]. They analyze different subgridding interpolation schemes, achieving spurious reflections of

$-60\,\mathrm{dB}$. In [115], the authors analyze the inaccuracy problems that arise in the one-sheet HSG in contrast to the two-sheet formulation.

Material traverse is another important source of inaccuracy, as it has been found in the literature [100]. In [118], the authors conclude that at least 4 cells should be left between the subgridding boundary and the material object.

2.3.4 Studies of efficiency and parallelization

Typically, all subgridding algorithms enhance the original FDTD efficiency. This is because, compared to a fine-only simulation, many cells are removed and replaced with one coarse cell. This already provides a decrease of cells of order $O(r^3)$ for each subgridding level, where r is the refinement ratio. However, when LTS is applied, fields belonging to the coarse region are processed fewer times, providing a decrease of processed fields of $O(r^4)$ to get to the same simulated time. This is shown in efficiency tables in many works in the literature [68, 70, 76, 77, 122, 144–147].

The efficiency of subgridding methods can be enhanced even further by designing a parallelization scheme. Typically, subgridding publications lack information about the possibility of parallelization, but some works can be found in the literature in this regard. Some examples are [62, 132, 148]. An interesting case can be found in [149], in which a multi-GPU cluster is used to parallelize an FDTD simulation implementing an explicit subgridding scheme. Another notable example is the recent work from Feng et al. [145], where a hybrid CPU-GPU system is used to process a subgridding simulation with two levels, in which CPU processes the coarse one and the GPU processes the fine ones.

Another interesting work related to efficiency was presented by Hartley et al. in [166]. In their work, the authors present the switched Huygens subgridding (SHSG) method, which removes the requirement of PML boundary conditions in the subgridding boundaries and therefore enhances the efficiency.

2.3.5 Modern subgridding schemes

Most classic subgridding methods found in the literature implement some interpolation rule in the subgridding boundary to communicate fine and coarse regions, while they utilize the original FDTD update algorithm in the inner part of the region. However, modern subgridding schemes typically apply more complex approaches. Within this section, some of the most relevant works in this regard are presented.

An interesting example of a modern subgridding approach is the separation of the spatial and temporal subgridding boundaries in simulations with LTS. This approach was presented by Xiao et al. in [95, 96]. When coming from the outer coarse grid, the domain first encounters a region where the time refinement is performed without any spatial refinement. Inside this region, a new boundary is found in which the opposite happens: a spatial, but not temporal, interpolation is performed. In [139], the authors suggest that long-term stability can be achieved by separating the temporal and spatial boundaries by at least 3 cells. More recent works based on this idea were presented in [100, 105, 167], including material traverse and achieving long-term stability.

Notable work was performed by Railton in [65, 125], in which, by applying the principles of the HSG method, he obtained a formulation for rotated fine regions within the computational domain in 2D FDTD. Later, in [126] his work was expanded to 3D simulations, and in [127] he implemented non-uniform fine regions.

An interesting approach is the domain decomposition. This scheme consists of running two separate simulations sequentially, one for the coarse grid and another one for the fine grid. The first simulation, which contains the sources, is run until a desired time step, and the values of the fields in the positions corresponding to the subgridding boundaries are stored for every discrete time instant. Then, the second simulation, which has a different cell size, is run, and the previously stored fields are injected as currents using an interpolation rule. This was originally the case presented by Kunz et al. in [150], where a source running in the coarse simulation was injected into a fine one. Later, a similar idea was reproduced by Pascaud et al. in [151, 152], but switching the order of the simulations. In this case, the authors simulated antennas in the fine grid and stored the field values at the boundaries to be reused in any potential coarse simulation.

An adaptative mesh refinement (AMR) scheme was presented by Li et al. in [147]. The authors begin by meshing the computational domain, placing the fine region around relevant parts of the simulation such as the excitation source. After letting the system evolve for a fixed amount of time, the implementation automatically re-meshes the entire computational domain, placing the fine regions in the most energetically dense areas. This process is repeated periodically, thus allowing the simulation to be always refined in the energetically relevant parts.

Higher-order methods are often used to enhance accuracy. This is the case found in [66], where authors run a higher-order FDTD in the coarse region and the conventional FDTD (2,2) in the fine one, adding an interlayer region in the coarse region in order to transition from one another. A recent similar example can be found in [75], where authors hybridize an spatially-filtered symplectic finite-difference time-domain (SF-SFDTD) in the fine grid and FDTD (2,2) in the coarse one.

Hybrid subgridding methods are also commonly found in the literature. These approaches mix the conventional FDTD with other CEM methods. ADI-FDTD is one of the most common methods to find in the literature due to its unconditional stability [68, 74, 77, 106, 113, 134–136, 145]. Similarly, Londersele et al. [164] propose a hybrid method with CN since it is also unconditionally stable. Wang et al. [70, 71, 79, 146] proposed a hybrid and stable method based on summation-by-parts simultaneous approximation term (SBP-SAT).

As a last example, we may find the work presented by Donderici and Teixeira for 2D simulations [143, 153], and later extended to 3D [138]. In their work, the authors implement the concept of domain overriding. This consists of a multi-layer overlap between the fine and coarse regions that extends further from the subgridding boundary. Within those layers, a process of filtering and phase compensation is applied to enhance the accuracy of the method. Additionally, a method of suppression of high frequencies is designed to prevent unstable modes from propagating from the fine to the coarse grid, which is achieved by separating the modes and propagating the high-frequency ones into a PML in one overlapped layer.

CHAPTER 3

The OI-SG method

The OI-SG method is the main subject of study of this dissertation, and thus this chapter constitutes the technical core of this work. Throughout this chapter, the reader can find:

- A fully detailed derivation of the method.
- A discussion on the time refinement: usage of LTS in contrast to GTS.
- The integration of subgridding boundaries with FDTD boundary conditions.
- Different methodologies to obtain the stability conditions of the method, including a long-term stability discussion via a spectral approach.
- A discussion of the dispersion relation and convergence to the analytical expression.
- Two method modifications aiming to enhance the limitations of the method.
- Some relevant details about the implementation and parallelization.

3.1 Method description

The OI-SG utilizes a generalization of the main-dual grid formalism of the FDTD to derive the update equations in the subgridding boundaries. In what follows, a detailed explanation of the procedure is provided: first, the spatial configuration of the subgridding boundary is explained; second, the general update equations are derived; and finally, these equations are applied to the fields present at the subgridding boundaries, thus obtaining the OI-SG method. Throughout this section, we assume usage of GTS.

3.1.1 Spatial refinement

Let us suppose two FDTD cells, as depicted in fig. 3.1 (a) and (c). The picture shows the main grid, which contains the discrete electric components, in blue; whereas the dual grid, containing the discrete magnetic components, is represented in red. As explained in section 2.1, the vertices of the dual grid are located at the geometrical center of the main grid cells, and vice-versa. Now, let us refine one of the main grid cells into two or more subcells in each Cartesian axis. Here, we define the refinement ratios r_x, r_y, and r_z as the number of subcells resulting in each respective direction. In general, the OI-SG method accepts arbitrary ratios for each direction, but throughout this explanation, we will assume $r_x = r_y = r_z = 2$ for simplicity. As a result, the original cell gets split into 8 subcells. We name them *fine cell*, whereas the ones of the original size are referred to as *coarse cells*. The surface that lies between the coarse cell and the set of fine cells is called *subgridding boundary*.

The resulting stencil does not preserve the original lattice structure, but it provides a natural placement for the vertices and edges of the new main grid. The dual grid configuration in the inner part of both coarse and fine regions is trivial, since we may just place the vertices at the geometrical centers of the main grid cells. However, the cells adjacent to the subgridding boundary, both fine and coarse, contain dangling vertices of the dual grid. This is solved by placing dual grid edges connecting said dangling vertices, thus fully connecting the fine and coarse grids. The resulting configuration corresponds to the red grid in fig. 3.1 (b) and (d).

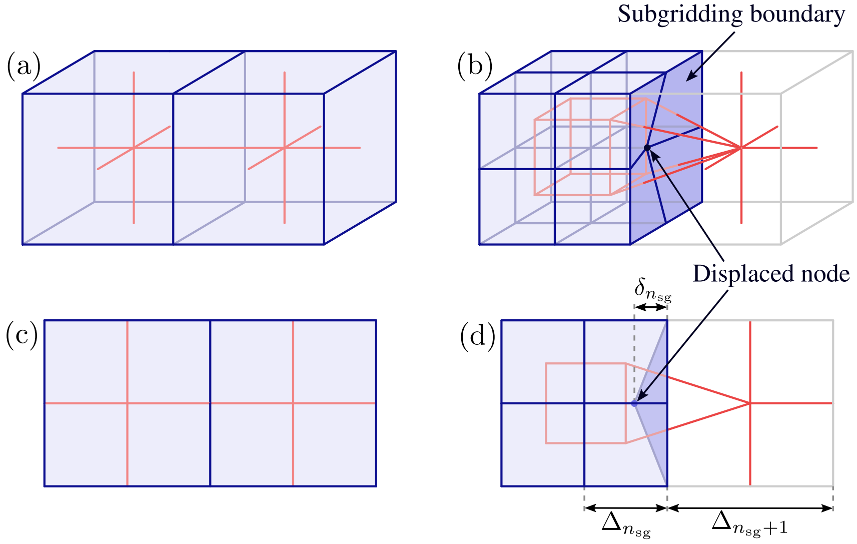
Subgridding boundary
(a)
(b)
Displaced node
$\delta_{n_{\mathrm{sg}}}$
(c)
(d)
$\Delta_{n_{\mathrm{sg}}}$
$\Delta_{n_{\mathrm{sg}}+1}$

Figure 3.1: Representation of the spatial subgridding scheme of the OI-SG. Represented are: (a) two usual FDTD cells, (b) the result after one of them is refined with a refinement ratio $r = 2$, and (c) and (d) the respective 2D projections. In all cases, the dual (magnetic) grid is represented in red. In (a) and (c), the main (electric) grid is represented in blue. In (b) and (d) the coarse magnetic grid is represented in gray and the fine one in blue.

The refinement can be applied to any set of coarse cells, thus creating a region of refined cells with one or multiple arbitrarily shaped subgridding boundaries. In this case, the main-dual grid considerations explained above only apply to the cells adjacent to a subgridding boundary. Additionally, the refinement procedure can be recursively applied to fine cells as well, thus creating an even finer region. This leads to the creation of multiple levels, each denoted by an integer number n_{sg}. The convention established for this work is as follows: the finest region has a level of $n_{\mathrm{sg}} = 0$, and each coarser region increases n_{sg} by one, until reaching the coarsest region, which has a level of $n_{\mathrm{sg}} = N_{\mathrm{sg}}$. Thus, $n_{\mathrm{sg}} \in \{0, 1, \ldots, N_{\mathrm{sg}}\}$. The OI-SG algorithm can be applied to arbitrary non-uniform grids, but for simplicity, throughout this work, we only consider cubic grids. Thus, the cell length of a given level n_{sg} is denoted $\Delta_{n_{\mathrm{sg}}}$, where

$$\Delta_{n_{\mathrm{sg}}} = 2\,\Delta_{n_{\mathrm{sg}}-1} = 2^{n_{\mathrm{sg}}}\,\Delta_0. \tag{3.1}$$

The time step of a given level n_{sg} is denoted $\Delta t_{n_{\mathrm{sg}}}$. Time steps of different levels can be equal or different by choice, further discussion of this is presented in section 3.2. For this section, we assume GTS, i.e. $\Delta t_{n_{\mathrm{sg}}} \equiv \Delta t$ across all the levels.

To end the spatial discussion of the OI-SG refinement, let us draw attention to the subgridding boundary that lies between two contiguous subgridding levels n_{sg} and $n_{\mathrm{sg}} + 1$. For each coarse cell adjacent to a subgridding boundary, we have four corresponding fine cells in the same situation. Between the coarse and the fine cells, a square surface lies, which is composed of four fine surfels. The method proposes a displacement of the central vertex towards the direction normal to the subgridding boundary [45] (depicted in fig. 3.1). This displacement is called the *orthogonalization displacement* and is denoted $\delta_{n_{\mathrm{sg}}}$. This magnitude is defined to be proportional to the cell length of its corresponding level, thus $\delta_{n_{\mathrm{sg}}} = \delta_{\mathrm{r}}\,\Delta_{n_{\mathrm{sg}}}$, where δ_{r} is an adimensional number named the *orthogonalization parameter*, and is a constant for all levels. The value of δ_{r} is free of choice, but it has an impact on stability, as is later discussed in section 3.4.

It is worth noting some considerations required if a refinement ratio $r \neq 2$ is being used. Let us suppose a coarse cell adjacent to a subgridding boundary where refinement ratios of r_1 and r_2 are being used in the parallel Cartesian directions. The dual grid dangling vertices are connected in the same manner, but now, we need to draw $r_1\,r_2$ dual grid edges instead of just 4. The second difference is that the orthogonalization displacement needs to be applied to $(r_1 - 1)(r_2 - 1)$ vertices instead of just one. Thus, depending on the configuration, we may want to apply a different displacement for each of them, therefore leading to more than one orthogonalization parameter δ_{r}.

3.1.2 Update equations

To derive the update equations of the OI-SG, first, we need to provide a discussion of an extended form of FDTD. In what follows, a different notation than the one used in section 2.1 is going to be employed to refer to the discrete electromagnetic components. The discrete electric and magnetic components present in a given simulation are denoted, respectively, $E|_i$ and $H|_i$. The amount of discrete electric and magnetic components are denoted, respectively, N_e and N_h. Thus, the sets containing all the discrete electric and magnetic components are respectively written as

$$\left\{E|_i\right\}_{i=1}^{N_e}, \quad \left\{H|_i\right\}_{i=1}^{N_h}. \tag{3.2}$$

These sets are not ordered in any particular way. When referring to a given field $U|_i$ evaluated at a discrete time instant $t = N\Delta t$, it is denoted as $U|_i^N$, where N can be integer or half-integer.

Let us assume a discrete magnetic component $H|_i$ contained within an integration surfel, similar to the one presented in fig. 2.2, but bounded by an arbitrary number of edges. We denote this surfel $S_h|_i$. This is depicted in fig. 3.2. Note that the edges are not necessarily coplanar, and thus the surfel is not well-defined. However, we may define any arbitrary surface as long as it is bounded by the mentioned edges and it will not affect the result, as is later shown. At the center of each edge, a discrete electric component is placed directed towards the direction of said edge. The discrete magnetic component is placed somewhere in the middle of the surfel and has an arbitrary orientation, thus we may write the corresponding discrete magnetic vector as $\vec{H}|_i = H|_i \hat{H}|_i$, where $\hat{H}|_i$ is a vector of unit-norm that marks the orientation of $\vec{H}|_i$.

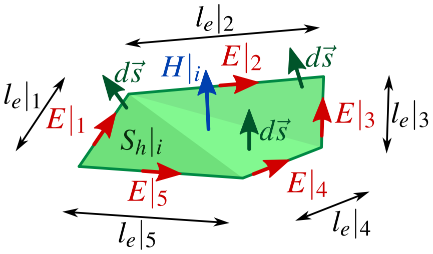

Figure 3.2: Arbitrary surfel used as integration surface to derive the OI-SG update equation for a discrete field component $H|_i$.

Now, we may approximate Faraday's Law (eq. (2.8)) on the surface $S_h|_i$. First, we need to evaluate the surface integrals. This is done by assuming that all physical magnitudes are constant throughout the integration surface: the magnetic field vector $\vec{H}|_i$, the magnetic current density $\vec{J_m}|_i$, the magnetic permeability $\mu|_i$, and the magnetic conductivity $\sigma_m|_i$. By doing this, the right-hand side of eq. (2.8) transforms into

$$-\mu|_i \frac{dH|_i}{dt}\,\hat{H}\Big|_i \cdot \left(\int_{S_h|_i} d\vec{s}\right) - \sigma_m\big|_i H|_i\,\hat{H}\Big|_i \cdot \left(\int_{S_h|_i} d\vec{s}\right) - J_m|_i\,\hat{J_m}\Big|_i \cdot \left(\int_{S_h|_i} d\vec{s}\right). \tag{3.3}$$

It is worth noting that the discrete magnetic orientations in the grid never change, as they are defined by the grid itself. Thus, we define the *equivalent surface* $\tilde{S_h}|_i$ as

$$\tilde{S_h}\Big|_i := \hat{H}\Big|_i \cdot \left(\int_{S_h|_i} d\vec{s}\right), \tag{3.4}$$

which is a constant associated with the discrete magnetic component $H|_i$. Similarly, we define the *equivalent current density* $\tilde{J_m}|_i$ as

$$\tilde{J_m}\Big|_i := J_m|_i \frac{\hat{J_m}\Big|_i \cdot \left(\int_{S_h|_i} d\vec{s}\right)}{\tilde{S_h}\Big|_i}, \tag{3.5}$$

where $\hat{J_m}|_i$ is a unit vector marking the orientation of $\vec{J_m}|_i$. Note that, unlike $\tilde{S_h}|_i$, $\tilde{J_m}|_i$ does have a dependency on time because both the current density magnitude $J_m|_i$ and the orientation $\hat{J_m}|_i$ may change. Now, we may substitute eqs. (3.4) and (3.5) into eq. (3.3) to get

$$-\left(\mu|_i \frac{dH|_i}{dt} + \sigma_m\big|_i H\big|_i + \tilde{J_m}|_i\right) \tilde{S_h}\Big|_i. \tag{3.6}$$

At this point, we must note that neither $\tilde{S_h}|_i$ nor $\tilde{J_m}|_i$ depend on the surface chosen for the surfel $S_h|_i$. In both cases, we have an expression with the following form:

$$\int_S \hat{u} \cdot d\vec{s}, \tag{3.7}$$

where $\hat{u}$ is a unit vector constant throughout the surface S. Now, let us write down Stoke's theorem,

$$\int_S \left(\vec{\nabla} \times \vec{F}\right) \cdot d\vec{s} = \int_{\partial S} \vec{F} \cdot d\vec{l}. \tag{3.8}$$

In order to apply Stoke's theorem to eq. (3.7), we just need to find a vector $\vec{F}$ such that $\hat{u} = \vec{\nabla} \times \vec{F}$. If we write $\hat{u} = (u_x, u_y, u_z)$, it is easy to prove that $\vec{F} = (u_y z, u_z x, u_x y)$ meets this condition. Therefore,

$$\int_S \hat{u} \cdot d\vec{s} = \int_{\partial S} (u_y z, u_z x, u_x y) \cdot d\vec{l}, \tag{3.9}$$

which only depends on the boundary ∂S, and not the chosen surface. In fact, it is not difficult to see that eq. (3.7) results in the orthogonally projected area of S onto a plane orthogonal to $\hat{u}$.

The circulation integral in the left-hand side of eq. (2.8) can be easily approximated by considering each discrete electric component constant through the edge:

$$\oint_{\partial S_h|_i} \vec{E} \cdot d\vec{l} \simeq \sum_{j \in \mathcal{N}_h|_i} \mathrm{sgn}_h|_{i,j}\, l_e|_j\, E|_j, \tag{3.10}$$

where $\mathcal{N}_h|_i$ is the *neighbors set* of $H|_i$, which contains all the discrete electric field indices that surround the particular component $H|_i$, $l_e|_j$ is the edge length of the discrete electric component $E|_j$, and $\mathrm{sgn}_h|_{i,j}$ is 1 if the electric field $E|_j$ has the same orientation as the integration direction and -1 otherwise. For the example depicted in fig. 3.2 we would obtain

$$\oint_{S_h|_i} \vec{E} \cdot d\vec{l} \simeq -\, l_e|_1\, E|_1 - l_e|_2\, E|_2 + l_e|_3\, E|_3 + l_e|_4\, E|_4 + l_e|_5\, E|_5. \tag{3.11}$$

Now, we must do a very similar procedure as the one performed in section 2.1 to derive the update equations. First, we substitute eq. (3.6) and eq. (3.10) into Faraday's Law (eq. (2.8)). Then, we evaluate the expression in $t = n\Delta t$ and apply the finite-difference technique in the time derivative (eq. (2.9)). Finally, we perform a temporal interpolation in the remaining discrete magnetic field analogous to the one performed in eq. (2.11). After doing all this, and by some algebra, we finally obtain

$$\begin{aligned} H|_i^{n+\frac{1}{2}} &= \frac{1 - \dfrac{\sigma_m|_i \Delta t}{2\,\mu|_i}}{1 + \dfrac{\sigma_m|_i \Delta t}{2\,\mu|_i}} H|_i^{n-\frac{1}{2}} \\ &+ \frac{\Delta t}{\mu|_i + \dfrac{\sigma_m|_i \Delta t}{2}} \left(-\frac{1}{\tilde{S}_h|_i} \sum_{j \in \mathcal{N}_h|_i} \mathrm{sgn}_h|_{i,j}\; l_e|_j\; E|_j^n + \tilde{J}_m|_i^n \right). \end{aligned} \tag{3.12}$$

By an analogous procedure, using an integration surfel $S_e|_i$ corresponding to an electric field $E|_i$ on Ampère's Law (eq. (2.13)), we obtain

$$E|_i^{n+1} = \frac{1-\dfrac{\sigma|_i \Delta t}{2\,\varepsilon|_i}}{1+\dfrac{\sigma|_i \Delta t}{2\,\varepsilon|_i}} E|_i^n + \frac{\Delta t}{\varepsilon|_i + \dfrac{\sigma|_i \Delta t}{2}} \left(\frac{1}{\tilde{S}_e|_i} \sum_{j \in \mathcal{N}_e|_i} \mathrm{sgn}_e|_{i,j}\ l_h|_j\ H|_j^{n+\frac{1}{2}} + \tilde{J}|_i^{n+\frac{1}{2}} \right), \tag{3.13}$$

where

- $\sigma|_i$ is the electric conductivity in the integration surfel,
- $\varepsilon|_i$ is the electric permittivity in the integration surfel,
- $\mathrm{sgn}_e|_{i,j}$ is analogous to $\mathrm{sgn}_h|_{i,j}$ but applied to the i-th electric surfel and the j-th discrete magnetic component,
- $l_h|_j$ is the dual grid edge length corresponding to the j-th discrete magnetic component,
- $\tilde{S}_e|_i$ is the equivalent surface of the surfel, defined as

$$\tilde{S}_e\big|_i := \hat{E}\big|_i \cdot \left(\int_{S_h|_i} d\vec{s} \right), \tag{3.14}$$

- $\tilde{J}|_i^{n+\frac{1}{2}}$ is the equivalent electric current density evaluated at the time instant $t = (n+\frac{1}{2})\Delta t$, where

$$\tilde{J}\big|_i := J|_i \frac{\hat{J}\big|_i \cdot \left(\int_{S_e|_i} d\vec{s} \right)}{\tilde{S}_e\big|_i}, \tag{3.15}$$

- $\hat{E}|_i$ is a unit vector marking the orientation of the i-th electric component, and
- $\hat{J}|_i$ is a unit vector marking the orientation of the i-th electric current density.

Eqs. (3.12) and (3.13) constitute an extended form of the FDTD update equations for arbitrarily shaped main-dual grids. It is easy to prove that they reduce to the usual FDTD equations when all the integration surfaces are rectangles and the surface-integrated field components are orthogonal to them. Nonetheless, eqs. (3.12) and (3.13) in this form are required to solve the discrete components located at the subgridding boundaries. For this work, all the subgridding boundaries are contained within free space without sources, and thus simplifications can be made by assuming $\varepsilon|_i = \varepsilon_0$, $\mu|_i = \mu_0$, $\sigma|_i = 0$, $\sigma_m|_i = 0$, $\tilde{J}|_i = 0$, $\tilde{J}_m|_i = 0 \forall i$:

$$H|_i^{n+\frac{1}{2}} = H|_i^{n-\frac{1}{2}} - \frac{\Delta t}{\mu_0 \left.\tilde{S}_h\right|_i} \sum_{j\in\mathcal{N}_h|_i} \mathrm{sgn}_h|_{i,j} \left.l_e\right|_j E|_j^n, \tag{3.16}$$

$$E|_i^{n+1} = E|_i^n + \frac{\Delta t}{\varepsilon_0 \left.\tilde{S}_e\right|_i} \sum_{j\in\mathcal{N}_e|_i} \mathrm{sgn}_e|_{i,j} \left.l_h\right|_j H|_j^{n+\frac{1}{2}}. \tag{3.17}$$

3.1.3 Subgridding boundary fields update

In order to apply eqs. (3.16) and (3.17) to the subgridding boundary, we need to define a main-dual grid configuration in which every main grid edge is surrounded by a number of dual grid edges, and vice-versa. This naturally creates integration surfels. Fortunately, the configuration provided in section 3.1.1 meets this condition exactly. The inner region of each subgridding level employs the original FDTD main-dual grid, thus they are trivially updated by the usual FDTD equations. Additionally, the configuration depicted in fig. 3.1 provides natural integration surfels for each discrete field component contained within the subgridding boundary or adjacent to it, although they require being updated by eqs. (3.16) and (3.17) and thus require special attention.

Let us take a closer look at the subgridding boundary between two given levels. We denote Δ_c to the space step of the coarse level and Δ_f to the space step of the fine level. Now, let us consider the fine discrete electromagnetic field components placed within this subgridding boundary or adjacent to it. Some of them can be trivially updated with the usual FDTD equations, as long as their equivalent surface fulfills $\tilde{S} = \Delta_f$ and it is surrounded by four other discrete components whose edge lengths are $l = \Delta_\mathrm{f}$. An analogous argument can be made for coarse discrete components with an equivalent surface of $\tilde{S} = \Delta_c$ and surrounded by four components with $l = \Delta_\mathrm{c}$. The rest of the discrete field components, which cannot be trivially updated, are denoted as *non-trivial field components*, and can be classified based on their values of $\tilde{S}$ and l. This classification, for a cubic grid with a refinement ratio of $r = 2$, provides 6 distinct electric field types, and 4 distinct magnetic field types. It is depicted in fig. 3.3, and their corresponding equivalent surfaces and edge lengths are written in table 3.1.

Table 3.1: OI-SG field classification for a cubic grid with a refinement ratio $r = 2$. For each field type, the equivalent surface $\tilde{S}$ and the edge length l are written. In each expression, Δ_f refers to the cell length of the fine level in the boundary considered. Field types correspond to fig. 3.3.

Electric		
Type	$\frac{\tilde{S}_e}{\Delta_\mathrm{f}^2}$	$\frac{l_e}{\Delta_\mathrm{f}}$
E-1	$\frac{3+\delta_\mathrm{r}}{4\sqrt{1+{\delta_\mathrm{r}}^2}}$	$\sqrt{1+{\delta_\mathrm{r}}^2}$
E-2	$\frac{9}{4}$	1
E-3	1	$1-\delta_\mathrm{r}$
E-4	3	1
E-5	2	1
E-6	$\frac{3}{2}$	1

Magnetic		
Type	$\frac{\tilde{S}_h}{\Delta_\mathrm{f}^2}$	$\frac{l_h}{\Delta_\mathrm{f}}$
H-1	$\frac{3+\delta_\mathrm{r}}{\sqrt{11}}$	$\frac{\sqrt{11}}{2}$
H-2	$1-\frac{\delta_\mathrm{r}}{2}$	1
H-3	$1-\delta_\mathrm{r}$	1
H-4	4	2

Each electric field type is assigned a number from 1 to 6, and similarly, each magnetic field type is assigned a number from 1 to 4. We denote E-i to the i-th electric field type and H-i to the i-th magnetic field type.

It is interesting to note that some field types only appear in the corners of the subgridding region, as can be extracted from fig. 3.3. Particularly, field types E-4 and H-3 appears only in convex corners, field type E-5 appears only in concave corners, and field type E-6 appears only in the overlap between two convex corners.

The magnitudes in table 3.1 are referred to the cell length Δ_f corresponding to the fine level in the subgridding boundary considered. It is worth noting, however, that the field type H-4 belongs to the coarse level and not to the fine one, as can be seen in fig. 3.3. Furthermore, the values of $\tilde{S}_h$ and l_h for this field type are the same as in the usual FDTD, and thus it would seem like it can be trivially updated. However, it is considered a non-trivial field for two reasons. First, it has 5 neighbor electric components: three coarse and two fine. Second, the field type E-2, which belongs to the fine level, uses it to update itself, therefore its edge length Δ_c is not trivial from the fine perspective.

Let us provide an example to better understand the update equations applied to the subgridding boundary fields. Let us take a closer look at the electric field type E-1 and H-1. Discrete field components of these types use each other in their update schemes. An example is depicted in fig. 3.4. Using the same notation as in the figure, we find that:

- $E|_1$ is updated using the discrete components $H|_1$, $H|_2$, and $H|_3$, and it is of type E-1,

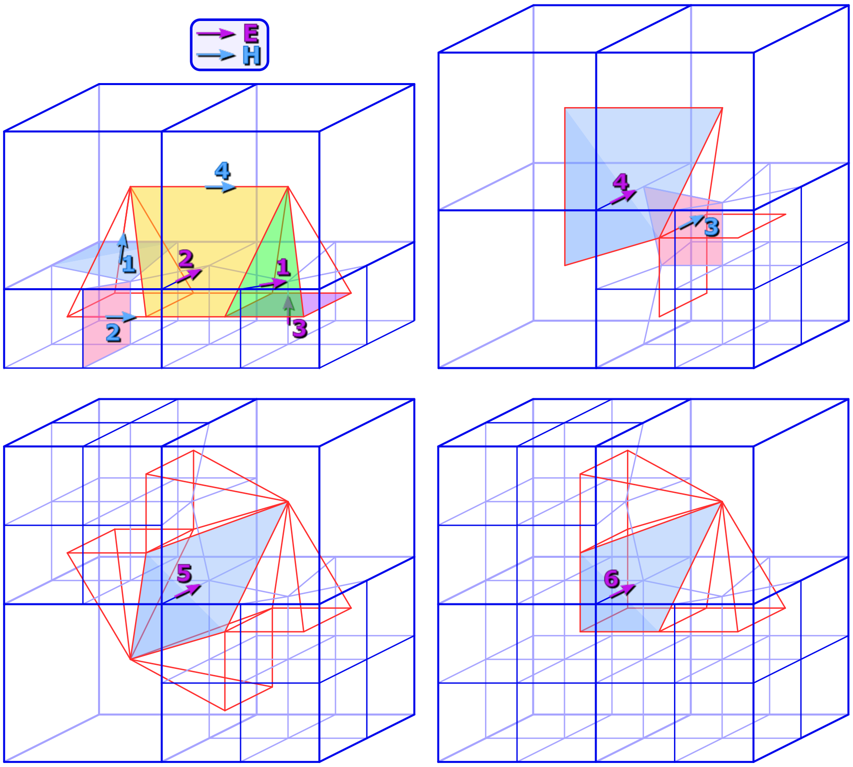

Figure 3.3: Illustration of the OI-SG field classification for a cubic grid with a refinement ratio $r = 2$. Electric fields are represented in purple and magnetic fields are represented in blue. Electric field types 1-3 are present in subgridded region faces, 4 in convex corners, 5 in concave corners, and 6 in the overlap of two convex corners. Magnetic field types 1-3 Field types correspond to table 3.1.

thus its equivalent surface is

$$\tilde{S}_e|_1 = \frac{3+\delta_{\rm r}}{4\sqrt{1+\delta_{\rm r}^2}}, \tag{3.18}$$

- $H|_1$ is updated using the discrete components $E|_1$, $E|_2$, $E|_3$, and $E|_3$, and it is of type H-1, thus its equivalent surface is

$$\tilde{S}_h|_1 = \frac{3+\delta_{\rm r}}{\sqrt{11}} \tag{3.19}$$

- $E|_1$ and $E|_4$ are of type E-1, thus $l_e|_1 = l_e|4 = \Delta_{\rm f}$,
- $E|_2$ and $E|_3$ are of type E-2, thus $l_e|_2 = l_e|3 = \sqrt{1+\delta_{\rm r}^2}\,\Delta_{\rm f}$,
- $H|_1$ and $H|_2$ are of type H-1, thus $l_h|_1 = l_h|2 = \sqrt{11}/2\,\Delta_{\rm f}$,
- $H|_3$ is of type H-2, thus $l_h|_3 = \Delta_{\rm f}$

Substituting this information into eqs. (3.16) and (3.17), we obtain

$$E|_1^{n+1} = E|_1^n + \frac{4\sqrt{1+\delta_\mathrm{r}^2}}{3+\delta_\mathrm{r}}\frac{\Delta t}{\varepsilon_0\,\Delta_\mathrm{f}}\left(\frac{\sqrt{11}}{2}H|_1^{n+\frac{1}{2}} - \frac{\sqrt{11}}{2}H|_2^{n+\frac{1}{2}} - H|_3^{n+\frac{1}{2}}\right), \tag{3.20}$$

$$H|_1^{n+\frac{1}{2}} = H|_1^{n-\frac{1}{2}} - \frac{\sqrt{11}}{3+\delta_\mathrm{r}}\frac{\Delta t}{\mu_0\,\Delta_\mathrm{f}}\left(-\sqrt{1+\delta_\mathrm{r}^2}\,E|_1^n + E|_2^n + E|_3^n - \sqrt{1+\delta_\mathrm{r}^2}\,E|_4^n\right). \tag{3.21}$$

The expressions shown in eqs. (3.20) and (3.21) are the OI-SG update equations applied to a particular example of discrete components of types E-1 and H-1. However, following the same procedure provides us with analogous expressions for any field in the computational domain of any simulation, with arbitrarily shaped subgridding regions and an arbitrary number of subgridding levels.

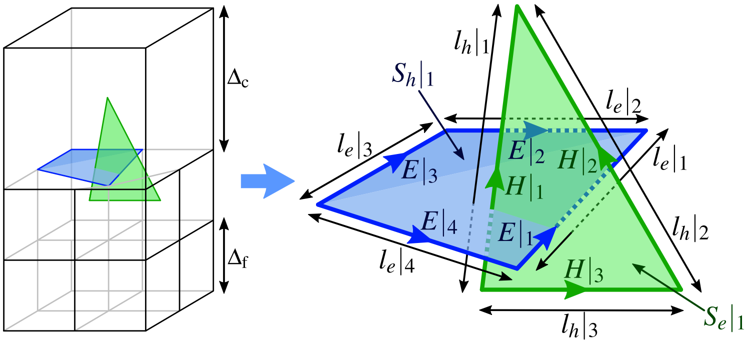

Figure 3.4: Illustration of the interlacing between two discrete field components of types E-1 and H-1. The update of the discrete component $E|_1$ requires the components $H|_1$, $H|_2$, and $H|_3$. The update of the discrete component $H|_1$ requires the components $E|_1$, $E|_2$, $E|_3$, and $E|_4$.

3.2 Local Time-Stepping

The expressions obtained in section 3.1 assume a simple GTS scheme. Although this is entirely possible, and simulations with a uniform time step are presented in the following chapters, it is not generally desirable. Reasons for this have already been discussed, but the main reason is that the CFL criterion constrains the time step in the finest level, making the simulation less efficient and inducing higher phase dispersion in the coarser levels (see fig. 2.8). For this reason, this section provides an LTS scheme that is used in diverse simulations throughout this work.

Let us assume a computational domain with $N_\mathrm{sg}+1$ levels, where the finest level is 0 and the coarsest level is N_{sg}. Each level uses a time step proportional to its cell length, $\Delta t_{n_{\mathrm{sg}}} \propto \Delta_{n_{\mathrm{sg}}}$. This means that, assuming a refinement ratio of $r = 2$, for each iteration of a level n_{sg}, the level $n_{\mathrm{sg}} - 1$ must perform two iterations, and therefore the discrete-time placements of the field components are different for each level. By convention, we assume that electric fields of all levels exist at times $t = n\Delta t_{N_{\mathrm{sg}}}$, where n is an integer. From here, each level must advance half a time step to meet the existence of the magnetic components, and then another half a time step to meet the electric components again. The resulting scheme, depicted in fig. 3.5, can be summarized as:

- Electric components of the coarsest level N_{sg} exist at times $t = n\,\Delta t_{N_{\mathrm{sg}}}$,
- Magnetic components of the coarsest level N_{sg} exist at times $t = \left(n + \frac{1}{2}\right)\Delta t_{N_{\mathrm{sg}}}$,
- Electric components of the level $N_{\mathrm{sg}} - 1$ exist at times
$$\frac{t}{\Delta t_{N_{\mathrm{sg}}}} \in \left\{ n, n + \frac{1}{2}, \right\}, \tag{3.22}$$
- Magnetic components of the level $N_{\mathrm{sg}} - 1$ exist at times
$$\frac{t}{\Delta t_{N_{\mathrm{sg}}}} \in \left\{ n + \frac{1}{4}, n + \frac{3}{4} \right\}, \tag{3.23}$$
- Electric components of the level $N_{\mathrm{sg}} - 2$ exist at times
$$\frac{t}{\Delta t_{N_{\mathrm{sg}}}} \in \left\{ n, n + \frac{1}{4}, n + \frac{1}{2}, n + \frac{3}{4} \right\}, \tag{3.24}$$
- Magnetic components of the level $N_{\mathrm{sg}} - 2$ exist at times
$$\frac{t}{\Delta t_{N_{\mathrm{sg}}}} \in \left\{ n + \frac{1}{8}, n + \frac{3}{8}, n + \frac{5}{8}, n + \frac{7}{8} \right\}, \tag{3.25}$$
- etc.

The main issue that we must now address is how the electric and magnetic fields from different levels use each other in their update equation. Let us assume a subgridding boundary between levels $n_{\mathrm{sg}} - 1$ and n_{sg}. First, we must note that the only discrete fine field components that require using coarse components are electric: discrete fields of types E-2, E-4 and E-6 require using components of type H-4. Furthermore, this situation is reciprocal:

the only discrete coarse field components that require using fine components are those of type H-4. This can be appreciated in fig. 3.3. In other words, fine electric components communicate with coarse magnetic ones, and vice-versa. No fine magnetic component sees the coarse level directly, and no coarse electric component sees the fine level directly.

Coarse magnetic components are placed at times $t = (n + \frac{1}{2})\Delta t_{n_{\text{sg}}}$, where n is an integer, and it expects using electric components placed at times $t = n\Delta t_{n_{\text{sg}}}$. Fine electric components are placed at times and $t = n\Delta t_{n_{\text{sg}}}$ and $t = (n + \frac{1}{2})\Delta t_{n_{\text{sg}}}$, therefore the update of coarse magnetic fields can be naturally done by using correctly time-placed fine electric components. This is represented by the green arrows in fig. 3.5. On the other hand, fine electric components require using coarse magnetic fields placed at times $t = (n - \frac{1}{4})\Delta t_{n_{\text{sg}}}$ and $t = (n + \frac{1}{4})\Delta t_{n_{\text{sg}}}$. It is clear that these coarse magnetic components do not exist within our scheme, so a different approach is required. Time interpolations and extrapolations can be used to obtain the required magnetic components, but doing this would remove the linearity of the OI-SG update scheme, and furthermore, this kind of approach is known to induce instabilities. For this reason, a different approach is devised in this work.

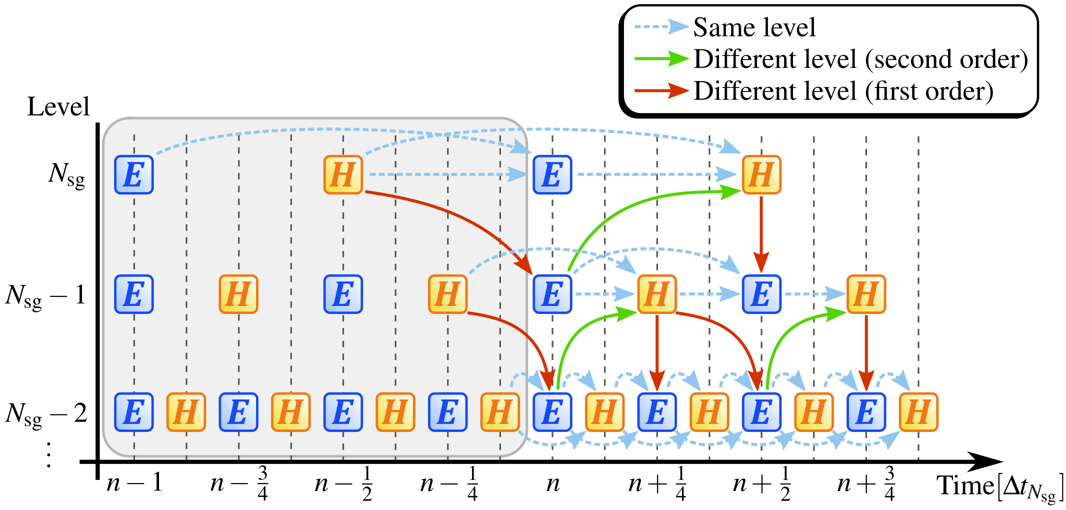

Figure 3.5: Scheme of the LTS employed for the OI-SG method. Each box marked E or H represents the set of all the electric or magnetic fields belonging to a particular level at a given time instant. Arrows represent field usage in the update: an arrow going from A to B implies that the update of some fields of B requires the usage of fields of A. Blue dashed arrows represent usage within the same level, green arrows represent different-level usage with second-order time derivatives, and red arrows represent different-level usage with first-order time derivatives.

Let us assume Faraday's Law (eq. (2.8)) in free space without sources for an integration patch containing an electric component that belongs to the level $n_{\text{sg}} - 1$. Now, let us evaluate it at a time instant $t = n\Delta t_{n_{\text{sg}}-1}$ and approximate the time-derivative by a non-centered finite difference:

$$\frac{dE|_i^n}{dt} \simeq \frac{E|_i^{n+1} - E|_i^n}{\Delta t_{n_{\mathrm{sg}}-1}}, \tag{3.26}$$

where the superindex n refers to the evaluation at time $t = n\Delta t_{n_{\mathrm{sg}}-1}$. By substituting it into Faraday's Law and applying the finite integration on the magnetic circulation, we obtain

$$E|_i^{n+1} = E|_i^n + \frac{\Delta t}{\varepsilon_0 \left.\tilde{S}_e\right|_i} \sum_{j \in \mathcal{N}_e|_i} \mathrm{sgn}_e|_{i,j} \left. l_h \right|_j H|_j^n. \tag{3.27}$$

Eq. (3.27) is formally identical to eq. (3.17) but with time-displaced magnetic components. By comparing eqs. (3.17) and (3.27), we may devise a combination where the magnetic components are placed at times $t = (n + \frac{1}{2})\Delta t_{n_{\mathrm{sg}}-1}$ if they belong to the same level as the electric component, and $t = n\Delta t_{n_{\mathrm{sg}}-1}$ if they belong to the immediate coarse level. Thus, we obtain

$$E|_i^{n+1} = E|_i^n + \frac{\Delta t_{n_{\mathrm{sg}}-1}}{\varepsilon_0 \left.\tilde{S}_e\right|_i} \left(\begin{array}{c} \displaystyle\sum_{\substack{j \in \mathcal{N}_e|_i \\ n_{\mathrm{sg},h}|_j = n_{\mathrm{sg}}-1}} \mathrm{sgn}_e|_{i,j} \left. l_h \right|_j H|_j^{n+\frac{1}{2}} \\ \displaystyle + \sum_{\substack{j \in \mathcal{N}_e|_i \\ n_{\mathrm{sg},h}|_j = n_{\mathrm{sg}}}} \mathrm{sgn}_e|_{i,j} \left. l_h \right|_j H|_j^n \end{array} \right), \tag{3.28}$$

where $n_{\mathrm{sg},h}|_j$ denotes the level of the j-th magnetic component.

Eq. (3.28) can be applied for discrete fine electric components placed at times $t = n\Delta t_{n_{\mathrm{sg}}}$ by using coarse magnetic components placed at $t = (n - \frac{1}{2})\Delta t_{n_{\mathrm{sg}}}$. However, it does not work for fine electric components placed at $t = (n + \frac{1}{2})\,\Delta t_{n_{\mathrm{sg}}}$, as it would require coarse magnetic components at $t = n\Delta t_{n_{\mathrm{sg}}}$, which does not exist either. This can be solved by repeating the same procedure with slight changes. First, we evaluate Faraday's Law at $t = n\Delta t_{n_{\mathrm{sg}}}$. Second, we employ a different non-centered finite difference for the time derivative:

$$\frac{dE|_i^{n+1}}{dt} \simeq \frac{E|_i^{n+1} - E|_i^n}{\Delta t_{n_{\mathrm{sg}}-1}}. \tag{3.29}$$

Now, by repeating exactly the same procedure, we get to

$$E\big|_i^{n+1} = E\big|_i^n + \frac{\Delta t_{n_{\text{sg}}-1}}{\varepsilon_0\, \tilde{S}_e\big|_i} \left(\begin{array}{c} \displaystyle\sum_{\substack{j \in \mathcal{N}_e|_i \\ n_{\text{sg},h}|_j = n_{\text{sg}}-1}} \operatorname{sgn}_e|_{i,j}\, l_h\big|_j\, H\big|_j^{n+\frac{1}{2}} \\ \displaystyle + \sum_{\substack{j \in \mathcal{N}_e|_i \\ n_{\text{sg},h}|_j = n_{\text{sg}}}} \operatorname{sgn}_e|_{i,j}\, l_h\big|_j\, H\big|_j^{n+1} \end{array} \right). \tag{3.30}$$

Eqs. (3.28) and (3.30) can be applied for discrete fine electric components placed at times $t = n\,\Delta t_{n_{\text{sg}}}$ and $t = (n + \frac{1}{2})\,\Delta t_{n_{\text{sg}}}$, respectively. The coarse-to-fine field usage for these cases is represented by the red arrows in fig. 3.5. A pseudocode implementation of the resulting OI-SG method with the proposed LTS is presented in algorithm 1. It is worth noting that this scheme reduces the time-derivative convergence order from $O(\Delta t^2)$ to $O(\Delta t)$ as the finite differences are non-centered, but provides a linear and explicit method, whose stability is demonstrated in section 3.4. Nonetheless, an LTS variant with time interpolations is also discussed in section 3.6.1.

Algorithm 1 Pseudocode implementation of the OI-SG LTS scheme.

```
procedure UpdateLevel($n_{\text{sg}}$)
    UpdateE($n_{\text{sg}}$)                          ▷ Update all electric fields from level $n_{\text{sg}}$
    if $n_{\text{sg}} > 0$ then UpdateLevel($n_{\text{sg}} - 1$)
    UpdateH($n_{\text{sg}}$)                          ▷ Update all magnetic fields from level $n_{\text{sg}}$
    if $n_{\text{sg}} > 0$ then UpdateLevel($n_{\text{sg}} - 1$)
end procedure
procedure PerformSimulation($N_{\text{sg}}$, $N_{\text{timesteps}}$)        ▷ Main procedure
    for $n_t = 1, \dots, N_{\text{timesteps}}$ do
        UpdateLevel($N_{\text{sg}}$)
    end for
end procedure
```

As a final note about the time-stepping scheme, it is important to note that the value of CFLN must always be defined by the most restrictive space step associated with a given time step. In executions with LTS, we have that $\Delta t_{n_{\text{sg}}} \propto \Delta_{n_{\text{sg}}}$, and therefore we can trivially define:

$$\text{CFLN}_{\text{LTS}} = \frac{\Delta t_{n_{\text{sg}}}}{\sqrt{3}\, c_0\, \Delta_{n_{\text{sg}}}}, \tag{3.31}$$

which is independent of n_{sg}. On the other hand, in executions with GTS, we have that $\Delta t_{n_{\text{sg}}} \equiv \Delta t$ is a constant, and thus CFLN must be defined by the finest-level space step, as it is the smallest one,

$$\mathrm{CFLN}_{\mathrm{GTS}} = \frac{\Delta t}{\sqrt{3}\, c_0\, \Delta_0}. \tag{3.32}$$

These definitions can be unified as

$$\mathrm{CFLN} = \frac{\Delta t_0}{\sqrt{3}\, c_0\, \Delta_0}. \tag{3.33}$$

3.3 Integration with boundary conditions

For some simulations, the subgridding boundary is required to extend until the computational domain boundary. When this happens, we need to integrate the OI-SG equations with the boundary conditions. Throughout this work, this integration is only required for three different boundary conditions: PEC, PMC, and periodic. The PEC case is trivial, as we only need to enforce the tangential electric fields at the boundaries to be zero. However, PMC and periodic BCs require special attention. Thus, this section provides the required knowledge to properly integrate them with the subgridding boundary.

3.3.1 PMC conditions

In some cases, we may require the subgridding boundary to reach a PMC boundary at the computational limit. As an example, this is required in cases where a truncated plane wave source (see section 2.1.3) propagates with normal incidence from a certain level towards a surface contained within a finer level.

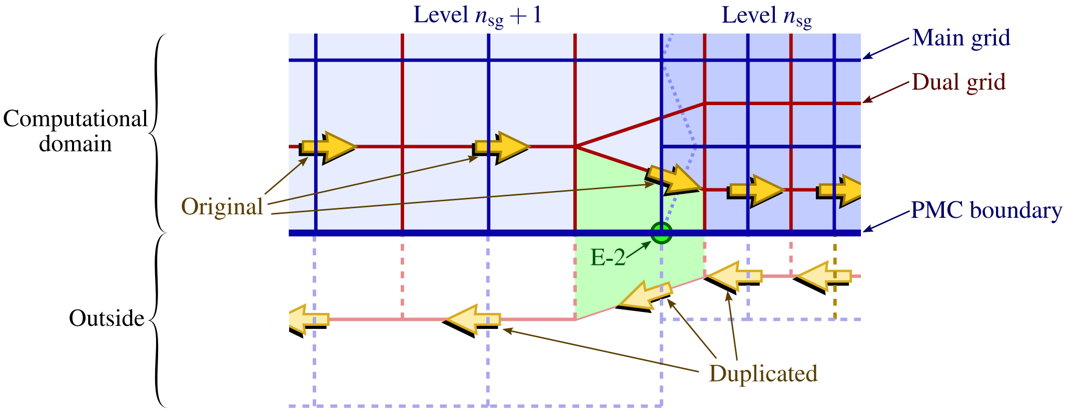

Figure 3.6: Scheme of the integration between a subgridding boundary and the PMC boundary conditions. The main grid is represented in blue, the dual grid is represented in red. The dual grid duplication is represented beyond the PMC boundary.

To simulate a PMC boundary, we are required to define a spatial configuration such that the interpolated tangential magnetic components are zero in the same way, in a similar way as explained in section 2.1.2. First, let us assume a subgridding boundary between two levels, n_{sg} and $n_{\text{sg}}+1$, that extends until reaching a PMC boundary at the computational limit. A cross-section is depicted in fig. 3.6. In the usual FDTD, this is solved by mirroring the discrete magnetic components that are placed half a cell from the PMC boundary and inverting their magnitude (a detailed explanation is provided in section 2.1.2). The approach devised for a case with a subgridding boundary consists of mirroring all the dual grid edges as they exist inside the computational domain and inverting the duplicated magnetic components, as depicted in fig. 3.6. This has several consequences:

- For the magnetic components that are not traversing the subgridding boundary, this is equivalent to the usual PMC boundary condition: the averaged magnetic field at the PMC boundary, which is tangential to it, is zero.
- The mirrored magnetic field of the components traversing the subgridding boundary is not entirely the opposite: only the component parallel to the PMC boundary is inverted, and the orthogonal component is preserved.
- The electric components tangential to both the subgridding and PMC boundaries are necessarily of type E-2.

3.3.2 Periodic conditions

Some of the numerical simulations performed in chapter 4 consist of periodic surfaces. They require periodic boundary conditions at the computational limits and, since they are confined to a refined region, the subgridding boundary is required to reach the periodic boundary as well.

As explained in section 2.1.2, boundaries with periodic conditions always come in pairs. In the usual FDTD, periodic boundaries can be implemented by identifying the tangential electric components and the normal magnetic components within the computational boundaries. However, if a subgridding boundary reaches at least one of the boundaries, a variety of situations can occur. This is depicted in fig. 3.7. To apply periodic conditions correctly, both the main and dual grids must adapt accordingly to match the grids of the opposite end. If, at some point, one end of the computational domain is refined and the opposite is not, this forces fine discrete components to appear at the coarse end. Furthermore, depending on the configuration of the subgridding regions, non-trivial fields appear at both ends, which must be taken into account. Once this is taken into account, we only need to identify the fields at both ends and apply the OI-SG update algorithm normally.

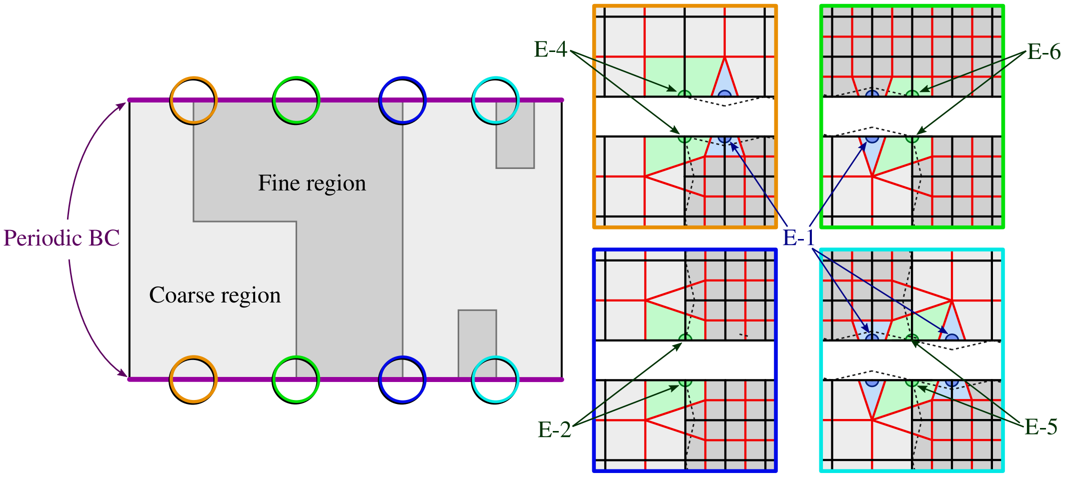

Figure 3.7: Cross-section of a simulation with subgridding and periodic boundaries. When a subgridding boundary reaches a periodic boundary, the main and dual grids must adapt to match the opposite limit of the computational domain, thus making non-trivial fields appear.

3.4 Stability

Stability is one of the main key points in TD numerical methods, and more specifically in algorithms that modify the standard FDTD equations, as is the case of subgridding schemes. As detailed in section 2.3, many subgridding methods present instabilities, and even late-time instabilities. Therefore, a stability analysis is fundamental to guarantee the robustness of the method: we need to determine whether it is stable in the long term and, if it is, what are the required conditions. The OI-SG method has a free parameter, δ_{r}, and the stability in FDTD-based methods is typically determined by the CFLN value. For this reason, we intend to obtain the maximum stable value of CFLN as a function of δ_{r}.

A rigorous analytical demonstration of the stability is usually far too complex, and for this reason, authors in the literature typically cannot provide them. Nonetheless, three different methodologies have been designed, implemented and tested to derive the stability conditions in this work. First, some reasoning is carried out on the OI-SG update equations, obtaining an analytical expression. Second, a set of minimal simulations are designed and their late-time stability is obtained by two independent approaches: a heuristic approach based on running many iterations, and a spectral approach based on obtaining the maximum magnitude eigenvalue of the update operator.

3.4.1 Analytical approximation

This methodology was presented in [46, 47] and was later given a different justification in [48]. The explanation presented here is based, with slight changes, on the works from [25, 26], which established a stability criterion on the non-structured cells in the conformal method.

First, let us take a look at the CFL criterion for a cubic grid (eq. (2.72)). This expression uses the speed of light in vacuum c_0, but this criterion must also hold for any material potentially present in a simulation, which could have a different speed of light. Thus, the CFL criterion can be rewritten as

$$\Delta t \leq \min_i \left\{ \frac{\Delta}{\sqrt{3}\, c_i} \right\}, \tag{3.34}$$

where c_i is the speed of light at the i-th material present in the simulation, and $i = 0$ represents free space. In general, all physical materials fulfill $c_i \leq c_0$, so eq. (3.34) is redundant for normal FDTD simulations. However, if we can show that the OI-SG equations are equivalent, to some extent, to non-physical superluminal materials, then eq. (3.34) can be applied to derive a stability criterion.

Now, let us rewrite the OI-SG update expression for the electric fields, eq. (3.17), as

$$E\big|_i^{n+1} = E\big|_i^n + \sum_{j \in \mathcal{N}_e|_i} \operatorname{sgn}_e\big|_{i,j} \frac{\Delta t}{\left(\varepsilon_0 \dfrac{\tilde{S}_e\big|_i}{l_h\big|_j \Delta} \right) \Delta} H\big|_j^{n+\frac{1}{2}}. \tag{3.35}$$

Let us define

$$\tilde{\varepsilon}\big|_{i,j} := \varepsilon_0 \frac{\tilde{S}_e\big|_i}{l_h\big|_j \Delta} \tag{3.36}$$

as the equivalent electric permittivity between the discrete components $E|_i$ and $H|_j$. Substituting in eq. (3.35), we obtain

$$E\big|_i^{n+1} = E\big|_i^n + \sum_{j \in \mathcal{N}_e|_i} \operatorname{sgn}_e\big|_{i,j} \frac{\Delta t}{\tilde{\varepsilon}\big|_{i,j} \Delta} H\big|_j^{n+\frac{1}{2}}. \tag{3.37}$$

By performing an analogous procedure on the OI-SG update equation for the magnetic field, eq. (3.16), we obtain

$$H\big|_i^{n+\frac{1}{2}} = H\big|_i^{n+\frac{1}{2}} - \sum_{j\in\mathcal{N}_h|_i} \mathrm{sgn}_h|_{i,j} \frac{\Delta t}{\tilde{\mu}|_{i,j}\Delta} E\big|_j^n, \tag{3.38}$$

where we have defined the magnetic permeability between the discrete components $H|_i$ and $E|_j$ as

$$\tilde{\mu}|_{i,j} := \varepsilon_0 \frac{\tilde{S}_h\big|_i}{l_e\big|_j \Delta}. \tag{3.39}$$

We note that eqs. (3.37) and (3.38) are both formally identical to the FDTD update equations in a cubic grid (eqs. (2.20)–(2.25)), but with electric permittivities $\tilde{\varepsilon}|_{i,j}$ and magnetic permeabilities $\tilde{\mu}|_{i,j}$. Now, let us consider two neighbor field components $E|_i$ and $H|_j$. We define the equivalent speed of light between these components as

$$\tilde{c}|_{i,j} := \frac{1}{\sqrt{\tilde{\varepsilon}|_{i,j}\,\tilde{\mu}|_{i,j}}} = c_0\sqrt{\frac{l_e|_i\Delta}{\tilde{S}_e\big|_i}\frac{l_h|_j\Delta}{\tilde{S}_h\big|_i}}. \tag{3.40}$$

Finally, substituting eq. (3.40) into eq. (3.34) we obtain

$$\Delta t \le \frac{\Delta}{\sqrt{3}\,c_0} \min_{\substack{i,j \\ E|_i, H|_j \,\text{neighbors}}} \left\{ \sqrt{\frac{\tilde{S}_e\big|_i}{l_e|_i\Delta}\frac{\tilde{S}_h\big|_i}{l_h\big|_j\Delta}} \right\} \tag{3.41}$$

If we take the usual definition of CFLN for a cubic grid from eq. (2.72), we obtain

$$\mathrm{CFLN} \le \min_{\substack{i,j \\ E|_i, H|_j \,\text{neighbors}}} \left\{ \sqrt{\frac{\tilde{S}_e\big|_i}{l_e|_i\Delta}\frac{\tilde{S}_h\big|_i}{l_h\big|_j\Delta}} \right\} \tag{3.42}$$

Particularly, if we look at fig. 3.3, it is possible to determine all the possible neighboring field types that may be present in a simulation. If we take the values of $\tilde{S}$ and l for these pairs and we plug them in eq. (3.42), we obtain the maximum stable CFLN value:

$$\mathrm{CFLN}_{\max}(\delta_\mathrm{r}) = \begin{cases} \dfrac{3+\delta_\mathrm{r}}{\sqrt{22\left(1+{\delta_\mathrm{r}}^2\right)}} & \text{if} \quad \delta_\mathrm{r} \le \dfrac{5}{13} \quad \text{(Field types: E-1, H-1)} \\[2ex] \sqrt{\dfrac{(3+\delta_\mathrm{r})(1-\delta_\mathrm{r})}{4\left(1+{\delta_\mathrm{r}}^2\right)}} & \text{if} \quad \delta_\mathrm{r} > \dfrac{5}{13} \quad \text{(Field types: E-1, H-3)} \end{cases} \tag{3.43}$$

This function is plotted in fig. 3.8, and its maximum value is found at $\delta_\text{r} = \frac{1}{3} \simeq 0.33$, which corresponds to CFLN $= \frac{2}{3} \simeq 0.67$.

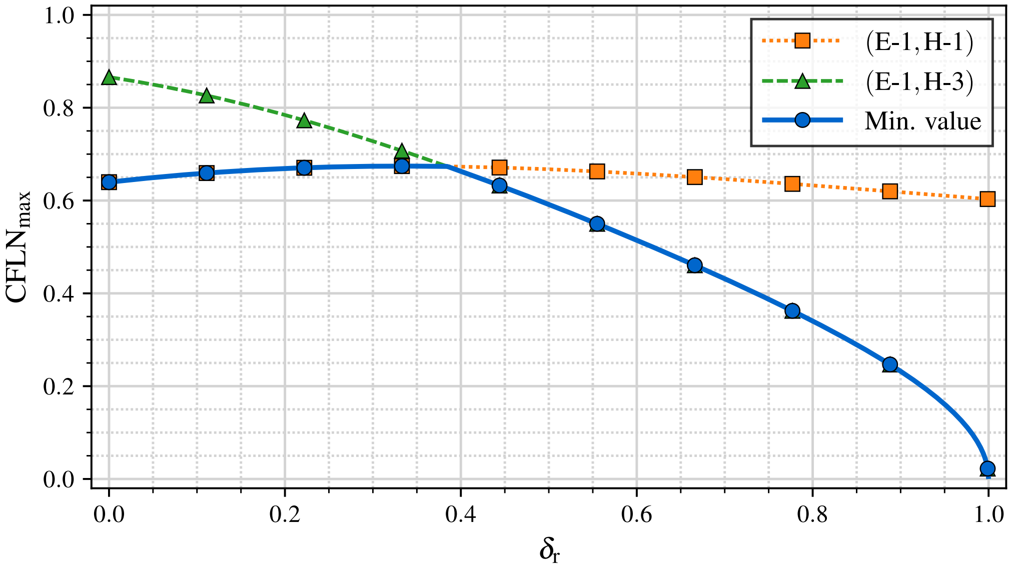

Figure 3.8: Maximum stable CFLN value as a function of δ_r found by the analytical method.

Eq. (3.42) tells us very valuable information about the sources of instabilities: they are related to large lines and small surfaces. It is worth noting that this is a known feature of the FDTD method [25, 26]. However, in the OI-SG case, we are not speaking just about areas, since we are dealing with equivalent surfaces $\tilde{S}$. Let us take a step back and look at the definition of $\tilde{S}$ in eqs. (3.14) and (3.4). We may see that the equivalent surface is the area of the integration patch's projection onto a plane normal to the discrete field vector, and the direction of this vector is defined by the grid edge it belongs to. Thus, it is clear that the projection's area is maximized when the edge that traverses through the integration patch is orthogonal to most of it. In other words: the more orthogonal the main and dual grids' surfaces are to each other's edges, the more stable the system is. Let us illustrate this with an example. If we look at the discrete component of type E-1 in fig. 3.3 or fig. 3.4, we may see that its surface consists of a triangle that does not change with δ_r. However, the main grid edge that traverses through it is indeed affected by δ_r, and thus if we write down the projected area $\tilde{S}$ (can be looked up at table 3.1), we have

$$\tilde{S}_\text{E-1} = \frac{3+\delta_\text{r}}{4\sqrt{1+\delta_\text{r}^{\,2}}}. \tag{3.44}$$

Interestingly, this expression finds its maximum value at $\delta_{\mathrm{r}} = \frac{1}{3}$, which indeed matches the maximum value found in eq. (3.43).

Additionally eq. (3.43), also tells us exactly which field types are the most critical for the stability: E-1, H-1, and, for greater values of δ_{r}, H-3. This information is used later in section 3.6.2 to derive a methodology that allows us to increase the maximum stable value of CFLN.

It is important to note that the procedure described in this section is not rigorous. $\tilde{\varepsilon}|_{i,j}$ does not act exactly as a normal electric permittivity. This is because these values are different for each one of the surrounding components $H|_j$, unlike in a normal bulk material, even for anisotropic ones. A similar argument can be made for $\tilde{\mu}|_{i,j}$.

3.4.2 Heuristic and spectral approaches

In many cases in the literature, as seen in section 2.3, authors provide heuristic approaches to demonstrate the stability of their methods. Typically, this approach consists of running one or more simulations for a large amount of iterations, usually over 10^6 iterations. This methodology is not rigorous as it depends on the chosen sources, and it can only identify instabilities that appear before the execution is stopped. For this reason, other authors present spectral analyses of their methods, which do not depend on the sources. In this section, both approaches are carried out for a set of small simulations with the condition that all field types are present, both with GTS and LTS.

Stability in LTI systems

First, let us present a theoretical justification for the spectral method. This approach is devised for any generic initial-value problem in linear time invariant (LTI) algorithms, which can be represented as

$$\vec{v}|^{n+1} = A\,\vec{v}|^{n} \tag{3.45}$$

$$\vec{v}|^{0} = \vec{v}_0, \tag{3.46}$$

where $\vec{v}|^n$ is the state of the system at the n-th iteration, $\vec{v}_0$ is the initial state, and A is the time-update linear operator, which is independent of time and updates the system in every iteration. In matrix representation, A is a matrix and the states are column vectors.

As explained in section 2.1.5, stability throughout this work refers to BIBO stability (eq. (2.81)), i.e. non-divergence. In LTI systems, we can distinguish two types of divergences [157, 158]. First, polynomial divergences may exist only when A is not diagonalizable, which is later demonstrated in this section. For example, let us consider the following 2D example,

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{3.47}$$

For an arbitrary initial state $\vec{v}_0 = (v_1, v_2)$, we find that

$$\vec{v}|^n = \begin{pmatrix} v_1 + n v_2 \\ v_2 \end{pmatrix}, \tag{3.48}$$

which grows linearly if $v_2 \neq 0$. The second type of divergences are exponential, they may exist in all cases and are directly related to the eigenvalues of A. This kind of divergence is discussed next.

Before continuing the explanation, let us make some definitions. Let us consider a linear operator A that acts on a space V of dimension N. We say that λ is an eigenvalue of A if an state $\vec{e}$ exists such that

$$A\vec{e} = \lambda\vec{e}. \tag{3.49}$$

Here, we say that e is an *eigenvector* or *eigenstate* of A associated to λ. The eigenvalues can be calculated as the roots of the characteristic polynomial $P_A(\lambda)$, which is defined as

$$P_A(\lambda) := |A - \lambda\, I_N|, \tag{3.50}$$

where I_N is the N-dimensional identity operator. According to the fundamental theorem of algebra, $P_A(\lambda)$ has exactly N roots, therefore A has exactly N eigenvalues. Thus, we may define the set of all eigenvalues of A as

$$\Lambda_A := \left\{\lambda_i\right\}_{i=1}^{N}. \tag{3.51}$$

We note that some eigenvalues may appear repeat. The amount of times an eigenvalue λ_i appears repeated in Λ_A is named the *arithmetic multiplicity* and is denoted $\mu_A(\lambda_i)$. By definition, each eigenvalue λ_i has one associated subspace of eigenvectors, named *eigenspace* and denoted $V_A(\lambda_i)$. The dimension of $V_A(\lambda_i)$ is called the *geometric multiplicity* of λ_i and is denoted $\gamma_A(\lambda_i)$. We find that $1 \leq \gamma_A(\lambda_i) \leq \mu_A(\lambda_i)$. Additionally, if $\gamma_A(\lambda_i) = \mu_A(\lambda_i)$

for every λ_i, then the summation of all the eigenspaces give V. In mathematical terms,

$$\gamma_A(\lambda_i) = \mu_A(\lambda_i) \quad \forall i \in \{1,\dots,N\} \quad \Rightarrow \quad \sum_{i=1}^{N} V_A(\lambda_i) = V, \tag{3.52}$$

where the summation symbol represents vector space summation. In this case, and only in this case, the union of the bases of all eigenspaces $V_A(\lambda_i)$ is on itself a base of V composed of eigenvectors only. The representation of A in this base is diagonal, and it is possible to prove that this is the only way to find a base in which A is diagonal. Therefore, we say that A is diagonalizable if and only if the arithmetic and geometric multiplicities of each eigenvalue are equal, or equivalently if we can find a base of V composed of eigenvectors only.

Let us consider now a diagonalizable operator A acting on a space V of dimension N. Since A is diagonalizable, we may define a base of eigenvectors as

$$\Gamma_A := \left\{\hat{e}_i\right\}_{i=1}^{N}, \tag{3.53}$$

where

$$A\,\hat{e}_i = \lambda_i\,\hat{e}_i \quad \forall i \in \{1,\dots,N\}, \tag{3.54}$$

and

$$\|\hat{e}_i\| = 1 \quad \forall i \in \{1,\dots,N\}. \tag{3.55}$$

Note that the base of eigenvectors is not unique, but, as stated previously, we can always find one as long as A is diagonalizable. We also note that the eigenvalues λ_i are, in general, complex. Therefore, let us write their polar form as

$$\lambda_i = |\lambda_i|\, e^{\mathbb{i}\,\theta_i}, \tag{3.56}$$

where $|\lambda_i|$ is the module, θ_i the phase, and $\mathbb{i}$ is the imaginary unit. We may write an arbitrary initial state $\vec{v}_0$ expressed in this base as

$$\vec{v}_0 = \sum_{i=1}^{N} v_i\,\hat{e}_i, \tag{3.57}$$

where v_i are the coefficients of $\vec{v}_0$ expressed in the base Γ_A. Using eqs. (3.57) and (3.54), and the linearity of A, the state after n iterations $\vec{v}|^n$ can be expressed as

$$\vec{v}|^n = \sum_{i=1}^{N} v_i \lambda_i^n \hat{e}_i. \tag{3.58}$$

Let us now calculate the squared norm $\| \vec{v}|^n \|^2$ on eq. (3.58). Using the notation from eq. (3.56), and after some algebra, we get to

$$\| \vec{v}|^n \|^2 = \sum_{i=1}^{N} |\lambda_i|^{2n} |v_i|^2 + 2 \sum_{\substack{i=1 \\ j>i}}^{N} |\lambda_i|^n \left|\lambda_j\right|^n \mathfrak{Re} \left[v_i^* \, v_j \, e^{jn(\theta_i - \theta_j)} \left(\hat{e}_i^* \cdot \hat{e}_j\right) \right], \tag{3.59}$$

where z^* denotes the complex conjugate of z and $\mathfrak{Re}(z)$ denotes the real part of z. It is worth noting that, in general, two eigenspaces $V_A(\lambda_i)$ and $V_A(\lambda_j)$ with $\lambda_i \neq \lambda_j$ are not necessarily orthogonal, and thus, $\hat{e}_i^* \cdot \hat{e}_j$ is not necessarily zero. However, from within the same eigenspace, we may always choose an orthogonal base, thus making $\hat{e}_i^* \cdot \hat{e}_j = 0$ if $\lambda_i = \lambda_j$ and $i \neq j$. In what follows, we assume this to be the case, as it simplifies some of the calculus.

By looking at eq. (3.59), it is possible to draw two conclusions. First, if $|\lambda_i| > 1$ for any i, then it is always possible to find initial states that grow exponentially. Particularly, if we take the initial state $\vec{v}_0 = \hat{e}_i$, we have

$$\| \vec{v}|^n \|^2 = |\lambda_i|^{2n} \to +\infty, \tag{3.60}$$

which is unstable in terms of BIBO stability (eq. (2.81)). Furthermore, even if $\hat{e}_i$ is not real, it is always possible to find an unstable real initial state as long as A is real. If A is real, then the characteristic polynomial (eq. (3.50)) is also necessarily real, and thus all the roots must come in pairs of complex conjugates. In other words, for each eigenvalue λ_i with an eigenvector $\hat{e}_i$, the complex conjugate λ_i^* is also necessarily an eigenvalue, and it is trivial to prove that the complex conjugate $\hat{e}^*$ is an eigenvector associated to it. If we take the initial state $\vec{v}| = \hat{e}_i + \hat{e}_i^*$, with $|\lambda_i| > 1$, we obtain

$$\| \vec{v}|^n \|^2 = 2 \, |\lambda_i|^{2n} \left[1 + \mathfrak{Re}\left(\hat{e}_i \cdot \hat{e}_i\right)\right], \tag{3.61}$$

which diverges if $\mathfrak{Re}\left(\hat{e}_i \cdot \hat{e}_i\right) > -1$. If we write

$$\hat{e}_i = \left(|e_{i1}| e^{\mathbb{i}\,\theta_{i1}}, |e_{i2}| e^{\mathbb{i}\,\theta_{i2}}, \ldots, |e_{iN}| e^{\mathbb{i}\,\theta_{iN}} \right), \tag{3.62}$$

then we have

$$\mathfrak{Re}\left(\hat{e}_i \cdot \hat{e}_i\right) = \sum_{j=1}^{N} |e_{ij}|^2 \cos\left(2\theta_{ij}\right) \geq -\sum_{j=1}^{N} |e_{ij}|^2 = -\|\hat{e}_i\|^2 = -1. \tag{3.63}$$

By looking at eq. (3.63), it is clear that $\mathfrak{Re}\left(\hat{e}_i \cdot \hat{e}_i\right) > -1$ unless each of the phases θ_{ij} are either $\frac{\pi}{2}$ or $\frac{3\pi}{2}$. However, if this happens, then $\hat{e}$ is necessarily a pure imaginary vector. Therefore, it is possible to divide it by $\mathbb{i}$, obtaining a pure real eigenvector with the same eigenvalue λ_i, which can be used as an initial state, leading us back to the divergence found in eq. (3.60).

The second conclusion that can be extracted from eq. (3.59) is that, if all eigenvalues fulfill $|\lambda_i| \leq 1$, then the system is necessarily BIBO stable. It can be proven by noting that all terms are bounded. First, we have that $|\lambda_i|^n \leq 1$ for any n, and therefore,

$$\sum_{i=1}^{N} |\lambda_i|^{2n} |v_i|^2 \leq \sum_{i=1}^{N} |v_i|^2 \equiv B_1. \tag{3.64}$$

Second, we have that

$$\begin{aligned} &\mathfrak{Re}\left(v_i^* \, v_j \, e^{\mathbb{i}\, n(\theta_i - \theta_j)} \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right) \\ &= \left|v_i^* \, v_j \, \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right| \cos\left\{\text{phase}\left[v_i^* \, v_j \, \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right] + n(\theta_i - \theta_j)\right\} \leq \left|v_i^* \, v_j \, \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right|, \end{aligned} \tag{3.65}$$

where phase(z) is the complex phase of z. This implies

$$\sum_{\substack{i=1 \\ j>i}}^{N} |\lambda_i|^n \left|\lambda_j\right|^n \mathfrak{Re}\left[v_i^* \, v_j \, e^{jn(\theta_i - \theta_j)} \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right] \leq \sum_{\substack{i=1 \\ j>i}}^{N} \left|v_i^* \, v_j \, \left(\hat{e}_i^* \cdot \hat{e}_j\right)\right| \equiv B_2, \tag{3.66}$$

and therefore

$$\| \vec{v}|^n \|^2 \leq B_1 + 2B_2, \tag{3.67}$$

where neither B_1 nor B_2 depend on the time iteration n.

To summarize, we have demonstrated that a diagonalizable update operator A complies

$$A \text{ is BIBO stable } \Longleftrightarrow \max_i \left\{|\lambda_i|\right\} \leq 1. \tag{3.68}$$

Additionally, eq. (3.59) also proves that, if some initial state is stable, its norm diverges exponentially. Therefore, if A is diagonalizable, polynomial divergences are not possible.

Late-time instabilities in diagonalizable LTI systems

Let us consider the greatest-norm eigenvalue of A and denote it $\lambda_{\max}$. This is,

$$|\lambda_i| \leq |\lambda_{\max}| \quad \forall i \in \{1,\ldots,N\}. \tag{3.69}$$

From the base Γ_A, let us take one of the unit-norm eigenvectors corresponding to $\lambda_{\max}$ and denote it $\hat{e}_{\max}$. Now, let us consider an arbitrary initial state $\vec{v}_0$ that, if represented in the base Γ_A, has a non-zero component $v_{\max}$ on $\hat{e}_{\max}$,

$$\vec{v}_0 = \left(\sum_{\substack{i=1 \\ \hat{e}_i \neq \hat{e}_{\max}}}^{N} v_i\, \hat{e}_i \right) + v_{\max}\, \hat{e}_{\max} \quad \text{with} \quad v_{\max} \neq 0. \tag{3.70}$$

Let us now assume that A is unstable, i.e. $|\lambda_{\max}| > 1$. Late-time instabilities are characterized by taking a large amount of iterations to become appreciably unstable. However, we must determine what "appreciably unstable" means in mathematical terms. Thus, let us define the following criterion: we consider that the execution of A on an initial state $\vec{v}_0$ has become appreciably unstable at the n-th iteration if

$$\frac{||\,\vec{v}|^n||}{||\vec{v}_0||} \geq 10^5. \tag{3.71}$$

Note that the number 10^5 must not be taken as a strict criterion, but rather as an approximate order of magnitude that we may use to determine when an execution has become unstable. If we look at eq. (3.59), it is clear that for large values of n, the norm is dominated by the term $|\lambda_{\max}|^{2n}$. Therefore, for a large n, we may write

$$||\,\vec{v}|^n||^2 \simeq |\lambda_{\max}|^{2n} \sum_{\substack{i=1 \\ \lambda_i = \lambda_{\max}}}^{N} |v_i|^2 . \tag{3.72}$$

If we run many different executions with arbitrary initial states, it is expected that, on average, the magnitude of the components of $\vec{v}_0$ on the base of Γ_A will tend to be equal, i.e. $|v_i| \simeq |v_j| \forall i, j$. Assuming this condition, we have

$$||\vec{v}_0||^2 \simeq N\, |v_i|^2 \quad \forall i \in \{1,\ldots,N\}, \tag{3.73}$$

and

$$\|\vec{v}|^n\|^2 \simeq |\lambda_{\max}|^{2n}\, \mu_A(\lambda_{\max})\, |v_i|^2 \quad \forall i \in \{1,\ldots,N\}, \tag{3.74}$$

which, in combination, gives

$$\frac{\|\vec{v}|^n\|}{\|\vec{v}_0\|} \simeq |\lambda_{\max}|^n \sqrt{\frac{\mu_A(\lambda_{\max})}{N}}. \tag{3.75}$$

Here, the square root term is a constant given A, and since we are dealing with the asymptotic behavior when $n \to +\infty$, let us neglect it and write down

$$\frac{\|\vec{v}|^n\|}{\|\vec{v}_0\|} \simeq |\lambda_{\max}|^n. \tag{3.76}$$

With this, in combination with eq. (3.71), we can finally state that, on average, a late-time unstable execution of A becomes appreciably unstable at the n_{ins}-th iteration if

$$|\lambda_{\max}|^{n_{\text{ins}}} \simeq 10^5. \tag{3.77}$$

By looking at eq. (3.77), it is clear that, the larger n_{ins} is, the closer $|\lambda_{\max}|$ must be to one. Thus, we can rewrite $|\lambda_{\max}|$ as

$$|\lambda_{\max}| = 1 + \frac{1}{\varepsilon}, \tag{3.78}$$

where ε is expected to be a very large positive number. If we substitute eq. (3.78) into eq. (3.77), we obtain

$$\left(1+\frac{1}{\varepsilon}\right)^{N_{\text{ins}}} = \left[\left(1+\frac{1}{\varepsilon}\right)^{\varepsilon}\right]^{N_{\text{ins}}/\varepsilon} \simeq e^{N_{\text{ins}}/\varepsilon} \simeq 10^5, \tag{3.79}$$

where we have used Bernoulli's limit:

$$\lim_{\varepsilon \to +\infty} \left(1+\frac{1}{\varepsilon}\right)^{\varepsilon} = e. \tag{3.80}$$

Now, operating with eq. (3.79), we get

$$N_{\text{ins}} \simeq 5\,\ln(10)\,\varepsilon \simeq 10\,\varepsilon \tag{3.81}$$

The obtained expression gives us crucial information about the order of the magnitudes that we must expect for the eigenvalues in late-time instabilities. For example, consider an LTI system that, on average, takes $n_{\mathrm{ins}} \simeq 10^7$ iterations to become visually unstable. By using eq. (3.81), we may now expect that the maximum eigenvalue magnitude is in the order of $|\lambda_{\max}| \simeq 1+10^{-6}$. Note that the specific numbers used for this explanation are, in some measure, arbitrary, and they will depend on the case, but it provides us with information about the order of the magnitudes.

Now, let us assume that we want to test the stability of a diagonalizable operator A by numerically obtaining $|\lambda_{\max}|$. If the instabilities of A are late-time, we necessarily require a very low tolerance on any algorithm we use to calculate $|\lambda_{\max}|$. In the previous example, we would require a tolerance of $10^{-4}\%$, or even lower for later-time instabilities. Furthermore, in many cases, the operator A might depend on certain parameters, such as the OI-SG equations depend on δ_{r} and CFLN, which leads us to a parametric study of the stability that requires obtaining $|\lambda_{\max}|$ for many operators and not just one. Numerical algorithms for obtaining eigenvalues are known for having high complexities, and additionally, the minimum dimensions of the linear operator for systems such as the OI-SG method are in the order of thousands, thus making this approach, in principle, not feasible. However, an alternate approach is proposed as follows.

Let us consider the linear operator A and another set of linear operators M_m defined as

$$\begin{aligned} M_m &= M_{m-1}^2 \quad \text{if} m > 0 \\ M_0 &= A. \end{aligned} \tag{3.82}$$

This implies that

$$M_m = A^{2^m}, \tag{3.83}$$

which effectively means that M_m is the operator that corresponds to an advance of 2^m iterations. This procedure allows us to obtain an advancement operator of an exponentially increasing amount of time steps, and more important, the set of eigenvalues of M_m is

$$\Lambda_{M_m} = \left\{ \lambda_{m,i} \right\}_{i=1}^{N}, \tag{3.84}$$

where

$$\lambda_{m,i} = \lambda_i^{2^m}, \tag{3.85}$$

and therefore

$$|\lambda_{m,i}| = |\lambda_i|^{2^m}. \tag{3.86}$$

We note that, as the exponentiation is a strictly increasing function, the sorting of the eigenvalues by magnitude will remain the same as in A, meaning that

$$|\lambda_{m,\max}| = |\lambda_{\max}|^{2^m}, \tag{3.87}$$

where $\lambda_{m,\max}$ the eigenvalue of M_m with the largest magnitude. This means that, for a late-time unstable simulation, even if $|\lambda_{\max}|$ is very close to 1, $|\lambda_{m,\max}|$ will become appreciably greater than 1 for a sufficiently large m. An example of this behavior is illustrated in fig. 3.9

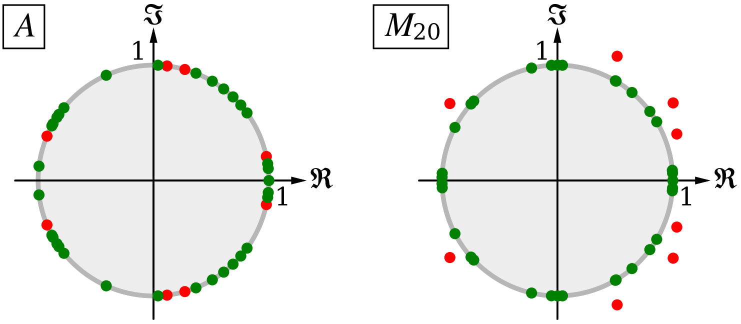

Figure 3.9: Illustration of the spectral radius of a given linear operator A that represents a late-time unstable algorithm. After enough iterations of eq. (3.82), the modulus of the eigenvalues becomes significantly greater than 1. Gray circle represents the subset $\{z \in \mathbb{C} / |z| \leq 1\}$. Green dots represent eigenvalues of unit modulus. Red dots represent eigenvalues with a modulus greater than one.

Let us go back to the previous example, where $|\lambda_{\max}| \simeq 1 + 10^{-6}$. If we use the procedure defined in eq. (3.82) for $m = 20$ iterations, we obtain that $|\lambda_{20,\max}| \simeq 2.85$. This means that, if we try to numerically solve the largest magnitude eigenvalue of M_{20} instead of A, the required tolerance may be much higher, for example of $\sim 10\%$, hence having much less impact on the computational cost. And after having numerically calculated $|\lambda_{m,\max}|$, we may obtain $|\lambda_{\max}|$ by simply doing

$$|\lambda_{\max}| = |\lambda_{m,\max}|^{-2^m}. \tag{3.88}$$

It is worth noting that the complex number $\lambda_{\max}$ cannot be calculated from $\lambda_{m,\max}$ as the inverse exponentiation is a multivalued function in complex analysis. However, to determine the stability we only require the magnitude $|\lambda_{\max}|$.

To finish this explanation, we only need to provide a criterion to establish how many iterations of eq. (3.82) are required for a given operator A, this is, how large m should be. Let us assume that we are interested in identifying any instability up to a given number of iterations n_{ins} of A. To do this, we need to choose m such that $|\lambda_{m,\max}|$ becomes large enough to be able to use a high tolerance. In general, we are not interested in getting the exact value of $|\lambda_{m,\max}|$, only to determine whether it is greater than one. Thus, let us choose m such that $|\lambda_{m,\max}| \simeq e$, which allows us to use a tolerance of $\sim 60\%$. Substituting in eq. (3.87) and using Bernoulli's limit again (eq. (3.80)), after some algebra, we obtain

$$m \simeq \log_2(\varepsilon) \simeq \log_2\left(\frac{n_{\text{ins}}}{10}\right). \tag{3.89}$$

This provides us with a very powerful method to establish late-time instabilities in diagonalizable LTI systems. For example, if we want to determine stability up until an iteration $n_{\text{ins}} \simeq 10^7$, we have that $m \simeq 20$. This means performing 20 matrix multiplications and determining $\lambda_{m,\max}$ with a high tolerance. In this work, this has been found to be much less expensive in practice than trying to calculate directly $\lambda_{\max}$ with the required low tolerance.

As a last note, it is relevant to point out that this procedure is affected by the floating-point representation of real numbers in computers. All the simulations performed in this work use real numbers with 8 bytes precision, which provides approximately 16 significant decimal positions. The matrix multiplications involved in eq. (3.82) require multiple sums of N floating-point numbers, which has an associated error that increases with m. This can be mitigated by employing precise floating-point summation algorithms such as Kahan's summation [168], however, the calculated eigenvalues stop being significant when we try to determine them for $|\lambda_{\max}| \simeq 1+10^{-16}$, as this is the precision limit of the machine. Thus, with 8 bytes floating-point numbers, the maximum value of m that makes sense in this procedure is $m \simeq 53$, which corresponds to $n_{\text{ins}} \simeq 10^{17}$.

Implementation for the OISG method

In order to apply the proposed methodology to the OI-SG method, first we need to write it as in eq. (3.46) to determine the operator A. First, let us consider a case with GTS. We define the set of all stored electric components at a given time $t = n\Delta t$ as $E|^n$ and, similarly, the set of all stored magnetic components at $t = (n+\frac{1}{2})\Delta t$ as $H|^n$. Let us assume that, at some point of the execution, we have the following state:

$$\begin{pmatrix} E|^{n} \\ H|^{n+\frac{1}{2}} \end{pmatrix}. \tag{3.90}$$

Since the magnetic state is evaluated at a later physical time, we need to update the electric fields. Assuming no external sources, this is a linear operation that can be described with a matrix denoted U_E,

$$U_E \begin{pmatrix} E|^{n} \\ H|^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} E|^{n+1} \\ H|^{n+\frac{1}{2}} \end{pmatrix}. \tag{3.91}$$

Similarly, the update of the magnetic fields can be written as a linear operator U_H,

$$U_H \begin{pmatrix} E|^{n+1} \\ H|^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} E|^{n+1} \\ H|^{n+\frac{3}{2}} \end{pmatrix}. \tag{3.92}$$

At this point, we may define the update operator

$$A := U_H U_E, \tag{3.93}$$

and the state at the n-th iteration

$$\vec{v}|^{n} := \begin{pmatrix} E|^{n} \\ H|^{n+\frac{1}{2}} \end{pmatrix}. \tag{3.94}$$

This, in combination with eqs. (3.91) and (3.92), gives

$$A\vec{v}|^{n} = \vec{v}|^{n+1}. \tag{3.95}$$

In other words, the operator required for the eigenvalue calculus is the composition of the electric and magnetic updates.

The case with LTS, on the other hand, requires more attention. First, let us denote $E_{n_{\text{sg}}}|^{N}$ and $H_{n_{\text{sg}}}|^{N}$, respectiely, to the set of all stored electric and magnetic components that belong to the level n_{sg} evaluated at the time $t = N\Delta t_{N_{\text{sg}}}$, where $\Delta t_{N_{\text{sg}}}$ is the coarsest-level time step. Note that N is not necessarily an integer nor a half-integer; the possible values depend on n_{sg}. Now, let us define the update operator $U_{E,n_{\text{sg}}}$ that performs the update of the discrete electric fields that belong to the level n_{sg}. Similarly, we define $U_{H,n_{\text{sg}}}$ as the operator that updates the discrete magnetic fields that belong to the level n_{sg}. We also define $A_{n_{\mathrm{sg}}}$ as a full update of the level n_{sg}, which is equivalent to calling the function UPDATELEVEL(n_{sg}) in algorithm 1. Noticeably, this implies updating the electric and magnetic fields of the level n_{sg}, but for each one, it also requires performing a full update of the level $n_{\mathrm{sg}}-1$ if $n_{\mathrm{sg}} \neq 0$. This can be written in mathematical terms as

$$A_{n_{\mathrm{sg}}} = \begin{cases} A_{n_{\mathrm{sg}}-1}\, U_{H,n_{\mathrm{sg}}} A_{n_{\mathrm{sg}}-1}\, U_{E,n_{\mathrm{sg}}} & \text{if} \quad n_{\mathrm{sg}} \neq 0 \\ \\ U_{H,n_{\mathrm{sg}}} U_{E,n_{\mathrm{sg}}} & \text{if} \quad n_{\mathrm{sg}} = 0 \end{cases} . \tag{3.96}$$

It is easy to see that $A_{n_{\mathrm{sg}}}$ performs an advance of $\Delta t_{n_{\mathrm{sg}}}$ on all the sets $E_{n_{\mathrm{sg}}'}$ and $H_{n_{\mathrm{sg}}'}$ such that $n_{\mathrm{sg}}' \leq n_{\mathrm{sg}}$, and leaves them unchanged if $n_{\mathrm{sg}}' > n_{\mathrm{sg}}$.ç Therefore, the operator $A_{N_{\mathrm{sg}}}$ performs an advance of $\Delta t_{N_{\mathrm{sg}}}$ on all the discrete fields in the simulation. If we define the state at the n-th iteration as

$$\vec{v}|^n := \begin{pmatrix} E_{N_{\mathrm{sg}}}\big|^n \\ H_{N_{\mathrm{sg}}}\big|^{n+\frac{1}{2}} \\ E_{N_{\mathrm{sg}}-1}\big|^{n+\frac{1}{2}} \\ H_{N_{\mathrm{sg}}-1}\big|^{n+\frac{3}{2}} \\ \vdots \\ E_0\big|^{n+1-2^{n_{\mathrm{sg}}-N_{\mathrm{sg}}}} \\ H_0\big|^{n+1-2^{n_{\mathrm{sg}}-N_{\mathrm{sg}}-1}} \end{pmatrix}, \tag{3.97}$$

it is easy to see that

$$A_{N_{\mathrm{sg}}}\, \vec{v}|^n = \vec{v}|^{n+1}. \tag{3.98}$$

and therefore $A \equiv A_{N_{\mathrm{sg}}}$ is the update operator whose eigenvalues we need to calculate. The operator $A_{N_{\mathrm{sg}}}$ and the states $\vec{v}|^n$ and $\vec{v}|^{n+1}$ are represented in fig. 3.10.

Once the update operators A are determined, we may obtain the evolution matrix for any given simulation case and perform the procedure described above. To do so, a set of small simulation cases have been prepared. Each case consists only of free space with arbitrary refined regions and periodic boundary conditions. For each one of them, we vary δ_{r} and CFLN in the interval $[0,1]$, and, for each pair $(\delta_{\mathrm{r}}, \mathrm{CFLN})$, we obtain the evolution matrix A and calculate $|\lambda_{\max}|$. It is worth noting that this procedure assumes that A is diagonalizable. A demonstration of the diagonalizability of the OI-SG update matrices, in general, cannot

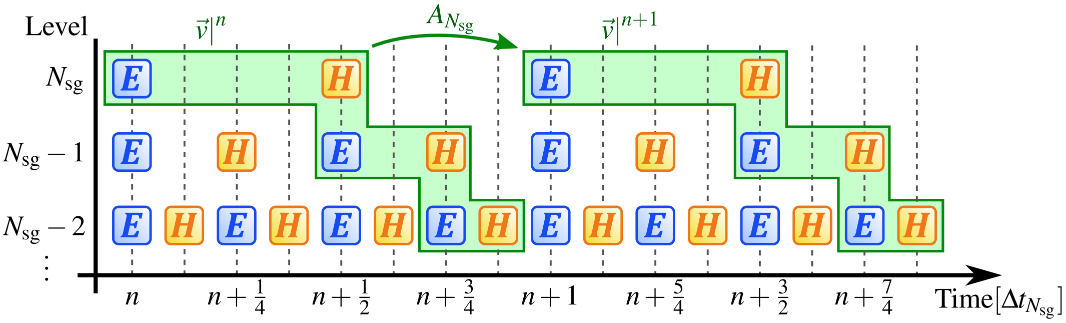

Figure 3.10: Illustration of the effect of the operator $A_{N_{sg}}$ on the stored states of the simulation at a given iteration. $A_{N_{sg}}$ performs an update of $\Delta t_{n_{sg}}$ on all components.

be provided. However, an alternate approach for non-diagonalizable systems is provided at the end of this section.

The spectral approach is backed by a heuristic methodology. It consists of setting a series of arbitrary sources and letting the execution run for up until 10^7 iterations. If, at some point, the energy contained within the simulation has grown by a factor greater than 10^5, we consider the simulation unstable and stop it. This methodology has been found to require much more CPU time, and therefore, it is only employed in one of the simulation cases. An image of this case is depicted in fig. 3.11, and the stability results are shown in fig. 3.12.

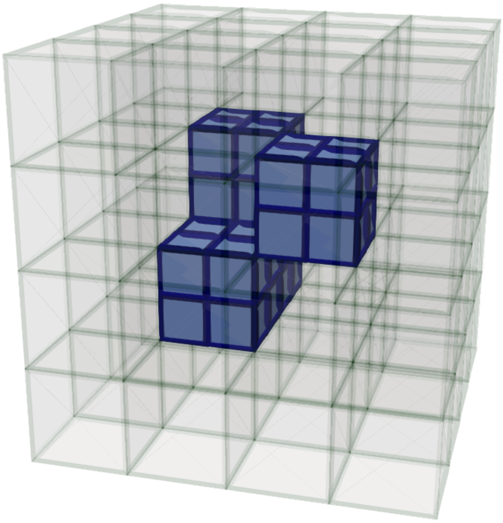

Figure 3.11: Image of a small simulation case used for the heuristic and spectral stability methods. Blue cells represent the fine level and transparent green cells represent the coarse level.

The first interesting thing that we can note from fig. 3.12 is that both the spectral and

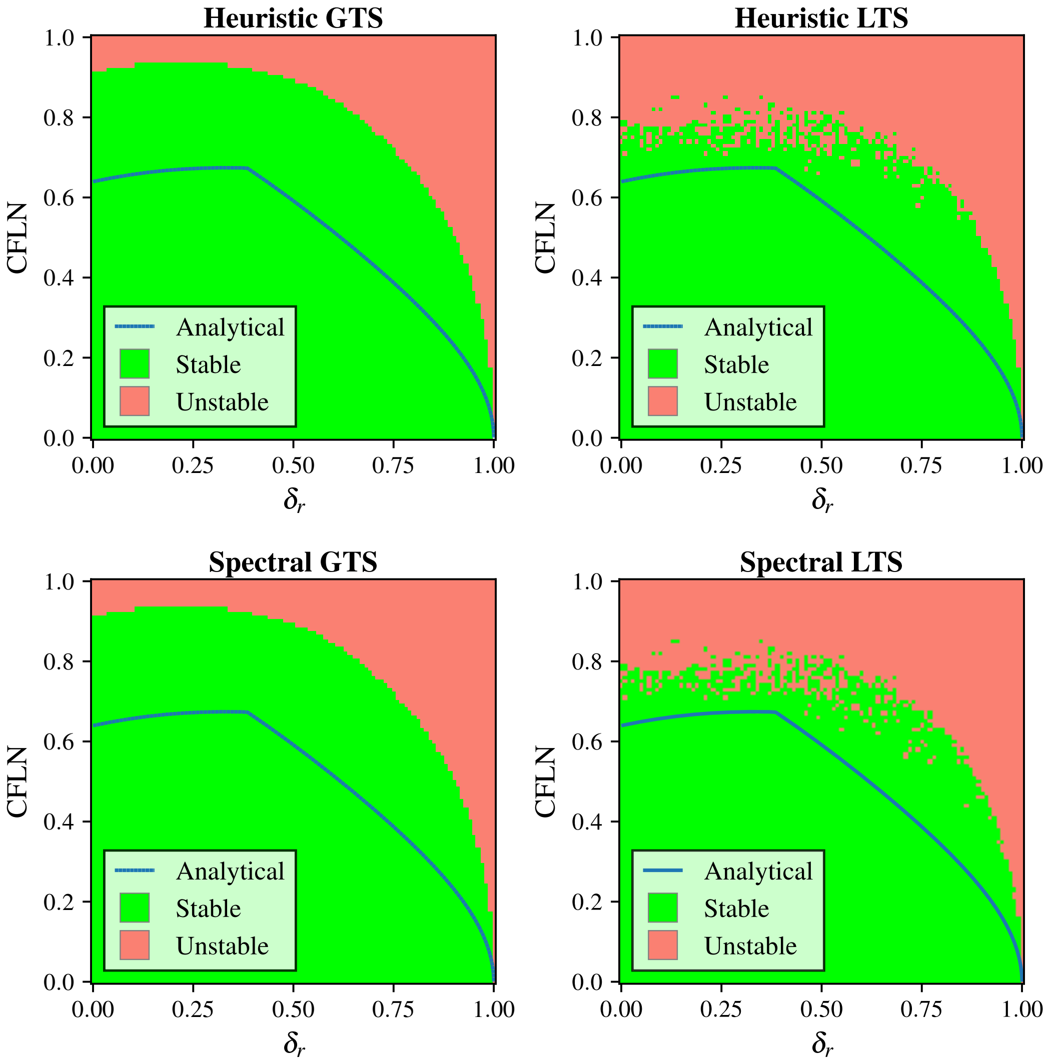

Figure 3.12: Stability of the simulation case depicted in fig. 3.11. Upper plots are heuristic and lower ones are spectral. Left plots employ LTS and right ones employ GTS. In all cases, the analytical expression from eq. (3.43) is plotted for reference. For executions with GTS, CFLN refers to the finest level.

heuristic methods roughly show the same behavior. Second, we see that the boundary between the stable and unstable regions in the case with GTS is very well-defined and smooth, whereas the case with LTS does not show a well-defined boundary. For illustration purposes, the calculated $|\lambda_{\max}|$ is plotted as a function of CFLN for $\delta_\mathrm{r} = 0.33$ in fig. 3.13. Late-time instabilities can be appreciated for the execution with LTS, and additionally, it is clear that $|\lambda_{\max}|$ exhibits an erratic behavior with stable regions surrounded by unstable regions. This is particularly rare for an FDTD submethod, as they typically are either unconditionally unstable, unconditionally stable, or a separation between stability and instability conditions is provided by CFLN. The causes of this behavior in fig. 3.12 for executions with LTS are unknown, but it is likely due to erratic behavior of the late-time instabilities, and therefore, stable regions surrounded by unstable ones for the same δ_r are likely unstable for other simulation cases. To test this, we may perform this same procedure for a set of other, more complex, simulation cases, and overlap the results with the following rule: if a pair $(\delta_\mathrm{r}, \mathrm{CFLN})$ is unstable for at least one case, we consider it to be unstable in general. Results of this methodology are provided in fig. 3.14.

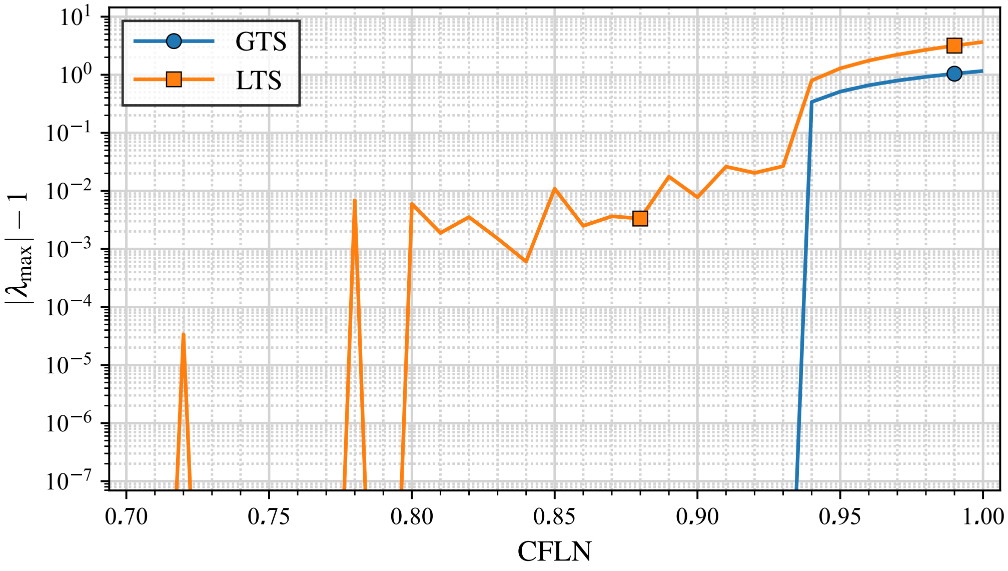

Figure 3.13: Plot of $|\lambda_{\max}| - 1$ as a function of CFLN for $\delta_\mathrm{r} = 0.33$ for the simulation case depicted in fig. 3.11.

By looking at fig. 3.14, we may see that the behavior remains unchanged for executions with GTS: the smooth separation between stable and unstable regions is still present for the same pairs $(\delta_\mathrm{r}, \mathrm{CFLN})$. On the other hand, for the executions with LTS, the overlap of the

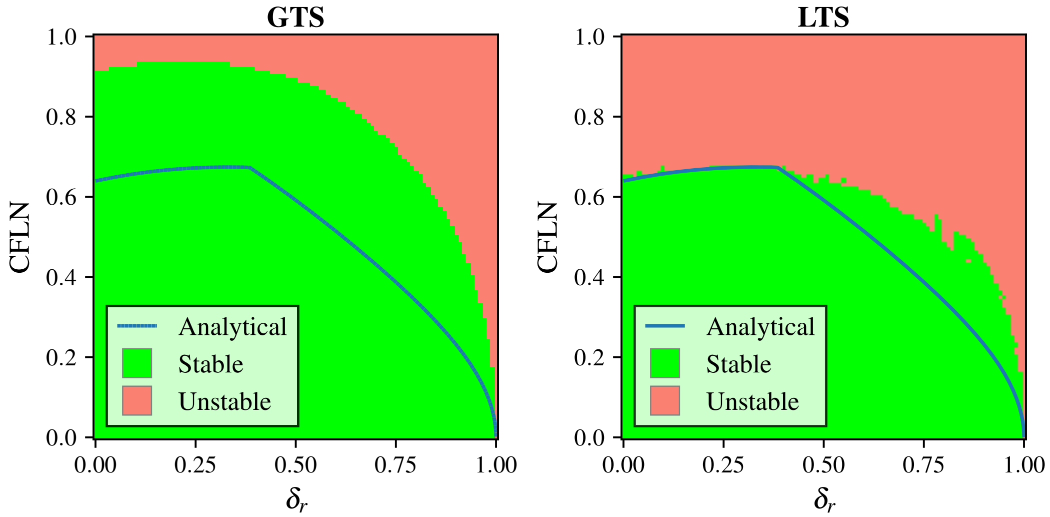

Figure 3.14: Overlapped stability obtained by the spectral method for various simulation cases as a function of δ_r and CFLN, with LTS (left) and GTS (right). The analytical expression from eq. (3.43) is plotted for reference. For executions with GTS, CFLN refers to the finest level.

different cases mostly solves the issue previously appreciated in fig. 3.12.

In both cases, we appreciate that an optimum value of δ_r exists for which the maximum stable CFLN is maximized. According to fig. 3.14, with GTS, the optimum value δ_r is in the interval $[0.11, 0.33]$, with the maximum stable CFLN in $[0.93, 0.94]$ the interval. By repeating the same procedure with steps of 0.001 in CFLN, we find an optimum value $\mathrm{CFLN} = 0.936$ for $\delta_\mathrm{r} = 0.21$. For the executions with LTS, obtaining these optimum values is not trivial because the curve is not well-defined. However, we appreciate that the analytical approach matches the stable-unstable separation for the most part for executions with LTS. Particularly, this is true for the interval $\delta_\mathrm{r} \in [0, 0.4]$, which is also where the stable CFLN is maximized. Thus, for executions with LTS, we may take the analytical result and consider the optimum $\delta_\mathrm{r} = \frac{1}{3}$, with a maximum stable $\mathrm{CFLN} = \frac{2}{3}$.

Possible non-diagonalizability of the update operator

Throughout this section, diagonalizability of A has been assumed, which is a required condition to apply the spectral method above described. However, a formal demonstration of the diagonalizability of A in the OI-SG method cannot be provided in this work. Nonetheless, it is possible to obtain a variant of the spectral methodology to establish the stability of a given update operator A without requiring it to be diagonalizable.

Let us consider yet again the operators M_m as defined in eq. (3.82). As has been already stablished, applying the operator M_m to an initial state $\vec{v}_0$ is equivalent to applying A 2^m times. If we denote Δt to the time step associated to A, and Δt_m to the time step associated to M_m, then we necessarily have $\Delta t_m = 2^m \Delta t$. Now, let us consider an initial state, denoted $\hat{v}_i$, such that every discrete component is zero-valued except for the i-th, which has a value of one. If we apply M_m to this state, the result is the impuse response of the i-th discrete field after 2^m iterations. If we denote $(M_m)_{i',j'}$ to the element of the matrix M_m at the i'-th row and the j'-th column, we have

$$M_m \hat{v}_i = \begin{pmatrix} (M_m)_{1,1} & \dots & (M_m)_{1,i} & \dots & (M_m)_{1,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (M_m)_{N,1} & \dots & (M_m)_{N,i} & \dots & (M_m)_{N,N} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} (M_m)_{1,i} \\ \vdots \\ (M_m)_{N,i} \end{pmatrix} = M_m|_i, \quad (3.99)$$

where N is the dimension of the space of states V and we have denoted $M_m|_i$ to the i-th column of M_m. In other words, eq. (3.99) tells us that applying M_m to $\hat{v}_i$ results in the i-th column of M_m.

If we define $\mathcal{V}$ as the set of all states $\hat{v}_i$, i.e.

$$\mathcal{V} := \left\{ \hat{v}_i \right\}_{i=1}^{N}, \quad (3.100)$$

we may see that $\mathcal{V}$ is trivially a base of V. Therefore, each possible initial state can be written as a linear combination of them. Thus, in this context let us write an arbitrary initial state $\vec{v}_0$ as

$$\vec{v}_0 = \sum_{i=1}^{N} a_i \hat{v}_i, \quad (3.101)$$

where a_i are the components of $\vec{v}_0$ in the base $\mathcal{V}$. The state after 2^m iterations, denoted $\vec{v}|^{2^m}$, can be written as

$$\vec{v}|^{2^m} = \sum_{i=1}^{N} a_i M_m|_i, \quad (3.102)$$

whose norm is

$$\left\| \vec{v}|^{2^m} \right\| = \left\| \sum_{i=1}^{N} a_i M_m|_i \right\| \leq \sum_{i=1}^{N} a_i \left\| M_m|_i \right\|, \quad (3.103)$$

where we have employed the triangle inequality. It is clear that, if none of the terms $||M_m|_i||$ diverges as $m \to +\infty$, then the norm of $\vec{v}|^{2^m}$ cannot diverge either. Therefore, checking stability up to a certain iteration 2^m can be reduced to checking the norm of all the columns of the matrices M_m: if none of the column norms are significantly increasing with m, then the system can be considered stable.

The methodology described allows us to determine whether a certain system A is stable for very late-time iterations. And, more importantly, we do not require A to be diagonalizable, therefore allowing us to catch polynomial divergences such as in the example presented in eq. (3.47). All the stability results presented in this section have been double-checked by this methodology, and the results found are identical.

3.5 Numerical dispersion, convergence and reflection

The dispersion relation for the usual FDTD method (eq. (2.64)) can be easily obtained for a uniform grid, as shown in section 2.1.4 or books such as [17]. This is because this uniformity naturally allows for the propagation of pure sinusoidal waves with specific frequencies. However, when non-uniformity is present, and more particularly, for subgridding, this dispersion relation is not trivial. The waveforms that can propagate in these grids are not pure sinusoidal waves, but rather a composition of several of them. This gets even more complicated when LTS is employed. Nonetheless, in this section, a deeper comprehension of the issues faced when studying this numerical phenomenon is provided. First, a simple 1D case is studied analytically with GTS to extract some information related to the numerical dispersion. From there, the dispersion of the 3D OI-SG method is studied numerically, along with the convergence.

3.5.1 Analytical study of a 1D subgridding case

Let us suppose a 1D FDTD setup with a non-uniform grid extending towards the x-axis, where electric fields are oriented in the y-axis and magnetic fields in the z-axis. In this scenario, discrete positions are denoted x_I, where I is an integer or half-integer. Discrete electric components are located at integer indices and magnetic ones are located at half-integer indices. Therefore, we may write $E_y|_i$ and $H_z|_{i+\frac{1}{2}}$, respectively, as the discrete electric and magnetic components placed at positions x_i and $x_{i+\frac{1}{2}}$, where i is an integer. Additionally, we may define the space steps $\Delta x|_I = x_{I+\frac{1}{2}} - x_{I-\frac{1}{2}}$ and the time step Δt. Electric components exist only at integer multiples of Δt and magnetic components exist only at half-integer multiples, thus we denote $E_y|_i^n$ to the component $E_y|_i$ evaluated at the time $t = n\Delta t$ and $H_z|_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ to the component $H_z|_{i+\frac{1}{2}}$ evaluated at $t = (n+\frac{1}{2})\Delta t$.

Given this configuration, we may write down the 1D FDTD update equations for a general non-uniform grid in free space:

$$E_y|_i^{n+1} = E_y|_i^n + \frac{\Delta t}{\varepsilon_0 \Delta_i}\left(H_z|_{i-\frac{1}{2}}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right), \tag{3.104}$$

$$H_z|_{i+\frac{1}{2}}^{n+\frac{1}{2}} = H_z|_{i+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\Delta t}{\mu_0 \Delta_{i+\frac{1}{2}}}\left(E_y|_i^n - E_y|_{i+1}^n\right), \tag{3.105}$$

To obtain the dispersion relation, we need to consider a monochromatic plane wave solution,

$$E_y|_i^n = E_0 e^{\mathbb{i}(\omega n\Delta t - kx_i)}, \tag{3.106}$$

$$H_z|_i^n = H_0 e^{\mathbb{i}(\omega n\Delta t - kx_i)}, \tag{3.107}$$

where $\mathbb{i}$ is the imaginary unit, ω is the angular frequency, k is the wavenumber, and E_0 and H_0 are the electric and magnetic amplitudes. By plugging eqs. (3.106) and (3.107) into equations eqs. (3.104) and (3.105), respectively, and after some manipulation, we get to the following relations:

$$E_{y,0}\sin\left(\frac{\omega\Delta t}{2}\right) = H_{z,0}\frac{\Delta t}{\varepsilon_0\Delta_i}\sin\left(\frac{k\Delta_i}{2}\right)e^{\mathbb{i}k\frac{\Delta_{(i-\frac{1}{2},i)} - \Delta_{(i,i+\frac{1}{2})}}{2}}, \tag{3.108}$$

$$H_{z,0}\sin\left(\frac{\omega\Delta t}{2}\right) = E_{y,0}\frac{\Delta t}{\mu_0\Delta_{j+\frac{1}{2}}}\sin\left(\frac{k\Delta_{j+\frac{1}{2}}}{2}\right)e^{\mathbb{i}k\frac{\Delta_{(j,j+\frac{1}{2})} - \Delta_{(j+\frac{1}{2},j+1)}}{2}}, \tag{3.109}$$

where we have defined $\Delta_{(i,j)} = x_j - x_i$. This results in a system of linear equations similar to eqs. (2.58)–(2.63). However, in this case, there are as many different equations as discrete electric and magnetic field components. Assuming non-trivial solutions ($E_{y,0} \neq 0$ and $H_{z,0} \neq 0$), multiplying eqs. (3.108) and (3.109) for two disctinct points $(i, j+\frac{1}{2})$ gets us to

$$\begin{aligned}\sin^2\left(\frac{\omega\Delta t}{2}\right) =& \frac{(c\Delta t)^2}{\Delta_i\Delta_{j+\frac{1}{2}}}\sin\left(\frac{k\Delta_i}{2}\right)\sin\left(\frac{k\Delta_{j+\frac{1}{2}}}{2}\right) \\ & e^{\mathbb{i}k\frac{\Delta_{(i-\frac{1}{2},i)} - \Delta_{(i,i+\frac{1}{2})} + \Delta_{(j,j+\frac{1}{2})} - \Delta_{(j+\frac{1}{2},j+1)}}{2}}.\end{aligned} \tag{3.110}$$

It is worth noting that equation eq. (3.110) becomes the uniform FDTD dispersion relation (obtained from eq. (2.64) by assuming $k_y = k_z = 0$) when all space steps are equal. However, it is trivial to see that, in the general non-uniform case, eq. (3.110) provides a different relation for each pair (i, j), and therefore the system is overdetermined. In other words, the fields provided in eqs. (3.106) and (3.107) cannot be a solution of this numerical scheme.

Let us now consider the particular case of a 1D subgridding scheme with a refinement ratio $r = 2$. This means that all space steps are equal until a certain transition index i_t, and half the space step afterward. In mathematical terms, $\Delta_I = \Delta$ if $I < i_t$ and $\Delta_I = \Delta/2$ if $I > i_t$. In the particular case of i_t, we obtain that $\Delta_{i_t} = 3\Delta/4$. This configuration is depicted in fig. 3.15.

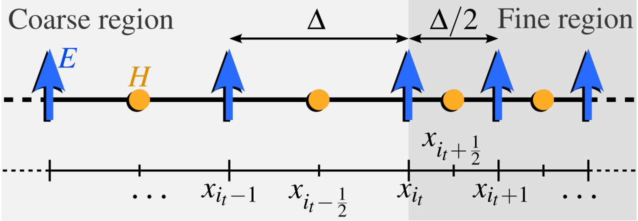

Figure 3.15: Spatial configuration of a 1D subgridding simulation.

For the same reason as in the general non-uniform case, a simple monochromatic wavefront is not a solution for our numerical problem. However, let us now consider a different field with the following form:

$$E_y|_i^n = \begin{cases} E_{\text{inc},0}\, e^{\mathbb{i}(\omega n\Delta t - k_1 x_i)} + E_{\text{ref},0}\, e^{\mathbb{i}(\omega n\Delta t + k_1 x_i)} & \text{if} \quad i < i_t \\ E_{\text{tr},0}\, e^{\mathbb{i}(\omega n\Delta t - k_2 x_i)} & \text{if} \quad i \geq i_t \end{cases}, \tag{3.111}$$

$$H_z|_I^N = \begin{cases} H_{\text{inc},0}\, e^{\mathbb{i}(\omega N\Delta t - k_1 x_I)} + H_{\text{ref},0}\, e^{\mathbb{i}(\omega N\Delta t + k_1 x_I)} & \text{if} \quad i < i_t \\ H_{\text{tr},0}\, e^{\mathbb{i}(\omega N\Delta t - k_2 x_I)} & \text{if} \quad i \geq i_t \end{cases}. \tag{3.112}$$

This is the stationary state of an incident monochromatic wavefront with an electric amplitude of $E_{\text{inc},0}$ that, upon reaching x_{i_t}, transmits a wavefront of amplitude $E_{\text{tr},0}$ and reflects another one of amplitude $E_{\text{ref},0}$. We have also considered different wavenumbers in each region, k_1 in the coarse region and k_2 in the fine one.

By plugging this solution in the finite-differences scheme, and with some manipulation, we get to equations analogous to eq. (3.108) and eq. (3.109):

$$(E_{\text{inc},0} + E_{\text{ref},0}) \sin\left(\frac{\omega \Delta t}{2}\right) = (H_{\text{inc},0} - H_{\text{ref},0}) \frac{\Delta t}{\varepsilon_0 \Delta} \sin\left(\frac{k_1 \Delta}{2}\right), \tag{3.113}$$

$$(H_{\text{inc},0} + H_{\text{ref},0}) \sin\left(\frac{\omega \Delta t}{2}\right) = (E_{\text{inc},0} - E_{\text{ref},0}) \frac{\Delta t}{\mu_0 \Delta} \sin\left(\frac{k_1 \Delta}{2}\right), \tag{3.114}$$

$$E_{\text{tr},0} \sin\left(\frac{\omega \Delta t}{2}\right) = H_{\text{tr},0} \frac{\Delta t}{\varepsilon_0 \Delta/2} \sin\left(\frac{k_2 \Delta/2}{2}\right), \tag{3.115}$$

$$H_{\text{tr},0} \sin\left(\frac{\omega \Delta t}{2}\right) = E_{\text{tr},0} \frac{\Delta t}{\mu_0 \Delta/2} \sin\left(\frac{k_2 \Delta/2}{2}\right). \tag{3.116}$$

It is worth noting that equation eq. (3.113) is obtained from any field E_i such that $i < i_t$, and analogously eq. (3.115) is obtained if $i > i_t$. If we repeat this process for E_{i_t} we get:

$$E_{\text{tr},0}\, 2\mathbb{i} \sin\left(\frac{\omega \Delta t}{2}\right) = \frac{\Delta t}{\varepsilon_0\, 3\Delta/4} \left(H_{\text{inc},0}\, e^{\mathbb{i} k_1 \frac{\Delta}{2}} + H_{\text{ref},0}\, e^{-\mathbb{i} k_1 \frac{\Delta}{2}} - H_{\text{tr},0}\, e^{-\mathbb{i} k_2 \frac{\Delta}{4}}\right), \tag{3.117}$$

Additionally, eq. (3.114) and eq. (3.116) require special attention for the fields $H_{i_t - \frac{1}{2}}$ and $H_{i_t + \frac{1}{2}}$, respectively. In both cases, we have assumed that the field E_{i_t} matches the same waveform as the electric fields in their respective regions. This can be ensured by imposing continuity in E_{i_t}. For simplicity, we assume $x_{i_t} = 0$, which leads us to:

$$E_{\text{inc},0} + E_{\text{ref},0} = E_{\text{tr},0}. \tag{3.118}$$

At this point, eqs. (3.113)–(3.118) form a linear system composed of 6 equations with 6 unknowns, meaning that non-trivial solutions can be found. From eqs. (3.113) and (3.114) we may get the dispersion relation for the coarse region, and similarly, from eqs. (3.115) and (3.116), we get the dispersion relation for the fine region:

$$\sin^2\left(\frac{\omega \Delta t}{2}\right) = \left(\frac{c_0 \Delta t}{\Delta}\right)^2 \sin^2\left(\frac{k_1 \Delta}{2}\right), \tag{3.119}$$

$$\sin^2\left(\frac{\omega \Delta t}{2}\right) = \left(\frac{c_0 \Delta t}{\Delta/2}\right)^2 \sin^2\left(\frac{k_2 \Delta/2}{2}\right). \tag{3.120}$$

Using this in eqs. (3.113) and (3.115) gives us the numerical impedances that relate the electric and magnetic amplitudes by the vacuum impedance (eq. (2.39)):

$$E_{\mathrm{inc},0} = \eta_0 H_{\mathrm{inc},0}\,, \tag{3.121}$$

$$E_{\mathrm{ref},0} = -\eta_0 H_{\mathrm{ref},0}\,, \tag{3.122}$$

$$E_{\mathrm{tr}\,,0} = \eta_0 H_{\mathrm{tr}\,,0}\,. \tag{3.123}$$

Using eqs. (3.121)–(3.123) and eq. (3.118) in eq. (3.117), and after some algebra, we get to

$$\frac{H_{\mathrm{inc},0} + H_{\mathrm{ref},0}}{H_{\mathrm{tr},0}} = \frac{\cos\left(\frac{k_2\Delta/2}{2}\right)}{\cos\left(\frac{k_1\Delta}{2}\right)}, \tag{3.124}$$

which, in combination with eq. (3.118) and eqs. (3.121)–(3.123), finally gets us to

$$R := \frac{E_{\mathrm{ref},0}}{E_{\mathrm{inc},0}} = \frac{\cos\left(\frac{k_1\Delta}{2}\right) - \cos\left(\frac{k_2\Delta/2}{2}\right)}{\cos\left(\frac{k_1\Delta}{2}\right) + \cos\left(\frac{k_2\Delta/2}{2}\right)} \tag{3.125}$$

$$T := \frac{E_{\mathrm{tr},0}}{E_{\mathrm{inc},0}} = \frac{2\cos\left(\frac{k_2\Delta/2}{2}\right)}{\cos\left(\frac{k_1\Delta}{2}\right) + \cos\left(\frac{k_2\Delta/2}{2}\right)} = 1 - R, \tag{3.126}$$

where R and T are, respectively, the numerical reflection and transmission coefficients.

We may now define the Points per Wavelength in the coarse and fine regions, respectively, as PPW_1 and PPW_2, which are adimensional magnitudes that tell us the electric size of the grids,

$$\mathrm{PPW}_1 := \frac{\lambda_1}{\Delta} = \frac{2\pi}{k_1\Delta}, \tag{3.127}$$

$$\mathrm{PPW}_2 := \frac{\lambda_2}{\Delta/2} = \frac{2\pi}{k_2\Delta/2}. \tag{3.128}$$

From eqs. (3.119) and (3.120), we may obtain the relation between them:

$$\sin\left(\frac{\pi}{\mathrm{PPW}_1}\right) = 2\sin\left(\frac{\pi}{\mathrm{PPW}_2}\right) \tag{3.129}$$

This allows us to re-write eqs. (3.125) and (3.126) as a function of these adimensional magnitudes:

$$R = \frac{\cos\left(\dfrac{\pi}{\mathrm{PPW}_1}\right) - \cos\left(\dfrac{\pi}{\mathrm{PPW}_2}\right)}{\cos\left(\dfrac{\pi}{\mathrm{PPW}_1}\right) + \cos\left(\dfrac{\pi}{\mathrm{PPW}_2}\right)} \tag{3.130}$$

$$T = \frac{2\cos\left(\dfrac{\pi}{\mathrm{PPW}_2}\right)}{\cos\left(\dfrac{\pi}{\mathrm{PPW}_1}\right) + \cos\left(\dfrac{\pi}{\mathrm{PPW}_2}\right)}, \tag{3.131}$$

We may define a discretization parameter h in the same way as in eq. (2.87) to assess the order of convergence,

$$h = \frac{1}{\mathrm{PPW}_1}. \tag{3.132}$$

If we write the first term of the Taylor polynomial of R as a function of h, we obtain

$$R(h) = \frac{3\pi^2}{16}h^2 + O(h^4) \tag{3.133}$$

Analytical study of a 1D case with LTS

Let us suppose the same case with LTS. Thus, maintaining the same value of $\mathrm{CFLN}_{\mathrm{1D}}$ in both regions, let us refer to the time step in the coarse region as Δt and the time step in the fine region as $\Delta t/2$. We also define $E_y|_i^N$ as the field $E_y|_i$ evaluated at time $t = N\Delta t$, where i is an integer, and N is an integer if $i < i_t$, or can be an integer or half-integer if $i \geq i_t$. Similarly, we define $H_z|_I^N$ as the field $H_z|_I$ evaluated at time $t = N\Delta t$, where I is a half-integer, and N is a half-integer if $I < i_t$, or a quarter or three quarters of integer if $I > i_t$. Following the same LTS scheme as in fig. 3.5, the update equations can be written normally for all discrete electromagnetic components except for $E_y|_{i_t}$. For the coarse region, we have

$$E_y|_i^{n+1} = E_y|_i^n + \frac{\Delta t}{\varepsilon_0 \Delta}\left(H_z|_{i-\frac{1}{2}}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) \qquad \text{if} \quad i < i_t, \tag{3.134}$$

$$H_z|_{i+\frac{1}{2}}^{n+\frac{1}{2}} = H_z|_{i+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\Delta t}{\mu_0 \Delta}\left(E_y|_i^n - E_y|_{i+1}^n\right) \qquad \text{if} \quad i < i_t, \tag{3.135}$$

where n must be an integer. For the fine region, we have

$$E_y\Big|_i^{N+\frac{1}{2}} = E_y\Big|_i^{N} + \frac{\Delta t}{\varepsilon_0 \Delta}\left(H_z\Big|_{i-\frac{1}{2}}^{N+\frac{1}{2}} - H_z\Big|_{i+\frac{1}{2}}^{N+\frac{1}{2}}\right) \qquad \text{if} \quad i > i_t, \tag{3.136}$$

$$H_z\Big|_{i+\frac{1}{2}}^{N+\frac{1}{4}} = H_z\Big|_{i+\frac{1}{2}}^{N-\frac{1}{4}} + \frac{\Delta t}{\mu_0 \Delta}\left(E_y\Big|_i^{N} - E_y\Big|_{i+1}^{N}\right) \qquad \text{if} \quad i \geq i_t, \tag{3.137}$$

where N must be an integer or half-integer. On the other hand, the update equation for $E_y|_{i_t}$ gets split into two: one for the case when the new field is contemporary to the coarse electric components ($t = n\Delta t$), and another one for the case when it is contemporary to the coarse magnetic components ($t = (n+\frac{1}{2})\,\Delta t$). These equations are written as

$$E_y\Big|_{i_t}^{n+\frac{1}{2}} = E_y\Big|_{i_t}^{n} + \frac{\Delta t}{\varepsilon_0\, 3\Delta/2}\left(H_z\Big|_{i_t-\frac{1}{2}}^{n+\frac{1}{2}} - H_z\Big|_{i_t+\frac{1}{2}}^{n+\frac{1}{4}}\right) \tag{3.138}$$

$$E_y\Big|_{i_t}^{n+1} = E_y\Big|_{i_t}^{n+\frac{1}{2}} + \frac{\Delta t}{\varepsilon_0\, 3\Delta/2}\left(H_z\Big|_{i_t-\frac{1}{2}}^{n+\frac{1}{2}} - H_z\Big|_{i_t+\frac{1}{2}}^{n+\frac{3}{4}}\right), \tag{3.139}$$

where n must be an integer.

We already know that monochromatic waves propagate naturally in both regions since they consist of uniform FDTD grids. This means that, yet again, the expected stationary solution of an incident wave must be the same as written in eqs. (3.111) and (3.112). Once again we need to assume continuity (eq. (3.118)), which allows us to substitute eqs. (3.111) and (3.112) into eqs. (3.134)–(3.137) to obtain the dispersion relations in both regions,

$$\sin^2\left(\frac{\omega\,\Delta t}{2}\right) = \left(\frac{c_0\,\Delta t}{\Delta}\right)^2 \sin^2\left(\frac{k_1\,\Delta}{2}\right), \tag{3.140}$$

$$\sin^2\left(\frac{\omega\,\Delta t/2}{2}\right) = \left(\frac{c_0\,\Delta t}{\Delta}\right)^2 \sin^2\left(\frac{k_2\,\Delta/2}{2}\right), \tag{3.141}$$

and the same impedance relations as in eqs. (3.121)–(3.123). Now, we may substitute eqs. (3.111) and (3.112) into eqs. (3.138) and (3.139) to get two expressions similar to eq. (3.117):

$$E_{\mathrm{tr}\,,0}\, 2\mathbb{i} \sin\left(\frac{\omega\,\Delta t/2}{2}\right) = \frac{\Delta t}{\varepsilon_0\, 3\Delta/2}\left[\begin{array}{l} \left(H_{\mathrm{inc},0}\, e^{\mathbb{i}\,\frac{k_1 \Delta}{2}} + H_{\mathrm{ref},0}\, e^{-\mathbb{i}\,\frac{k_1 \Delta}{2}}\right) e^{\mathbb{i}\,\frac{\omega\Delta t/2}{2}} \\ -H_{\mathrm{tr}\,,0}\, e^{-\mathbb{i}\,\frac{k_2 \Delta/2}{2}} \end{array}\right], \tag{3.142}$$

$$E_{\mathrm{tr}\,,0}\, 2\mathbb{i} \sin\left(\frac{\omega\,\Delta t/2}{2}\right) = \frac{\Delta t}{\varepsilon_0\, 3\Delta/2} \left[\begin{array}{l} \left(H_{\mathrm{inc},0} e^{\mathbb{i}\frac{k_1\Delta}{2}} + H_{\mathrm{ref},0} e^{-\mathbb{i}\frac{k_1\Delta}{2}}\right) e^{-\mathbb{i}\frac{\omega\Delta t/2}{2}} \\ -H_{\mathrm{tr}\,,0} e^{-\mathbb{i}\frac{k_2\Delta/2}{2}} \end{array} \right]. \tag{3.143}$$

Substracting eqs. (3.142) and (3.143) lead to

$$\left(H_{\mathrm{inc},0} e^{\mathbb{i}\frac{k_1\Delta}{2}} + H_{\mathrm{ref},0} e^{-\mathbb{i}\frac{k_1\Delta}{2}}\right) \sin\left(\frac{\omega\,\Delta t/2}{2}\right) = 0. \tag{3.144}$$

To fulfill eq. (3.144), we require either of the following conditions to be true:

$$\frac{H_{\mathrm{ref},0}}{H_{\mathrm{inc},0}} = -e^{\mathbb{i}\, k_1\,\Delta}, \tag{3.145}$$

$$\omega\,\Delta t = 4m\pi, \quad m \in \mathbb{Z}. \tag{3.146}$$

However, neither condition can be true. Eq. (3.145) cannot be true because upon substitution in either eqs. (3.142) or (3.143) we get to an expression that contradicts the impedance condition eq. (3.123). Eq. (3.146) cannot be true because it either requires $\Delta t = 0$ or breaks the Nyquist-Shannon sampling theorem.

At this point, we have proven analytically that complex sinusoidal monochromatic waves cannot propagate when LTS is used, strictly speaking. However, it is interesting to see what happens in the limit $\omega\Delta t \to 0$. In this case, both eqs. (3.142) and (3.143) can be approximated to the same equation:

$$E_{\mathrm{tr}\,,0}\, 2\mathbb{i} \frac{\omega\,\Delta t/2}{2} = \frac{\Delta t}{\varepsilon_0\, 3\Delta/2} \left(H_{\mathrm{inc},0} e^{\mathbb{i}\frac{k_1\Delta}{2}} + H_{\mathrm{ref},0} e^{-\mathbb{i}\frac{k_1\Delta}{2}} - H_{\mathrm{tr}\,,0} e^{-\mathbb{i}\frac{k_2\Delta/2}{2}}\right). \tag{3.147}$$

Finally, by using the impedance relations eqs. (3.121)–(3.123), the continuity condition from eq. (3.118), and the dispersion relations eqs. (3.119) and (3.120) into eq. (3.117), and following an analogous procedure as in the case with LTS, we get to expressions identical to the reflection and transmission coefficients in eqs. (3.125) and (3.126). However, in this case, the dependence between PPW_1 and PPW_2 gets more complex, involving the one-dimensional CFL number, $\mathrm{CFLN}_{1\mathrm{D}}$,

$$\arcsin\left[\mathrm{CFLN} \sin\left(\frac{\pi}{\mathrm{PPW}_1}\right)\right] = 2 \arcsin\left[\mathrm{CFLN} \sin\left(\frac{\pi}{\mathrm{PPW}_2}\right)\right], \tag{3.148}$$

where

$$\mathrm{CFLN}_{1\mathrm{D}} := \frac{c_0\,\Delta t}{\Delta}. \tag{3.149}$$

The reflection coefficient R is plotted in fig. 3.16 as a function of h with LTS and with GTS, and for different values of $\mathrm{CFLN}_{1\mathrm{D}}$. Interestingly, reflection in all cases has the same order of convergence, $O(h^2)$, and they are very similar.

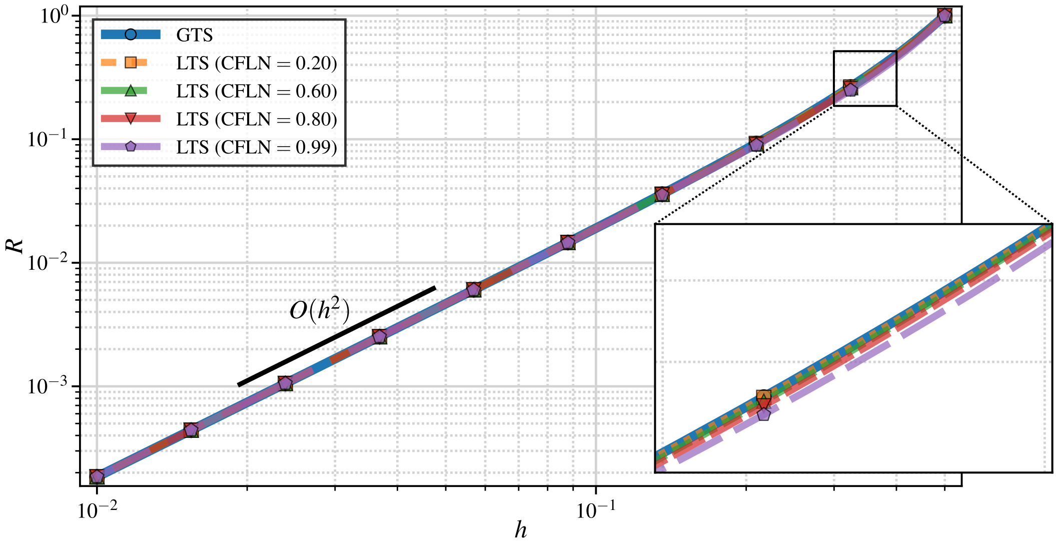

Figure 3.16: Numerical reflection of a 1D subgridding case with LTS and with GTS.

Two main conclusions are meant to be extracted from this analytical study. First, the dispersion relation in a uniform grid may be particularly simple due to the uniformity, but non-uniformity induces more complex numerical phenomena that involve numerical reflections, even in a simple 1D case. Second, when LTS is introduced, monochromatic traveling waves are strictly not a solution of this scheme, and only for low values of PPW one may expect to see a convergence to the analytical solution. In the 3D case, this kind of analytical study is much more complex, and more particularly the non-trivial fields introduced by the OI-SG scheme make it not feasible. Nonetheless, a numerical study of dispersion, convergence, and reflection is done in what follows.

3.5.2 Dispersion and convergence in 3D subgridding

Analytical studies of numerical dispersion typically consider a complex sinusoidal monochromatic wave such as the one written in eqs. (2.52)–(2.57). Afterward, a system of equations is obtained that, to find non-trivial solutions, requires satisfying a certain condition between the numerical frequency ω, the numerical wavenumber vector $\vec{k}$, and the spatiotemporal steps, thus providing the dispersion relation. Let us denote a generic discrete field component in 3D evaluated at the time $t = n\Delta t$ as $U|_i^n$. For a monochromatic traveling wave, we would have

$$U|_i^n = U_0\, e^{\mathbb{i}\left(\omega n \Delta t - \vec{k}\cdot\vec{r}_i\right)}, \tag{3.150}$$

where U_0 is the wave amplitude and $\vec{r}_i$ is the physical position of $U|_i$ in space. However, in a more general case, a sinusoidal wave might not necessarily be a solution to our system, and therefore we require thinking of a more general approach involving a spectral analysis of the system.

As presented in section 3.4.2, the OI-SG equations can always be written as an LTI system by defining a matrix A that is applied recursively on the numerical state. Regardless of whether GTS or LTS is used, A always consists of an advancement of a certain time-step for all discrete components. Let us denote it Δt. Now, let us assume an eigenvalue of A and denote it λ, and one associated eigenvector $\hat{e}$,

$$A\,\hat{e} = \lambda\,\hat{e}. \tag{3.151}$$

If A is stable and non-dissipative, then we can find a λ such that $|\lambda| = 1$, and therefore we may write it as

$$\lambda = e^{\mathbb{i}\,\theta}. \tag{3.152}$$

It is easy to see that, if our system allowed for the propagation of monochromatic sinusoidal waves, then eq. (3.150) would be not just a solution, but an eigenstate of A, since

$$U|_i^{n+1} = U_0\, e^{\mathbb{i}\left(\omega\,(n+1)\,\Delta t - \vec{k}\cdot\vec{r}_i\right)} = e^{\mathbb{i}\,\omega\,\Delta t}\, U|_i^n, \tag{3.153}$$

and thus its eigenvalue would be

$$\lambda = e^{\mathbb{i}\,\omega\,\Delta t}. \tag{3.154}$$

As seen in the previous section, sinusoidal monochromatic waves are not always necessarily a solution for our numerical system. However, analytically, Maxwell's equations allow their propagation, and therefore any numerical system should be able to, at least, approximate such waves. Furthermore, in the asymptotic limit when the discretization length goes to zero, we should be able to find an eigenstate that converges to the sinusoidal monochromatic wave.

Let us now suppose a 3D computational domain with lengths L_x, L_y, and L_z in each respective Cartesian axis, such as depicted in fig. 3.17. Let us suppose that this domain has periodic boundary conditions in all directions. This, analytically, would imply that the only monochromatic modes that can exist are harmonic modes of the box, which propagate in one Cartesian direction and must fit exactly a whole number of wavelengths. We denote as $\{\alpha, m\}$ to the m-th mode in the α-axis, where $\alpha \in \{x, y, z\}$ and m is an integer. The associated angular frequency is denoted $\omega_{\{\alpha,m\}}$, and the wavenumber is denoted $k_{\{\alpha,m\}}$, where

$$k_{\{\alpha,m\}} = \frac{2m\pi}{L_\alpha}, \tag{3.155}$$

and, analytically, we have

$$\omega_{\{\alpha,m\}} = c_0 k_{\{\alpha,m\}}. \tag{3.156}$$

The associated wave can be analytically written as

$$U_{\{\alpha,m\}}(\vec{r},t) = U_0 e^{\mathbb{i}\left(\omega_{\{\alpha,m\}} t + k_{\{\alpha,m\}} \alpha\right)}, \tag{3.157}$$

which is depicted in fig. 3.17.

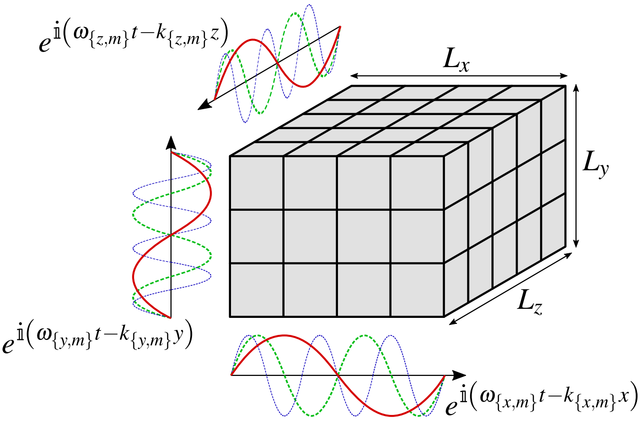
$e^{\mathbb{i}\left(\omega_{\{z,m\}}t-k_{\{z,m\}}z\right)}$
L_x
L_y
L_z
$e^{\mathbb{i}\left(\omega_{\{y,m\}}t-k_{\{y,m\}}y\right)}$
$e^{\mathbb{i}\left(\omega_{\{x,m\}}t-k_{\{x,m\}}x\right)}$

Figure 3.17: Computational box with periodic boundary conditions in all directions. The analytical harmonic modes are illustrated.

Numerically, we cannot assume that the harmonic modes propagate within the described domain. However, given a specific analytical harmonic mode, we should be able to find an eigenstate that converges to it if we decrease the space step while maintaining the physical limits of the domain. For this purpose, a very simple simulation case has been created (depicted in fig. 3.18), which has a coarse cell length of $\Delta = 2\,\mathrm{m}$, a long side in the x-axis of length $L = 6\,\mathrm{m}$ and two short sides in the y- and z-axes. Taking this case as a seed, we create multiple copies of it, keeping the same physical limits but decreasing the cell lengths by a factor d. For each case, the evolution matrix A and all its eigenstates are obtained. Then, we choose the eigenstate that best fits a sinusoidal wave directed towards the x-axis of the simulation with a wavelength equal to L. In other words, we obtain the eigenvalue that best matches the harmonic mode $\{x, 1\}$, also known as the fundamental mode, although it is worth noting that this process can be done for any other harmonic mode. A plot of some of the obtained eigenstates with LTS is shown in fig. 3.19 and with GTS in fig. 3.20. Interestingly, we may see that the fundamental harmonic modes fit almost perfectly the analytical wave.

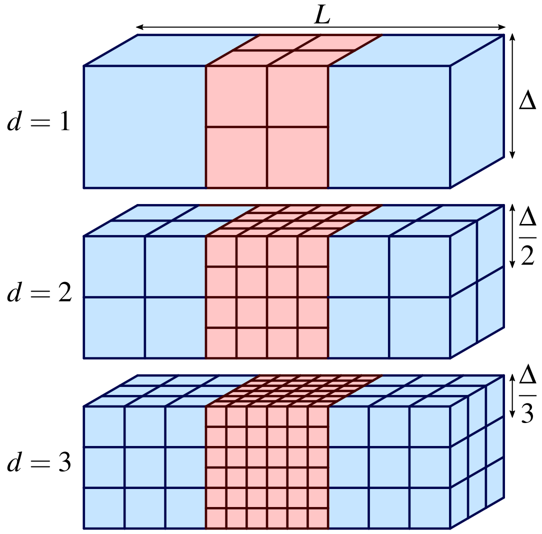

Figure 3.18: Illustration of the simulation cases designed to obtain the eigenstates of the harmonic modes. Each case has an associated magnitude $d \in \mathbb{N}$ and a coarse cell length of Δ/d.

The next part of the procedure is based on the work presented in [169, Ch. 4]. Once the best-match eigenstate is obtained, we may calculate the numerical frequency $\tilde{\omega}$ associated to it from its corresponding eigenvalue λ,

$$\tilde{\omega} = -\mathbb{i}\frac{\ln(\lambda)}{\Delta t}. \tag{3.158}$$

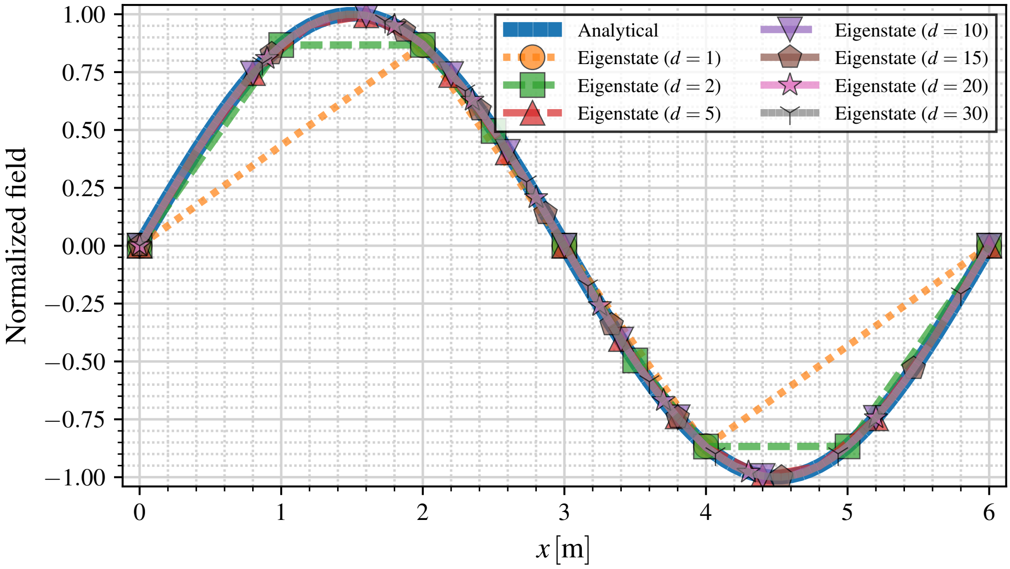

Figure 3.19: Spatial plot of the eigenstates that best match the harmonic mode $\{x, 1\}$ for the case depicted in fig. 3.18 for different values of d with LTS.

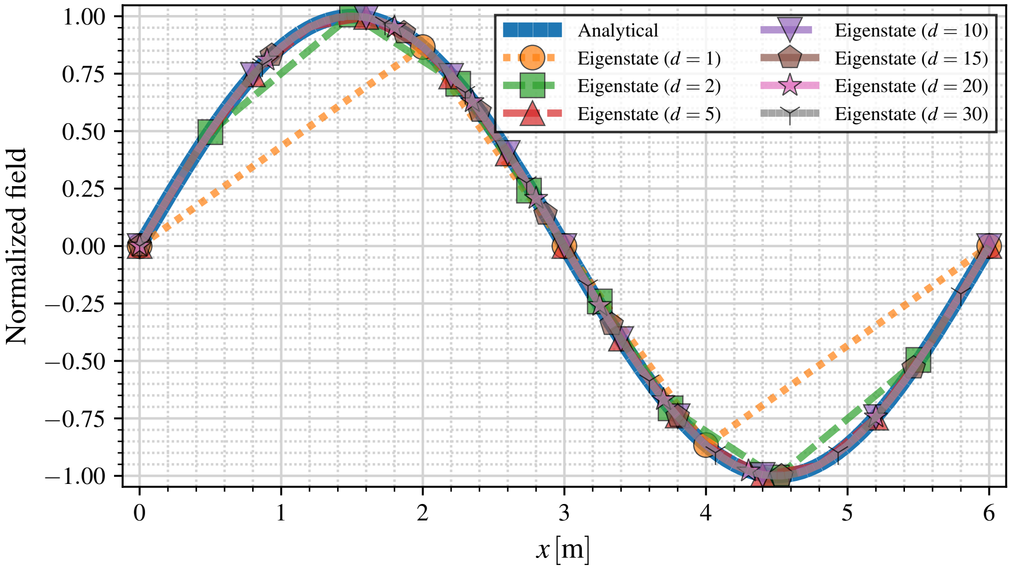

Figure 3.20: Spatial plot of the eigenstates that best match the harmonic mode $\{x, 1\}$ for the case depicted in fig. 3.18 for different values of d with GTS.

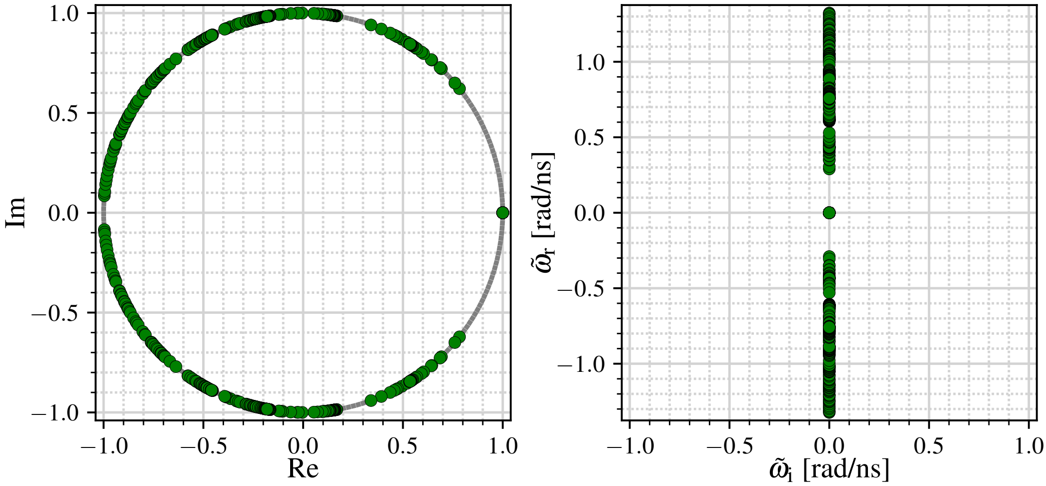

Figure 3.21: Example of the eigenvalues obtained for an evolution matrix (left) and the associated complex frequencies (right).

Analytically, the expected frequency ω would be

$$\omega = \frac{2\pi c_0}{L}, \tag{3.159}$$

and thus we may now calculate the error of $\tilde{\omega}$. However, we must bear in mind that $\tilde{\omega}$ is, in general, a complex number. Let us denote $\tilde{\omega}_\mathrm{r}$ and $\tilde{\omega}_\mathrm{i}$ to the real and imaginary parts of $\tilde{\omega}$. Thus, the following errors are defined:

$$\mathrm{Err}_\mathrm{real} = \left|\frac{\tilde{\omega}_\mathrm{r} - \omega}{\omega}\right|, \tag{3.160}$$

$$\mathrm{Err}_\mathrm{imag} = \left|\frac{\tilde{\omega}_\mathrm{i}}{\omega}\right|. \tag{3.161}$$

Additionally, we define a discretization parameter h as

$$h := \frac{\Delta}{2dL} = \frac{1}{\mathrm{PPW}_\mathrm{fine}}, \tag{3.162}$$

where $\mathrm{PPW}_\mathrm{fine}$ is the Points Per Wavelength in the fine grid. An example of the calculated eigenvalues and complex frequencies is depicted in fig. 3.21. The errors from eqs. (3.160) and (3.161) are plotted as a function of h in fig. 3.22.

By looking at fig. 3.22, we may draw several interesting conclusions. First, the error on the imaginary part is negligible in all cases, meaning that $\tilde{\omega}$ is almost real. By looking at

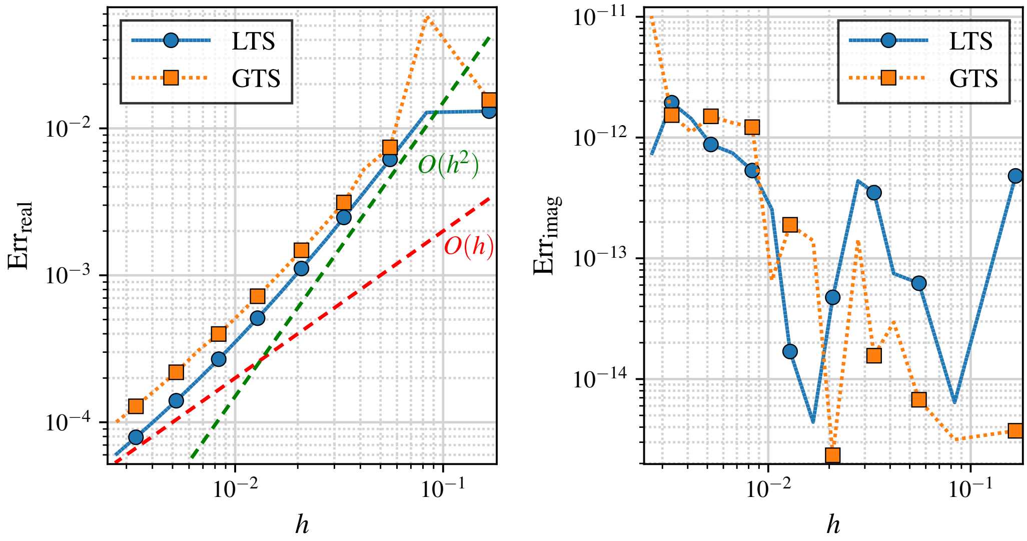

Figure 3.22: Numerical error of the angular frequency obtained from the eigenstates that best match the fundamental harmonic mode $\{x, 1\}$ in the case from fig. 3.18.

eq. (3.158), it is easy to see that $\tilde{\omega}_{\mathrm{i}}$ is directly related to the magnitude of the eigenvalue as

$$\tilde{\omega}_{\mathrm{i}} = -\frac{\ln|\lambda|}{\Delta t}. \tag{3.163}$$

Therefore, the first conclusion that we draw is that $|\lambda| \simeq 1$, meaning that the obtained mode is stable and non-dissipative. A second conclusion that can be drawn is that, in all cases, $\tilde{\omega}$ converges to ω with order $O(h^2)$ for large values of h that decays to $O(h)$ for lower values. It is unclear whether this tendency is maintained for lower values of h. This could be solved by providing more data points, however, the computational time required to calculate the eigenvalues exceeded the available capabilities for this work.

3.5.3 Numerical reflection in 3D subgridding

In section 3.5.1, numerical reflection is obtained analytically for the 1D subgridding case. In this section, it is complemented by a numerical study on the 3D reflection. To do so, a family of simulation cases has been arranged as depicted in fig. 3.23. Each simulation has $N_{\mathrm{sg}} + 1$ subgridding levels, where 0 is the finest one and N_{sg} is the coarsest, and the computational domain has a length of one coarse cell in the y- and z-axis, whereas the x-axis is much longer. A plane wave source is placed near the lower-x boundary directed toward the positive x-axis in such a way that it enters and then exits several nested refined regions.

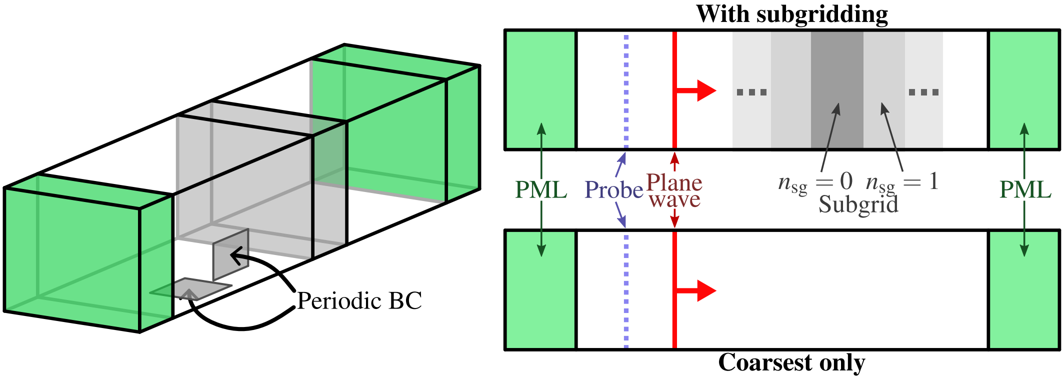

Figure 3.23: Illustration of the simulation setup used to measure the numerical reflection of the OI-SG method. The number of subgridding levels and their respective depths vary between simulations. In each case, two simulations are performed, one with the subgridding region and the other one with the coarsest level only. The resulting fields are subtracted to remove numerical artifacts due to plane wave insertion and the PML.

Periodic boundary conditions are used in the y and z limits to simulate an indefinite subgridding boundary, and PML with 10 layers are used in the x limits to absorb the traveling plane wave and its reflection. A field probe is placed behind the plane wave source to measure the reflection produced by the refined region. In normal circumstances in which we were interested in measuring the reflection of a given object, this setup would be sufficient. However, the reflection of the subgridding boundary is expected to be exceptionally low because it should converge to zero. Numerical backpropagation of the plane wave source or spurious reflections of the PML boundary conditions would be negligible in other circumstances, however, in this scenario, they can affect the measurement. For this reason, a separate simulation is run for each case containing only the coarse level. Due to the linearity of the FDTD and the OI-SG methods, we may subtract the results from both simulations to isolate the effects due to subgridding.

Simulations have been performed for $N_{\text{sg}} \in 1,2,3,4$, and the depth of the subgridded regions have been varied between 1 and 1000 cells. In all cases, we used the same finest-level spatial step of $\Delta_0 = 1\,\text{m}$. The temporal pulse shape for the plane wave is a Gaussian with a decay of 3 dB on the frequency whose PPW corresponding to the coarsest level ($\text{PPW}_{\text{coarse}} = \lambda/\Delta_{N_{\text{sg}}}$) is $\text{PPW}_{\text{coarse}} = 5$. The discretization parameter defined for this case is $h = 1/\text{PPW}_{\text{coarse}}$ Results for the different values of N_{sg} with LTS are shown in fig. 3.24, and with GTS in fig. 3.25. Resonant peaks can be appreciated in all cases, but all reflection curves follow a general envelope that marks the reflection tendencies. These envelopes are plotted in fig. 3.26 and compared against the analytical reflection of the 1D subgridding case from eq. (3.125). We may observe that the convergence order drops from $O(h^2)$ to $O(h)$ and, interestingly, executions with GTS provide appreciably larger reflection.

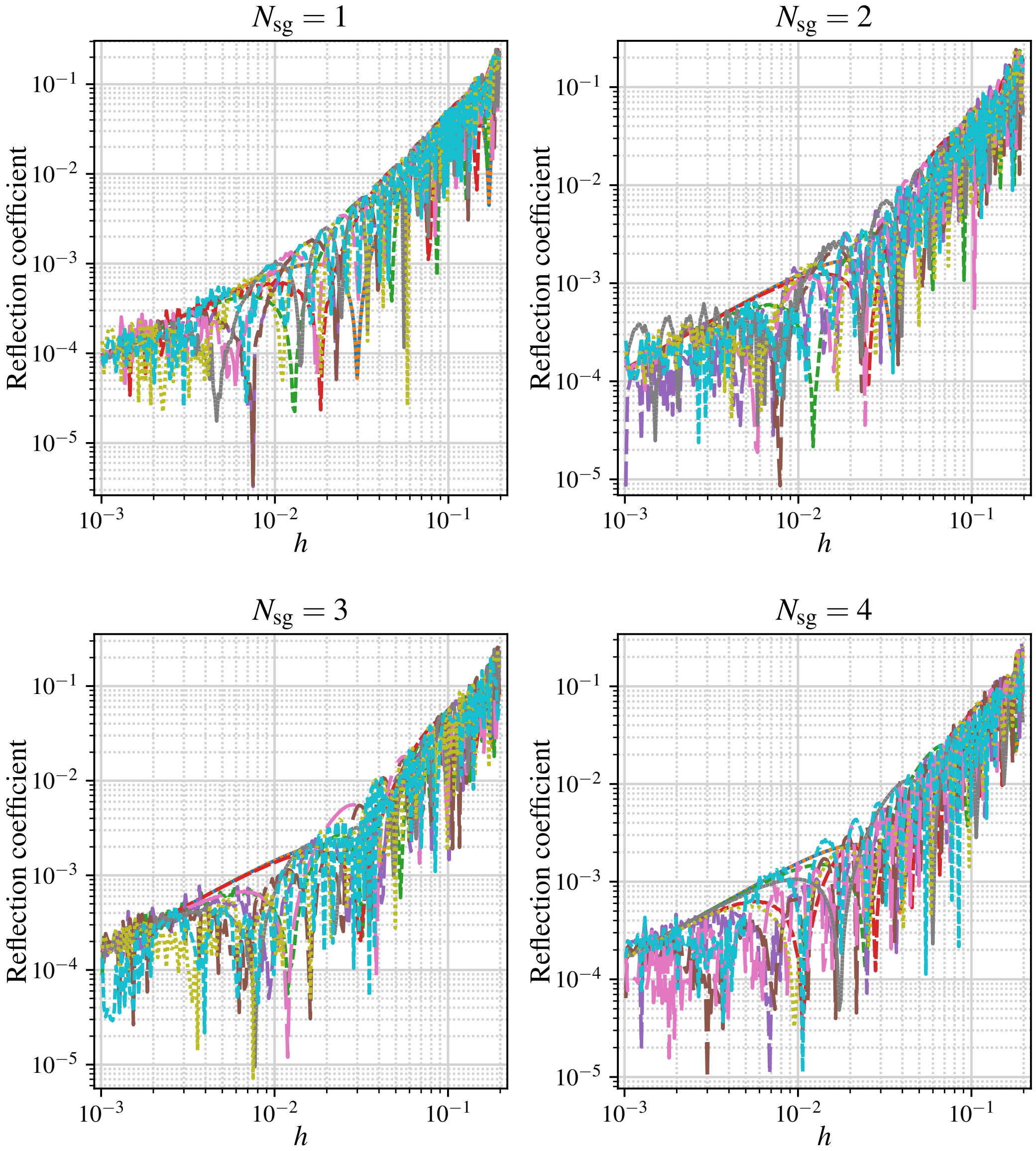

Figure 3.24: Reflection coefficient of the subgridding boundaries with LTS for $N_{sg} \in \{1,2,3,4\}$. Simulations have been performed with subgridded region depths between 1 and 1000 cells.

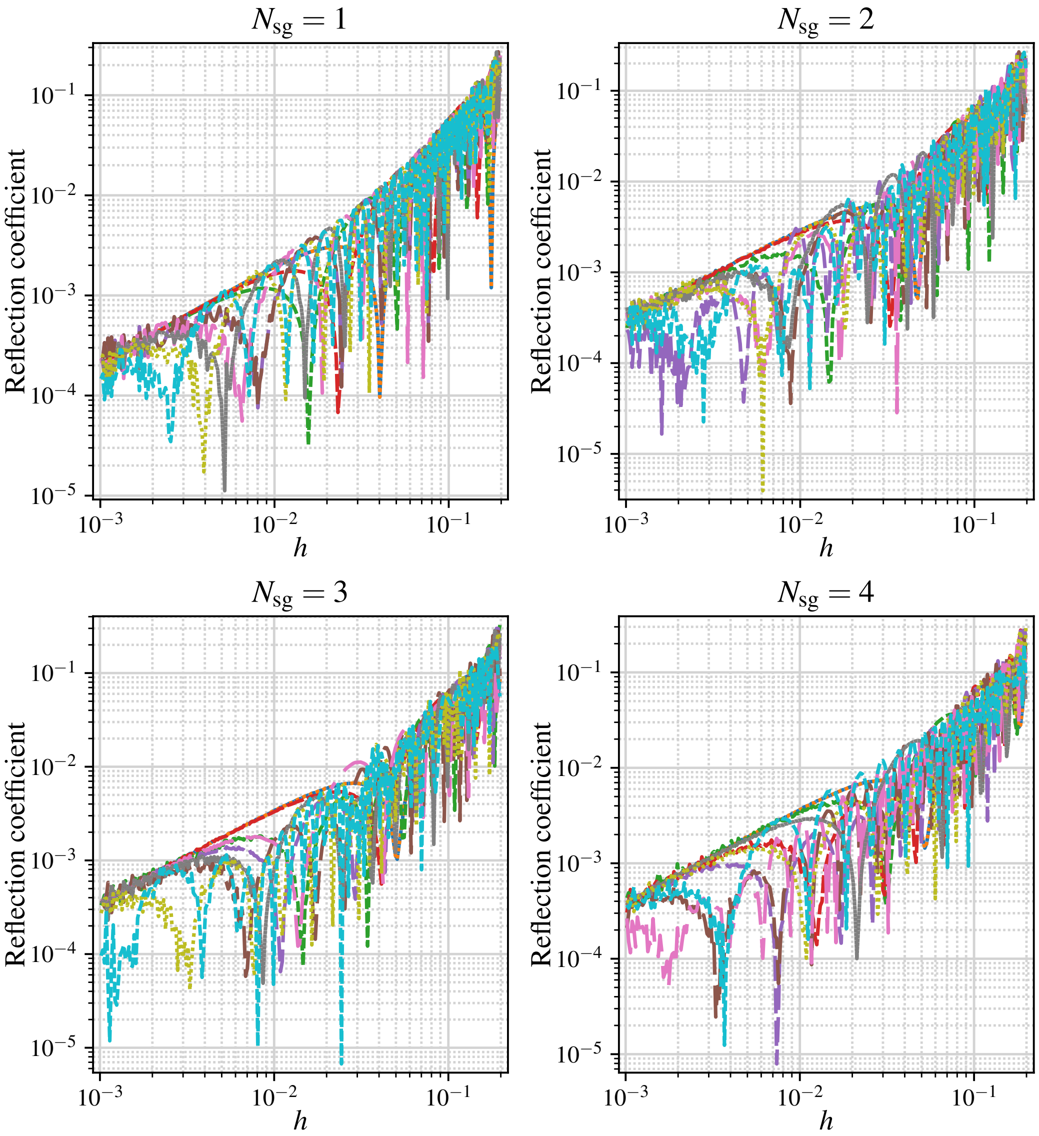

Figure 3.25: Reflection coefficient of the subgridding boundaries with GTS for $N_{\mathrm{sg}} = 1$ (upper-left), $N_{\mathrm{sg}} = 2$ (upper-right), $N_{\mathrm{sg}} = 3$ (bottom-left), and $N_{\mathrm{sg}} = 4$ (bottom-right). Simulations have been performed with subgridded region depths between 1 and 1000 cells.

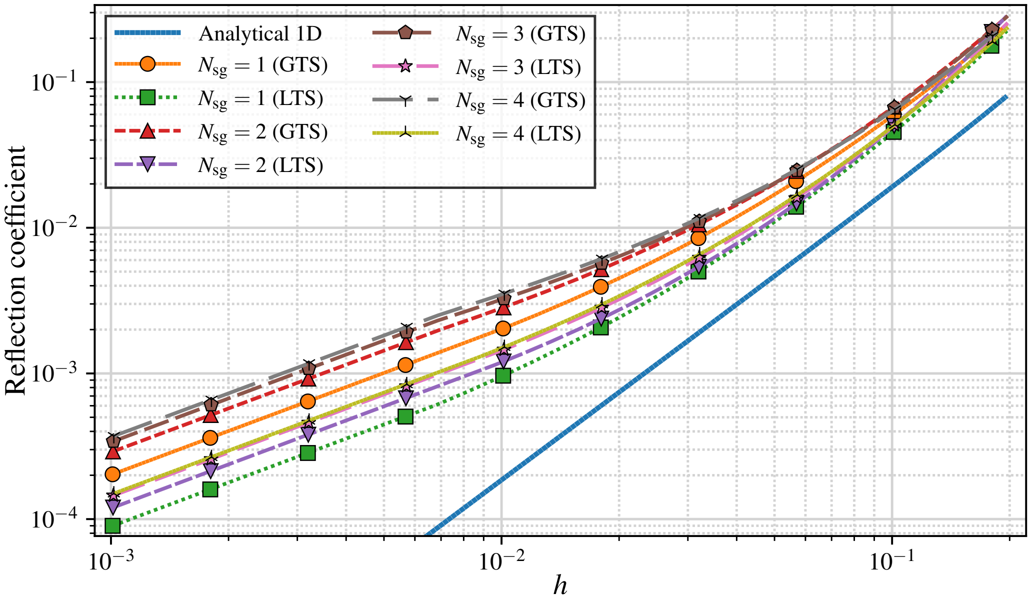

Figure 3.26: Envelop of the reflection coefficients of the subgridding boundaries with LTS for $N_{sg} \in \{1,2,3,4\}$. The analytical expression for the 1D subgridding reflection is drawn for reference.

3.6 Method modifications

Some modifications aiming to enhance some of the characteristics of the OI-SG are proposed in this section. These consist of

- An alternative LTS scheme involving time interpolations and extrapolation that prevents the non-centered finite differences in the time derivatives.
- A locally enlarged cell technique (LECT) technique that modifies some of the update parameters in the OI-SG scheme, thus permitting a larger CFLN value.

3.6.1 Interpolation/extrapolation LTS scheme

As seen in section 3.2, the LTS methodology proposed for the OI-SG method requires involving time derivatives of order $O(\Delta t)$, as written in eqs. (3.26) and (3.29), contrary to the centered finite differences of order $O(\Delta t^2)$. However, a different workaround involving time interpolations and extrapolations has been designed and implemented in this work. The scheme is depicted in fig. 3.27.

Let us get back to the LTS scheme depicted in fig. 3.5 and recapitulate the time mismatch explained in section 3.2. Let us consider an electric component of type E-2 or E-4 that belongs to a certain level $n_{\text{sg}}-1$. This electric field, in each update, requires the usage of at least one magnetic component of type H-4 belonging to the level n_{sg}. In what follows, we define $\Delta t_{n_{\text{sg}}}$ as the time step of the level n_{sg}, and the variable n always refers to an integer. We also denote $E_{\text{f}}|^{N}$ and $H_{\text{c}}|^{N}$, respectively, to the fine electric and coarse magnetic components evaluated at $t = N\,\Delta t_{n_{\text{sg}}}$, where N is not necessarily an integer nor a semi-integer. The problem can be summarized as:

- $E_{\text{f}}|^{n}$ requires $H_{\text{c}}|^{n-\frac{1}{4}}$.
- $E_{\text{f}}|^{n-\frac{1}{2}}$ requires $H_{\text{c}}|^{n-\frac{3}{4}}$.
- The only coarse magnetic components that exist are $H_{\text{c}}|^{n+1/2}$.

In general, the value of $H_{\text{c}}|^{N}$ can be obtained from the interpolation of any two other values $H_{\text{c}}|^{N_1}$ and $H_{\text{c}}|^{N_2}$ as

$$H_{\text{c}}|^{N} = \frac{(N_2-N)\,H_{\text{c}}|^{N_1} + (N-N_1)\,H_{\text{c}}|^{N_2}}{N_2-N_1}, \tag{3.164}$$

where $N_1 < N < N_2$. Using this, we may write

$$H_{\text{c}}|^{n-\frac{3}{4}} = \frac{1}{4}H_{\text{c}}|^{n-\frac{3}{2}} + \frac{3}{4}H_{\text{c}}|^{n-\frac{1}{2}}. \tag{3.165}$$

The only requirement to apply eq. (3.165) is that both components $H_{\text{c}}|^{n-\frac{3}{2}}$ and $H_{\text{c}}|^{n-\frac{1}{2}}$ must be already known at the moment of performing the interpolation. This can be done right after calculating $H_{\text{c}}|^{n-\frac{1}{2}}$, but before calculating $E_{\text{f}}|^{n-\frac{1}{2}}$, without any additional considerations. This interpolation is represented with green arrows in fig. 3.27.

On the other hand, $H_{\text{c}}|^{n-\frac{1}{4}}$ could be interpolated from $H_{\text{c}}|^{n-\frac{1}{2}}$ and $H_{\text{c}}|^{n+\frac{1}{2}}$. However, this is not possible if we look at the OI-SG algorithm (algorithm 1). Calculating the value of $H_{\text{c}}|^{n+\frac{1}{2}}$ necessarily requires $E_{\text{f}}|^{n}$ to be already known, but if we use the interpolation scheme, then $E_{\text{f}}|^{n}$ is calculated from $H_{\text{c}}|^{n+\frac{1}{2}}$, thus entering in a circular dependence. The only possible way to solve this is to use extrapolation instead of interpolation. For this case, extrapolation can be expressed in the same way as eq. (3.164), but with $N_1 < N_2 < N$. Thus, we may write

$$H_{\text{c}}|^{n-\frac{1}{4}} = -\frac{1}{4}H_{\text{c}}|^{n-\frac{3}{2}} + \frac{5}{4}H_{\text{c}}|^{n-\frac{1}{2}}. \tag{3.166}$$

The extrapolation is represented with red arrows in fig. 3.27.

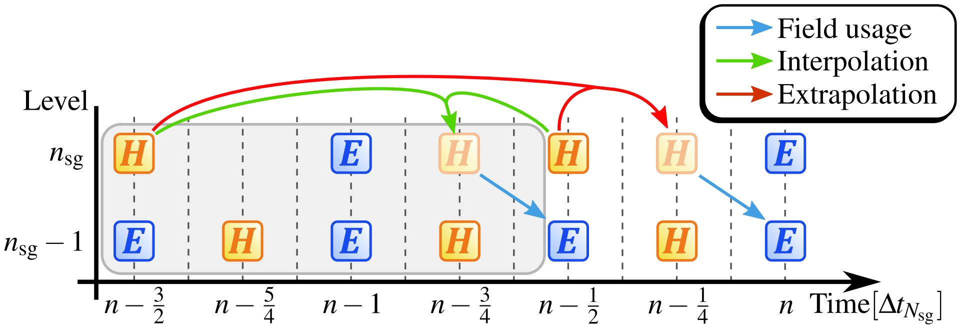

Figure 3.27: Illustration of the LTS scheme with time interpolations and extrapolations.

Regarding the stability of this methodology, first, it is worth pointing out that the spectral method described in section 3.4.2 cannot be used. This is because the OI-SG equations with the interpolation/extrapolation LTS cannot be written as an LTI system (eq. (3.46)). An LTI obtains the next state $\vec{v}|^{n+1}$ based on the state in the immediate previous iteration $\vec{v}|^{n}$. It is not difficult to see that, regardless of how we define $\vec{v}|^{n}$, the interpolation/extrapolation scheme requires using magnetic field components belonging to $\vec{v}|^{n-1}$ to obtain $\vec{v}|^{n+1}$. Therefore, it is written as

$$\vec{v}|^{n+1} = A\,\vec{v}|^{n} + B\,\vec{v}|^{n-1}, \tag{3.167}$$

contrary to eq. (3.46). The heuristic methodology, on the other hand, can be used to determine the stability conditions. However, the conclusions achieved have determined that the interpolation/extrapolation LTS scheme is unconditionally unstable, i.e. simulations are unstable for any pair $(\delta_{\text{r}}, \text{CFLN})$ with CFLN > 0.

As a final note, alternative higher-order interpolation/extrapolation schemes may be used, requiring values from even previous iterations. Additionally, time or spatial filtering techniques can be used as in other examples in the literature (see section 2.3) to filter the highest-frequency modes, which typically constitute the main source of instabilities. However, these have not been considered in the present work as they significantly increase the complexity of the analysis and the efficiency of the implementation.

3.6.2 Locally enlarged cell technique (LECT)

In section 3.4.1, an expression telling the maximum CFLN value as a function of the orthogonalization parameter δ_{r} was derived (eq. (3.42)). This condition, although not rigorous, is supported by the stability results obtained in section 3.4.2 with LTS, achieving a maximum stable value of CFLN $\simeq 0.67$. However, this value is particularly low for an FDTD submethod and enforces a decrease in the time step of a 67% compared to the CFL criterion, which impacts the efficiency (we require more time steps to get to the same physical time) and the numerical dispersion (see section 2.1.4). On the other hand, eq. (3.42) tells us that the sources of instabilities in the OI-SG are directly related to the combination of neighboring fields with large integration lines l and low equivalent surfaces $\tilde{S}$ (see table 3.1 and fig. 3.3). Inspired by this conclusion, and following a methodology based on the work from [25], a computational zero-cost methodology has been developed to permit larger stable values of CFLN, thus allowing a larger time step. In this section, a derivation of this methodology and its impact on stability are provided. Results from this methodology have been published in [48].

Let us consider the electric update equation of the OI-SG method in free space as written in eq. (3.17). If we substitute the time step Δt using eq. (2.72), we obtain

$$E\big|_i^{n+1} = E\big|_i^n + \frac{\text{CFLN}}{\tilde{S}_e\big|_i}\frac{\Delta}{\sqrt{3}}\sqrt{\frac{\mu_0}{\varepsilon_0}}\sum_{j\in\mathcal{N}_e|_i}\text{sgn}_e|_{i,j}\, l_h\big|_j\, H\big|_j^{n+\frac{1}{2}}. \tag{3.168}$$

As shown in section 3.4.1, the maximum stable CFLN is topped by a maximum value that depends on δ_r. In other words, if a simulation is unstable, we may stabilize it by lowering CFLN. If we take a look at eq. (3.168), for a given field update, lowering the value of CFLN is equivalent to increasing its equivalent surface $\tilde{S}_e|_i$. Thus, we may artificially increase $\tilde{S}_e|_i$ to permit a larger value of CFLN. This approach is known as locally enlarged cell technique (LECT). An analogous procedure can be performed with the magnetic update equation from eq. (3.16), concluding that artificially increasing $\tilde{S}_h|_i$ also allows a larger value of CFLN.

Let us define $\tilde{S}_{e,\text{new}}|_i$ as the modified value of $\tilde{S}_e|_i$, which will be used in a simulation with LECT. Similarly, we define $\xi_e|_i$ as the LECT factor, where

$$\tilde{S}_{e,\text{new}}|_i = \xi_e|_i \tilde{S}_e,|_i. \tag{3.169}$$

Analogously, we define the modified magnetic equivalent surface $\tilde{S}_{h,\text{new}}|_i$ and the magnetic LECT factor $\xi_h|_i$, where

$$\tilde{S}_{h,\text{new}}|_i = \xi_h|_i \tilde{S}_h,|_i. \tag{3.170}$$

Additionally, from eq. (3.43), we know which field types are the most critical ones concerning stability: E-1, H-1, and H-3. Therefore, we may modify the equivalent surfaces of these field types only to permit a larger CFLN value. Furthermore, when the modified values $\tilde{S}_{e,\text{new}}|_i$ and $\tilde{S}_{h,\text{new}}|_i$ have already been set, we may plug them into eq. (3.42) to yet again obtain whether other field types have now become the most critical ones concerning stability, and modify them as well.

Table 3.2: Values of ξ_e and ξ_h for the LECT configurations designed. The field types correspond to those in table 3.1 and fig. 3.3. $\xi = 1$ implies no modification in the equivalent surface.

Field type	ξ (No LECT) (CFLN=0.66)	ξ (LECT-1) (CFLN=0.8)	ξ (LECT-2) (CFLN=0.9)
E-1	1.0	1.7	3.0
E-2	1.0	1.0	3.0
E-3	1.0	1.0	3.0
E-4	1.0	1.0	3.0
E-5	1.0	1.0	3.0
E-6	1.0	1.0	3.0
H-1	1.0	1.5	3.0
H-2	1.0	1.5	3.0
H-3	1.0	1.5	3.0

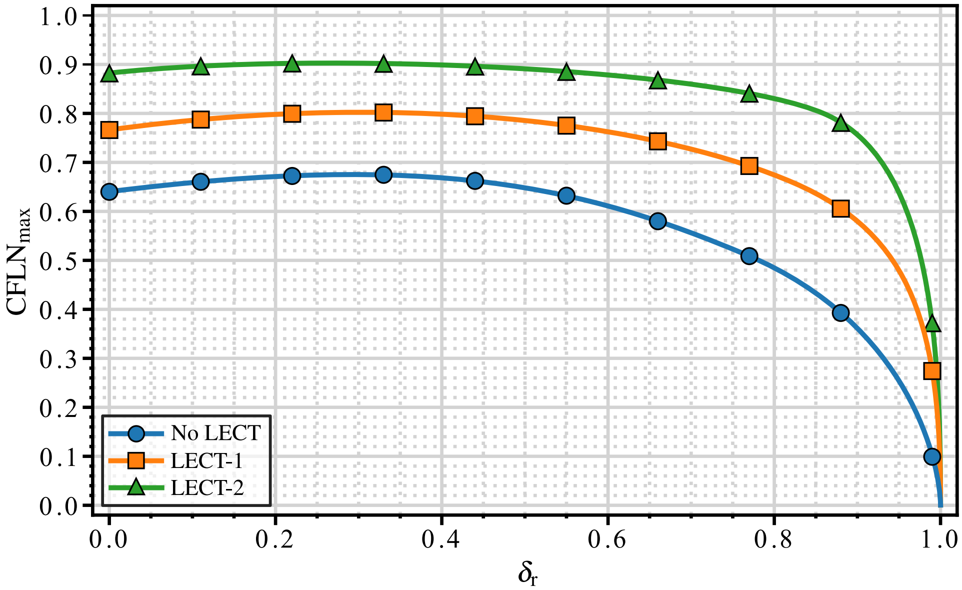

Figure 3.28: Maximum stable CFLN value as a function of δ_{r} by the spectral analysis method for different LECT configurations. We show the original one (without LECT), one that reaches CFLN = 0.8 and another that reaches CFLN = 0.9.

For all the field types described in table 3.1, different LECT parameters have been tried, focusing on those field components with smaller values of $\tilde{S}$. Finally, two different configurations have been designed: first, one that allows us to achieve CFLN $= 0.8$ with LTS (denoted LECT-1); and second, a more aggressive one that allows us to reach CFLN $= 0.9$ (denoted LECT-2). The LECT factors of these configurations are provided in table 3.2.

To test the maximum value of CFLN, the spectral methodology from section 3.4.2 has been employed. The results are shown in fig. 3.28 and compared against the non-LECT case. The figure was obtained using the spectral analysis method from section 3.4.2.

By looking at eq. (3.168), it is immediate to verify that this methodology is equivalent to locally modifying the media values of ε and μ. Nonetheless, only a few fields (non-trivially updated ones) are affected, and their modification can be very slight depending on the desired value of CFLN. Furthermore, this artificial material does not introduce numerical losses; therefore, only large LECT factors affect the precision. In addition, the LECT method does not involve modifying the OI-SG advance equations; therefore, it is zero-cost in memory and CPU usage. The impact on accuracy is tested in chapter 4.

3.7 Implementation

Some of the most relevant details regarding the computational implementation are shown in this section. These include:

- A meshing algorithm that adapts the computational domain to the objects present in the simulation, wrapping the desired subgridding level around each of them.
- A hybrid OMP–MPI parallelization scheme designed to further enhance the efficiency of the implementation.

3.7.1 Adaptative mesh refinement (AMR)

Specific simulation cases can be implemented by hand, allowing us to get results quickly. However, a deep comprehension of the OI-SG method requires performing many different simulations in various scenarios. Implementing so many cases by hand is not a trivial task, not to mention that it is exceptionally error-prone. For this reason, an automatic mechanism must be implemented such that arbitrary simulations can be performed. For this work, an adaptative mesh refinement (AMR) algorithm has been implemented, and it is presented in this section.

The implemented AMR algorithm is depicted in fig. 3.29 and expressed in pseudocode in algorithm 2. This algorithm takes the list of objects present in the simulation and the maximum subgridding level N_{sg} as input, and then creates the computational domain. Each object has a subgridding level n_{sg} associated, implying that said object must be embedded in a region of level n_{sg}. In most cases, all objects in a simulation are placed at the lowest level $n_{\text{sg}} = 0$ because it is of interest to have as much refinement as possible. Nonetheless, this algorithm permits optionally placing objects at different levels, which can be useful in some test simulations. The AMR algorithm begins by creating the lowest level region around the objects that belong to it. Then, the next coarser level $n_{\text{sg}} = 1$ is also created around the objects that belong to it, but also around the immediate finer level $n_{\text{sg}} = 0$. This process is repeated until reaching the coarsest level N_{sg}, which just extends until the computational limit. Since all corner cases are considered in the field classification in table 3.1 and fig. 3.3, this AMR can adapt to any arbitrary geometry, thus allowing all the simulations that are performed in chapter 4.

Algorithm 2 Pseudocode implementation of the AMR algorithm.

```
procedure CreateMesh(list_of_objects, $N_{\text{sg}}$)
    for $n_{\text{sg}} = 0,\dots,N_{\text{sg}}-1$ do
        if $n_{\text{sg}} > 0$ then
            RefineAroundLowerLevel($n_{\text{sg}}$)
        end if
        for object in list_of_objects do
            if ObjectBelongsToLevel(object, $n_{\text{sg}}$) then
                RefineAroundObject(object, $n_{\text{sg}}$)
            end if
        end for
    end for
    CreateCoarsestLevel()
end procedure
```

Buffering

During the AMR algorithm execution, refined regions of different levels are created around the material objects as desired. However, we need to define how much distance is left between the region boundaries and the material objects. This distance is called the *buffering distance* and is denoted d_{buf}. It can be also measured by the number of cells that it corresponds to, denoted n_{buf}. Due to the nature of the FDTD grid, d_{buf} is calculated as the Chebyshev distance. Given two points $\vec{x} = (x_1, x_2, x_3)$ and $\vec{y} = (y_1, y_2, y_3)$, the Chebyshev distance between them is

$$\text{dist}(\vec{x}, \vec{y}) = \max_{i=1}^{3}\{|x_i - y_i|\}. \tag{3.171}$$

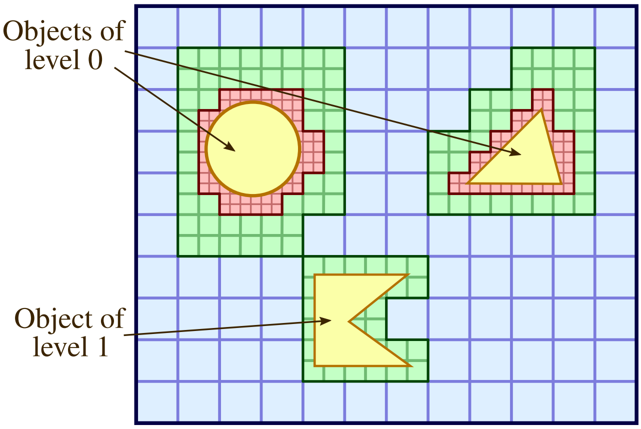

Figure 3.29: Illustration of the adaptative mesh implemented for this work. Each object has an associated level and the computational domain adapts to it.

A 2D buffering example with $n_{\text{buf}} = 4$ is illustrated in fig. 3.30. It is clear that, on a subgridding boundary, each coarse cell must be adjacent to a set of 2×2 fine cells. Therefore, the distance left between material cells and the subgridding boundary is of at least n_{buf}, but can be greater if required by the grid configuration. This is illustrated by the pink cells in fig. 3.30.

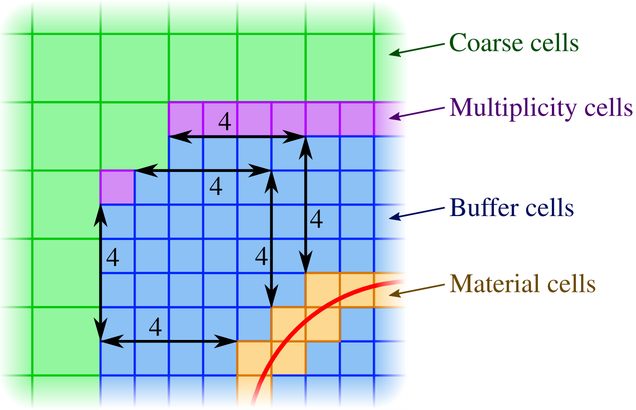

Figure 3.30: Example of the buffering methodology applied on a given object with $n_{\text{buf}} = 4$. A minimum of n_{buf} cells are left between the object and the finest-level subgridding boundary, according to the Chebyshev distance.

In previous bibliography [118], authors concluded that the buffering distance may significantly impact the accuracy of subgridding methods. In this work, this is validated for the OI-SG algorithm in various simulations in chapter 4. Optionally, buffering distance can also be applied to the space left between different subgridding boundaries, however, no significant difference has been found in this regard for $n_{\text{buf}} > 1$.

3.7.2 Hybrid parallelization OMP–MPI

All subgridding methods in FDTD are designed with efficiency in mind. Subgridding algorithms reduce the overall amount of FDTD cells in the computational domain compared to execution with finest-only cells. However, the original FDTD has a major advantage that is lost in subgridding schemes: due to its structured nature, it is exceptionally simple to parallelize. For this reason, a parallelization scheme suited for the OI-SG is required in order to further enhance the efficiency. The parallelization scheme designed for the OI-SG hybridizes two different kinds of parallelism: shared memory open multi-processing (OMP) parallelization and distributed memory message passing interface (MPI) parallelization.

The OMP specification is an interface that extends the languages C, C++, and Fortran. It provides a simple set of directives and routines to distribute tasks in threads within the same process. Typically, the computational domain of FDTD implementations is a rectangular cuboid. This makes the update equations trivial to implement as a triple for-loop for each field component, and whose iteration limits are known at the beginning of the execution. OMP provides a directive that automatically parallelizes this kind of loops, and thus FDTD can be trivially parallelized in this way [170]. Now, let us consider the region composed of all the cells of a given level n_{sg} in a simulation with subgridding. We name it *level domain*. In general, this region is arbitrarily shaped and thus the field updates cannot be written as a simple triple for-loop. In consequence, the OMP parallelization cannot be trivially applied. For this reason, the first step of the parallelization scheme consists of dividing each level domain into boxes shaped as rectangular cuboids. Note that this division is not unique, but the number of boxes should be as low as possible. A 2D example of this division is shown in fig. 3.31. Each one of these boxes is allocated in memory separately, and, since the OI-SG update equations are just a generalization of the FDTD equations, the triple for-loop approach can be applied in each box separately. This now allows us to implement the OMP parallelization scheme in each box trivially. It is worth noting that this scheme also requires performing certain communications between the different regions, thus providing some latency. However, this latency is only proportional to the number of fields located at the surfaces between the different boxes. Similarly, communication between regions of different levels is also required.

The MPI specification permits the execution of distributed tasks across different processes that do not necessarily share the same computer memory. For this work, we consider N_{MPI} different processes, where each one is assigned an integer number from 0 to $N_{\text{MPI}} - 1$. As stated previously, each level domain is divided into boxes in which the OI-SG algorithm

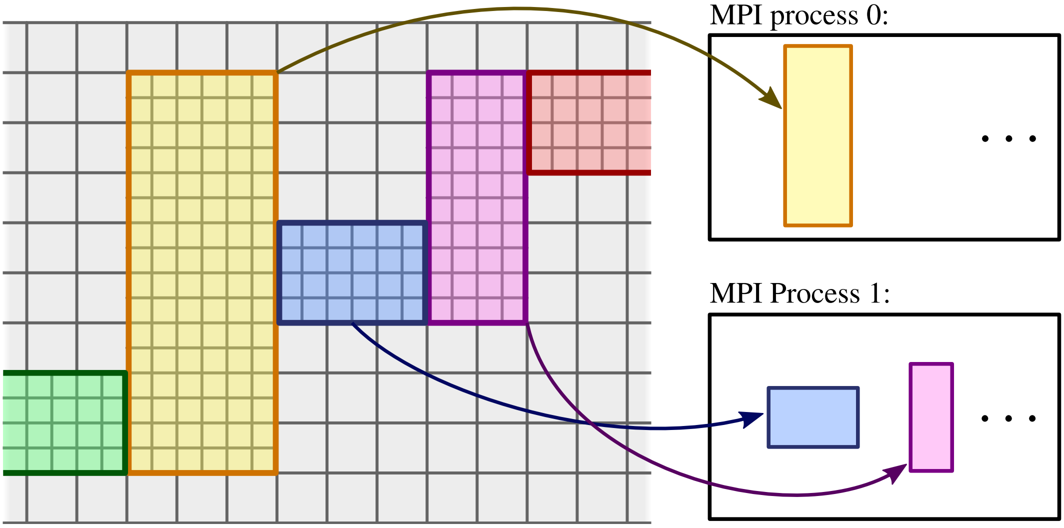

Figure 3.31: Illustration of the MPI parallelization implementation. The computational domain is divided into boxes and they are distributed between the MPI processes.

is applied separately. To apply the MPI parallelization, each region is assigned to a different MPI process at the beginning of the execution. This is depicted in fig. 3.31. Then, each process allocates in its memory the required fields of the boxes that it has been assigned. During the time-stepping execution, each MPI process takes care of updating its allocated fields. The distribution of the different boxes between the MPI processes is performed trying to balance the workload as much as possible. Similarly to the OMP parallelization, certain communication is required between the different MPI processes.

CHAPTER 4

Numerical validations

4.1 Introduction

This chapter is devoted to validating the OI-SG algorithm in actual numerical scenarios. In this regard, different simulation setups have been prepared and executed with a variety of parameters:

- A refined spherical shell without any material. This is the only case that does not correspond to any real scenario but allows us to study the numerical scattering provided by the subgridding boundaries.
- A frequency-selective surface (FSS). This structure, upon normal incidence, only allows a specific profile of frequencies to pass through it. Calculating the transmission and reflection coefficients allows us to measure the effects of the buffer spacing and the LECT methodology.
- An all-angle negative refraction (AANR) metasurface. Metasurfaces typically consist of very small repeating patterns, thus making them an illustrating example of multiscale problems.
- A PEC sphere. This is a very well-known canonical case. Its backscattering can be analytically calculated by the Mie series, thus allowing us to test the convergence of the OI-SG method.
- The NASA almond. Measuring the scattering of this case is a well-known LO problem. As has been seen in previous chapters, subgridding algorithms and non-uniform grids provide certain reflections, making LO cases particularly challenging. With this simulation, the OI-SG method will be tested in such a scenario.

- The FLAMME stealth object. Measuring the monostatic scattering of this case is another LO problem, and therefore a challenge for the OI-SG method.

- A conductive spherical shell. The shielding effectiveness (SE) can be calculated and thus we may observe the potential shielding effects that the OI-SG provides.

- Thin slot modeling in reverberant chambers. The SE of a conductive box with thin curved slots is measured. The slots can be modeled by the usual FDTD, by subcell methods, or by using subgridding. Results are compared with real measures.

- EV55 aircraft. An aeronautic study case for which the transfer function between fields and induced currents is measured. Results are obtained and compared between a conformal mesh, a structured mesh with subgridding, and a combination of both.

Typically, the most relevant parameters are the maximum subgridding level N_{sg}, the finest-level space step Δ_0, the coarsest-level space step $\Delta_{N_{\text{sg}}}$, the CFL number CFLN and the buffer spacing measured by the number of cells n_{buf} or the distance d_{buf}. Unless otherwise indicated, the default buffer spacing has $n_{\text{buf}} = 1$.

It is worth noting that, in the examples provided in this chapter, the PPW cannot be analytically calculated from the frequency since the numerical dispersion relation is unknown or possibly not even well-defined, as is shown in the 1D subgridding example with LTS in section 3.5.1. However, in some of the graphs provided, results are plotted as a function of the PPW corresponding to the finest or coarsest grids, which are respectively denoted as PPW_{fine} and $\text{PPW}_{\text{coarse}}$. In these occasions, the wavelength, and therefore PPW, is calculated using the analytical free-space dispersion relation. This is done to provide an approximate understanding of the dependence of the studied phenomena on the spatial resolution of the grid.

In what follows, several relevant concepts are explained to provide a full understanding of the simulations performed.

4.1.1 Radar Cross Section

In certain simulations, we may be interested in measuring how an object scatters a given incident field. We name this object the *scatterer*. To prepare this setup, a plane wave source is set directed toward the scatterer. Assuming that no other sources or objects exist within the computational domain, the total EM field can be split into two: the plane wave and the scattered field. The plane wave source is prepared in such a way that, outside its region, only the scattered field exists (this is explained in detail in section 2.1.3). This allows us to measure it, which is usually performed by calculating the radar cross section (RCS). This magnitude measures the asymptotic far-field projected by the scatterer in a given direction. It is defined as

$$\mathrm{RCS}(\theta,\phi) = \lim_{r\to+\infty}\left\{4\pi r^2 \frac{\left|\vec{E}_{\mathrm{scatter}}(r,\theta,\phi)\right|^2}{\left|\vec{E}_{\mathrm{inc}}\right|^2}\right\} = 4\pi r^2 \frac{\left|\vec{E}_{\mathrm{FF}}(r,\theta,\phi)\right|^2}{\left|\vec{E}_{\mathrm{inc}}\right|^2}, \tag{4.1}$$

where r, θ and ϕ are the spherical coordinates of the measured position in space, $\vec{E}_{\mathrm{scatter}}$ is the scattered electric field, $\vec{E}_{\mathrm{FF}}$ is the electric field in the asymptotic far-field approximation, and $|E_{\mathrm{inc}}|$ is the electric field of the incident plane wave. It is interesting to note that eq. (4.1) only depends on the measured direction and not on r because $|E_{\mathrm{FF}}|$ is inversely proportional to r. The RCS can be measured in FDTD by using the near-to-far-field (NTFF) algorithm [17, Ch. 8].

4.1.2 Shielding Effectiveness

The shielding effectiveness (SE) is a magnitude that quantifies the protection that a given enclosure offers from the effects of external EM fields. As time passes, old mechanical systems are substituted by more modern electronic devices, thus making them subject to EM interference. For this reason, enclosures must be designed and built to protect the most critical systems, thus making SE a useful magnitude within EMC analysis.

Let us assume a given structure and set of incident fields provided by the environment, $E_{\mathrm{inc}}(f,\vec{r})$, where f is the frequency and $\vec{r}$ the position in space. The illuminated structure could be, for example, a material shell with arbitrary shape and some given physical properties, or a PEC box with slots. Now, let us measure the electric field at a given point $\vec{r}_0$ inside this structure. Clearly, if this structure was not present, and assuming no other sources or objects exist, the measured field would be $E_{\mathrm{inc}}(f,\vec{r}_0)$. However, the structure interacts with the incident field and therefore the actual measured field is $E_{\mathrm{measured}}(f,\vec{r}_0)$, which must be proportional to $E_{\mathrm{inc}}(f,\vec{r}_0)$ due to the linearity of Maxwell's equations. Thus, we may define the SE as the ratio

$$\mathrm{SE} = \frac{E_{\mathrm{inc}}(f,\vec{r}_0)}{E_{\mathrm{measured}}(f,\vec{r}_0)}. \tag{4.2}$$

This expression provides higher values when the measured field is lower compared to the incident field, thus higher values of SE imply better shielding.

4.1.3 Effective materials

To better capture the geometry of bulk materials, an effective material technique is employed based on the one presented in [31]. This technique consists of calculating the effective physical magnitudes, such as the electric permittivity and conductivity, of a given discrete position by ponderating the surrounding volume around it. To do so, a box with the same shape as an FDTD cell is placed around a given discrete component, and then the effective magnitudes are calculated from the materials that lie within the box by ponderating them by the volume they occupy.

$$\sigma_{\text{eff}} = \sigma_{\text{mat}} \frac{V_{\text{mat}}}{V_{\text{box}}}, \tag{4.3}$$

where σ_{eff} is an effective arbitrary physical parameter, σ_{mat} is said parameter in the material region, V_{box} is the ponderation box, and V_{mat} is the part of that volume that is occupied by the material region. An illustration is shown in fig. 4.1.

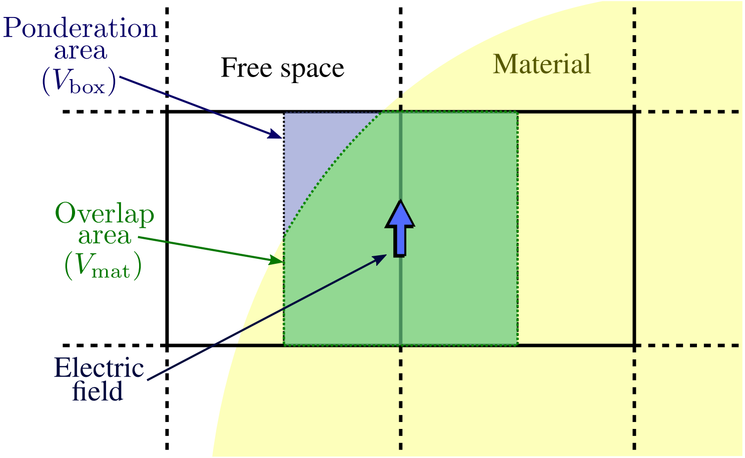
Ponderation area (V_{box})
Free space
Material
Overlap area (V_{mat})
Electric field

Figure 4.1: Scheme of the effective material technique employed for this work.

It is worth noting that the technique here described is a very simple approach, and more precise methodologies exist in the literature [171]. However, this work is only focused on the validation of the OI-SG method. Therefore, our main interest is to compare the behavior of different refinement levels against each other regardless of the technique used to calculate the effective materials.

4.1.4 Performance measurement

To measure the efficiency of the OI-SG algorithm in a given simulation case, first, we need to distinguish between two different times that can be evaluated: the CPU time is the time that the machine takes to perform a given set of tasks, whereas the EM time is the simulated time within a given simulation. The EM time can be viewed just as the time step multiplied by the number of iterations. Let us define the cells per electromagnetic time (CPET) as the amount of FDTD cells that are required to be processed to get to a specific EM time. It can be calculated as

$$\mathrm{CPET} = \sum_{n_{\mathrm{sg}}=0}^{N_{\mathrm{sg}}} \frac{N_{\mathrm{cells},n_{\mathrm{sg}}}}{\Delta t_{n_{\mathrm{sg}}}}, \tag{4.4}$$

where $N_{\mathrm{cells},n_{\mathrm{sg}}}$ and $\Delta t_{n_{\mathrm{sg}}}$ are, respectively, the number of cells and the time step of the subgridding level n_{sg}. We may also define the cells per CPU time (CPCT) as the average amount of cells that are processed within a given CPU time frame. If we execute a given simulation until reaching a certain EM time T_{EM}, we have that the required CPU time T_{CPU} is

$$T_{\mathrm{CPU}} = T_{\mathrm{EM}} \frac{\mathrm{CPET}}{\mathrm{CPCT}}. \tag{4.5}$$

From here, we can conclude that more efficient simulations relate to low values of CPET and high values of CPCT.

The OI-SG affects CPET in several ways. First, as seen in section 3.4, the CFL criterion is more restrictive for the OI-SG than for the original FDTD, thus decreasing the time steps $\Delta t_{n_{\mathrm{sg}}}$ and, in consequence, increasing CPET. However, if we compare a simulation with subgridding against another one with the finest-level grid only, the number of cells is necessarily decreased, therefore reducing CPET. Furthermore, if we apply LTS then $\Delta t_{n_{\mathrm{sg}}}$ is increased for $n_{\mathrm{sg}} > 0$, thus reducing CPET even further. In the following examples, it is shown that the net value of CPET is reduced by the OI-SG method. On the other hand, CPCT depends on various factors such as the quality of the implementation, the employed compiler, the parallelization scheme, the number of threads, the machine power, and the latency induced by other procedures required to be executed aside from the fields update.

Let us suppose one simulation without subgridding and another one with $N_{\mathrm{sg}} > 0$ which simulates the same case, both sharing the same finest level grid of space step Δ_0. In this context, when we say that two executions share the same finest-level grid, we mean that their finest-level region (or only region if $N_{\mathrm{sg}} = 0$) has the same space step. Clearly, the simulation with subgridding is expected to be more efficient because some parts of its computational domain are coarser than the simulation without subgridding, and therefore we require processing less discrete components to achieve the same EM time. As stated before, this has a direct impact on the CPET. If the CPCT remained constant, then the gain in CPU time would be exactly equal to the gain in CPET, as can be extracted from eq. (4.5). However, subgridding inevitably decreases CPCT because of the latency induced by the communications described in section 3.7, and additionally the parallelization scheme is necessarily more complex and not as efficient as in the standard FDTD. Thus, the gain in CPU time will necessarily be less than the gain in CPET in the considered scenario. In an ideal case with a perfect implementation that did not decrease CPCT, the gain of both magnitudes would be equal. In the performance tables shown in this chapter, both CPET and T_{CPU} are displayed to show this behavior.

Another important part of performance analysis is memory consumption. This magnitude is often measured by the amount of FDTD cells present in the simulation N_{cells}. On average, we may approximate that each cell contains six discrete components: E_x, E_y, E_z, H_x, H_y, and H_z. Therefore, it is an accurate representation of the memory required for a given execution. When subgridding is used, the number of total cells can be calculated just as the summation of the cells in each level,

$$N_{\mathrm{cells}} = \sum_{n_{\mathrm{sg}}=0}^{N_{\mathrm{sg}}} N_{\mathrm{cells},n_{\mathrm{sg}}}. \tag{4.6}$$

This magnitude is displayed in all the performance tables shown in this chapter.

4.2 Numerical scattering of the subgridding boundaries

This simulation case is the only one in this chapter that does not belong to a real physical object. The aim of this section is rather to study the numerical scattering produced by the subgridding boundaries per se. For this purpose, the simulation setup prepared contains a refined region shaped as a spherical shell, which does not contain any material aside from free space. The refined region is recursively embedded within coarser levels until reaching the coarsest region. An illustration of this is shown in fig. 4.2. PML boundary conditions are applied at the coarsest level. In the coarsest region, a plane wave source is set directed towards the z-positive direction with electric polarization in the x-positive direction (depicted in fig. 4.3). The scattering produced by the subgridding boundaries is then measured in all cartesian axes. For the same reasons described in section 3.5.3, a second simulation with the coarsest-grid only is performed to subtract any error due to the plane wave source or PML boundary conditions.

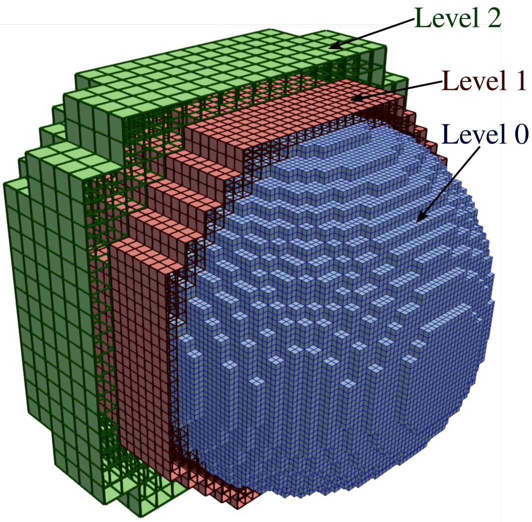

Figure 4.2: Example of the AMR around a spherical-shaped region. Levels 1 and 2 have been cropped for illustration purposes.

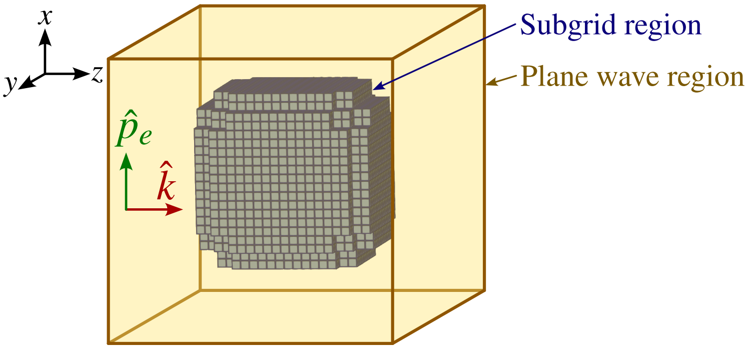

Figure 4.3: Illumination scheme of the spherical region.

A family of simulation cases has been prepared by varying certain parameters and taking all the combinations:

- Maximum subgridding level: $N_{\text{sg}} \in \{1,2,3,4\}$.
- Radius of the spherical shell: $r_s \in \{4,8,16,32,64\}$ cells.
- Distance between subgridding boundaries $\in \{1,5,10\}$ cells.

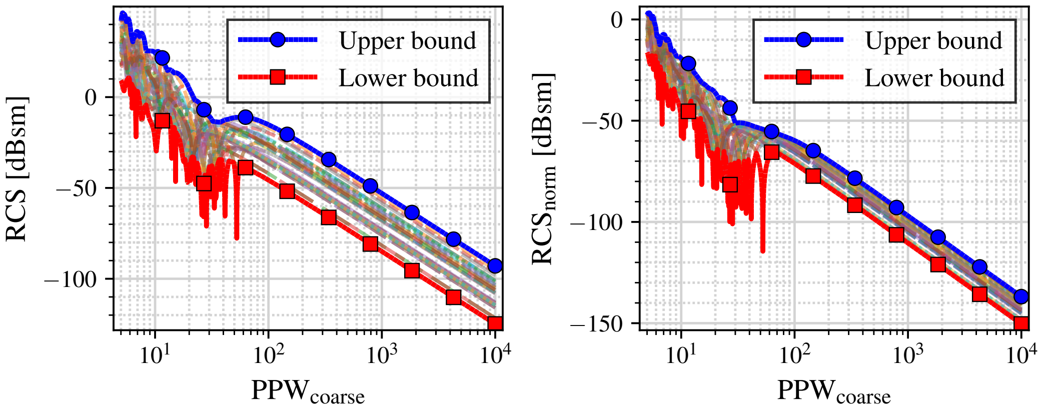

Figure 4.4: Backscattering ($\theta = \pi$) of the spherical shells with LTS, measured with RCS (left) and RCS_{norm} (right).

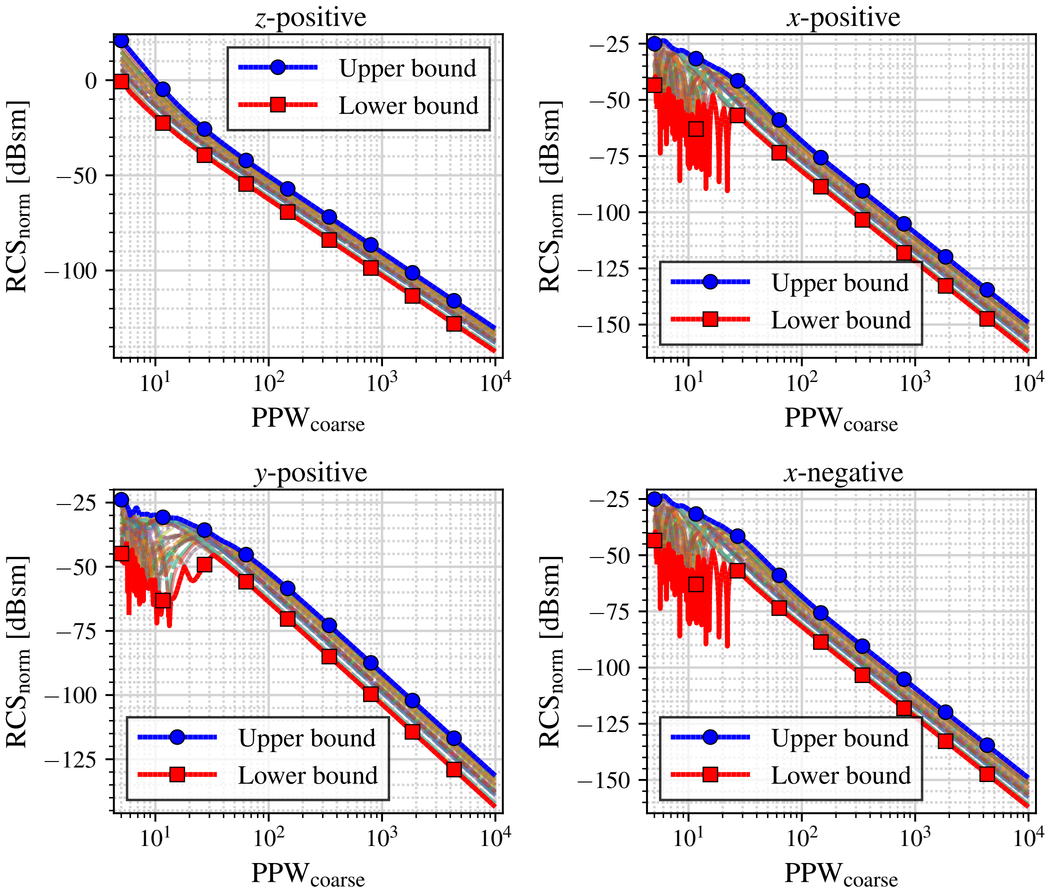

Figure 4.5: Normalized RCS of the spherical shells with GTS, measured for the directions *z*-positive, *x*-positive, *y*-positive, and *x*-negative.

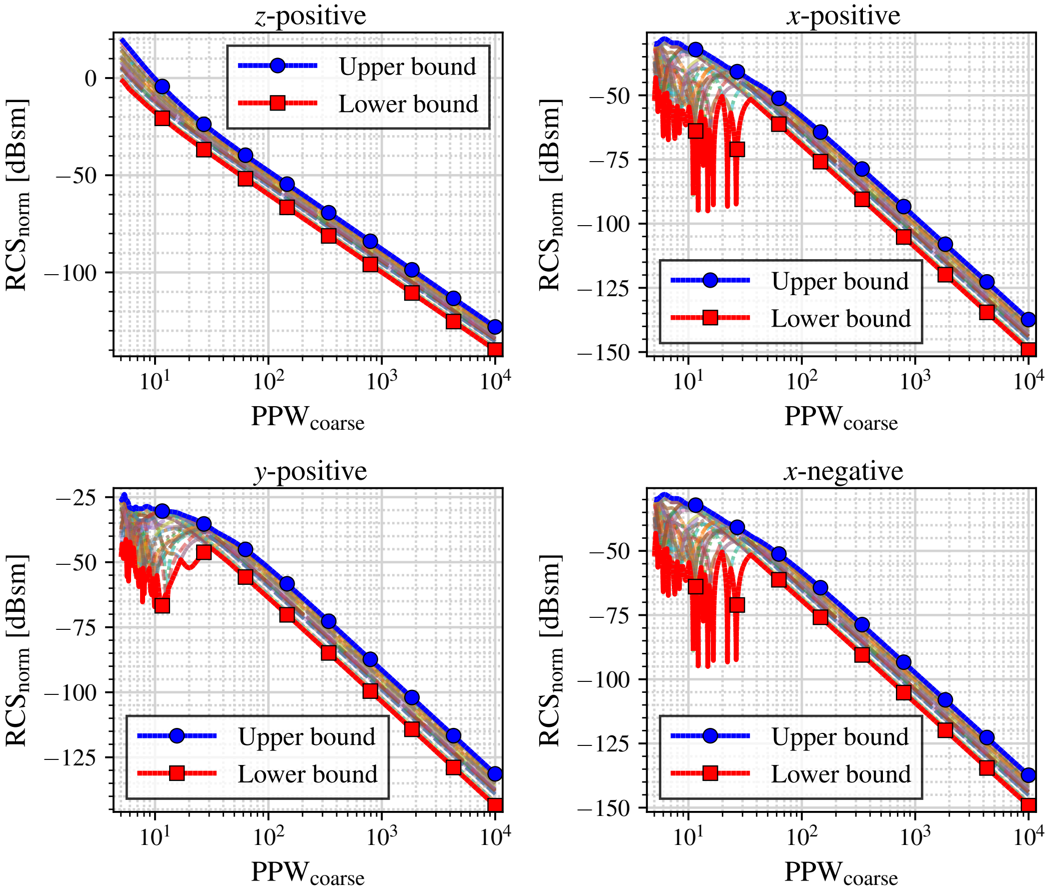

Figure 4.6: Normalized RCS of the spherical shells with LTS, measured for the directions z-positive, x-positive, y-positive, and x-negative.

Each simulation has been performed with GTS and LTS. The finest-level space step used is $\Delta_0 = 1\,\text{m}$ in all cases.

The RCS, as it is defined in eq. (4.1), does not depend on r. However, it does depend on the size of the scatterer: the larger the object the higher the scattered energy. Therefore, plotting the measured RCS is not going to provide meaningful results. To solve this, we require normalizing the RCS to the size of the object. This can be simply done by evaluating the scattered field at a distance proportional to the object size, which in this case we defined by the radius r_s,

$$\text{RCS}_{\text{norm}}(\theta,\phi) = 4\pi \frac{\left|\vec{E}_{\text{FF}}(r_s,\theta,\phi)\right|^2}{\left|\vec{E}_{\text{inc}}\right|^2}. \tag{4.7}$$

Additionally, we need to determine the appropriate variable against which to plot the RCS.

According to the results obtained in the previous chapter, we have determined that the convergence of the method depends on the resolution of the grids in a given subgridding boundary. In other words, the larger the value of PPW, the lower the numerical error. This implies that the most relevant parameter regarding numerical scattering is parameter is the resolution of the coarsest level, determined by $\mathrm{PPW}_{\mathrm{coarse}}$. For this reason, the RCS is measured against the value of $\mathrm{PPW}_{\mathrm{coarse}}$.

In fig. 4.4, the backscattering ($\theta = \pi$) of the simulations with LTS is plotted, comparing the measurements of RCS and $\mathrm{RCS}_{\mathrm{norm}}$. It is interesting to note how the normalized RCS brings together the different simulations. The normalized RCS is plotted in fig. 4.5 for the executions with GTS and in fig. 4.6 for executions with LTS. The normalized RCS has been taken in the directions z-positive, x-positive, y-positive, and x-negative. In all cases, aside from high-frequency resonances, a convergence of $O(h^2)$ is appreciated, where $h = (\mathrm{PPW}_{\mathrm{coarse}})^{-1}$.

4.3 Surface transmission test cases

4.3.1 Transmission and reflection of a frequency-selective surface

This simulation consists of a frequency-selective surface (FSS). Upon incidence of an EM wave, an FSS lets pass only a frequency band with a specific profile and reflects the rest of the spectrum. The FSS used for this simulation consists of a PEC surface with periodic slots. A scheme is depicted in fig. 4.7. The unit cell of the surface is a rectangle with dimensions $L_x = 0.14\,\mathrm{m}$ and $L_z = 0.12\,\mathrm{m}$. The slots are rectangles centered within the unit cell, with dimensions $l_x = 0.1\,\mathrm{m}$ and $l_z = 0.02\,\mathrm{m}$. This FSS provides the maximum transmission frequency (also denoted *peak frequency*) at $f_0 \simeq 1.5\,\mathrm{GHz}$, which corresponds to a wavelength of $\lambda_0 \simeq 5\,\mathrm{m}$. In all the simulations, the FSS is embedded within the finest-level grid, which has a space step of $\Delta_0 = 2.5\,\mathrm{mm}$. To replicate an infinitely large surface, only one unit cell is simulated and the computational domain is truncated with periodic boundary conditions and PML with 10 layers as depicted in fig. 4.7. A truncated plane wave is set at the coarsest level directed towards the FSS with normal incidence oriented towards the y-axis. The electric field is oriented towards the z-axis and has the following Gaussian profile:

$$E_z(t) = (1\,\mathrm{V/m})\, e^{-\left(\frac{t-t_0}{\tau}\right)^2}, \tag{4.8}$$

with $\tau = 0.23\,\mathrm{ns}$ and $t_0 = 4\,\tau$. Results of this simulation case with a different subgridding algorithm were presented by Xu et al. in [106], and with the OI-SG method in [48].

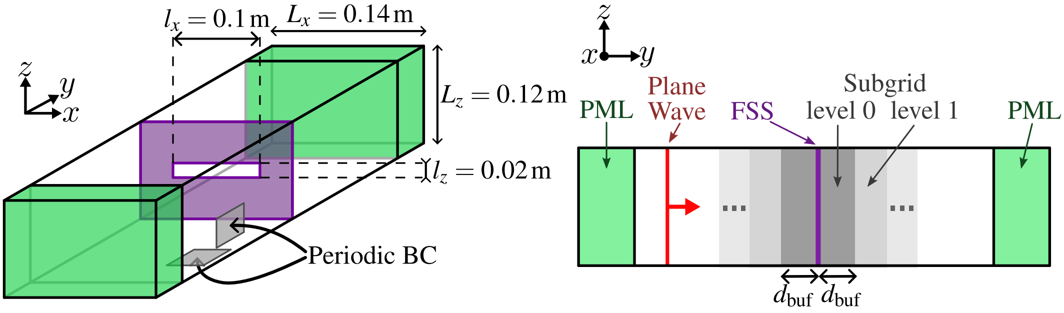

Figure 4.7: Scheme and dimensions of the FSS simulation setup.

With this simulation, several phenomena are intended to be studied. First, the differences in accuracy between GTS and LTS; second, the effect of the LECT technique from section 3.6.2 on a real simulation case; and third, the effect of the buffer space explained in section 3.7.1. For these purposes, different simulations have been prepared with maximum subgridding levels $N_{\mathrm{sg}} \in \{0, 1, 2, 3\}$. Each simulation with subgridding has been performed with GTS and LTS, and with a buffering space of $n_{\mathrm{buf}} \in \{1, 10\}$ cells. Additionally, some simulations with LTS have also been run with the configurations LECT-1 and LECT-2 from table 3.2. Since all simulations share the same finest-level space step, we can take $N_{\mathrm{sg}} = 0$ as a reference case. Any simulation with $N_{\mathrm{sg}} > 1$ is expected to have less accuracy than the reference, although they are also expected to be much more efficient due to the subgridding methodology. The magnitudes measured for this simulation are the reflection coefficient R and the transmission coefficient T, which are measured as

$$R(f) = \frac{|E_{\text{reflected}}(f)|^2}{|E_{\text{incident}}(f)|^2}, \tag{4.9}$$

$$T(f) = \frac{|E_{\text{transmitted}}(f)|^2}{|E_{\text{incident}}(f)|^2}, \tag{4.10}$$

where f is the frequency. Results for different LECT configurations are shown in fig. 4.8, and the corresponding performance table is shown in table 4.1. Results of the transmission and reflection coefficients for different values of n_{buf} are shown in fig. 4.9 with GTS and in fig. 4.10 with LTS. Their respective performance tables can be found in table 4.2 and table 4.3.

The first piece of information that can be extracted from tables 4.1–4.1 is that the memory consumption is highly reduced when applying subgridding: around a 15% is re-

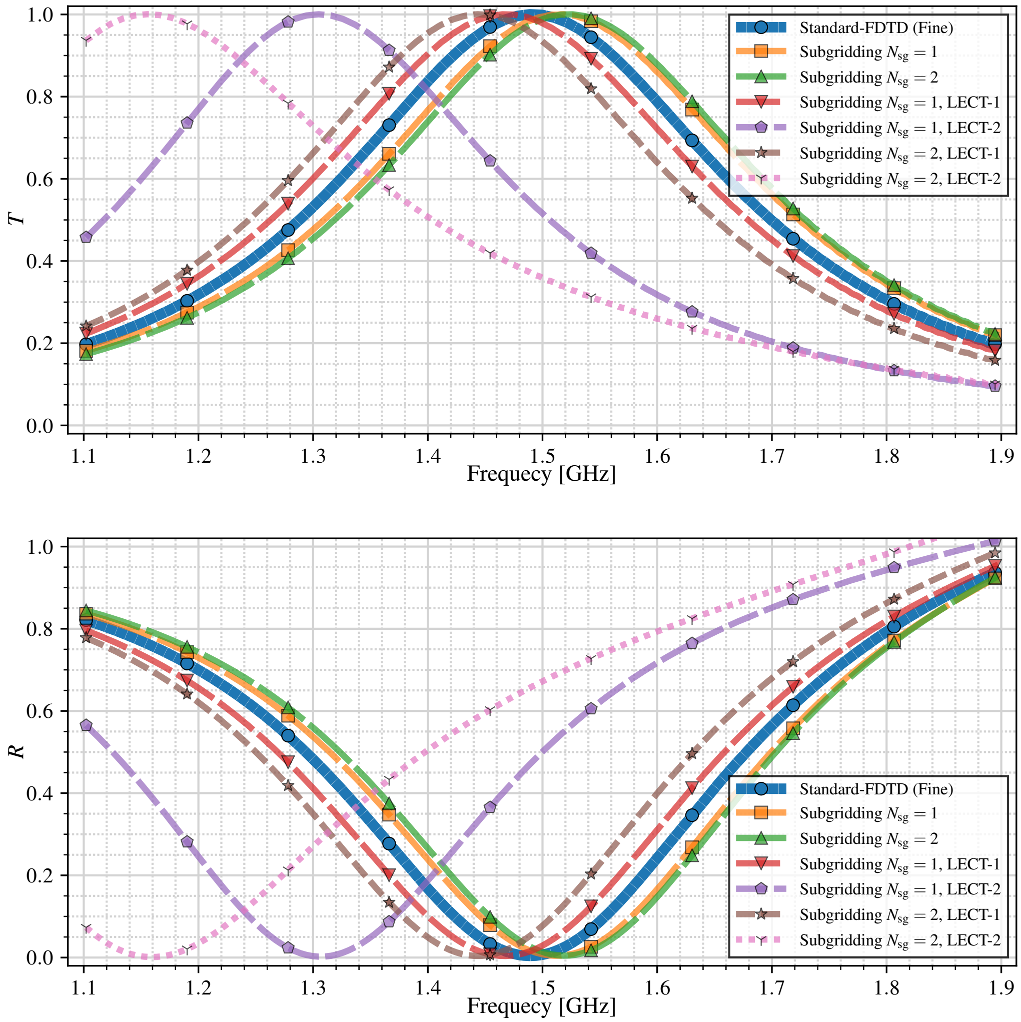

Figure 4.8: T and R of the FSS for different LECT configurations with LTS.

Table 4.1: Performance table of the FSS with LTS for the different LECT configurations with a common finest-level grid of $\Delta_0 = 2.5\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\text{sg}}}$ [mm]	LECT	N_{cells}	CPET	T_{CPU}
0	0.99	2.5	N/A	2967552	1.0000	1.0000
1	0.66	5	No	387840	0.0991	0.1038
1	0.80	5	LECT-1	387840	0.0830	0.0797
1	0.90	5	LECT-2	387840	0.0738	0.0711
2	0.66	10	No	63096	0.0122	0.0128
2	0.80	10	LECT-1	63096	0.0102	0.0114

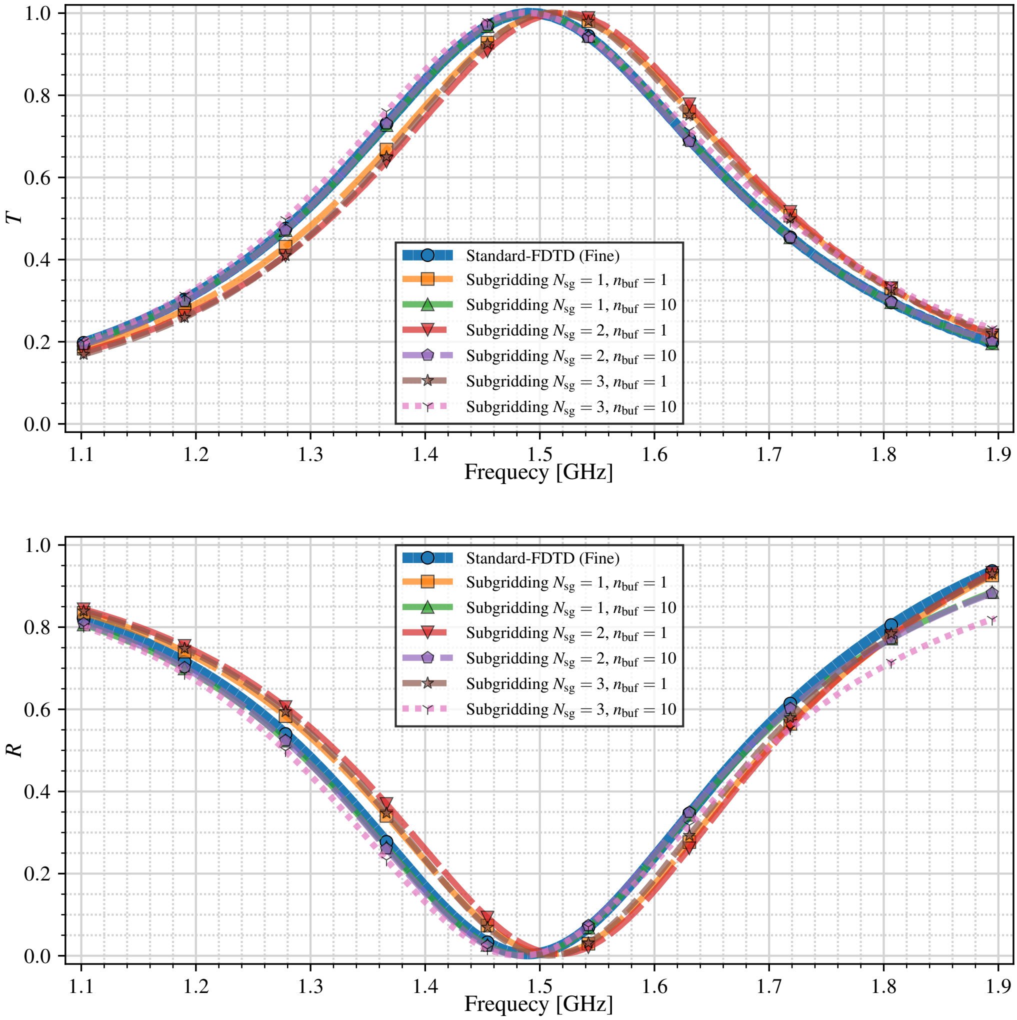

Figure 4.9: T and R of the FSS for different buffer spacings with GTS.

Table 4.2: Performance table of the FSS with GTS for different buffer spacings with a common finest-level grid of $\Delta_0 = 2.5\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\text{sg}}}$	n_{buf}	N_{cells}	CPET	T_{CPU}
0	0.99	2.5	N/A	2967552	1.0000	1.0000
1	0.93	5	1	387840	0.1391	0.1878
1	0.93	5	10	442848	0.1589	0.2376
2	0.93	10	1	63096	0.0226	0.0327
2	0.93	10	10	118272	0.0424	0.0612
3	0.93	20	1	21768	0.0078	0.0313
3	0.93	20	10	76860	0.0276	0.0512

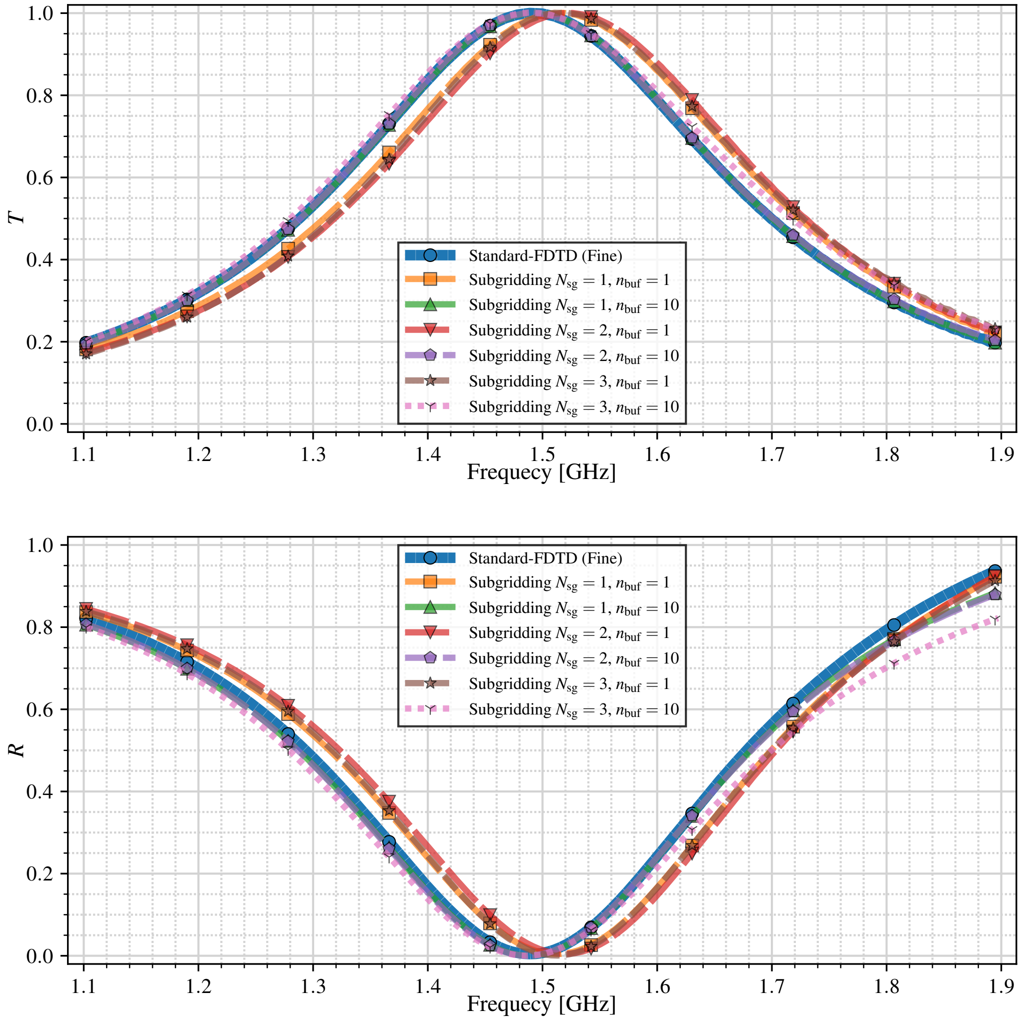

Figure 4.10: T and R of the FSS for different buffer spacings with LTS.

Table 4.3: Performance table of the FSS with LTS for different buffer spacings with a common finest-level grid of $\Delta_0 = 2.5\,\mathrm{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\mathrm{sg}}}$ [mm]	n_{buf}	N_{cells}	CPET	T_{CPU}
0	0.99	2.5	N/A	2967552	1.0000	1.0000
1	0.66	5	1	387840	0.0991	0.1038
1	0.66	5	10	442848	0.1250	0.1366
2	0.66	10	1	63096	0.0122	0.0128
2	0.66	10	10	118272	0.0380	0.0327
3	0.66	20	1	21768	0.0066	0.0100
3	0.66	20	10	76860	0.0324	0.0228

quired for $N_{\mathrm{sg}} = 1$, 5% for $N_{\mathrm{sg}} = 2$ and even less than 1% for $N_{\mathrm{sg}} = 3$. This effect is reduced in the presence of buffer spacing, which increases the amount of FDTD cells. But even in this circumstance, the required memory can be lowered to 4% for $N_{\mathrm{sg}} = 3$.

By looking at table 4.1, we note that all LECT configurations provide a performance gain in terms of both CPET and CPU time. This gain observed is between 12% to 23% for LECT-1, and 31% for LECT-2, compared to the same simulations without LECT. However, by looking at fig. 4.8, we may also observe that LECT-2 is too aggressive and induces large inaccuracy. On the other hand, although LECT-1 also provides a shift in the peak frequency of the transmission coefficient, it is much less aggressive. We may note that this shifting has a greater impact for $N_{\mathrm{sg}} = 2$, and therefore LECT-1 with $N_{\mathrm{sg}} = 1$ might be suitable for some simulations where great accuracy is not required. In this simulation, we may observe that this configuration provides a CPU time gain of $\sim 23\%$.

By looking at figs. 4.9 and 4.10, we observe that the profile shape of the transmission and reflection coefficients remains unchanged for all simulations. However, executions with subgridding provide a certain shift of the peak frequency to higher values. Interestingly, this would seem to be contradictory because it implies that certain frequencies that should not be able to pass through the FSS are actually passing. This can be explained by the near-field distortions due to the subgridding boundaries: the plane wave with normal incidence gets slightly distorted when entering the subgridding boundary, and therefore the FSS does not receive a perfectly normal incidence. It can be seen in figs. 4.9 and 4.10 that this gets solved when some buffer space is introduced: all the curves become nearly identical to the one corresponding to $N_{\mathrm{sg}} = 0$, except for the highest frequencies, especially with $N_{\mathrm{sg}} = 3$. This can be explained because the electric size of the cells at the subgridding boundaries becomes larger as we increase the frequency.

To better visualize the error, the RMS and peak frequency errors are defined as

$$\text{Error RMS} = \sqrt{\sum \left(\frac{T - T_{\mathrm{ref}}}{T_{\mathrm{ref}}} \right)^2}, \tag{4.11}$$

$$\text{Error } f_{\mathrm{peak}} = \left| \frac{f_{\mathrm{peak}} - f_{\mathrm{peak,ref}}}{f_{\mathrm{peak,ref}}} \right|, \tag{4.12}$$

where f_{peak} is the frequency of maximum transmission, and the case taken as reference is the one without subgridding ($N_{\mathrm{sg}} = 0$). The defined errors are plotted in fig. 4.11 as a function of the buffer distance d_{buf}.

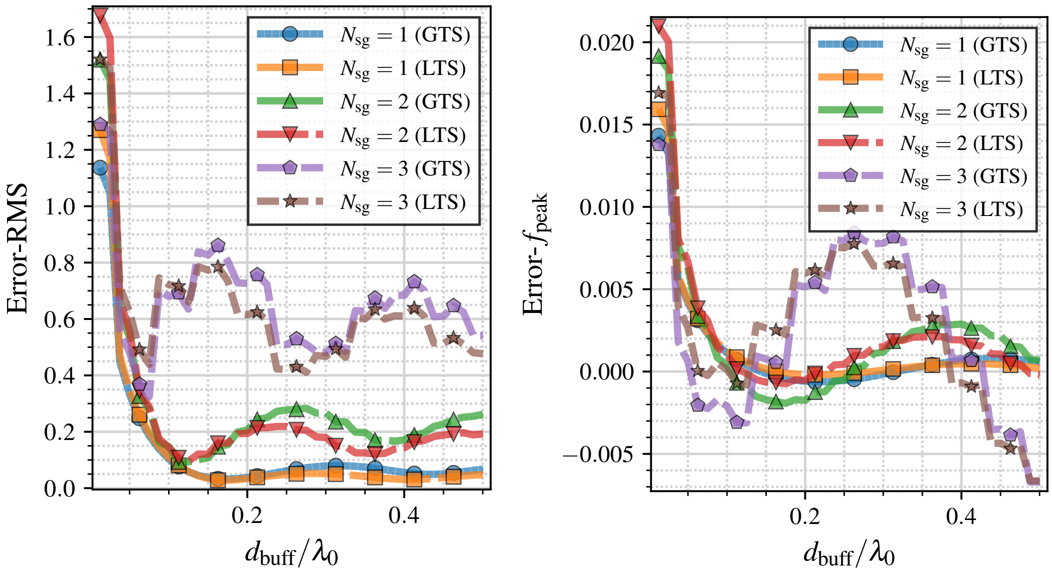

Figure 4.11: RMS (left) and peak frequency (right) errors of the transmission coefficients as a function of d_{buf} for the FSS simulation case.

In all the cases plotted in fig. 4.11, we observe that the error significantly decreases for $n_{\mathrm{buf}} > 1$. More specifically, the case that requires a larger buffer spacing to stabilize is $N_{\mathrm{sg}} = 1$, whose minimum error is achieved at around 10 cells, which corresponds to $d_{\mathrm{buf}} \simeq 0.12\lambda_0$. These plots show that the near-field distortions have an impact on the accuracy and it can be solved by applying buffer spacing. We also observe that GTS has a slightly lower accuracy than LTS, although the discrepancy is low.

4.3.2 All-angle negative refraction metasurface

A use case for subgridding algorithms is metasurfaces. Typically, metasurfaces consist of surfaces with small periodic geometrical patterns that, macroscopically, exhibit behaviors that are not achievable by conventional materials, such as negative refraction indices. The fine details of the metasurface typically require a much finer grid compared to the larger structures that may be present in the simulation. Therefore, this makes it a good example to illustrate multiscale problems. In this section, an all-angle negative refraction (AANR) metasurface is simulated with the OI-SG method.

This metasurface was previously studied in [39] and also simulated using a different FDTD subgridding algorithm in [143]. Results with the OI-SG method were presented in [48]. It has a thickness e and is made of a dielectric material of electric permittivity ε. It contains cylindrical holes with radius r, which are arranged in a square lattice with a distance of d between them (see fig. 4.12). The values taken for this simulations are $e = 0.916667\,\text{m}$, $r = 0.0625\,\text{m}$, $d = 0.18333\,\text{m}$, and $\varepsilon = 12\varepsilon_0$. To best capture its geometry, the metasurface is fully embedded within the finest-level grid, which has a cell size of $\Delta_0 = 12.5\,\text{mm}$. A buffer space is left between the metasurface and the finest subgridding boundary, which is measured by the number of cells n_{buf} or the distance d_{buf}. The rest of the levels are wrapped around the previous ones until reaching the coarsest one, where periodic boundary conditions are applied in the z-axis and PML boundary conditions with 10 layers in the two remaining axes. Note that this is a 2D simulation, and therefore the z-axis is only one coarse cell long. A vertical electric nodal source (directed towards the x axis) is placed next to the metasurface and is excited with the following modulated sinusoidal profile $E_{\text{s}}(t)$:

$$E_{\text{s}}(t) = \begin{cases} \sin\left(\frac{2\pi c}{\lambda}t\right)\exp\left(\frac{(t-t_0)^2}{2\,\sigma^2}\right) & \text{if} t \leq t_0 \\ \\ \sin\left(\frac{2\pi c}{\lambda}t\right) & \text{if} t > t_0 \end{cases} \tag{4.13}$$

with a wavelength $\lambda = 1\,\text{m}$, $\sigma = \lambda/c_0$ and $t_0 = 5\sigma$. A screenshot of this simulation without subgridding is shown in fig. 4.12. The physical constants of the discrete fields have been calculated using the effective material methodology explained in section 4.1.3.

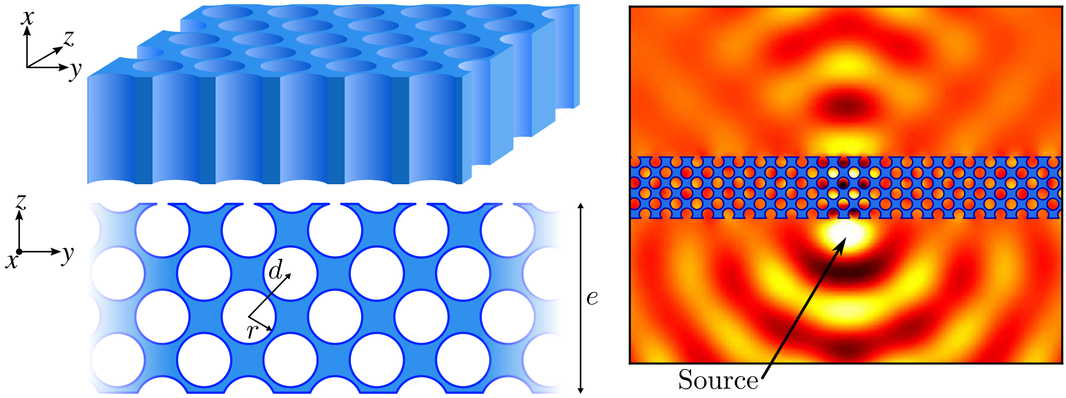

Figure 4.12: Scheme of the AANR metasurface simulated with the OI-SG and a screenshot of the stationary state without subgridding.

In this section, the aim is to measure the effects of the near-field distortions produced by the AANR metasurface. To do so, a nodal probe is placed at the other side of the metasurface from the source. Simulations are performed with LTS for maximum levels $N_{\text{sg}} \in \{0, 1, 2, 3\}$ and different buffering spaces, $n_{\text{buf}} \in \{1, 5, 10, 20\}$ for the simulations with subgridding. Then, the error of the probe is measured taking as reference the fine-only case ($N_{\text{sg}} = 0$).

The error is calculated as

$$\text{Error } E_x\,[\text{dB}[\text{V/m}]] = 10\log_{10}|E_x(\text{V/m}) - E_{x,\text{ref}}[\text{V/m}]|\,, \tag{4.14}$$

where $E_{x,\text{ref}}$ is the reference value. The results are depicted in fig. 4.13 and the corresponding performance table is shown in table 4.4.

Table 4.4: Performance table of the AANR metasurface. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	n_{buf}	N_{cells}	CPET	T_{CPU}
0	0.99	N/A	475600	0.6983	0.9385
1	0.66	1	227408	0.2414	0.6000
1	0.66	5	241712	0.2624	0.5846
1	0.66	10	253872	0.2803	0.5590
1	0.66	20	284464	0.3252	0.6872
2	0.66	1	266944	0.3319	0.9128
2	0.66	5	298592	0.3762	0.8923
2	0.66	10	326016	0.4142	0.6718
2	0.66	20	394992	0.5097	0.6308
3	0.66	1	483056	0.6432	1.0000
3	0.66	5	546352	0.7317	0.7026
3	0.66	10	602800	0.8083	0.6718
3	0.66	20	742416	1.0000	0.8308

By looking at fig. 4.13, we may appreciate that, similar to the FSS case from section 4.3.1, the buffer space reduces the error. Interestingly, by looking at table 4.4, we may see that $N_{\text{sg}} = 3$ has a worse performance than the finest-only case. This is to be expected since, as stated previously, this is a 2D simulation that has a thickness of one coarse cell in the z-axis. This means that the finest level region must have a thickness of $2^{N_{\text{sg}}}$ cells, which, when N_{sg} is large enough, seriously penalizes the efficiency.

4.4 Scattering test cases

4.4.1 Backscattering of a PEC sphere

This section aims to study the canonical case of the PEC sphere backscattering. The sphere taken for these simulations has a radius of 1 m. It is embedded within the finest level ($n_{\text{sg}} = 0$) and each coarser level embeds the finer regions until reaching the coarsest level N_{sg}, which extends until the computational limits, where PML boundary conditions with 10 layers are applied. In the coarsest level, a plane wave source is set directed towards the

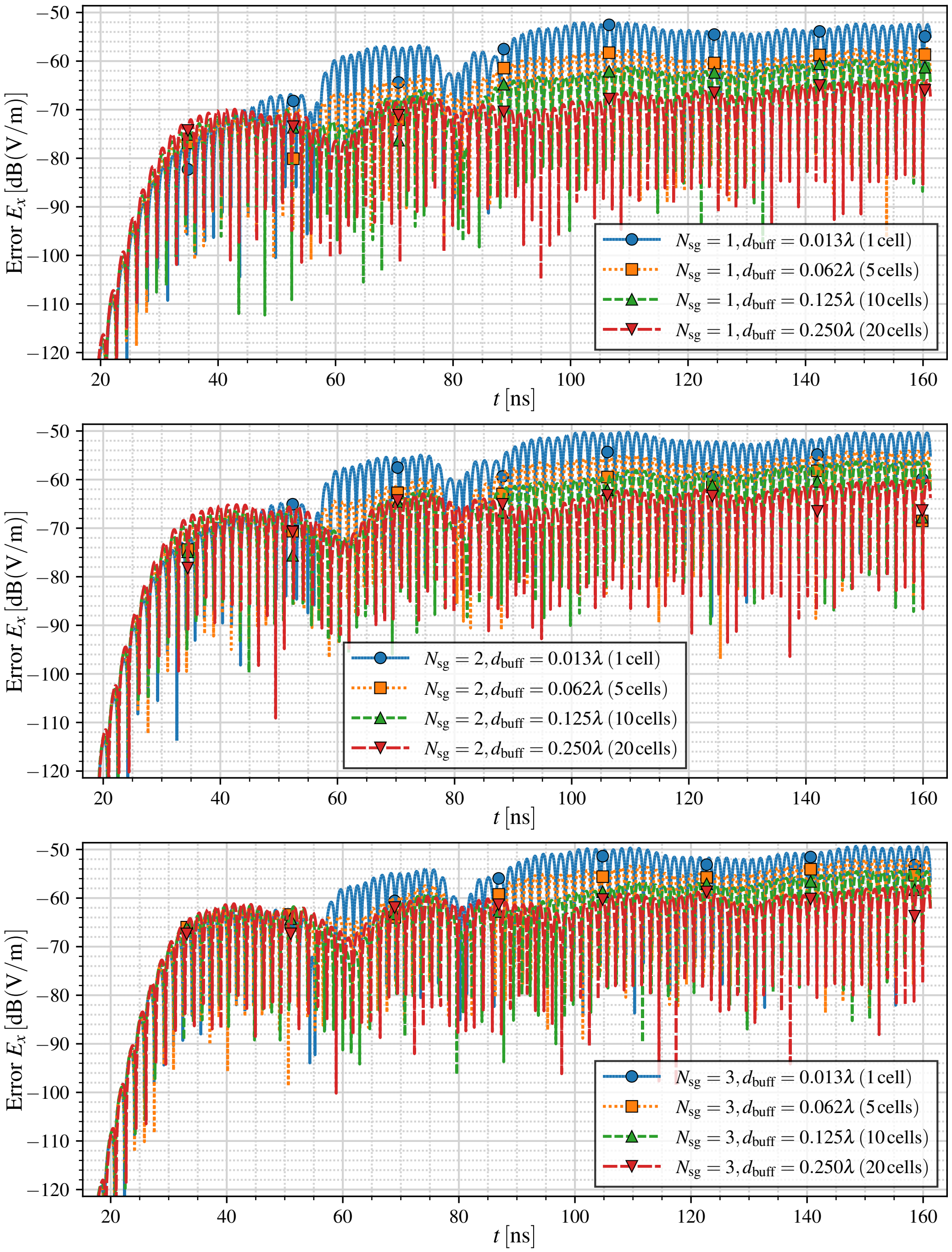

Figure 4.13: Error in the nodal probe results as a function of the time, for maximum subgrid levels $N_{\mathrm{sg}} = 1$, 2 and 3. All simulations have a common finest-level cell size of $\Delta_0 = 12.5\,\mathrm{mm}$.

sphere, and the backscattering is measured using the RCS formula from eq. (4.1). The theoretical value of the monostatic RCS of a PEC sphere can be calculated with the Mie series. Results for the OI-SG method were presented in [48].

To validate the OI-SG algorithm, two different approaches have been designed. First, a set of simulations is executed with a common finest-level grid of $\Delta_0 = 12.5\,\mathrm{mm}$, and with maximum subgrid levels $N_{\mathrm{sg}} \in \{0,1,2,3\}$. Second, another set of simulations is performed with a common coarsest-level grid of $\Delta_{N_{\mathrm{sg}}} = 100\,\mathrm{mm}$, and with maximum subgridding levels $N_{\mathrm{sg}} \in \{0,1,2,3\}$. Additionally, each simulation is run with $n_{\mathrm{buf}} \in \{1,10\}$. The backscattering RCS is measured in every case and is compared to the theoretical curve calculated from the Mie series. All simulations have been performed with LTS. Results are shown in figs. 4.14 and 4.15, and performance tables are displayed in tables 4.5 and 4.6.

By looking at fig. 4.14, it is interesting to note that the RCS of the PEC sphere is particularly well-solved in all cases where $N_{\mathrm{sg}} \leq 2$. Some induced error may be appreciated as the coarsest-level space step increases, and the case with $\Delta_{N_{\mathrm{sg}}} = 100\,\mathrm{mm}$ is visibly worsened, especially for higher frequencies. However, it is also worth noting that these frequencies correspond to $\mathrm{PPW}_{\mathrm{fine}} \simeq 100$, and for $N_{\mathrm{sg}} = 3$ implies $\mathrm{PPW}_{\mathrm{coarse}} \simeq 12.5$, which is a very low resolution that has been shown to provide larger numerical reflections in section 3.5.3 and section 4.2. By looking at fig. 4.15, we may observe that decreasing the finest-level grid coming from a common coarsest-level grid does indeed reduce the discrepancy between numerical results and the theoretical values obtained from the Mie series. It must be noted that is true for $N_{\mathrm{sg}} < 3$, but a greater error can be appreciated for $N_{\mathrm{sg}} = 3$, implying that an excess of subgridding levels can lead to worse accuracy. Regarding the efficiency, table 4.5 shows a decrement in N_{cells} between 69% with $N_{\mathrm{sg}} = 1$ and 87% with $N_{\mathrm{sg}} = 3$ if no buffer space is applied, and between 43% and 61% if $n_{\mathrm{buf}} = 10$. Similarly, T_{CPU} decrements between 72% and 87% if $n_{\mathrm{buf}} = 1$, and between 37% and 52% if $n_{\mathrm{buf}} = 10$, thus demonstrating strong performance in this case.

It is interesting to note that, from the results shown in figs. 4.14 and 4.15, no clear conclusion can be drawn about the effect of the buffer spacing. For this reason, additional simulations with $\Delta_0 = 25\,\mathrm{mm}$ have been performed for various values of n_{buf} and maximum subgridding levels $N_{\mathrm{sg}} \in \{0,1,2\}$. Taking the case with $N_{\mathrm{sg}} = 0$ as a reference, we define the accumulate RMS error as

$$\text{Error RMS Accumulate}(f) = \sqrt{\int_{f_0}^{f} \left(\frac{\mathrm{RCS}(f') - \mathrm{RCS}_{\mathrm{ref}}(f')}{\mathrm{RCS}_{\mathrm{ref}}(f')} \right)^2 df'}, \tag{4.15}$$

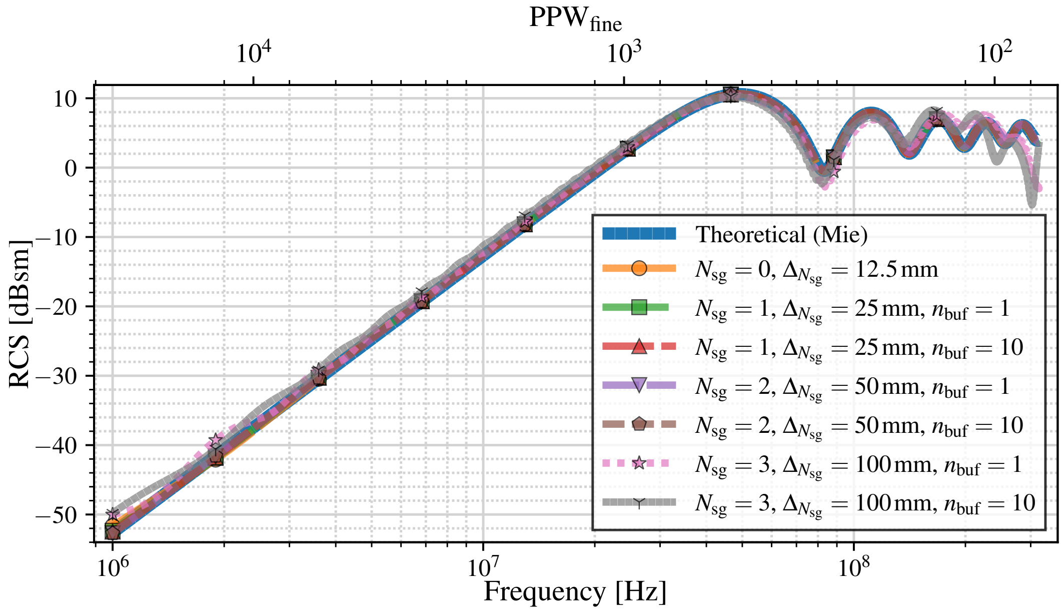

Figure 4.14: Backscattering RCS of a PEC sphere of radius 1 m for a common finest-level grid of $\Delta_0 = 12.5\,\mathrm{mm}$.

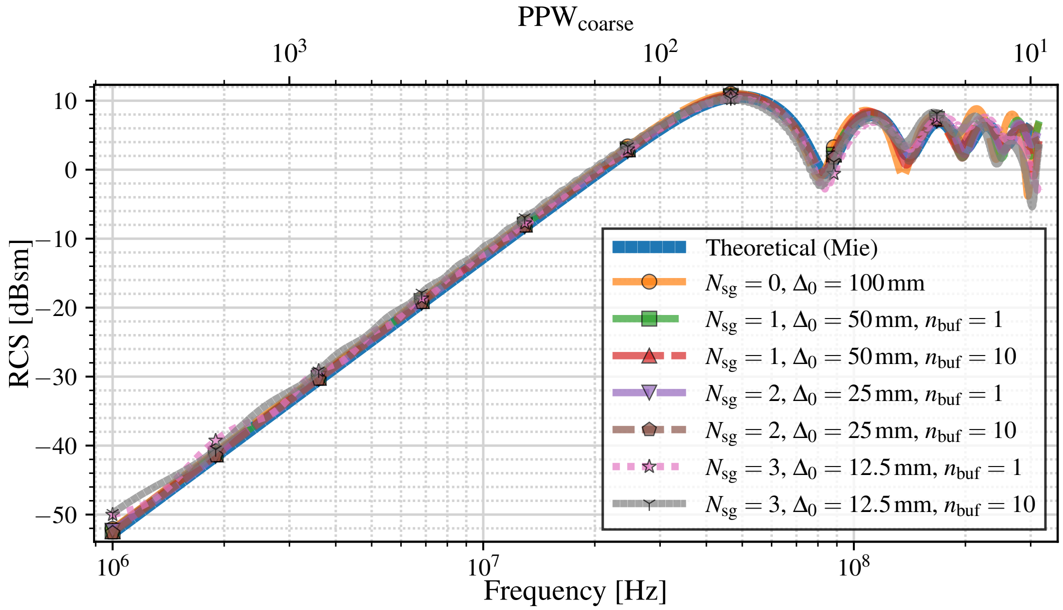

Figure 4.15: Backscattering RCS of a PEC sphere of radius 1 m for a common coarsest-level grid of $\Delta_{N_{\mathrm{sg}}} = 100\,\mathrm{mm}$.

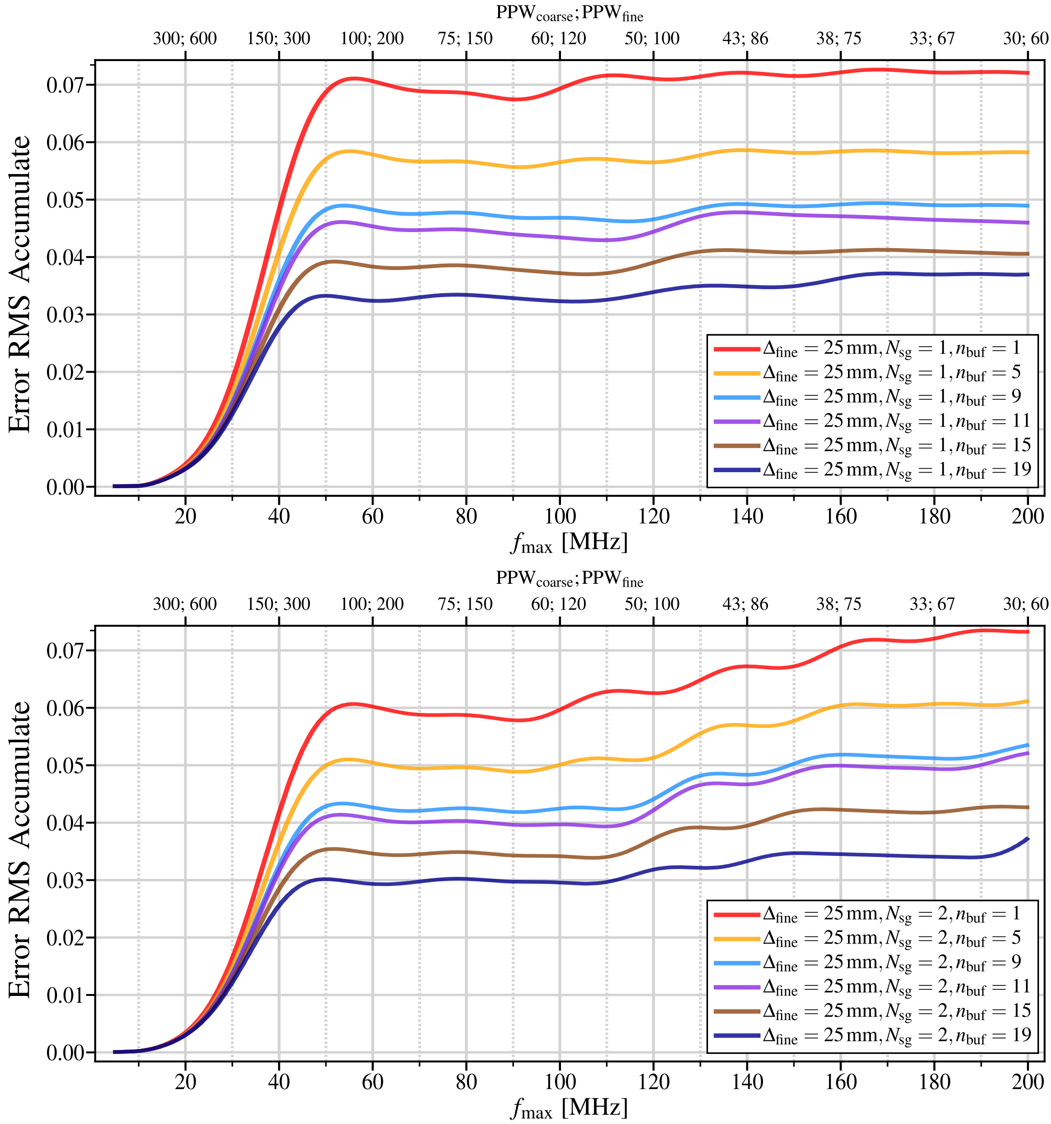

Figure 4.16: Accumulate RMS error of the PEC sphere RCS as a function of the frequency for various values of n_{buf}. Results are shown for $N_{\mathrm{sg}} = 1$ (left) and $N_{\mathrm{sg}} = 2$ (right).

Table 4.5: Performance table of a PEC sphere of radius 1 m for a finest-level grid of $\Delta_0 = 10\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\text{sg}}}$ [mm]	n_{buf}	N_{cells}	CPET	T_{CPU}
0	0.99	12.5	N/A	15438249	1.0000	1.0000
1	0.66	25	1	4575976	0.1781	0.1377
1	0.66	25	10	8614024	0.4203	0.1637
2	0.66	50	1	2354168	0.0879	0.0302
2	0.66	50	10	6364928	0.3270	0.0470
3	0.66	100	1	1795327	0.0749	0.0116
3	0.66	100	10	5767257	0.3130	0.0245

Table 4.6: Performance table of a PEC sphere of radius 1 m for a coarsest-level grid of $\Delta_{N_{\text{sg}}} = 100\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	Δ_0 [mm]	n_{buf}	N_{cells}	CPET	T_{CPU}
0	0.99	100	N/A	328509	0.0068	0.3179
1	0.66	50	1	444712	0.0100	0.3645
1	0.66	50	10	753800	0.0198	0.3751
2	0.66	25	1	605416	0.0229	0.2392
2	0.66	25	10	1338088	0.0760	0.3403
2	0.66	12.5	1	1795327	0.2393	0.4753
2	0.66	12.5	10	5767257	1.0000	1.0000

where f and f' are frequencies, and RCS_{ref} is the RCS of the reference case, and $f_0 = 3\,\text{kHZ}$. The integral is numerically obtained by the trapeze finite integration. Results are shown in fig. 4.16.

In all cases, we may observe that the accumulated error increases with the frequency. However, the increase rate becomes visibly lower at a certain value, and it is clear that the error is significantly lower as we increase n_{buf}. Therefore, we may conclude that the buffer spacing has a relevant effect in increasing the accuracy of this simulation.

4.4.2 Scattering of the NASA almond

The NASA almond, along with ogives in general [172, 173], is a very well-known low observability (LO) case. It is entirely made of PEC material. In particular, the NASA almond used for this case is the one described in [174]. It is 2.5 m long and has an elliptical profile of diameters 185.888 mm and 386.666 mm. A scheme is depicted in fig. 4.17 (left).

LO cases are known for providing low scattering upon the incidence of an EM pulse, which is particularly challenging for subgridding algorithms since they provide spurious reflections that can be comparable to the object scattering. For this reason, this section

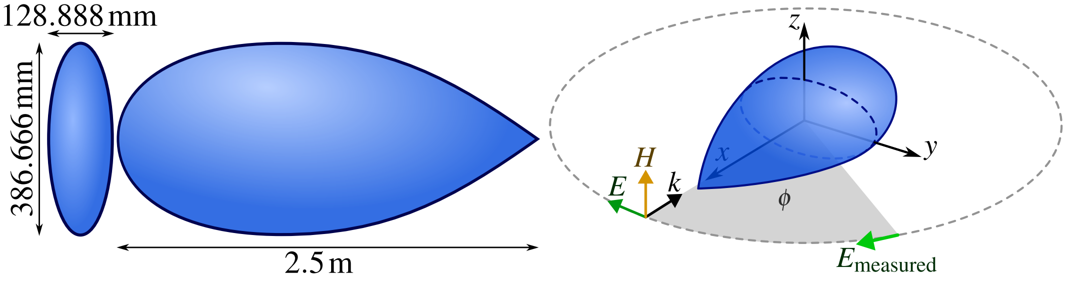

Figure 4.17: Scheme and dimensions of the NASA almond and plane wave incidence conditions. The geometry shown is only a scheme and not an accurate representation.

aims to test the OI-SG against such a case. For this work, the object has been meshed with different space steps: 1.25 mm, 2.5 mm, 5 mm, and 10 mm. In all simulations, the NASA almond is embedded within the finest level grid, and each coarser level is wrapped around the finer one, until reaching the coarsest level, which extends until the computational limits. An illustration of the grids with $\Delta_0 = 10$ mm is depicted in fig. 4.18. In the coarsest level, a plane wave source is set with horizontal polarization (spherical ϕ-axis) directed towards the tip of the NASA almond. Then, the bistatic RCS of the horizontal electric field ($\text{RCS}_{E_{hh}}$) is measured across ϕ. This scheme is depicted in fig. 4.17 (right). The plane wave source has a Gaussian profile with a decay of 3 dB in the frequency that corresponds to $\text{PPW}_{\text{coarse}} = 5$. Nonetheless, the frequency studied in this case is exclusively $f = 1$ GHz, which corresponds to $\lambda \simeq 0.3$ m. For the space steps used, the maximum and minimum values of PPW are, respectively, 240 (for $\Delta = 1.25$ mm) and 30 (for $\Delta = 10$ mm). An accurate result of the RCS as a function of ϕ has been obtained using method of moments (MoM) with a mesh of 41040 triangles with an average length of 8 mm and standard deviation of 5 mm.

To test the accuracy and performance of the OI-SG method in this simulation, two different approaches have been used. First, the finest-level grid is taken with $\Delta_0 = 1.25$ mm, and simulations are performed growing the coarse grid around it with $N_{\text{sg}} \in \{0,1,2,3\}$. Second, the coarsest-level grid is taken with $\Delta_{N_{\text{sg}}} = 10$ mm and simulations are performed refining the NASA almond mesh with $N_{\text{sg}} \in \{0,1,2,3\}$. All simulations have been performed with GTS and LTS. Additionally, it is interesting to observe the effect of the buffer spacing in a LO case such as this. From the results obtained in section 4.3.1, we conclude that a distance $d_{\text{buf}} = 0.12\,\lambda$ is most desirable, which corresponds to different values of n_{buf} depending on the finest-level grid. Finally, it is also intended to provide the error offered by each simulation. For the executions that share the same finest-level space step, we hope to minimize the discrepancies with the case with $N_{\text{sg}} = 0$, and therefore it is taken as the reference. On the other hand, for the executions that share the same coarsest-level space step,

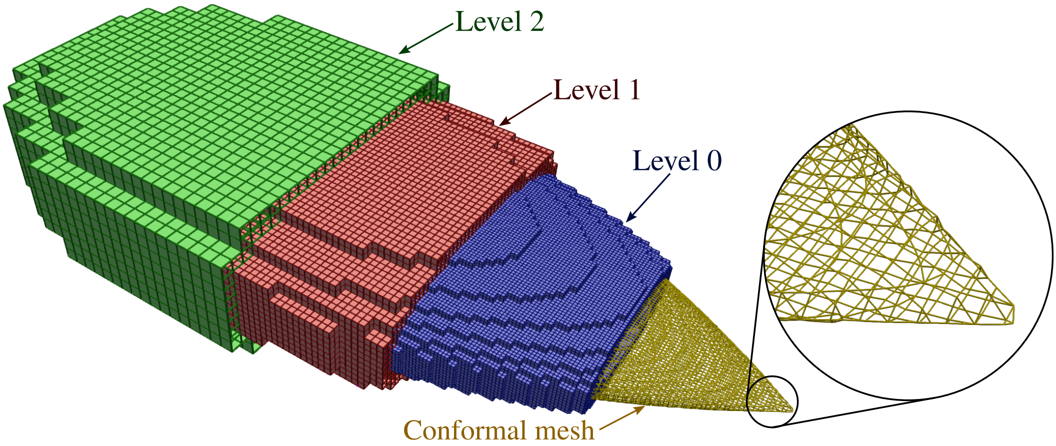

Figure 4.18: Example of the AMR around a NASA almond with a conformal mesh and $\Delta_0 = 10\,\text{mm}$. Levels 0, 1 and 2 have been cropped for illustration purposes.

simulations should get closer to the analytical solution, and therefore the MoM solution is taken as reference. The error is calculated as

$$\text{Error RCS}_{E_{hh}}[\text{dBsm}] = |\text{RCS}_{E_{hh}}[\text{dBsm}] - \text{RCS}_{E_{hh},\text{ref}}[\text{dBsm}]| \,. \tag{4.16}$$

For a common finest-level grid with $\Delta_0 = 1.25\,\text{mm}$: comparison between GTS and LTS is shown in fig. 4.19; comparison between $n_{\text{buf}} = 1$ and $d_{\text{buf}} = 0.12\,\lambda$ is shown in fig. 4.23 for executions with GTS, and in fig. 4.27 for executions with LTS; and the respective errors are shown in fig. 4.20, fig. 4.24, and fig. 4.28. For a common coarsest-level grid with $\Delta_{N_{\text{sg}}} = 10\,\text{mm}$: comparison between GTS and LTS is shown in fig. 4.21; comparison between $n_{\text{buf}} = 1$ and $d_{\text{buf}} = 0.12\,\lambda$ is shown in fig. 4.25 for executions with GTS, and in fig. 4.29 for executions with LTS; and the respective errors are shown in fig. 4.22, fig. 4.26, and fig. 4.30. The respective performance tables of all the mentioned comparisons are shown, in the same order, in tables 4.7–4.12.

The first thing that we may observe in all the provided graphs is that the maximum and minimum values of RCS are not displaced in ϕ regardless of the configuration used. If we compare executions with GTS and LTS by looking at figs. 4.19–4.22, we observe that the error is very similar for lower values of N_{sg}. However, for $N_{\text{sg}} = 3$ we may appreciate that executions with LTS have a significantly larger error. In this regard, it is interesting to note the contrast between the results NASA Almond and other simulation cases with normal incidence, for example, the FSS or the numerical reflection obtained in fig. 3.26, where the error with GTS was slightly worse.

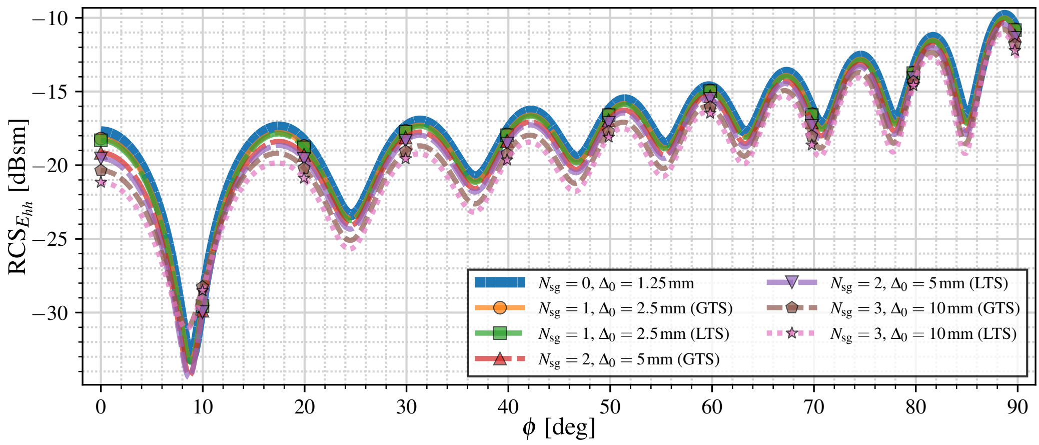

Figure 4.19: Backscattering RCS of the NASA almond for a common finest-level with $\Delta_0 = 1.25$ mm with GTS and LTS.

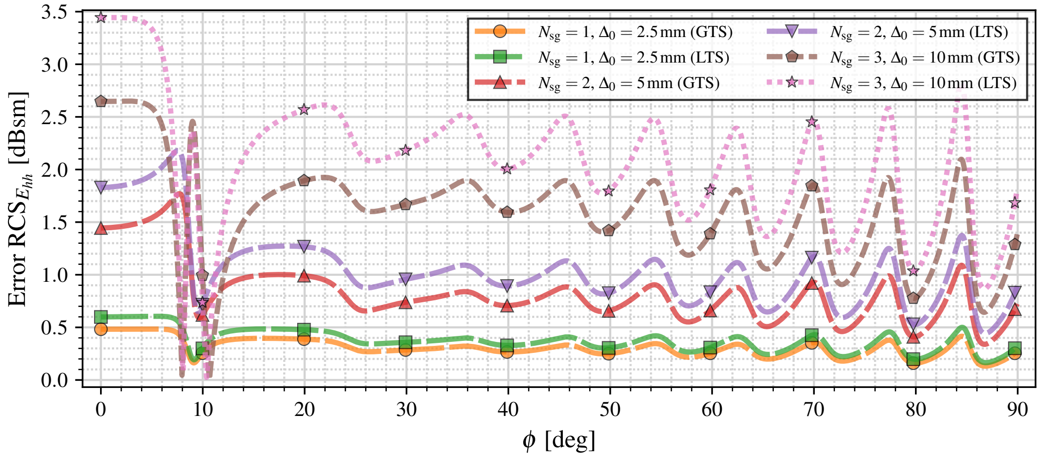

Figure 4.20: Backscattering RCS error of the NASA almond for a common finest-level with $\Delta_0 = 1.25$ mm with GTS and LTS.

Table 4.7: Performance table of the NASA almond with a common finest-level of $\Delta_0 = 1.25$ mm with GTS and LTS. CPET and T_{CPU} have been normalized.

N_{sg}	GTS/LTS	CFLN	$\Delta_{N_{\mathrm{sg}}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	1.25	519388716	1.0000	1.0000
1	GTS	0.93	2.5	113981192	0.2336	0.3270
1	LTS	0.66	2.5	113981192	0.1851	0.2597
2	GTS	0.93	5	52326688	0.1072	0.4258
2	LTS	0.66	5	52326688	0.0949	0.1599
3	GTS	0.93	10	41251394	0.0845	0.1765
3	LTS	0.66	10	41251394	0.0865	0.1577

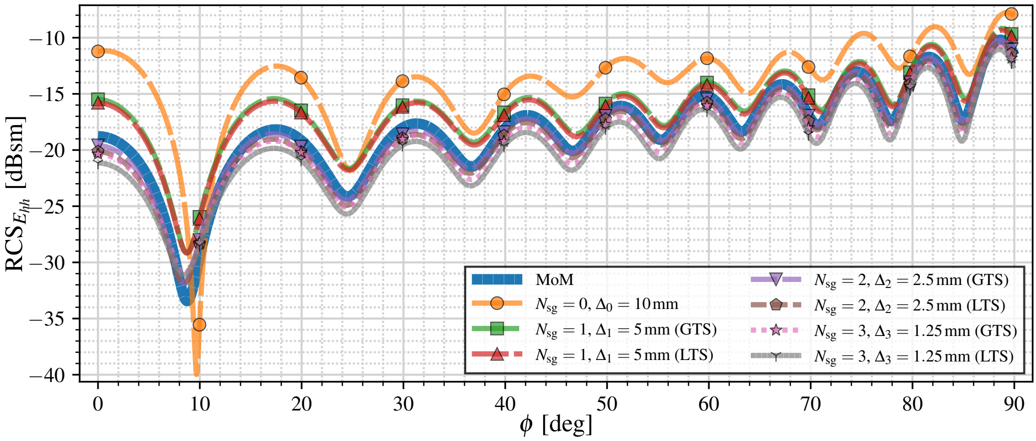

Figure 4.21: Backscattering RCS of the NASA almond for a common coarsest-level with $\Delta_{N_{\mathrm{sg}}} = 10\,\mathrm{mm}$ with GTS and LTS.

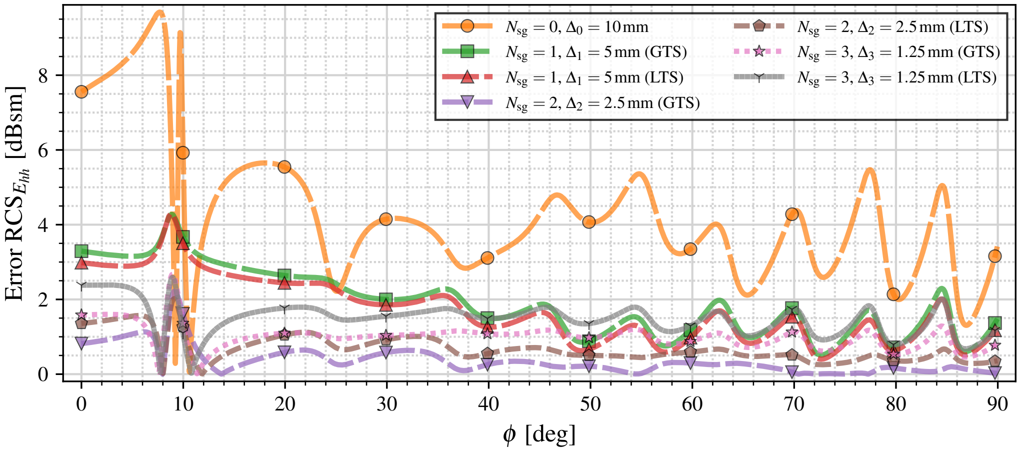

Figure 4.22: Backscattering RCS error of the NASA almond for a common coarsest-level with $\Delta_{N_{\mathrm{sg}}} = 10\,\mathrm{mm}$ with GTS and LTS.

Table 4.8: Performance table of the NASA almond with a common coarsest-level of $\Delta_{N_{\mathrm{sg}}} = 10\,\mathrm{mm}$ with GTS and LTS. CPET and T_{CPU} have been normalized.

N_{sg}	GTS/LTS	CFLN	Δ_0 [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	10	3604146	0.0100	0.0091
1	GTS	0.93	5	5784456	0.0342	0.0274
1	LTS	0.66	5	5784456	0.0284	0.0213
2	GTS	0.93	2.5	13001856	0.1534	0.1395
2	LTS	0.66	2.5	13001856	0.1369	0.1079
3	GTS	0.93	1.25	41251394	0.9735	1.0000
3	LTS	0.66	1.25	41251394	1.0000	0.8970

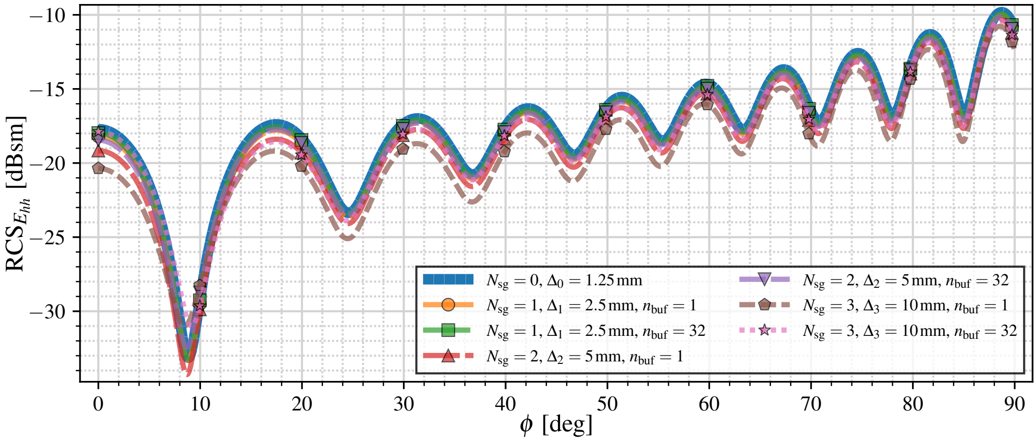

Figure 4.23: Backscattering RCS of the NASA almond for a common finest-level with $\Delta_0 = 1.25$ mm with GTS, with and without buffer spacing.

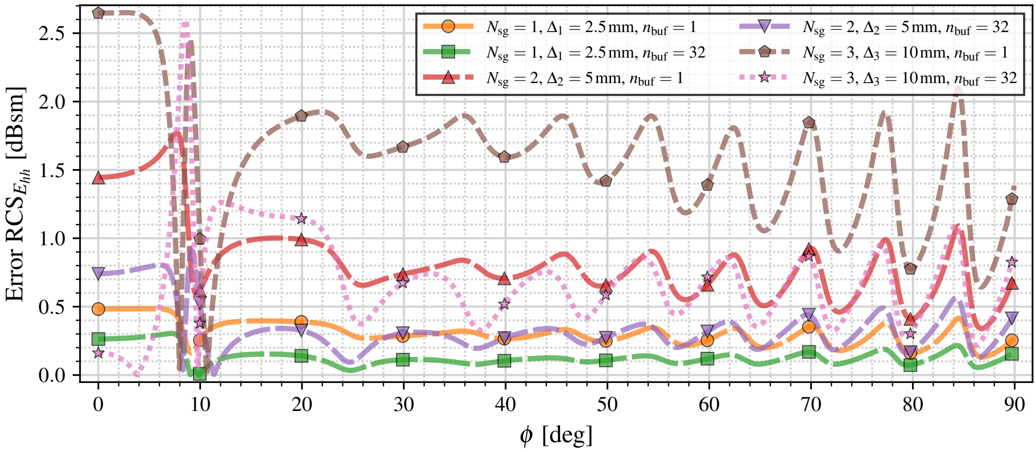

Figure 4.24: Backscattering RCS error of the NASA almond for a common finest-level with $\Delta_0 = 1.25$ mm with GTS, with and without buffer spacing.

Table 4.9: Performance table of the NASA almond with a common $\Delta_0 = 1.25$ mm with GTS, with and without buffer spacing. CPET and T_{CPU} have been normalized.

N_{sg}	n_{buf}	CFLN	$\Delta_{N_{sg}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	1.25	519388716	1.0000	0.9596
1	1	0.93	2.5	113981192	0.2336	0.3138
1	32	0.93	2.5	341353608	0.6996	1.0000
2	1	0.93	5	52326688	0.1072	0.1875
2	32	0.93	5	283493472	0.5810	0.9475
3	1	0.93	10	41251394	0.0845	0.1693
3	32	0.93	10	272319490	0.5581	0.9688

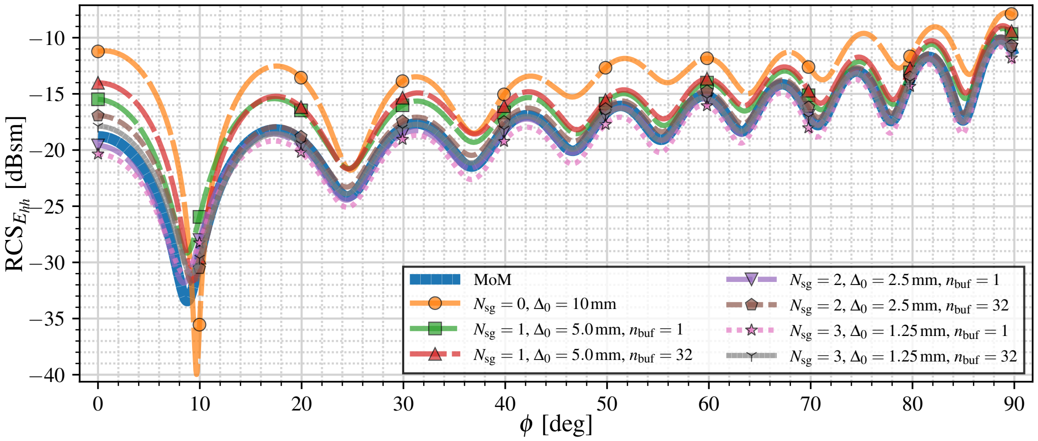

Figure 4.25: Backscattering RCS of the NASA almond for a common coarsest-level with $\Delta_{N_{sg}} = 10$ mm with GTS, with and without buffer spacing.

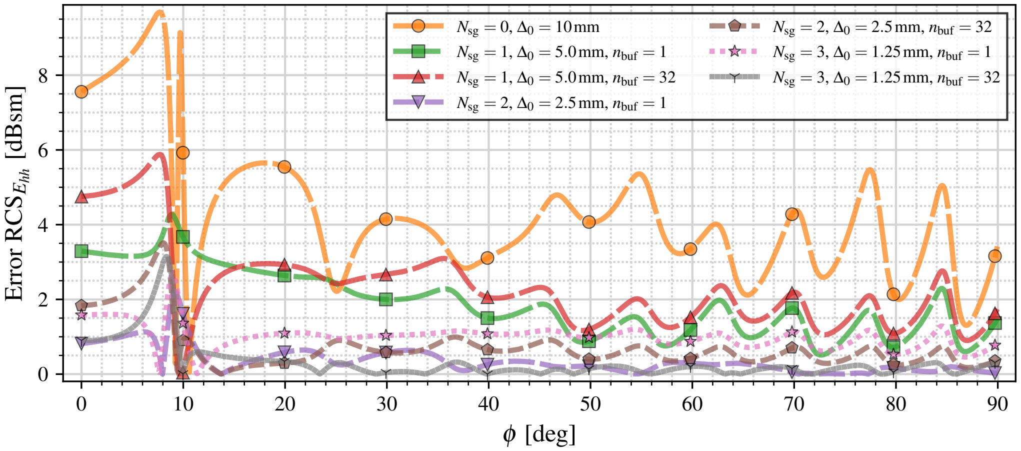

Figure 4.26: Backscattering RCS error of the NASA almond for a common coarsest-level with $\Delta_{N_{sg}} = 10$ mm with GTS, with and without buffer spacing.

Table 4.10: Performance table of the NASA almond with a common $\Delta_{N_{sg}} = 10$ mm with GTS, with and without buffer spacing. CPET and T_{CPU} have been normalized.

N_{sg}	n_{buf}	CFLN	Δ_0 [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	10	3604146	0.0016	0.0017
1	1	0.93	5	5784456	0.0053	0.0048
1	10	0.93	5	10113256	0.0093	0.0080
2	1	0.93	2.5	13001856	0.0239	0.0244
2	16	0.93	2.5	41131424	0.0755	0.0708
3	1	0.93	1.25	41251394	0.1515	0.1748
3	32	0.93	1.25	272319490	1.0000	1.0000

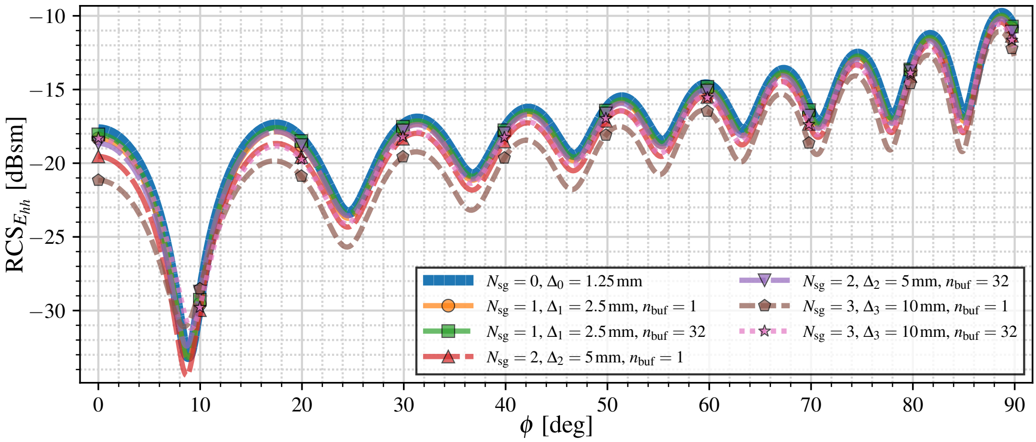

Figure 4.27: Backscattering RCS of the NASA almond for a common finest-level with $\Delta_0 = 1.25\,\mathrm{mm}$ with LTS, with and without buffer spacing.

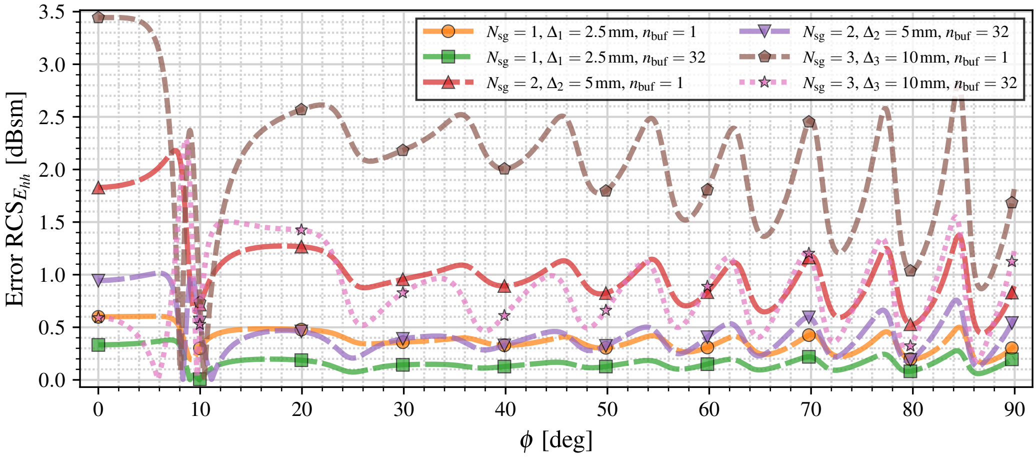

Figure 4.28: Backscattering RCS error of the NASA almond for a common finest-level with $\Delta_0 = 1.25\,\mathrm{mm}$ with LTS, with and without buffer spacing.

Table 4.11: Performance table of the NASA almond with a common $\Delta_0 = 1.25\,\mathrm{mm}$ with LTS, with and without buffer spacing. CPET and T_{CPU} have been normalized.

N_{sg}	n_{buf}	CFLN	$\Delta_{N_{\mathrm{sg}}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	1.25	519388716	1.0000	0.9508
1	1	0.66	2.5	113981192	0.1851	0.2469
1	32	0.66	2.5	341353608	0.7404	1.0000
2	1	0.66	5	52326688	0.0949	0.1521
2	32	0.66	5	283493472	0.6525	0.9584
3	1	0.66	10	41251394	0.0865	0.1499
3	32	0.66	10	272319490	0.6438	0.9489

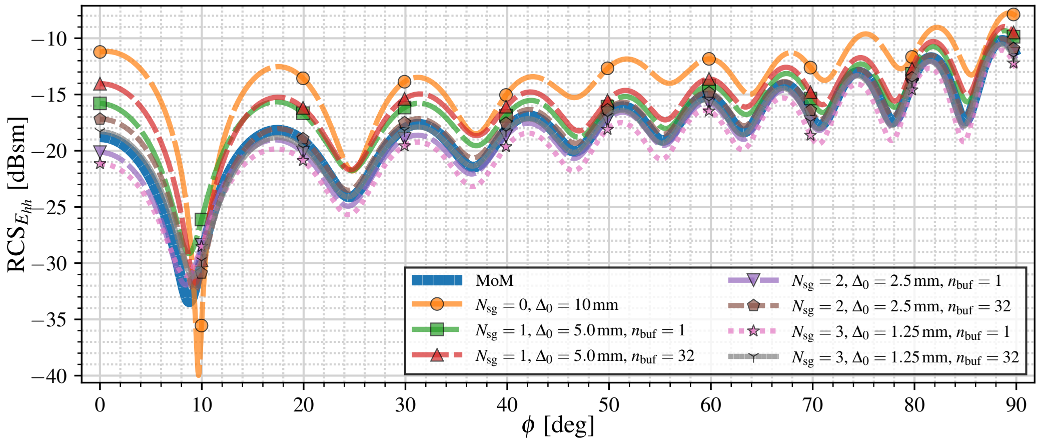

Figure 4.29: Backscattering RCS of the NASA almond for a common coarsest-level with $\Delta_{N_{sg}} = 10$ mm with LTS, with and without buffer spacing.

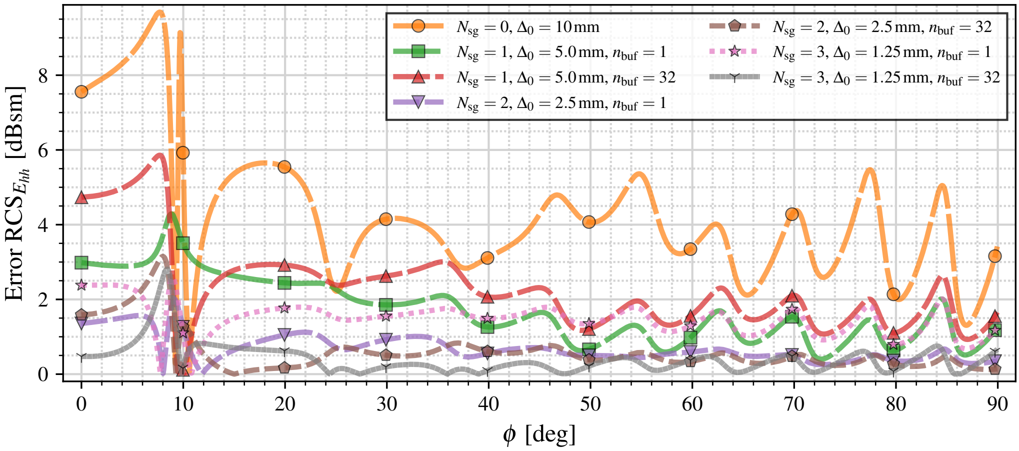

Figure 4.30: Backscattering RCS error of the NASA almond for a common coarsest-level with $\Delta_{N_{sg}} = 10$ mm with LTS, with and without buffer spacing.

Table 4.12: Performance table of the NASA almond with a common $\Delta_{N_{sg}} = 10$ mm with LTS, with and without buffer spacing. CPET and T_{CPU} have been normalized.

N_{sg}	n_{buf}	CFLN	Δ_0 [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	10	3604146	0.0013	0.0016
1	1	0.66	5	5784456	0.0038	0.0037
1	10	0.66	5	10113256	0.0078	0.0070
2	1	0.66	2.5	13001856	0.0184	0.0190
2	16	0.66	2.5	41131424	0.0708	0.0692
3	1	0.66	1.25	41251394	0.1344	0.1580
3	32	0.66	1.25	272319490	1.0000	1.0000

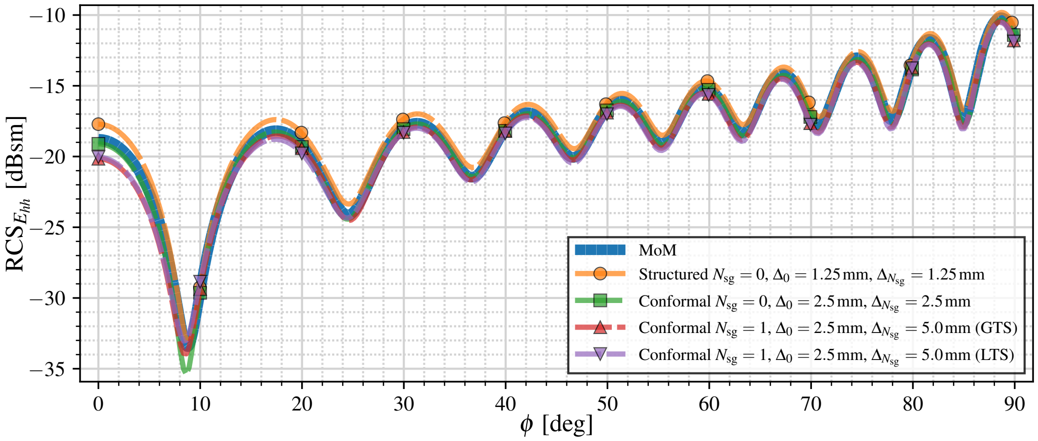

Figure 4.31: Backscattering RCS of the NASA almond for various simulations: (1) standard FDTD with $\Delta = 1.25$ mm; (2) conformal mesh with $\Delta = 2.5$ mm; (3) conformal mesh combined with subgridding using $N_{sg} = 1$, GTS, $\Delta_0 =$ mm and $\Delta_1 = 5$ mm, and (4) same as (3) but using LTS.

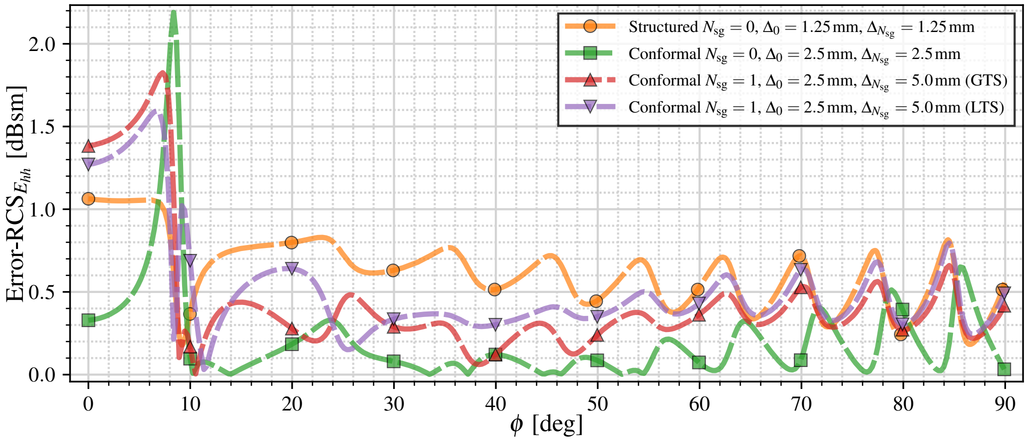

Figure 4.32: Backscattering RCS error of the NASA almond for various simulations with GTS: (1) standard FDTD with $\Delta = 1.25$ mm; (2) conformal mesh with $\Delta = 2.5$ mm; (3) conformal mesh combined with subgridding using $N_{sg} = 1$, GTS, $\Delta_0 =$ mm and $\Delta_1 = 5$ mm, and (4) same as (3) but using LTS.

Table 4.13: Performance table of the NASA almond for various simulations with GTS: (1) standard FDTD with $\Delta = 1.25$ mm; (2) conformal mesh with $\Delta = 2.5$ mm; (3) conformal mesh combined with subgridding using $N_{sg} = 1$, GTS, $\Delta_0 =$ mm and $\Delta_1 = 5$ mm, and (4) same as (3) but using LTS.

N_{sg}	GTS/LTS	CFLN	Δ_0 [mm]	$\Delta_{N_{sg}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	N/A	0.99	1.25	1.25	519388716	1.0000	1.0000
0	N/A	0.66	2.5	2.5	96608160	0.1395	0.6996
1	GTS	0.66	2.5	5.0	42778584	0.0618	0.7378
1	LTS	0.66	2.5	5.0	42778584	0.0409	0.5460

The maximum error is provided by the executions with the larger grid sizes. Particularly, the worst case is $N_{\text{sg}} = 0$ with $\Delta_0 = 10\,\text{mm}$. This error decreases if we refine the NASA Almond sharing the same coarsest-level grid, however, it is also worth noting that the error may increase slightly for $N_{\text{sg}} = 3$ compared to $N_{\text{sg}} = 2$ for $n_{\text{buf}} = 1$, as can be appreciated by looking at fig. 4.22. Interestingly, by observing figs. 4.26 and 4.30, we may note that this effect disappears with an increased value of n_{buf}, as the execution with $n_{\text{buf}} = 32$ and $N_{\text{sg}} = 3$ has an error comparable to the execution with $n_{\text{buf}} = 32$ and $N_{\text{sg}} = 2$. In general terms, we observe that increasing the buffer spacing decreases the error significantly.

For simulations with a common finest-level of $\Delta = 1.25\,\text{mm}$ (figs. 4.19, 4.23, and 4.27), it is clear that the error increases as we reach larger sizes of the coarsest-level grid. This is to be expected because the subgridding boundaries of larger sizes are expected to provide more distortions as $\text{PPW}_{\text{coarse}}$ decreases, and the minimum value reached is $\text{PPW}_{\text{coarse}} = 30$. In these cases, we are interested in looking at the efficiency provided by the subgridding usage. To do so, we need to look at tables 4.7, 4.9, and 4.11. For executions with $n_{\text{buf}} = 1$, memory usage is reduced between a 78% and a 92%, depending on N_{sg}. When applying a buffering distance of $n_{\text{buf}} = 32$, we find that this reduction is lowered to approximately 34% to 48%, which is still an appreciable reduction. The reduction in CPU, on the other hand, depends on the time-stepping scheme. For $n_{\text{buf}} = 1$, T_{CPU} offers a reduction of 59% to 78% for executions with GTS, and 74% to 84% with LTS. Especially for the latest case, these reductions are very significant to prove the efficiency of the OI-SG method. However, when a buffering distance of $n_{\text{buf}} = 32$ is applied, these reductions disappear completely. If we look at tables 4.9 and 4.11, we may observe that T_{CPU} increases when using subgridding in all cases, with only the exception of $N_{\text{sg}} = 3$ with LTS, where a reduction of 0.2% is appreciated.

The sharp geometry present at the tip of the NASA Almond is particularly difficult to capture with a standard FDTD mesh. As has been shown, the usage of subgridding allows us to get more accurate results, however, the buffer spacing required to achieve the best results decreases the efficiency significantly. For this reason, this simulation case has also been executed with a conformal mesh of size $\Delta_0 = 2.5\,\text{mm}$ and $N_{\text{sg}} \in \{0, 1\}$. The execution with $N_{\text{sg}} = 1$ has been run with a buffering distance of $n_{\text{buf}} = 10$. The obtained results are compared with the standard FDTD with $\Delta = 1.25\,\text{mm}$. The errors provided by all cases are also plotted taking as reference the results obtained with MoM. Results and errors are plotted, respectively, in figs. 4.31 and 4.32. The respective performance table is displayed in table 4.13. It is interesting to note that the conformal mesh with a size $\Delta_0 = 2.5\,\text{mm}$

coarser than the standard FDTD with $\Delta = 1.25\,\mathrm{mm}$ already provides significantly better results for most of the values of ϕ. Adding subgridding to the conformal mesh results in a slight increase in the numerical error, which is to be expected since the fine mesh remains unchanged, but the error remains lower than the standard FDTD despite having a lower resolution. Regarding the memory usage, if we compare to the standard FDTD execution, we find that the usage of a conformal mesh with a larger mesh size drops the required memory a 82%, and a 92% when using subgridding. On the other hand, the CPU time is reduced only a 36% with the conformal mesh, and the executions with subgridding depend on the time-stepping scheme: execution with GTS offers a worse performance with a T_{CPU} reduction of 33%, whereas LTS provides better performance with a reduction of 45%. In general terms, we may conclude that the combination of subgridding and conformal meshes provides a clear improvement in both accuracy and efficiency, especially regarding memory usage, as they allow us to use coarser mesh sizes with better geometry capture.

4.4.3 Scattering of the FLAMME stealth object

The simulation case used for this section is known as the FLAMME stealth object [175, 176]. It was designed and shown to the public in 1993 by ONERA as a demonstration of a stealth object. Measurement of the backscattering RCS of the FLAMME object from different angles constitutes a LO problem. Similar to the NASA Almond, this constitutes a challenge to subgridding algorithms such as the OI-SG method.

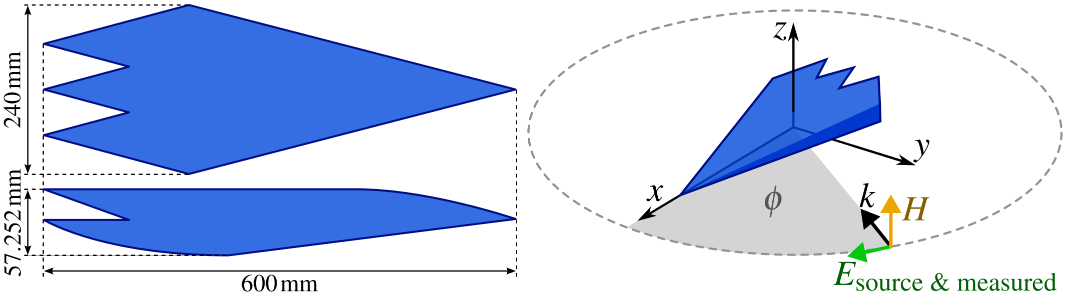

Figure 4.33: Scheme and dimensions (left), and illumination and measurement scheme (right) of the FLAMME stealth object. The geometry shown is only a scheme and not an accurate representation.

A scaled model of the FLAMME object with a length of 600 mm is used for this work. A scheme is depicted in fig. 4.33 (left). The object is entirely made of PEC material. In all simulations, the object is embedded within the finest level grid. The coarsest level has PML boundary conditions with 10 layers. The object is illuminated by a plane wave with a Gaussian profile that decays 3 dB in the frequency that corresponds to $\mathrm{PPW}_{\mathrm{coarse}} = 5$.

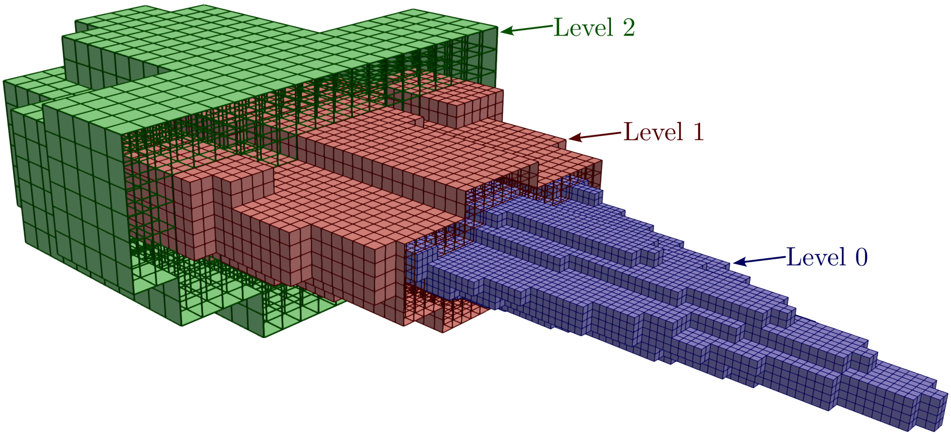

Figure 4.34: Example of the AMR around the FLAMME stealth object with $\Delta_0 = 4.3\,\text{mm}$. Levels 1 and 2 have been cropped for illustration purposes.

The vector that describes the direction of propagation of the plane wave is contained within the XY plane, varying the spherical coordinate ϕ. The electric field of the plane wave source is oriented in the ϕ-direction. For each plane wave source, the backscattering RCS of the ϕ component is measured. This scheme is depicted in fig. 4.33 (right). In short, the backscattering is measured as a function of ϕ, which implies that each point in each graph requires a different simulation, making this case particularly expensive. The only frequency studied in this case is $f = 2.714\,\text{GHz}$, which corresponds to $\lambda \simeq 110\,\text{mm}$. Additionally, for comparison, an accurate solution has been obtained using MoM with a mesh of 472154 triangles with an average length of 1 mm and a standard deviation of $6 \cdot 10^{-2}\,\text{mm}$.

To test the OI-SG method, two approaches have been designed. First, a common finest level grid with $\Delta_0 = 0.5\,\text{mm}$ has been used for $N_{\text{sg}} \in \{1,2,3,4\}$, which corresponds to $\text{PPW}_{\text{fine}} \simeq 221$. Results are shown in fig. 4.35 and the error is plotted in fig. 4.36 calculated using eq. (4.16) and taking as reference the solution obtained with MoM. Second, a common coarsest level grid with $\Delta_0 = 2.0\,\text{mm}$ has been used for $N_{\text{sg}} \in \{1,2,3,4\}$, which corresponds to $\text{PPW}_{\text{coarse}} \simeq 55$. All simulations in this section have been performed with LTS. Results are shown in fig. 4.37 and the error is plotted in fig. 4.38 taking as reference the solution obtained with MoM.

By looking at figs. 4.35 and 4.37, we may observe that the backscattering has a minimum peak of $-60\,\text{dB}$ at approximately $\phi \simeq 63°$. Fig. 4.35 show that this peak is accurately reproduced by the OI-SG method in all cases for a mesh of $\Delta_0 = 0.5\,\text{mm}$. If we look at fig. 4.35, we may observe a low error of $\sim 1\,\text{dBsm}$ for all the frequencies except for the mentioned LO peak. The greatest errors observed are of $5 \sim 7\,\text{dB}$ and correspond to

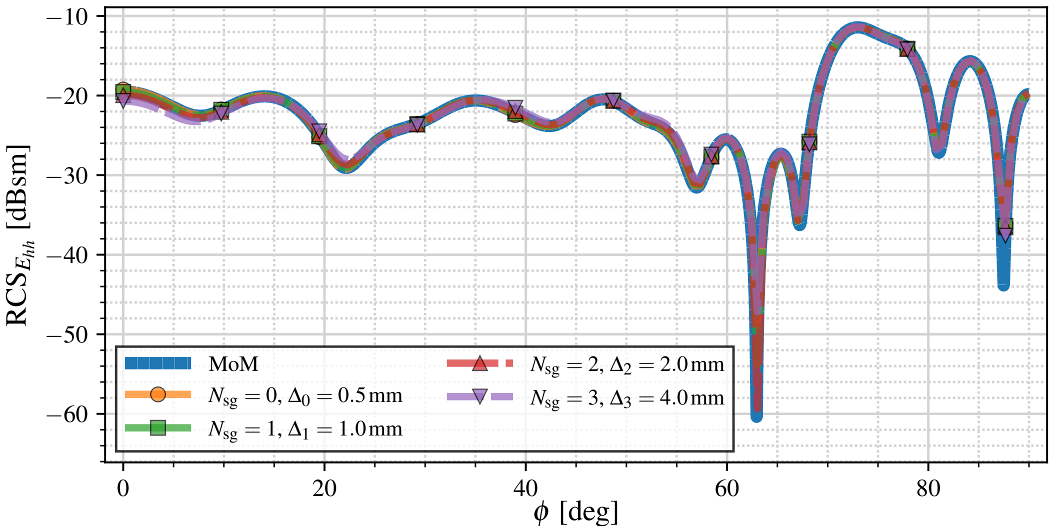

Figure 4.35: Backscattering RCS of the FLAMME object for a common finest level grid of $\Delta_0 = 0.5\,\mathrm{mm}$.

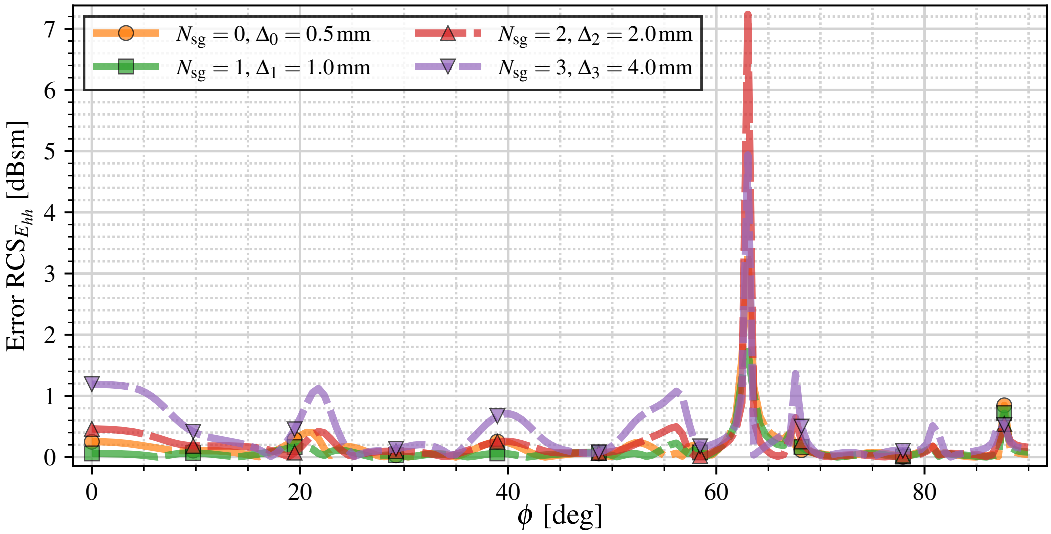

Figure 4.36: Backscattering RCS error of the FLAMME object for a common finest level grid of $\Delta_0 = 0.5\,\mathrm{mm}$.

Table 4.14: Performance table of the FLAMME stealth object for a common finest level of $\Delta_0 = 0.5\,\mathrm{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\mathrm{sg}}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	0.99	0.5	116193912	1.0000	1.0000
1	0.66	1.0	25801280	0.1849	0.3200
2	0.66	2.0	11730949	0.0864	0.2752
3	0.66	4.0	8667145	0.0750	0.2543

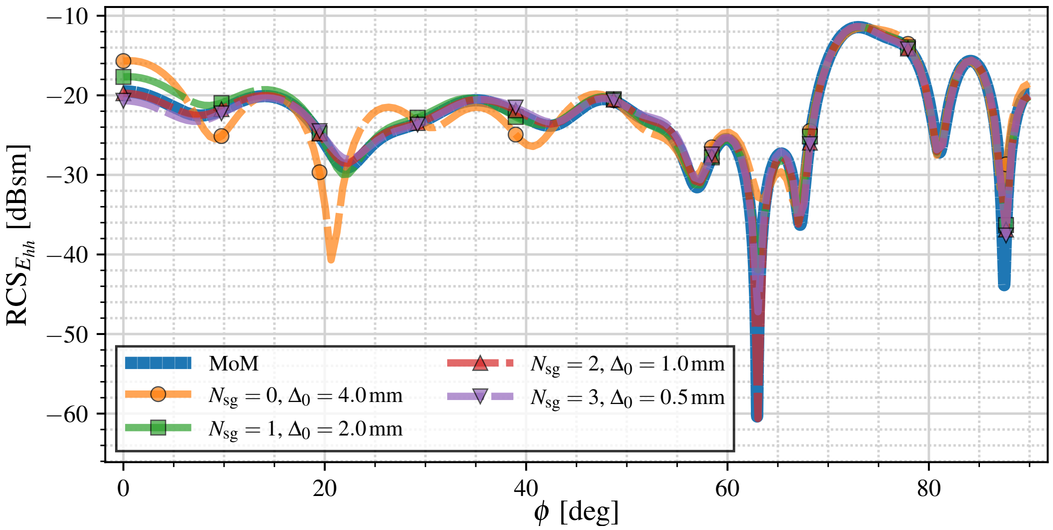

Figure 4.37: Backscattering RCS of the FLAMME object for a common coarsest level grid of $\Delta_{N_{sg}} = 4.0\,\mathrm{mm}$.

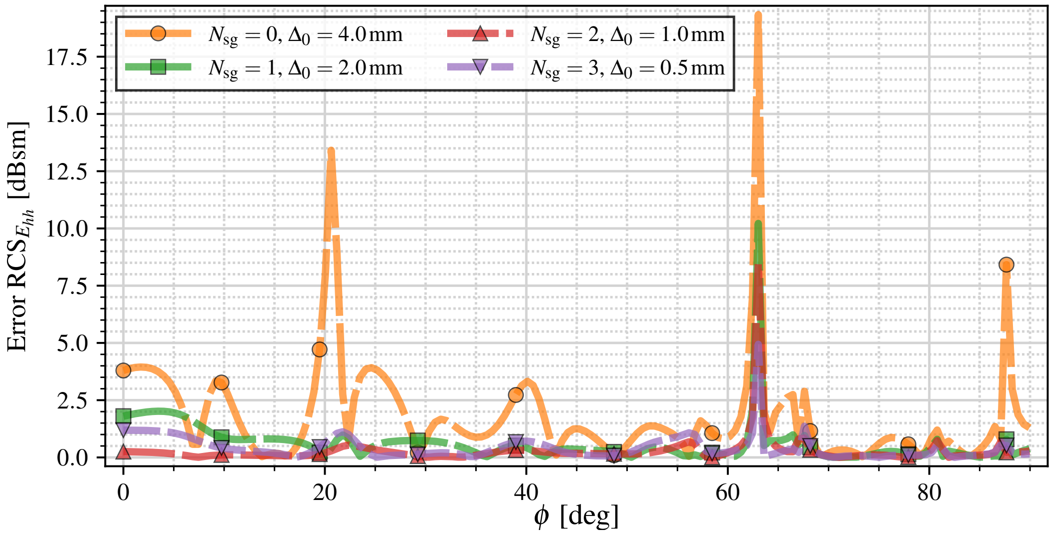

Figure 4.38: Backscattering RCS error of the FLAMME object for a common coarsest level grid of $\Delta_{N_{sg}} = 4.0\,\mathrm{mm}$.

Table 4.15: Performance table of the FLAMME stealth object for a common coarsest level of $\Delta_{N_{sg}} = 4.0\,\mathrm{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	Δ_0 [mm]	N_{cells}	CPET	T_{CPU}
0	0.99	4.0	1580531	0.0211	0.0120
1	0.66	2.0	1887368	0.0459	0.0379
2	0.66	1.0	3125657	0.1425	0.1442
3	0.66	0.5	8667145	1.0000	1.0000

$N_{\text{sg}} \in \{2,3\}$, whereas $N_{\text{sg}} = 1$ offers a lower error of 3 dB. However, it is worth noting that these errors correspond to narrow resonance peaks, and therefore they are not particularly significant. In general terms, the observed errors are low and the LO peaks can be accurately resolved, making the OI-SG method suitable for this case.

By looking at fig. 4.37, it is clear that simulations with more refinement resolve the LO RCS more accurately than the case with $N_{\text{sg}} = 0$, regardless of the number of subgridding levels. It is interesting to note, however, that for lower values of ϕ, fig. 4.36 shows that $N_{\text{sg}} = 2$ behaves slightly better than $N_{\text{sg}} = 3$ despite having a coarser mesh. However, the lowest peak, present at $\phi \simeq 63^\circ$, finds the lowest error for $N_{\text{sg}} = 3$, with a discrepancy of approximately 5 dB compared to the MoM solution.

4.5 EMC test cases

4.5.1 Shielding effectiveness of a conductive spherical shell

This simulation case consists of a spherical shell with thickness 20 mm and conductivity 5 S/m. Results of this case for the OI-SG algorithm were presented in [46] and [48]. The shell is entirely contained within the finest level grid. PML boundary conditions with 10 layers are applied at the coarsest level. The spherical shell is illuminated by a plane wave directed towards the z-axis and an electric polarization of $\hat{p}_e = (1,1,0)/\sqrt{2}$. The plane wave has a Gaussian pulse that decays 3 dB on the frequency corresponding to a $\text{PPW}_{\text{coarse}} = 5$. Then, the SE is calculated by taking a probe at the geometrical center of the sphere and employing eq. (4.2).

A conductive spherical shell with these characteristics can be simulated in various ways. First, it can be modeled directly with a standard structured FDTD using the effective material technique from section 4.1.3. Additionally, the shell can be modeled using the SGBC thin panel method explained in section 2.2.2, which can be done using a purely structured mesh or a conformal mesh. For this section, all three approaches have been simulated for comparison with the OI-SG method. Simulations with subgridding have been performed only with LTS and $n_{\text{buf}} = 1$. First, to test the convergence of the subgridding refinement to the analytical solution, results in fig. 4.39 are shown for $N_{\text{sg}} \in \{1,2,3,4\}$ with a common coarsest level grid with $\Delta_{N_{\text{sg}}} = 40\,\text{mm}$. This corresponds to $\text{PPW}_{\text{coarse}} \simeq 7.5$ for the maximum measured frequency 1 GHz. Results are compared with the analytical solution (obtained from [177]), structured with effective materials, structured SGBC, and conformal SGBC. Second, to test the effect of the orthogonalization parameter δ_{r} on the accuracy, results are shown in fig. 4.40 for $\delta_{\rm r} \in \{0, 0.33, 0.5\}$. In this case, all simulations have $N_{\rm sg} = 2$ and $\Delta_0 = 10\,{\rm mm}$, which corresponds to $\text{PPW}_{\rm fine} \simeq 30$ for the maximum measured frequency. Results are compared with the analytical solution. Third, the effect of LECT is tested by obtaining results of the various combinations of $N_{\rm sg}$ and the proposed LECT configurations in table 3.2, which are plotted in fig. 4.41. These simulations also have a common finest level grid with $\Delta_0 = 10\,{\rm mm}$. Results are compared with the analytical solution. A performance table is shown in table 4.16.

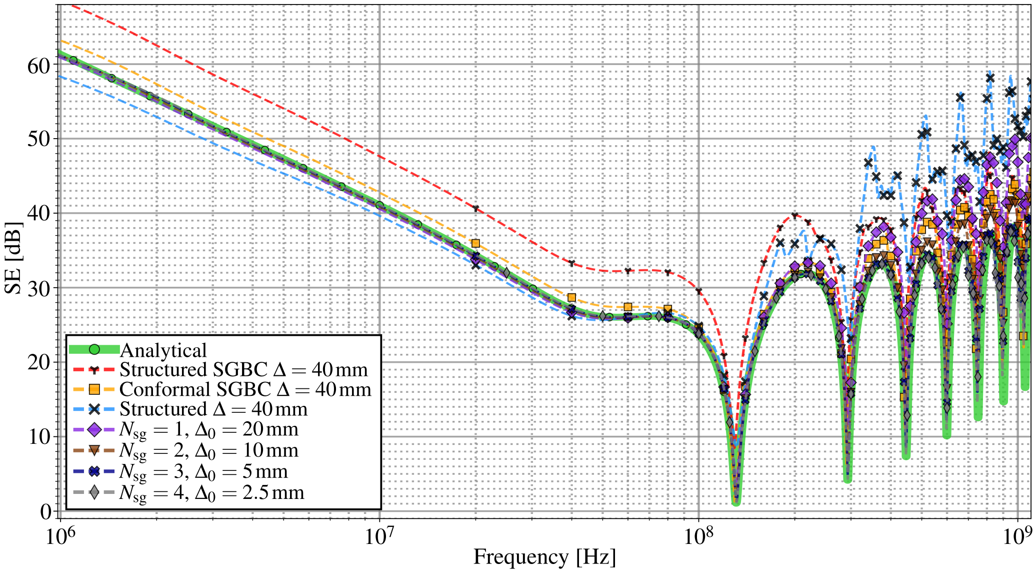

Figure 4.39: SE results for the conductive sphere for $N_{\rm sg} \in \{1,2,3,4\}$ with LTS and $n_{\rm buf} = 1$. Results are compared with the analytical solution, structured SGBC, conformal SGBC and usual FDTD. All simulations have a common coarsest level grid with $\Delta_{N_{\rm sg}} = 40\,{\rm mm}$.

Table 4.16: Performance table of the conductive sphere. CPET and $T_{\rm CPU}$ have been normalized.

$N_{\rm sg}$	LECT	CFLN	Δ_0 [mm]	$\Delta_{N_{\rm sg}}$ [mm]	$N_{\rm cells}$	CPET	$T_{\rm CPU}$
0	No	0.99	40	40	857375	0.0011	0.0013
1	No	0.66	20	40	1364480	0.0035	0.0038
2	No	0.66	10	40	3213688	0.0178	0.0189
3	No	0.66	5	40	11268605	0.1416	0.1425
4	No	0.66	2.5	40	38565597	1.0000	1.0000
1	LECT-1	0.80	10	20	4935680	0.0217	0.0178
2	LECT-1	0.80	10	40	3213688	0.0149	0.0158
1	LECT-2	0.90	10	20	4935680	0.0192	0.0158

By looking at fig. 4.39, we may observe that the result converges to the analytical

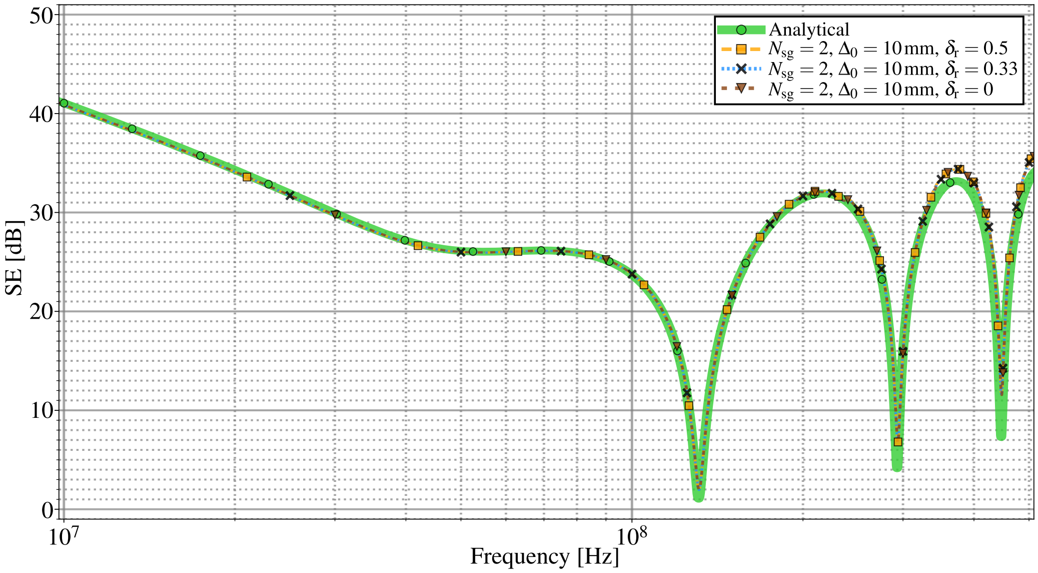

Figure 4.40: SE results for the conductive sphere for $\delta_r \in \{0, 0.33, 0.5\}$ for $N_{sg} = 2$ and $\Delta_0 = 10\,\mathrm{mm}$ with LTS. Results are compared with the analytical solution.

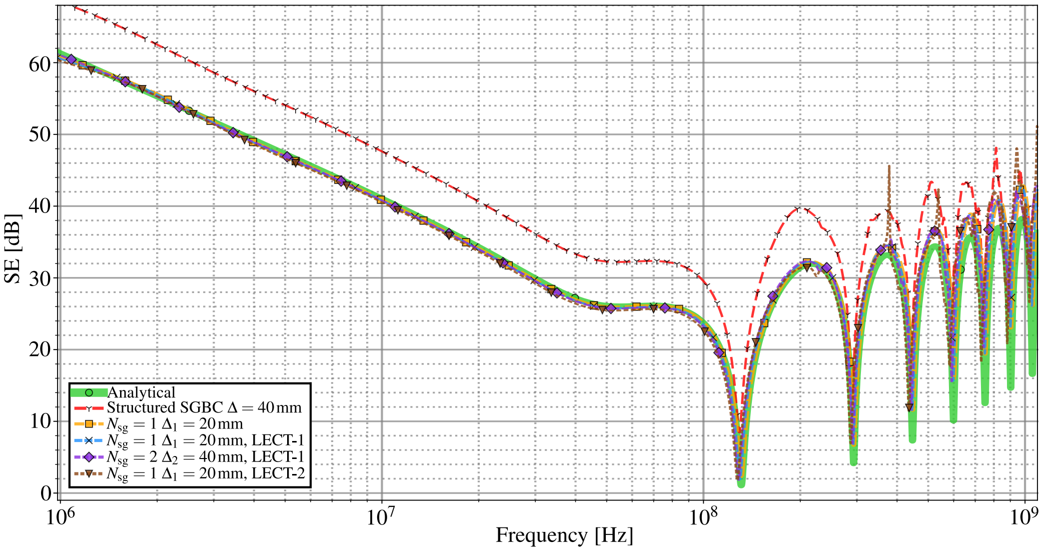

Figure 4.41: SE results for the conductive sphere for various combinations of N_{sg} and LECT with LTS. Results are compared with the analytical solution and structured SGBC. All simulations with subgridding have a common finest level grid with $\Delta_0 = 10\,\mathrm{mm}$.

solution as the subgridding refinement increases. Furthermore, we also observe that the SGBC thin panel model is particularly inaccurate when used in a structured mesh, whereas it provides a better behavior than the effective material technique when used in a conformal mesh. However, the OI-SG algorithm offers more accurate results when a fine enough refinement is used, which can be particularly observed between the highest-frequencies resonances for $N_{\text{sg}} \in \{3,4\}$. In fig. 4.40, it is observed that the difference in accuracy between various values of δ_{r} is negligible. Finally, from fig. 4.41, we may observe that LECT-1 provides negligible discrepancies compared to the simulations without LECT. On the other hand, LECT-2 shows some additional non-physical peaks in the SE, which makes it not suitable for this case.

4.5.2 Shielding effectiveness of a cage with thin curved slots

This case consists of measuring the SE provided by an enclosing PEC box with circular thin slots. This geometry was originally proposed in [178], and was built and measured at INTA facilities. A scheme and a real photograph of the box are depicted in fig. 4.42. The experimental measurements and the photograph were provided courtesy of INTA.

Results of the real measurements and the SE obtained by different numerical methods, including the OI-SG algorithm are published in [30]. The box has dimensions $(480 \times 480 \times 1000)$ mm. Two slots with a semi-circular arc shape are located at one of the square sides of the box and separated 20 mm. Both slots have an internal and external radius of, respectively, 198.75 mm and 201.25 mm, implying slots of width 2.5 mm. For the measurements setup, the box is confined within a reverberant chamber, which is simulated in FDTD by PEC boundary conditions. The box is illuminated by a set of plane wave sources with random directions of propagation to simulate a generic environmental EM influence and the SE is calculated using eq. (4.2).

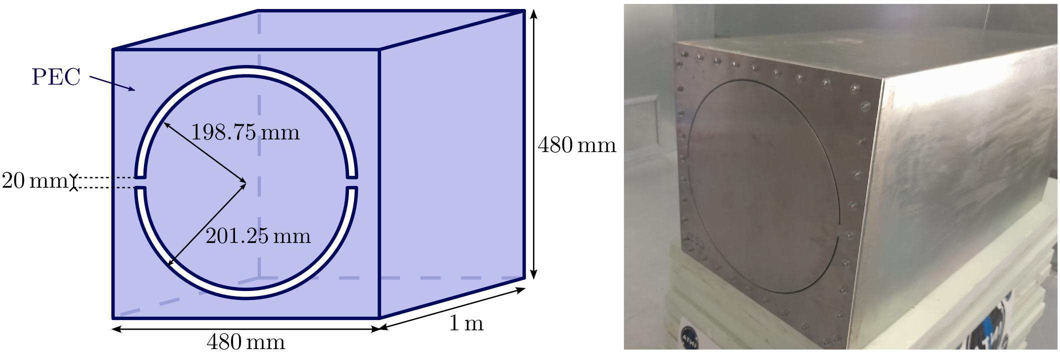

Figure 4.42: Scheme (left) and photo (right) of the PEC box simulated whose SE is measured.

To model the slot, different approaches are used and compared. First, a standard FDTD grid is used with a space step of $\Delta = 1.25\,\text{mm}$. Second, a more efficient simulation with subgridding is used with maximum level $N_{\text{sg}} = 1$, and space steps $\Delta_0 = 1.25\,\text{mm}$ and $\Delta_1 = 2.5\,\text{mm}$. Third, a simulation with a coarser grid of $\Delta = 5\,\text{mm}$ is used, modeling the slots using the DMMA model [30]. Last, a simulation with the same grid $\Delta = 5\,\text{mm}$ is used to model the slots with a conformal mesh (see section 2.2.1). Additionally, all computational results are compared by experimental measurements. Results are shown in fig. 4.43 and a performance table is shown in table 4.17. It is worth noting that the executions with conformal mesh and the DMMA model have performances with negligible differences.

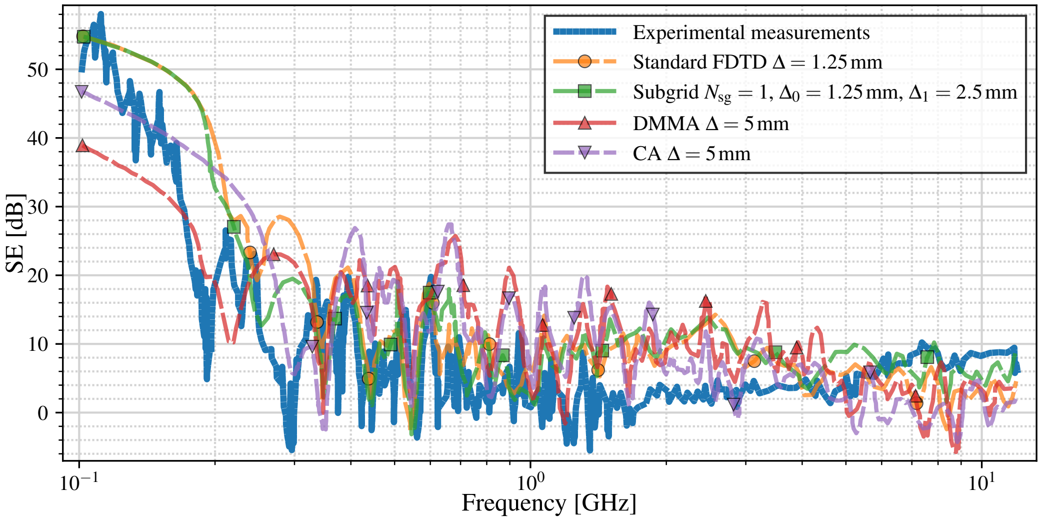

Figure 4.43: Results of the SE of the PEC box with thin slots in a reverberant chamber.

Table 4.17: Performance table of the PEC box with thin slots in a reverberant chamber. CPET and T_{CPU} have been normalized.

N_{sg}	Δ_0 [mm]	Δ_1 [mm]	CFLN	N_{cells}	CPET	T_{CPU}
0	1.25	1.25	0.99	233694720	1.0000	1.0000
0	5	5	0.80	3651480	0.0048	0.0044
1	1.25	2.5	0.66	30297376	0.1151	0.1508

Similar tendencies can be observed between the lines that correspond to the standard FDTD and the execution with subgridding in fig. 4.43. Since they share the same finest-level domain, an improvement in efficiency is expected to be found. This can be confirmed in table 4.17, where a $\sim 88\%$ reduction in CPET and a $\sim 84\%$ reduction in CPU have been reported. However, we must note that the executions with DMMA and conformal with

$\Delta = 5\,\mathrm{mm}$ provide a $\sim 99.6\%$ reduction in CPU time, thus making the OI-SG method less competitive in this example.

4.5.3 An aeronautical application: HIRF effects in the EV55 aircraft

As time passes, modern aircraft increasingly depend on electronic circuitry for communication and automation. Approaches such as the so-called fly-by-wire (FBW) systems, which aim to replace older mechanical systems to automate many of the tasks related to aircraft control, are becoming more common. All these systems are susceptible to external EM interference, thus making EMC studies particularly relevant within the context of aircraft design. In this context, several types of EM contamination are distinguished: lightning indirect effects (LIE) are currents induced by lightning strikes; high intensity radiated field (HIRF) are interferences due to external communication signals such as TV, radio, 5G, radars, etc.; and non-nuclear electromagnetic pulse (NN-EMP) are intentional destructive pulses meant to be used as a weapon. These EM pulses can potentially create interferences in the inner part of the aircraft by inducing currents in the fuselage or by penetrating through the apertures. The capacity of an aircraft to keep functionality in HIRF events is a necessary condition to obtain certification by the authorities, and for this reason the aircraft certification guide [179] proposes different tests to measure the HIRF effects. In particular, this section focuses on simulating a *radiated test*, which consists of illuminating the model by a set of specific EM fields and measuring the transference to fields and currents in different parts of the fuselage or the inner region of the aircraft.

The used model in this section was provided by EVEKTOR under the HIRF-SE project, for which other results have been published in the past [58, 180]. It has dimensions $16.5897\,\mathrm{m} \times 4.4253\,\mathrm{m} \times 16.1072\,\mathrm{m}$. In the interior of the aircraft, some grounded cables are placed with cylindrical shape of radius $30\,\mathrm{mm}$ and thickness $2\,\mathrm{mm}$, which is made of a material with a linear conductivity density of $1.263 \cdot 10^{-3}\,\Omega/\mathrm{m}$. The fuselage is modeled by a PEC conformal mesh and the cables by conformal SGBC. The geometry is embedded within the finest-level grid, which is wrapped by the successive coarser levels until reaching the coarsest-level N_{sg} (an example is depicted in fig. 4.44). At the coarsest level, a plane wave source is set with a Gaussian pulse with a decay of $3\,\mathrm{dB}$ at the frequency that corresponds to $\mathrm{PPW}_{\mathrm{coarse}} = 5$. The orientation of the plane wave is such that it comes from the front of the aircraft, tilted 45° upwards (depicted in fig. 4.45). The computational domain is truncated with PML boundary conditions with 10 layers in all boundaries.

To measure the HIRF effects, based on the work presented in [58, Ch. 5], two probes

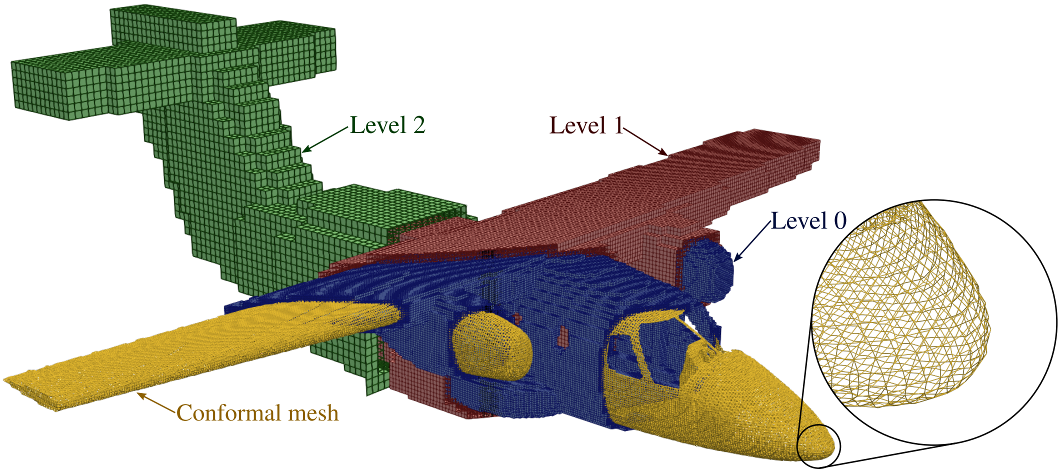

Figure 4.44: Example of the AMR around the EV55 aircraft with $\Delta_0 = 32\,\text{mm}$. Levels 0, 1 and 2 have been cropped for illustration purposes.

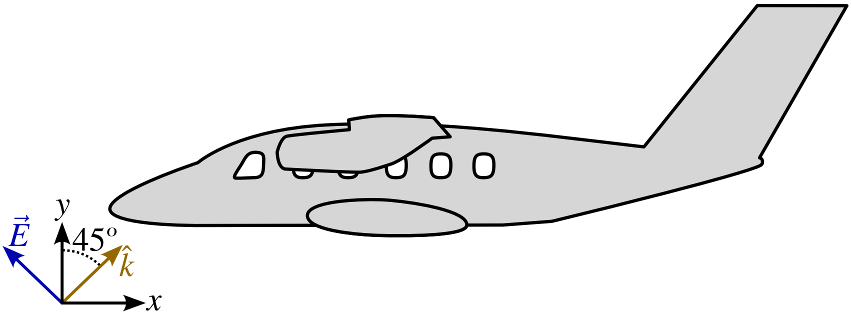

Figure 4.45: Illumination scheme of the EV55 aircraft.

have been placed and measured in different parts of the aircraft. These probes are depicted in fig. 4.46. The first probe measures the bulk current amperage I_{BC} that flows through one of the cables located at the front of the aircraft, inside it. It is calculated from the surrounding magnetic fields as

$$I_{\text{BC}} = \oint \vec{H} \cdot d\vec{l}, \tag{4.17}$$

which is approximated by finite integration. The second probe measures the surface current density J_{s} induced on a point of the fuselage located at the top of one wing. It is calculated from the magnetic fields $\vec{H}_1$ and $\vec{H}_2$ located at both sides of surface:

$$J_{\text{s}} = \left| \left(\vec{H}_2 - \vec{H}_1 \right) \times \hat{n} \right|, \tag{4.18}$$

where $\hat{n}$ is a unit-norm vector normal to the surface.

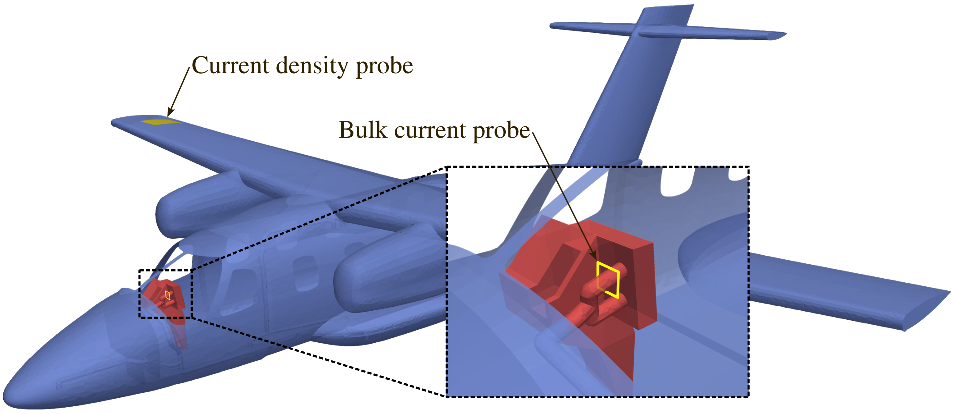

Figure 4.46: Illustration of the probes used in the EV55 simulation case.

In both cases, a transfer function is calculated to measure the effect as a function of the external radiation. For the bulk current probe, the transfer function T_{BC} is defined as

$$T_{\mathrm{BC}}(f) = \frac{|I_{\mathrm{BC}}(f)|^2}{|E_{\mathrm{PW}}(f)|^2}, \tag{4.19}$$

where f is the frequency and E_{PW} is the electric field of the incident plane wave. For the surface current density probe, the transfer function T_{s} is defined as

$$T_{\mathrm{s}} = \frac{|J_{\mathrm{s}}|^2}{|H_{\mathrm{PW}}|^2}. \tag{4.20}$$

It is worth noting that eq. (4.20) is adimensional, but eq. (4.19) is not. Nonetheless, the relation between the incident plane wave field and the induced current must be linear, and thus eq. (4.19) does not depend on the intensity of the incident plane wave.

Two sets of simulations have been prepared for this section. In all cases, simulations have been executed with CFLN $= 0.5$ due to limitations provided by the conformal mesh, and all of them use LTS.

The first set consists of simulations with a common finest-level grid with $\Delta_0 = 8\,\mathrm{mm}$ and $N_{\mathrm{sg}} \in \{0,1,2,3\}$. Additionally, the error has been calculated following eq. (4.16) taking as a reference the case with $N_{\mathrm{sg}} = 0$. Results and errors for T_{BC} are shown, respectively, in figs. 4.47 and 4.48; and results and errors for T_{s} are shown, respectively, in figs. 4.49 and 4.50. The performance table is shown in table 4.18.

The second set consists of simulations with a common finest-level grid with $\Delta_{N_{\mathrm{sg}}} = 32\,\mathrm{mm}$ and $N_{\mathrm{sg}} \in \{0,1,2\}$. Results for T_{BC} are shown in fig. 4.47, and for T_{s} in fig. 4.52.

The performance table is shown in table 4.19.

Table 4.18: Performance table of the EV55 simulation with LTS and a common finest-level grid of $\Delta_0 = 8\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	$\Delta_{N_{\text{sg}}}$ [mm]	N_{cells}	CPET	T_{CPU}
0	0.5	8	2186832132	1.0000	1.0000
1	0.5	16	357287296	0.0904	0.1413
2	0.5	32	95384512	0.0249	0.0741
3	0.5	64	56146102	0.0199	0.0730

Table 4.19: Performance table of the EV55 simulation with LTS and a common finest-level grid of $\Delta_0 = 8\,\text{mm}$. CPET and T_{CPU} have been normalized.

N_{sg}	CFLN	Δ_0 [mm]	N_{cells}	CPET	T_{CPU}
0	0.5	32	45017280	0.2066	0.0754
1	0.5	16	58955520	0.3118	0.1835
2	0.5	8	95384512	1.0000	1.0000

The behavior in HIRF scenarios is typically divided into low, medium and high frequencies. As is discussed next, this behavior is clearly shown in figs. 4.47, 4.49, 4.51 and 4.52.

At low frequencies ($< 10\,\text{MHz}$), EM waves cannot propagate inside the plane, and thus cannot penetrate the apertures of the vehicle. At these frequencies, the currents induced on the cables can be modeled by an LR series circuit [181–183]: when reaching the aircraft, EM waves induce a current on apertures such as the windows by Lenz's Law, which then creates a dissipative field inside the aircraft that further induces currents onto the cables. This LR circuit is characterized by a cutoff frequency f_c,

$$f_c = \frac{R}{2\pi L}, \tag{4.21}$$

where R and L are, respectively, the circuit resistance and inductance. It is worth noting that this model comes from a simplified version of the proposed scenario, and it is therefore limited, especially as we get closer to medium frequencies. If we look at the lowest frequencies from figs. 4.47, 4.49, 4.51 and 4.52, we observe a constant value of the transmission functions, therefore suggesting that $f_c < 1\,\text{MHz}$. In this regard, all the graphs exhibit the same behavior with very low discrepancies, regardless of N_{sg}.

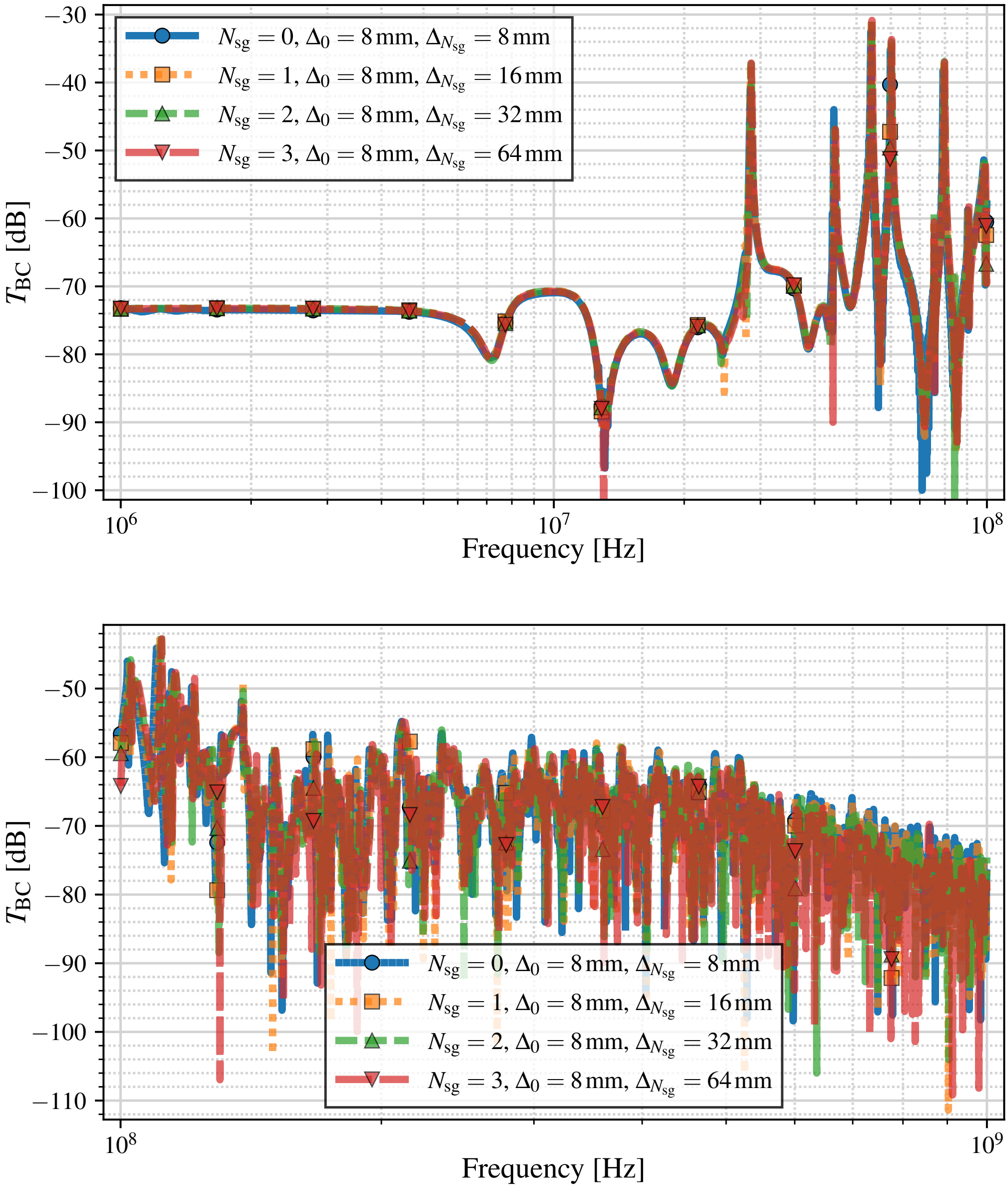

Figure 4.47: T_{BC} for a common finest-level grid of $\Delta_0 = 8\,mm$. Plots are split into low and medium frequencies (upper), and high frequencies (lower).

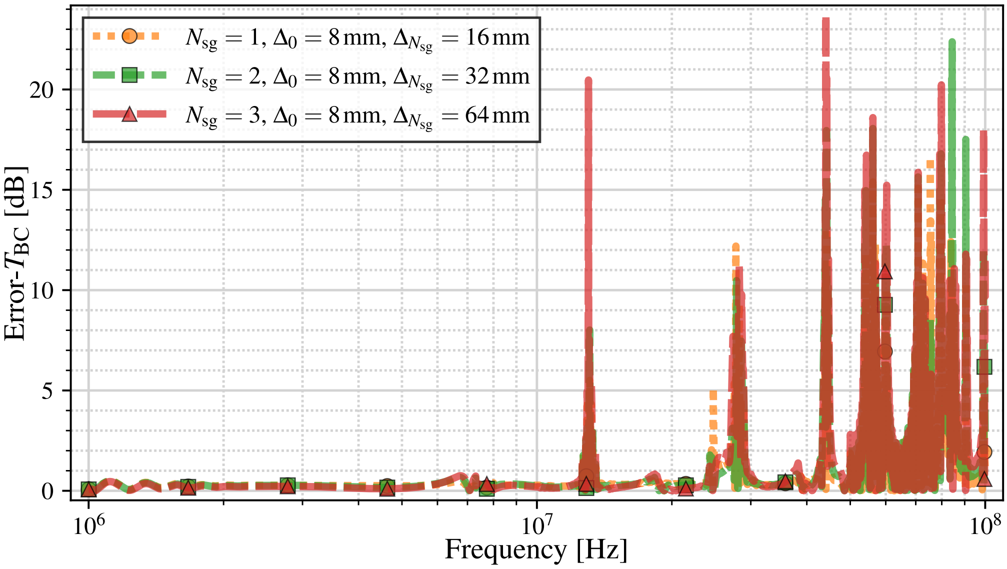

Figure 4.48: T_{BC} for a common finest-level grid of $\Delta_0 = 8\,mm$.

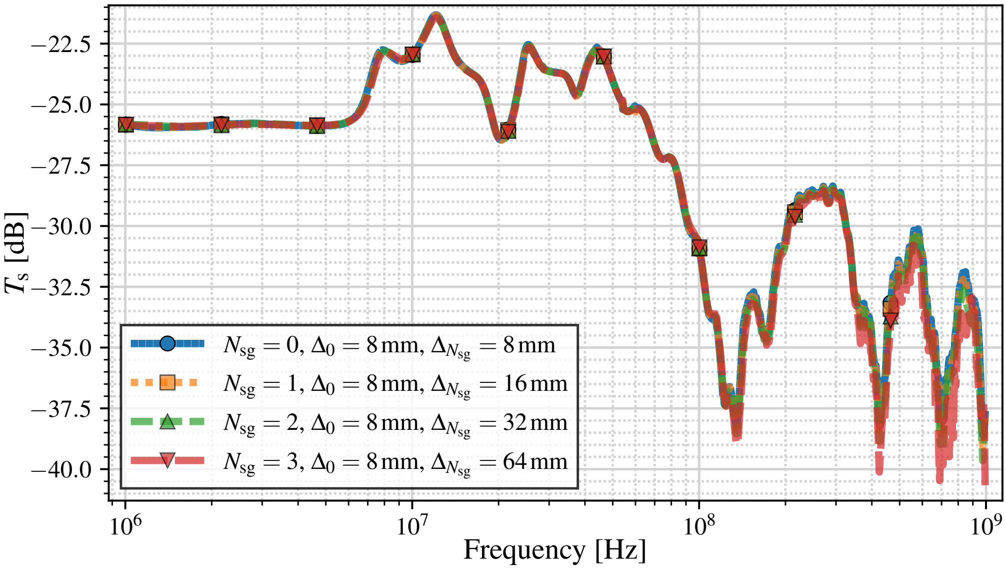

Figure 4.49: T_s for a common finest-level grid of $\Delta_0 = 8\,mm$. Plots are split into low and medium frequencies (upper), and high frequencies (lower).

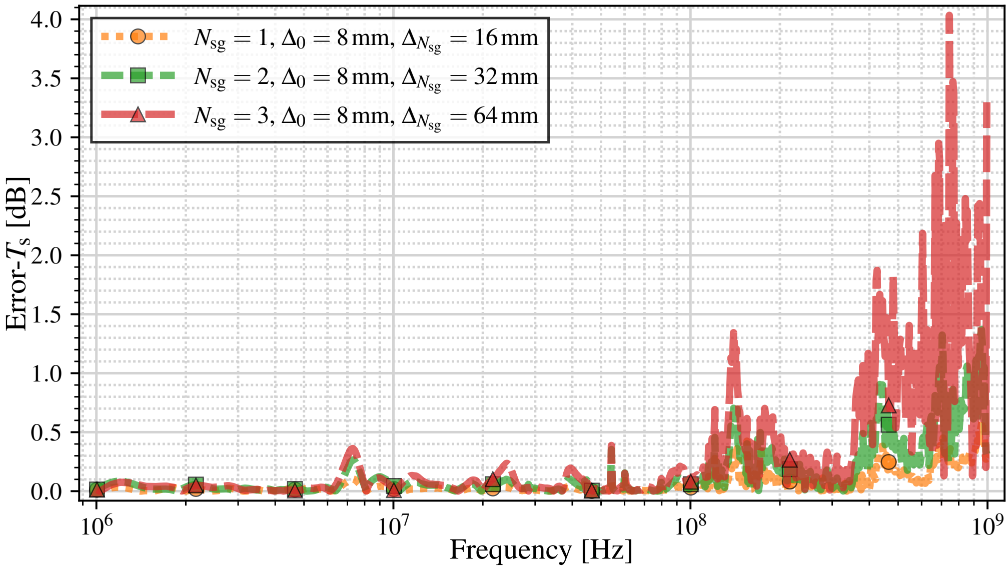

Figure 4.50: T_s for a common finest-level grid of $\Delta_0 = 8\,\text{mm}$.

At medium frequencies ($10 - 100\,\text{MHz}$), the first fundamental modes of the aircraft can propagate, and therefore the first and strongest resonances are observed. By looking at fig. 4.47, we may see that the frequencies associated with these resonances do not show any appreciable discrepancy. On the other hand, a slight displacement in the first resonance ($f \simeq 30\,\text{MHz}$) can be observed in fig. 4.51. Particularly, this discrepancy is observed for the execution with $\Delta_0 = 32$, which is a significantly large mesh length compared to the radius of the cable, which is 30 mm. Even with a conformal mesh, this results in a worse capture of the geometry, therefore creating this kind of discrepancy. Figs. 4.49 and 4.52 are not affected by the mentioned resonances because the surface current density is measured in the outer region. By looking at fig. 4.50, we may observe that the error is less than 0.5 dB in this range.

At high frequencies ($> 100\,\text{MHz}$), the incident waves can fully penetrate inside the aircraft. The behavior at these frequencies is comparable to a reverberant chamber, where a large amount of constructive and destructive interferences are created. At this point, the currents induced inside the aircraft become very sensitive to slight changes in frequency, thus showing many resonances, as can be seen in figs. 4.47 and 4.51. The behavior of the surface current (figs. 4.47 and 4.51), on the other hand, is not affected by this behavior as it is located on the outside of the aircraft. Here, we may observe that the surface current density decreases as we increase the frequency as the probe is located in the wing shadow.

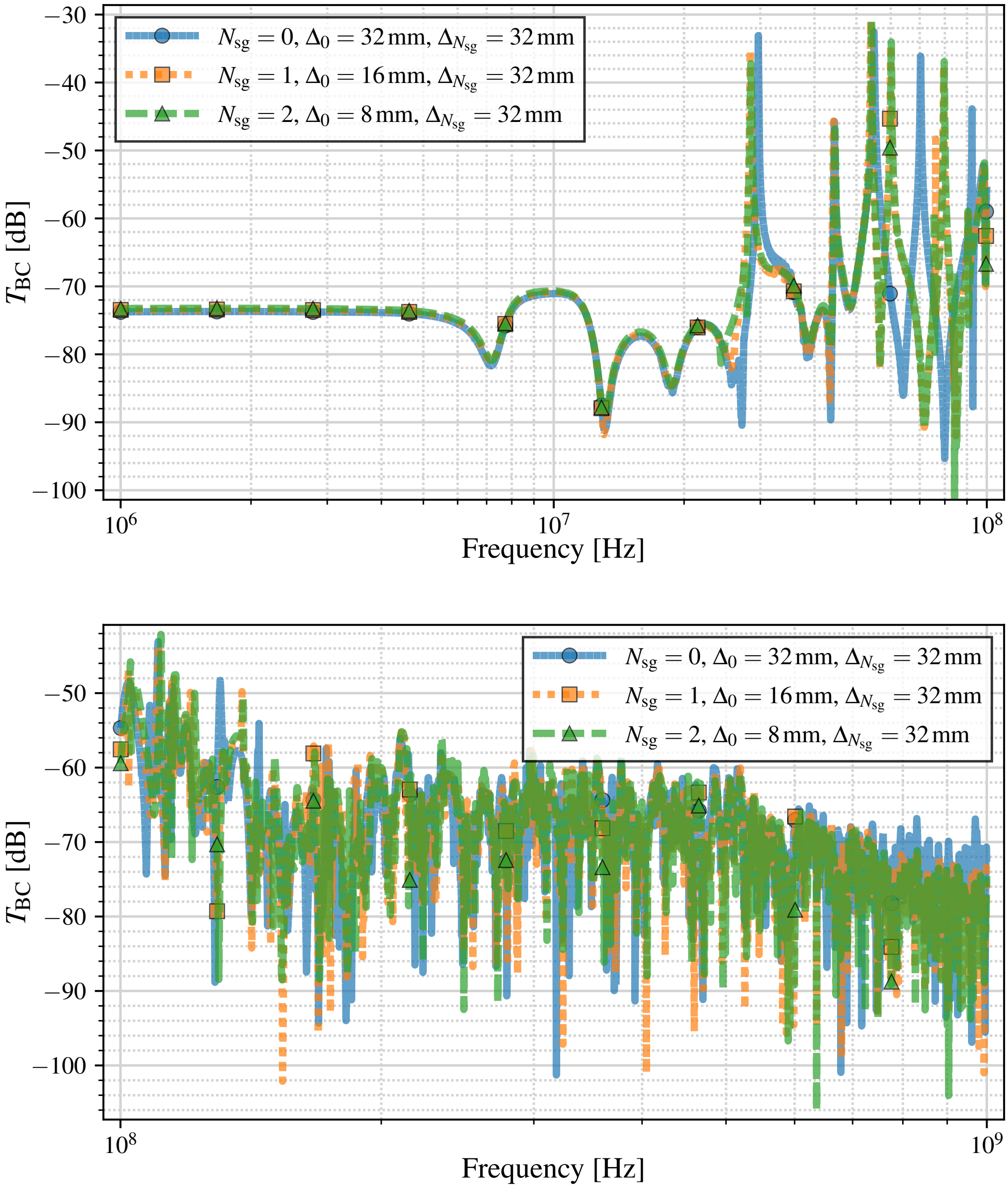

Figure 4.51: Error of T_{BC} for a common coarsest-level grid of $\Delta_{N_{sg}} = 8\,mm$. Plots are split into low and medium frequencies (upper), and high frequencies (lower).

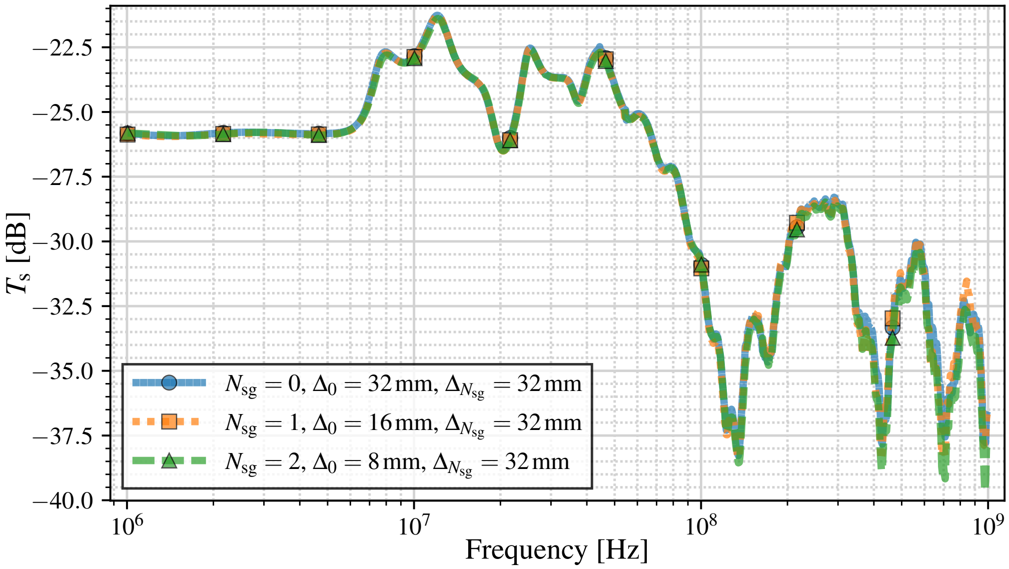

Figure 4.52: Error of T_s for a common coarsest-level grid of $\Delta_{N_{sg}} = 8$ mm. Plots are split into low and medium frequencies (upper), and high frequencies (lower).

By looking at table 4.18, we observe that the amount of cells is reduced from an 84% for $N_{sg} = 1$ to a 97% with $N_{sg} = 3$. Similarly, the CPU time required is reduced from a 86% with $N_{sg} = 1$ to a 93% with $N_{sg} = 2$. Interestingly, the value of T_{CPU} is barely reduced from $N_{sg} = 2$ to $N_{sg} = 3$, thus making it not as efficient in terms of CPU time, but almost twice as efficient in memory usage. By looking at table 4.19, we may see that N_{cells} and T_{CPU} increase, respectively, a 112% and a 384%, to get from $\Delta_0 = 32$mm to $\Delta_0 = 8$ mm. Nonetheless, this has allowed us to refine the EV55 model without requiring a full fine-only grid, thus increasing the accuracy without sacrificing the performance as much.

CHAPTER 5

Conclusions and future work

Three main pillars should be considered in the study of any subgridding algorithm in FDTD: the stability and the presence of late-time instabilities; the accuracy and spurious reflections; and the efficiency improvement provided by the method. Throughout this work, all three of them have been successfully addressed in various ways. Furthermore, the OI-SG algorithm has been implemented for any arbitrary simulation case and its capabilities have been successfully demonstrated in a variety of scenarios. The method has been proven to be stable and the trade-off between accuracy and efficiency has been shown to behave significantly better than the standard FDTD. Additionally, an optional LTS scheme has been designed and proven to be suitable and efficient for usage in real simulations. Although this approach requires the usage of non-centered finite differences in time derivative approximations, its stability and accuracy have been demonstrated.

The stability of the OI-SG method has been analyzed by three different approaches. Late-time instabilities have been reported in the LTS variant. Nonetheless, the required conditions to ensure late-time stability have been successfully determined in both time-stepping variants: $\delta_\mathrm{r} = 0.21$ and CFLN $= 0.93$ for executions with GTS, and $\delta_\mathrm{r} = 1/3$ and CFLN $= 0.66$ for executions with LTS. Furthermore, the analytical expression has been found to provide an accurate description of the stability conditions for simulations with LTS, thus allowing us to determine the most critical discrete field components regarding instability.

The numerical dispersion, convergence, and numerical reflections have been properly studied and identified. The complex frequency has been shown to converge with order $O(h)$ to the analytical solution. Similarly, numerical reflections have been found to converge with $O(h)$ in 3D, in contrast to $O(h^2)$ in the 1D subgridding.

Based on the conclusions derived from the stability analysis, a zero-cost LECT-based methodology has been successfully developed to increase the CFLN value to 0.8 and 0.9. The proposed configurations have resulted in a net decrease of the CPU time of 16% for LECT-1 and 26% for LECT-2, as has been also checked in some of the performance tables in chapter 4.

In executions with a common coarsest-level grid, the accuracy has been found to increase when with N_{sg}, and therefore with higher refinement of the finest level. The error analyses provided in chapter 4 have shown that this tendency is maintained in most cases, although for larger values $N_{\mathrm{sg}} \in \{3,4\}$ the error may slightly increase due to higher amount of subgridding boundaries. However, in general, we may conclude the accuracy is mostly related to the capability of the coarsest level to resolve the frequency of interest (larger values of $\mathrm{PPW}_{\mathrm{coarse}}$) and the capability of the finest level to resolve the geometry of the problem. The optimum value of $\mathrm{PPW}_{\mathrm{coarse}}$ mostly depends on the specific case of study, LO ones being the most critical ones.

The buffer spacing has been shown to appreciably decrease the numerical error. Particularly, in the FSS case, we conclude that an optimum buffer spacing could be defined by $d_{\mathrm{buf}} \simeq 0.12\lambda$, where λ is the wavelength corresponding to the frequency of interest. However, applying buffer spacing necessarily decreases the efficiency since it requires more finest-level cells. In a case such as the FSS, the impact is not particularly significant, but in other cases such as the NASA almond from section 4.4.2 we have reported a net in CPU time when applying buffer spacing.

LTS is naturally more efficient than GTS as levels with $N_{\mathrm{sg}} > 0$ require less time steps. Both schemes show similar accuracies, although GTS can behave appreciably better for greater values of N_{sg} according to the results presented in chapter 4. On the other hand, LTS provides slightly lower distortions in conditions of normal incidence and reflection, as shown in section 3.5.3 and section 4.3.1. This can be explained by the impact on the dispersion relations in fine and coarse regions due to the mismatch in the values of CFLN.

An efficient parallelization scheme has been developed in section 3.7 and tested in the simulation cases presented in chapter 4. According to the performance tables that share the same finest-level grid, the efficiency, in terms of both CPET and T_{CPU}, is always increased when introducing subgridding. This gain is particularly significant in simulations that contain larger chunks of empty space, such as in the FSS case, in which the CPU time reduction is around 81% for $N_{\mathrm{sg}} = 1$ and 97% for $N_{\mathrm{sg}} = 3$. This reduction is lower in cases with less empty space such as the scattering examples, in which the time reduction can vary between 60% and 80% for executions with GTS and 75% to 90% for executions with LTS.

We have also found that the OI-SG may reduce the efficiency in 2D cases such as the AANR metasurface. The LECT configurations provided further increase the efficiency but at the expense of the numerical error. Specifically, LECT-2 has been found to be too aggressive, thus making it unsuitable for most cases.

Another major point in the analysis of subgridding techniques is memory consumption. In this regard, the usage of subgridding always decreases the amount of FDTD cells compared to a fine-only execution. For $N_{\text{sg}} = 3$, the memory usage reduction has been reported to be up to 99.63% without buffer spacing and 97.4% with buffer spacing. In challenging cases such as the NASA almond with $N_{\text{sg}} = 1$ and buffer spacing, although the CPU time gain may be negative, the memory gain is a positive 34%.

The main advantages of the OI-SG method can be summarized as follows:

- It is long-term stable with both time-stepping variants: GTS and LTS.
- It allows the nesting of multiple levels.
- In general terms, it offers a significant net increase in efficiency. When compared to a fine-only execution with standard FDTD, the memory usage is always decreased and the CPU time required is significantly reduced for 3D simulations with large regions of empty space.
- When refining the geometry of the problem coming from a given coarse grid, the accuracy is increased. This effect is further enhanced if we apply buffer spacing.
- The algorithm is fully compatible with other subcell methods such as conformal meshes.

The main weak points found in this work can be summarized as follows:

- The maximum stable value of CFLN with LTS is 0.67, thus reducing the efficiency in terms of CPET and inducing phase dispersions.
- Subgridding boundaries provide spurious reflections that depend on the resolution of the grid.
- The LTS scheme requires non-centered finite differences, and the alternative approach involving time interpolations/extrapolations is unconditionally unstable.
- The implementation necessarily induces latency due to secondary procedures aside from field updates. This reduces the efficiency in terms of CPCT and highly depends on the quality of the implementation.

- The LECT methodology has a severe impact on accuracy, thus making it unsuitable in most contexts.

It is worth noting that most of the aforementioned weak points also apply to other subgridding methods. In general terms, the OI-SG algorithm is a suitable method for multiscale problems in the context of FDTD and entails a promising method for future lines of research, including:

- Finding new applications of the OI-SG method in multiscale problems in FDTD. Furthermore, studying its behavior in combination with other subcell methods.

- Implementing the algorithm for non-cubic and non-uniform grids. Although these grids are not common, providing a subgridding scheme on them could be proven useful in specific contexts.

- Implementing the method for other refinement ratios such as $r = 3$. Odd refinement ratios have been found to provide better numerical properties in some cases in the literature, making this an interesting topic of research.

- Revisiting the instabilities of this method by applying spatial filters that have been proven in the literature to enhance stability. Particularly, this could be an interesting approach for the interpolation/extrapolation LTS scheme.

Glossary

AANR	all-angle negative refraction.
ABCs	absorbing boundary conditions.
ADI-FDTD	alternating-direction-implicit finite-difference time-domain.
AMR	adaptative mesh refinement.
BC	boundary conditions.
BIBO	bounded-input bounded-output.
CEM	computational electromagnetic.
CFL	Courant-Friedrichs-Lewy.
CFLN	Courant-Friedrichs-Lewy number.
CN	Crank Nicolson.
CPCT	cells per CPU time.
CPET	cells per electromagnetic time.
CPML	convolutional perfectly matched layer.
CPU	central processing unit.
DGTD	discontinuous galerkin time domain.
DMMA	dispersive magnetic material approximation.
EM	electromagnetic.
EMC	electromagnetic compatibility.
FBW	fly-by-wire.
FD	frequency-domain.
FDFD	finite-difference frequency-domain.
FDTD	finite-difference time-domain.
FEM	finite element method.
FIT	finite integration technique.
FSS	frequency-selective surface.

FVTD	finite volume time domain.
GO	geometrical optics.
GPU	graphics processing unit.
GTD	geometric theory of diffraction.
GTS	global time-stepping.
HF	high frequency.
HIRF	high intensity radiated field.
HSG	Huygens subgridding.
INTA	Instituto Nacional de Técnica Aeroespacial.
LECT	locally enlarged cell technique.
LIE	lightning indirect effects.
LO	low observability.
LTI	linear time invariant.
LTS	local time-stepping.
MoM	method of moments.
MPI	message passing interface.
NASA	National Aeronautics and Space Administration.
NN-EMP	non-nuclear electromagnetic pulse.
NTFF	near-to-far-field.
OI-SG	orthogonalized integral-based subgridding.
OMP	open multi-processing.
ONERA	Office National d’Études et de Recherches Aérospatiales.
PEC	perfect electric conductor.
PMC	perfect magnetic conductor.
PML	perfectly matched layer.
PO	physical optics.
PTD	physical theory of diffraction.
RCS	radar cross section.
RMS	root mean square.
SBP-SAT	summation-by-parts simultaneous approximation term.

SE	shielding effectiveness.
SF	spatial filtering.
SF-SFDTD	spatially-filtered symplectic finite-difference time-domain.
SG	subgridding.
SGBC	subgridding boundary conditions.
SHSG	switched Huygens subgridding.
TD	time-domain.
TLM	transmission-line-matrix.
UTD	uniform theory of diffraction.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] J. C. Maxwell, "A dynamical theory of the electromagnetic field," *Philosophical Transactions of the Royal Society of London*, vol. 155, pp. 459–513, 1865.

[2] D. Bouche, F. Molinet, and R. Mittra, *Asymptotic methods in electromagnetics*. Springer Science & Business Media, 2012.

[3] J. B. Keller, "Geometrical theory of diffraction," *JOSA*, vol. 52, no. 2, pp. 116–130, 1962.

[4] R. M. Lewis and J. Boersma, "Uniform asymptotic theory of edge diffraction," *Journal of Mathematical physics*, vol. 10, no. 12, pp. 2291–2305, 1969.

[5] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proceedings of the IEEE*, vol. 62, pp. 1448–1461, Nov 1974.

[6] G. Thiele and T. Newhouse, "A hybrid technique for combining moment methods with the geometrical theory of diffraction," *IEEE Transactions on Antennas and Propagation*, vol. 23, pp. 62–69, Jan 1975.

[7] R. F. Harrington, *Field Computation by Moment Methods*. Wiley-IEEE Press, 1993.

[8] D. B. Davidson, *Computational Electromagnetics for RF and Microwave Engineering*. Cambridge University Press, 2 ed., 2010.

[9] A. Orlandi, "Lightning induced transient voltages in presence of complex structures and nonlinear loads," *IEEE Transactions on Electromagnetic Compatibility*, vol. 38, no. 2, pp. 150–155, 1996.

[10] A. Bretones, R. Mittra, and R. Martin, "A hybrid technique combining the method of moments in the time domain and FDTD," *Microwave and Guided Wave Letters, IEEE*, vol. 8, pp. 281–283, Aug 1998.

[11] R. Rumpf, *Electromagnetic and Photonic Simulation for the Beginner: Finite-Difference Frequency-Domain in MATLAB*. Artech, 2022.

[12] V. Shankar, A. H. Mohammadian, and W. F. Hall, "A time-domain, finite-volume treatment for the Maxwell equations," *Electromagnetics*, vol. 10, no. 1-2, pp. 127–145, 1990.

[13] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Publishing Company, Incorporated, 1st ed., 2007.

[14] C. Christopoulos, *The Transmission-Line Modeling (TLM) Method in Electromagnetics*. IEEE Press, 1995.

[15] T. Weiland, "A discretization method for the solution of Maxwell's equations for six-component fields," *Archiv fuer Elektronik und Uebertragungstechnik*, vol. 31, pp. 116–120, March 1977.

[16] M. Clemens and T. Weiland, "Discrete electromagnetism with the finite integration technique," *Journal of Electromagnetic Waves and Applications*, vol. 15, no. 1, pp. 79–80, 2001.

[17] A. Taflove and S. C. Hagness, *Computational Electrodynamics The Finite-Differences Time Domain Method*. Artech House, 2005.

[18] Y. Hao, *FDTD Modeling of Metamaterials: Theory and Applications*. Artech House, 2008.

[19] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302–307, May 1966.

[20] N. V. Kantartzis, K. P. Prokopidis, T. D. Tsiboukis, and E. E. Kriezis, "A nonorthogonally-oriented higher-order fdtd technique for 3-d waveguide and antenna structures on curvilinear grids," in *Antennas, Propagation and EM Theory, 2000. Proceedings. ISAPE 2000. 5th International Symposium on*, pp. 65–68, Aug 2000.

[21] K.-P. Hwang, "Computational efficiency of Fang's fourth-order FDTD schemes," *Electromagnetics*, vol. 23, no. 2, pp. 89–102, 2003.

[22] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM J. Res. Dev.*, vol. 11, pp. 215–234, March 1967.

[23] F. Teixeira, C. Sarris, Y. Zhang, D.-Y. Na, J.-P. Berenger, Y. Su, M. Okoniewski, W. Chew, V. Backman, and J. Simpson, "Finite-difference time-domain methods," *Nature Reviews Methods Primers*, vol. 3, 10 2023.

[24] S. Dey and R. Mittra, "A locally conformal finite-difference time-domain (FDTD) algorithm for modeling three-dimensional perfectly conducting objects," *Microwave and Guided Wave Letters, IEEE*, vol. 7, pp. 273–275, September 1997.

[25] S. Benkler, N. Chavannes, and N. Kuster, “A new 3-D conformal PEC FDTD scheme with user-defined geometric precision and derived stability criterion,” *Antennas and Propagation, IEEE Transactions on*, vol. 54, pp. 1843–1849, June 2006.

[26] M. R. Cabello, L. D. Angulo, J. Alvarez, A. R. Bretones, G. G. Gutierrez, and S. G. Garcia, “A new efficient and stable 3D conformal FDTD,” *IEEE Microwave and Wireless Components Letters*, vol. 26, pp. 553–555, Aug 2016.

[27] I. D. Flintoft, J. F. Dawson, L. Dawson, A. C. Marvin, J. Alvarez, and S. G. Garcia, “A modular test-suite for the validation and verification of electromagnetic solvers in electromagnetic compatibility applications,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, pp. 111–118, Feb. 2017.

[28] M. R. Cabello, L. D. Angulo, J. Alvarez, A. R. Bretones, and S. G. Garcia, “A new conformal fdtd for lossy thin panels,” *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 12, pp. 7433–7439, 2019.

[29] M. R. N. Cabello, A. J. Martín Valverde, B. P. Gallardo, M. Frövel, D. P. Martinez, A. R. Bretones, and S. G. García, “A subcell fdtd scheme implementation for thin slot modeling,” in *2022 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pp. 1–4, 2022.

[30] M. R. Cabello, A. J. Martín Valverde, B. Plaza, M. Frövel, D. Poyatos, A. R. Bretones, A. G. Bravo, and S. G. García, “A subcell finite-difference time-domain implementation for narrow slots on conductive panels,” *Applied Sciences*, vol. 13, no. 15, 2023.

[31] S. Dey and R. Mittra, “A conformal finite-difference time-domain technique for modeling cylindrical dielectric resonators,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, pp. 1737–1739, Sept. 1999.

[32] K.-P. Hwang and A. C. Cangellaris, “Effective permittivities for second-order accurate FDTD equations at dielectric interfaces,” *IEEE Microwave and Wireless Components Letters*, vol. 11, pp. 158–160, Apr. 2001.

[33] R. Holland and L. Simpson, “Finite-difference analysis of EMP coupling to thin struts and wires,” *Electromagnetic Compatibility, IEEE Transactions on*, vol. EMC-23, pp. 88 –97, may 1981.

[34] J. P. Berenger, “A multiwire formalism for the FDTD method,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 42, pp. 257–264, Aug. 2000.

[35] C. Guiffaut, A. Reineix, and B. Pecqueux, “New oblique thin wire formalism in the FDTD method with multiwire junctions,” *Antennas and Propagation, IEEE Transactions on*, vol. 60, pp. 1458–1466, March 2012.

[36] M. Sarto, “A new model for the FDTD analysis of the shielding performances of thin composite structures,” *Electromagnetic Compatibility, IEEE Transactions on*, vol. 41, pp. 298 –306, Nov. 1999.

[37] M. R. Cabello, L. Diaz Angulo, J. Alvarez, I. D. Flintoft, S. Bourke, J. F. Dawson, R. Gómez Martín, and S. G. Garcia, "A hybrid crank–nicolson fdtd subgridding boundary condition for lossy thin-layer modeling," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 5, pp. 1397–1406, 2017.

[38] L. D. Angulo, M. R. Cabello, J. Alvarez, A. R. Bretones, and S. G. García, "From microscopic to macroscopic description of composite thin panels: A road map for their simulation in time domain," *IEEE Transactions on Microwave Theory and Techniques*, 2018.

[39] C. Luo, S. G. Johnson, J. D. Joannopoulos, and J. B. Pendry, "All-angle negative refraction without negative effective index," *Phys. Rev. B*, vol. 65, p. 201104, May 2002.

[40] Z. Yang, Z. Lu, L. Wu, P. Zhang, and M. Zhao, "A novel helical metamaterial absorber: Simulation study with the fdtd method," in *2012 International Workshop on Metamaterials (Meta)*, pp. 1–3, 2012.

[41] A. F. Mota, A. Martins, V. M. Pepino, E. R. Martins, J. Weiner, F. L. Teixeira, and B.-H. V. Borges, "2d semi-analytical model for optimizing the radiation emission of quantum emitters embedded in a bounded nano-patterned hyperbolic metamaterial," in *2018 12th International Congress on Artificial Materials for Novel Wave Phenomena (Metamaterials)*, pp. 128–130, 2018.

[42] J. Alvarez, L. Angulo, M. Bandinelli, H. Bruns, M. Francavilla, S. Garcia, R. Guidi, G. Gutierrez, C. Jones, M. Kunze, J. Martinaud, I. Munteanu, M. Panitz, J. Parmantier, P. Pirinoli, Z. Reznicek, G. Salin, A. Schroder, P. Tobola, and F. Vipiana, "EV55: A numerical workbench to test TD/FD codes in HIRF EMC assessment," in *Proc. EUROEM European Electromagnetic*, (Toulouse (France)), 2012.

[43] L. D. Angulo, S. Greco, M. R. Cabello, S. G. García, and M. S. Sarto, "Fdtd techniques to simulate composite air vehicles for emc," in *Advanced Electromagnetics Symposium, AES*, 2012.

[44] M. R. Cabello, S. Fernández, M. Pous, E. Pascual-Gil, L. D. Angulo, P. López, P. J. Riu, G. G. Gutierrez, D. Mateos, D. Poyatos, M. Fernandez, J. Alvarez, M. F. Pantoja, M. Añón, F. Silva, A. R. Bretones, R. Trallero, L. Nuño, D. Escot, R. G. Martin, and S. G. Garcia, "Siva UAV: A case study for the EMC analysis of composite air vehicles," *IEEE Transactions on Electromagnetic Compatibility*, vol. PP, no. 99, pp. 1103–1113, 2017.

[45] J. Ritter and F. Arndt, "A generalized 3d subgrid technique for the finite-difference time domain method," in *Microwave Symposium Digest, 1997., IEEE MTT-S International*, vol. 3, pp. 1563–1566 vol.3, June 1997.

[46] A. M. Valverde, M. R. Cabello, C. C. Sánchez, A. R. Bretones, and S. G. Garcia, "On the effect of grid orthogonalization in stability and accuracy of an fdtd subgridding method," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 11, pp. 10769–10776, 2022.

[47] A. J. M. Valverde, R. C. N. Miguel, A. R. Bretones, and S. G. García, "An analysis of the stability of a general-purpose 3d subgridding method," in *2022 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pp. 1–3, 2022.

[48] A. J. M. Valverde, N. Miguel Ruiz-Cabello, A. R. Bretones, A. G. Bravo, and S. G. García, "Analysis and improvement of the stability of a 3d fdtd subgridding method by applying a lect-based technique," *IEEE Transactions on Antennas and Propagation*, pp. 1–1, 2023.

[49] J. A. Roden and S. D. Gedney, "Convolution PML (CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media," *Microwave and Optical Technology Letters*, vol. 27, pp. 334–339, December 2000.

[50] P. D. Lax and R. D. Richtmyer, "Survey of the stability of linear finite difference equations," *Communications on pure and applied mathematics*, vol. 9, no. 2, pp. 267–293, 1956.

[51] Q. He, H. Gan, and D. Jiao, "Explicit time-domain finite-element method stabilized for an arbitrarily large time step," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 11, pp. 5240–5250, 2012.

[52] M. Gaffar and D. Jiao, "An explicit and unconditionally stable fdtd method for electromagnetic analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2538–2550, 2014.

[53] M. Gaffar and D. Jiao, "Alternative method for making explicit fdtd unconditionally stable," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 12, pp. 4215–4224, 2015.

[54] M. Zhou, Z. David Chen, W. Fan, X. Bo, and W. Wei, "A subgridding scheme with the unconditionally stable explicit fdtd method," in *2016 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, pp. 1–3, 2016.

[55] X.-K. Wei, N. Diamanti, X. Zhang, A. P. Annan, and C. D. Sarris, "Spatially-filtered fdtd subgridding for ground penetrating radar numerical modeling," in *2017 9th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, pp. 1–4, 2017.

[56] M. Gkatzianas, C. Balanis, and R. Diaz, "The gilbert-holland fdtd thin slot model revisited: An alternative expression for the in-cell capacitance," *IEEE Microwave and Wireless Components Letters*, vol. 14, pp. 219–221, May 2004. Funding Information: Manuscript received August 1, 2003; revised January 16, 2004. This work was supported in part by U.S. Army Contract DAAH10-01-C-0016, Ft. Eustis, VA and the AHE Program. The review of this letter was arranged by Associate Editor A. Weisshaar. The authors are with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-7206 USA. Digital Object Identifier 10.1109/LMWC.2004.827843.

[57] S. Benkler, *Robust Conformal Subcell Modeling for Electromagnetic Simulations in Time Domain*. PhD thesis, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 2006.

[58] M. D. R.-C. Núñez, *Subcell FDTD techniques for electromagnetic compatibility assessment in aeronautics*. PhD thesis, University of Granada, Granada, Spain, 2017.

[59] I. Kim and W. Hoefer, "A local mesh refinement algorithm for the time domain-finite difference method using maxwell's curl equations," *IEEE Transactions on Microwave Theory and Techniques*, vol. 38, no. 6, pp. 812–815, 1990.

[60] S. Wang, F. L. Teixeira, R. Lee, and J.-F. Lee, "Optimization of subgridding schemes for FDTD," *IEEE Microwave and Wireless Components Letters*, vol. 12, pp. 223–225, June 2002.

[61] V. Douvalis, Y. Hao, and C. Parini, "Reduction of late time instabilities of the finite difference time domain method in curvilinear coordinates," in *The Fourth International Conference on Computation in Electromagnetics, 2002. CEM 2002 (Ref. No. 2002/063)*, pp. 2 pp.–, 2002.

[62] Z. Meglicki, S. K. Gray, and B. Norris, "Multigrid fdtd with chombo," *Computer Physics Communications*, vol. 176, no. 2, pp. 109–120, 2007.

[63] Y. Zhang, A. A. Kishk, A. B. Yakovlev, and A. W. Glisson, "Analysis of wideband dielectric resonator antenna arrays for waveguide-based spatial power combining," *IEEE Transactions on Microwave Theory and Techniques*, vol. 55, no. 6, pp. 1332–1340, 2007.

[64] S. Yang, Y. Yu, Z. Chen, and W. Y. Yin, "A subgridding scheme using hybrid one-step leapfrog adi-fdtd and fdtd methods," in *2012 IEEE/MTT-S International Microwave Symposium Digest*, pp. 1–3, 2012.

[65] C. J. Railton, "A tilted subgrid for two dimensional fdtd," in *2015 9th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, 2015.

[66] L. Zhou, F. Yang, R. Long, F. Yan, W. Han, and M. Gao, "A hybrid method of higher-order fdtd and subgridding technique," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1261–1264, 2016.

[67] M. Le, M. Hadi, and A. Elsherbeni, "Quantifying subgridding errors when modeling multiscale structures with fdtd," in *2019 International Applied Computational Electromagnetics Society Symposium (ACES)*, pp. 1–2, 2019.

[68] L. Deng, Y. Wang, X. Wang, X. Zhang, and S. Yang, "A stable hybrid implicit-explicit fdtd subgridding method for te analysis," in *2022 International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–2, 2022.

[69] Y. Cheng, Y. Wang, H. Liu, L. Li, X.-H. Wang, X. Zhang, Z. Chen, and S. Yang, "A stable fdtd subgridding scheme with sbp-sat for transient tm analysis," *Journal of Computational Physics*, vol. 494, p. 112510, 2023.

[70] Y. Wang, Y. Cheng, X.-H. Wang, S. Yang, and Z. Chen, “An sbp-sat fdtd subgridding method using staggered yee’s grids without modifying field components for tm analysis,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 71, no. 2, pp. 579–592, 2023.

[71] Y. Wang, L. Deng, H. Liu, Z. Chen, and S. Yang, “Toward the 2-d stable fdtd subgridding method with sbp-sat and arbitrary grid ratio,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 72, no. 3, pp. 1591–1605, 2024.

[72] F. Bekmambetova, X. Zhang, and P. Triverio, “A dissipative systems theory for fdtd with application to stability analysis and subgridding,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 2, pp. 751–762, 2017.

[73] X. He, B. Wei, and K. Fan, “An explicit newmark-fdtd algorithm with subgridding technique,” *IEEE Antennas and Wireless Propagation Letters*, vol. 19, no. 12, pp. 2383–2387, 2020.

[74] Z. Song, G. Xie, Z. Huang, G. Hou, and P. Pan, “The application of the adi-fdtd method in the 3d hybrid subgrid technique,” in *2022 International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–3, 2022.

[75] G. Xie, N. Feng, M. Fang, M. S. Tong, W. E. I. Sha, and Z. Huang, “A stability extensible symplectic fdtd(4,4) method and its application in subgrid technique for arbitrary grid ratios,” *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 11, pp. 9124–9129, 2023.

[76] M. Chi, X. Ma, and L. Ma, “A switched-huygens-subgridding-based combined fdtd–pitd method for fine structures,” *IEEE Microwave and Wireless Technology Letters*, vol. 33, no. 7, pp. 947–950, 2023.

[77] J. Feng, M. Fang, G. Xie, K. Song, W. Chen, Z. Huang, and X. Wu, “An efficient fdtd method based on subgridding technique and one-step leapfrog adi-fdtd,” *IEEE Microwave and Wireless Technology Letters*, vol. 33, no. 4, pp. 375–378, 2023.

[78] F. Kaburcuk and A. Z. Elsherbeni, “Sub-gridding fdtd algorithm for 3d numerical analysis of em scattering and radiation problems,” *Electromagnetic Science*, vol. 1, no. 4, pp. 1–8, 2023.

[79] Y. Wang, L. Deng, and S. Yang, “Towards the stable fdtd subgridding method with sbp-sat technique and arbitrary grid ratio,” in *2023 International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–3, 2023.

[80] F. Bekmambetova, X. Zhang, and P. Triverio, “Acceleration of shielding effectiveness analysis using stable fdtd subgridding,” in *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, pp. 1–3, 2017.

[81] F. Bekmambetova, X. Zhang, and P. Triverio, “A dissipation theory for three-dimensional fdtd with application to stability analysis and subgridding,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7156–7170, 2018.

[82] P. Monk, “Sub-gridding fdtd schemes,” *Applied Computational Electromagnetics Society*, vol. 11, pp. 37–46, 1996.

[83] M. Okoniewski, M. Mrozowski, and M. Stuchly, “Simple treatment of multi-term dispersion in fdtd,” *IEEE Microwave and Guided Wave Letters*, vol. 7, no. 5, pp. 121–123, 1997.

[84] P. Thoma and T. Weiland, “Numerical stability of finite difference time domain methods,” *IEEE Transactions on Magnetics*, vol. 34, no. 5, pp. 2740–2743, 1998.

[85] O. Podebrad, M. Clemens, and T. Weiland, “New flexible subgridding scheme for the finite integration technique,” *IEEE Transactions on Magnetics*, vol. 39, no. 3, pp. 1662–1665, 2003.

[86] M. Celuch-Marcysiak and J. Rudnicki, “On the effect of total reflection on subgridded fdtd meshes,” in *IEEE Antennas and Propagation Society Symposium, 2004.*, vol. 1, pp. 65–68 Vol.1, 2004.

[87] H. Ding and Q.-X. Chu, “An improved subgridding method with the second-order accurate fdtd technique at the dielectric interface,” in *2008 Asia-Pacific Symposium on Electromagnetic Compatibility and 19th International Zurich Symposium on Electromagnetic Compatibility*, pp. 52–55, 2008.

[88] W. Tierens, “Explicit and provably stable spatiotemporal fdtd refinement,” *Journal of Computational Physics*, vol. 375, pp. 901–917, 2018.

[89] D. M. Pederson and L. L. Raja, “A stable finite-difference time-domain scheme for local time-stepping on an adaptive mesh,” *Journal of Computational Physics*, vol. 394, pp. 456–476, 2019.

[90] S. S. Zivanovic, K. S. Yee, and K. K. Mei, “A subgridding method for the time-domain finite-difference method to solve maxwell’s equations,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 39, pp. 471–479, March 1991.

[91] M. White and M. Iskander, “A new 3d fdtd multi-grid technique with dielectric-traverse capabilities,” in *IEEE Antennas and Propagation Society International Symposium 1997. Digest*, vol. 4, pp. 2160–2163 vol.4, 1997.

[92] K. Krishnaiah and C. Railton, “A stable subgridding algorithm and its application to eigenvalue problems,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 5, pp. 620–628, 1999.

[93] X. Xiaoli and W. Wenbing, “Modeling of interstitial applicators used for microwave hyperthermia of bone cancer based on subgridding fdtd method,” in *6th International SYmposium on Antennas, Propagation and EM Theory, 2003. Proceedings. 2003*, pp. 645–648, 2003.

[94] K. Xiao, D. Pommerenke, and J. Drewniak, “A three-dimensional fdtd subgridding algorithm based on interpolation of current density,” in *2004 International*

Symposium on Electromagnetic Compatibility (IEEE Cat. No.04CH37559), vol. 1, pp. 118–123 vol.1, 2004.

[95] K. Xiao, D. J. Pommerenke, and J. L. Drewniak, "A three-dimensional fdtd subgridding method with separate spatial and temporal subgridding interfaces," in *Electromagnetic Compatibility, 2005. EMC 2005. 2005 International Symposium on*, vol. 2, pp. 578–583, IEEE, 2005.

[96] K. Xiao, D. J. Pommerenke, and J. L. Drewniak, "A three-dimensional fdtd subgridding algorithm with separated temporal and spatial interfaces and related stability analysis," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 7, pp. 1981–1990, 2007.

[97] Y. Zhang, A. Kishk, A. Yakovlev, and A. Glisson, "Dra analysis using simple and flexible fdtd subgridding scheme," in *2006 IEEE Antennas and Propagation Society International Symposium*, pp. 2767–2770, 2006.

[98] L. Kulas and M. Mrozowski, "A technique of stabilizing fdtd-subgridding algorithm using reciprocity principle," in *MIKON 2008 - 17th International Conference on Microwaves, Radar and Wireless Communications*, pp. 1–2, 2008.

[99] K. Ramli, R. Abd-Alhameed, C. See, D. Zhou, E. Elkhazmi, and P. Excell, "Precise computation of the induced fields inside biological cell tissue using floquet boundary and subgridding quasi-static fdtd method," in *2009 Asia Pacific Microwave Conference*, pp. 1533–1536, 2009.

[100] Y. Wang, S. Langdon, and C. Penney, "Analysis of accuracy and stability of fdtd subgridding schemes," in *The 40th European Microwave Conference*, pp. 1297–1300, 2010.

[101] T. Ohtani, Y. Kanai, and N. V. Kantartzis, "A 3-d interlayer-based fdtd/ns-fdtd connection technique combined with a stable subgrid model for low-cost simulations," *IEEE Transactions on Magnetics*, vol. 50, no. 2, pp. 153–156, 2014.

[102] T. Ohtani, Y. Kanai, and N. V. Kantartzis, "A 4-d subgrid scheme for the ns-fdtd technique using the cns-fdtd algorithm with the shepard method and a gaussian smoothing filter," *IEEE Transactions on Magnetics*, vol. 51, no. 3, pp. 1–4, 2015.

[103] A. Zadehgol, "Deterministic reduced-order macromodels for computing the broadband radiation-field pattern of antenna arrays in fdtd," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 6, pp. 2418–2430, 2016.

[104] A. Zadehgol, "An impedance transfer function formulation for reduced-order macromodels of subgridded regions in fdtd," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 1, pp. 401–404, 2017.

[105] Z. Ye, C. Liao, X. Xiong, and M. Zhang, "A novel fdtd subgridding method with improved separated temporal and spatial subgridding interfaces," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 1011–1015, 2017.

[106] J. Xu and G. Xie, “A novel hybrid method of spatially filtered fdtd and subgridding technique,” *IEEE Access*, vol. 7, pp. 85622–85626, 2019.

[107] A. Zakharian, M. Brio, C. Dineen, and J. Moloney, “Second-order accurate fdtd space and time grid refinement method in three space dimensions,” *IEEE Photonics Technology Letters*, vol. 18, no. 11, pp. 1237–1239, 2006.

[108] N. V. Venkatarayalu, R. Lee, Y.-B. Gan, and L.-W. Li, “A stable fdtd subgridding method based on finite element formulation with hanging variables,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 907–915, 2007.

[109] A. Monorchio and R. Mittra, “A novel subgridding scheme based on a combination of the finite-element and finite-difference time-domain methods,” *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 9, pp. 1391–1393, 1998.

[110] D. Prescott and N. Shuley, “A method for incorporating different sized cells into the finite-difference time-domain analysis technique,” *IEEE Microwave and Guided Wave Letters*, vol. 2, no. 11, pp. 434–436, 1992.

[111] W. Yu and R. Mittra, “A new subgridding method for the finite-difference time-domain (fdtd) algorithm,” *Microwave and Optical Technology Letters*, vol. 21, no. 5, pp. 330–333, 1999.

[112] W. Yu and R. Mittra, “A new higher-order subgridding method for finite difference time domain (fdtd) algorithm,” in *IEEE Antennas and Propagation Society International Symposium. 1998 Digest. Antennas: Gateways to the Global Network. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No.98CH36)*, vol. 1, pp. 608–611 vol.1, 1998.

[113] B.-Z. Wang, Y. Wang, W. Yu, and R. Mittra, “A hybrid 2-d adi-fdtd subgridding scheme for modeling on-chip interconnects,” *IEEE Transactions on Advanced Packaging*, vol. 24, no. 4, pp. 528–533, 2001.

[114] A. Vaccari, R. Pontalti, C. Malacarne, and L. Cristoforetti, “A robust and efficient subgridding algorithm for finite-difference time-domain simulations of maxwell’s equations,” *Journal of Computational Physics*, vol. 194, no. 1, pp. 117–139, 2004.

[115] H. AL-Tameemi, J.-P. Bérenger, and F. Costen, “Singularity problem with the one-sheet huygens subgridding method,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, no. 3, pp. 992–995, 2017.

[116] R. Kopecky and M. Persson, “Subgridding method for FDTD modeling in the inner ear,” in *Microwave and Optical Technology 2003* (J. Pistora, K. Postava, M. Hrabovsky, and B. S. Rawat, eds.), vol. 5445, pp. 398 – 401, International Society for Optics and Photonics, SPIE, 2004.

[117] L. Kulas and M. Mrozowski, “Low-reflection subgridding,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 5, pp. 1587–1592, 2005.

[118] M. White, M. Iskander, and Z. Huang, “Development of a multigrid fdtd code for three-dimensional applications,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 10, pp. 1512–1517, 1997.

[119] L. Kulas and M. Mrozowski, “A fast high-resolution 3-d finite-difference time-domain scheme with macromodels,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 9, pp. 2330–2335, 2004.

[120] L. Kulas and M. Mrozowski, “A simple high-accuracy subgridding scheme,” in *33rd European Microwave Conference Proceedings (IEEE Cat. No.03EX723C)*, vol. 1, pp. 347–350 Vol.1, 2003.

[121] L. Kulas, P. Sypek, J. Podwalski, and M. Mrozowski, “Model order reduction for subgriddding in fdtd scheme,” in *2006 17th International Zurich Symposium on Electromagnetic Compatibility*, pp. 18–21, 2006.

[122] L. Deng, Y. Wang, C. Tian, H. Liu, X. Wang, G. Chen, X.-H. Wang, X. Zhang, Z. Chen, and S. Yang, “A symmetric fdtd subgridding method with guaranteed stability and arbitrary grid ratio,” *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 12, pp. 9207–9221, 2023.

[123] L. Deng, N. Gong, G. Diao, and S. Yang, “A stable and symmetric fdtd subgridding method with arbitrary grid ratio,” in *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, pp. 1435–1436, 2023.

[124] L. Deng, Y. Wang, and S. Yang, “A provably stable fdtd subgridding method with arbitrary grid ratio for multisale structures,” in *2023 International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–3, 2023.

[125] C. J. Railton, “A rotated subgrid for 3d fdtd,” in *2015 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, pp. 330–331, 2015.

[126] C. J. Railton, “Rotated subgrids in the fdtd method,” *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 7, pp. 3047–3054, 2016.

[127] C. J. Railton, “Rotated nonuniform subgrids in the fdtd method with application to a hemispherical antenna array,” *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 5, pp. 2460–2466, 2017.

[128] P. Chow, T. Yamagajo, T. Kubota, and T. Namiki, “A novel he-coupling for explicit multigrid-fdtd,” in *IEEE/ACES International Conference on Wireless Communications and Applied Computational Electromagnetics, 2005.*, pp. 269–272, 2005.

[129] P. Park, D. Lee, S. Choi, S. Lee, S. Lim, and J. Choi, “A novel 3d sub-gridding method with odd interpolation ratio and magic location scheme,” in *2007 IEEE Antennas and Propagation Society International Symposium*, pp. 3061–3064, 2007.

[130] K. Zeng and D. Jiao, “Symmetric positive semi-definite fdtd subgridding algorithms in both space and time for accurate analysis of inhomogeneous problems,” *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 4, pp. 3047–3059, 2020.

[131] C.-C. Chang and S.-K. Jeng, “A multilevel subgridding scheme for two-dimensional finite-difference time-domain method,” in *IEEE Antennas and Propagation Society International Symposium. 2001 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No.01CH37229)*, vol. 1, pp. 36–39 vol.1, 2001.

[132] P. Chow, R. Nobes, T. Kubota, and T. Namiki, “Advanced methods for electromagnetic simulation,” *Fujitsu Scientific and Technical Journal*, vol. 43, pp. 524–531, 11 2007.

[133] W. Tierens and D. De Zutter, “Bor-fdtd subgridding based on finite element principles,” *Journal of Computational Physics*, vol. 230, no. 12, pp. 4519–4535, 2011.

[134] Z. Chen and I. Ahmed, “3d hybrid adi-fdtd/fdtd subgridding scheme applied to rf/microwave and optical structures [waveguide/cavity examples],” in *IEEE Antennas and Propagation Society Symposium, 2004.*, vol. 2, pp. 1684–1687 Vol.2, 2004.

[135] W. Pernice, F. Payne, and D. Gallagher, “Simulation of metallic nano-structures by using a hybrid fdtd-adi subgridding method,” in *2007 International Conference on Electromagnetics in Advanced Applications*, pp. 633–636, 2007.

[136] Z. Huang and G. Pan, “A novel subgridding technique for unconditionally stable time domain method,” in *2010 IEEE Antennas and Propagation Society International Symposium*, pp. 1–4, 2010.

[137] M. Chevalier, R. Luebbers, and V. Cable, “Fdtd local grid with material traverse,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 411–421, 1997.

[138] B. Donderici and F. Teixeira, “Domain-overriding and digital filtering for 3-d fdtd subgridded simulations,” *IEEE Microwave and Wireless Components Letters*, vol. 16, no. 1, pp. 10–12, 2006.

[139] P. Chow, T. Kubota, and T. Namiki, “A stable fdtd subgridding method for both spatial and temporal spaces,” in *2008 IEEE Antennas and Propagation Society International Symposium*, pp. 1–4, 2008.

[140] J.-P. Bérenger, “The huygens subgridding for the numerical solution of the maxwell equations,” *Journal of Computational Physics*, vol. 230, no. 14, pp. 5635–5659, 2011.

[141] J. P. Berenger, “A huygens subgridding for the fdtd method,” *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 12, pp. 3797–3804, 2006.

[142] H. El-Raoufk, E. El-Diwani, A.-H. Ammar, and F. El-Hefnawi, “A 3d modified fdtd (2,4) algorithm for improving phase accuracy with low resolution,” in *Proceedings of the Sixteenth National Radio Science Conference. NRSC’99 (IEEE Cat. No.99EX249)*, pp. B4/1–B4/9, 1999.

[143] B. Donderici and F. Teixeira, “Improved fdtd subgridding algorithms via digital filtering and domain overriding,” *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 9, pp. 2938–2951, 2005.

[144] Y. Cheng, L. Li, X.-H. Wang, S. Yang, and Z. Chen, “A provably stable fdtd subgridding technique for transient electromagnetic analysis,” in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pp. 669–670, 2022.

[145] J. Feng, M. Fang, X. Deng, Z. Li, G. Xie, Z. Huang, and X. Wu, “Fdtd modeling of nonlocality in a nanoantenna accelerated by a cpu–gpu heterogeneous architecture and subgridding techniques,” *IEEE Transactions on Antennas and Propagation*, vol. 72, no. 2, pp. 1708–1720, 2024.

[146] Y. Wang, G. Diao, H. Ni, and S. Yang, “A provably stable sbp-sat fdtd subgridding method without modifying yee’s grids,” in *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, pp. 1433–1434, 2023.

[147] H. Li, H. Zhou, X. Bao, and X. Chen, “Multi-scale electromagnetic simulation through parallel adaptive mesh refinement fdtd method with symmetric positive semidefinite subgridding scheme,” in *2023 International Applied Computational Electromagnetics Society Symposium (ACES-China)*, pp. 1–3, 2023.

[148] L. Catarinucci, P. Palazzari, and L. Tarricone, “A parallel variable-mesh fdtd algorithm for the solution of large electromagnetic problems,” in *19th IEEE International Parallel and Distributed Processing Symposium*, p. 8, 2005.

[149] J. Ritter, M. Benjes, M. Murso, D. Wulf, and S. Lange, “A stable subgridding finite difference time domain method on multi-gpu cluster,” in *2015 IEEE International Symposium on Electromagnetic Compatibility (EMC)*, pp. 507–511, 2015.

[150] K. S. Kunz and L. Simpson, “A technique for increasing the resolution of finite-difference solutions of the maxwell equation,” *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-23, no. 4, pp. 419–422, 1981.

[151] R. Pascaud, R. Gillard, R. Loison, J. Wiart, and M. Wong, “Dual-grid finite-difference time-domain scheme for the fast simulation of surrounded antennas,” *IET Microwaves, Antennas & Propagation*, vol. 1, pp. 700–706(6), June 2007.

[152] R. Pascaud, R. Gillard, R. Loison, J. Wiart, and M. F. Wong, “Multiresolution fdtd with switchable grid,” in *2006 First European Conference on Antennas and Propagation*, pp. 1–5, 2006.

[153] B. Donderici and F. Teixeira, “Subgridding with domain overriding for fdtd,” in *IEEE Antennas and Propagation Society Symposium, 2004.*, vol. 1, pp. 61–64 Vol.1, 2004.

[154] J.-P. Berenger, “Three dimensional huygens subgridding for fdtd,” in *2009 IEEE Antennas and Propagation Society International Symposium*, pp. 1–4, 2009.

[155] J.-P. Berenger, “Extension of the fdtd huygens subgridding algorithm to two dimensions,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 12, pp. 3860–3867, 2009.

[156] P. Thoma and T. Weiland, “A subgridding method in combination with the finite integration technique,” in *1995 25th European Microwave Conference*, vol. 2, pp. 770–774, 1995.

[157] S. Wang and F. Teixeira, “Some remarks on the stability of time-domain electromagnetic simulations,” *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 895–898, 2004.

[158] A. Van Londersele, D. De Zutter, and D. Vande Ginste, “An in-depth stability analysis of nonuniform fdtd combined with novel local implicitization techniques,” *Journal of Computational Physics*, vol. 342, pp. 177–193, 2017.

[159] S. Gedney and J. Roden, “Numerical stability of nonorthogonal fdtd methods,” *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 2, pp. 231–239, 2000.

[160] J. Yan and D. Jiao, “A symmetric positive semi-definite fdtd subgridding algorithm for arbitrary grid ratios with uncompromised accuracy,” in *2017 IEEE MTT-S International Microwave Symposium (IMS)*, pp. 687–690, 2017.

[161] J. Yan and D. Jiao, “A general approach to ensuring the stability of unsymmetrical fdtd subgridding schemes,” in *2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, pp. 2403–2404, 2017.

[162] J. Yan and D. Jiao, “Symmetric positive semidefinite fdtd subgridding algorithms for arbitrary grid ratios without compromising accuracy,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 12, pp. 5084–5095, 2017.

[163] J. Yan and D. Jiao, “An unsymmetric fdtd subgridding algorithm with unconditional stability,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 8, pp. 4137–4150, 2018.

[164] A. Van Londersele, D. De Zutter, and D. Vande Ginste, “Provably stable local application of crank-nicolson time integration to the fdtd method with nonuniform gridding and subgridding,” in *2018 International Applied Computational Electromagnetics Society Symposium (ACES)*, pp. 1–2, 2018.

[165] M. Okoniewski, E. Okoniewska, and M. Stuchly, “Three-dimensional subgridding algorithm for fdtd,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 422–429, 1997.

[166] J. Hartley, A. Giannopoulos, and N. Davidson, “Switched huygens subgridding for the fdtd method,” *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 8, pp. 6872–6882, 2022.

[167] Z. Ye, X. Liao, and J. Zhang, "A novel three-dimensional fdtd subgridding method for the coupling analysis of shielded cavity excited by ambient wave," *IEEE Transactions on Electromagnetic Compatibility*, vol. 62, no. 6, pp. 2441–2449, 2020.

[168] W. Kahan, "Pracniques: further remarks on reducing truncation errors," *Commun. ACM*, vol. 8, p. 40, jan 1965.

[169] J. Alvarez, *A Discontinuous Galerkin Finite Element Method for the Time-Domain Solution of Maxwell Equations*. PhD thesis, Universidad de Granada, 2013.

[170] M. Ruiz-Cabello N, M. Abaļenkovs, L. M. Diaz Angulo, C. Cobos Sanchez, F. Moglie, and S. G. Garcia, "Performance of parallel fdtd method for shared- and distributed-memory architectures: Application tobioelectromagnetics," *Plos one*, vol. 15, no. 9, p. e0238115, 2020.

[171] K. Tekbaş and J.-P. Bérenger, "Finite-difference time-domain (fdtd) method with non-homogeneous cells filled with voxels," *Journal of Computational Physics*, vol. 489, p. 112266, 2023.

[172] A. Woo, H. Wang, M. Schuh, and M. Sanders, "Em programmer's notebook-benchmark radar targets for the validation of computational electromagnetics programs," *Antennas and Propagation Magazine, IEEE*, vol. 35, pp. 84–89, Feb 1993.

[173] M. J. S. A. C. Woo, H. T. G. Wang, "Benchmark radar targets for the validation of computational electromagnetics programs," *IEEE Antennas and Propagation Magazine*, vol. 35, pp. 84 – 89, February 1993.

[174] J. Alvarez, J. M. Alonso-Rodriguez, H. Carbajosa-Cobaleda, M. R. Cabello, L. D. Angulo, R. Gomez-Martin, and S. G. Garcia, "DGTD for a class of low-observable targets: A comparison with mom and (2,2) FDTD," *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 241–244, 2014.

[175] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: Measurement and calculation of the bistatic radar cross section of a stealth target," *Radio Science*, vol. 38, no. 3, 2003.

[176] G. Schnattinger, R. A. M. Mauermayer, and T. F. Eibert, "Monostatic radar cross section near-field far-field transformations by multilevel plane-wave decomposition," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 8, pp. 4259–4268, 2014.

[177] F.-M. Tesche, "Electromagnetic field shielding of a spherical shell–revisited," *Forum for Electromagnetic Research Methods and Application Technologies (FERMAT)*, 2008.

[178] P. G. de Francisco, D. Poyatos Martinez, B. P. Gallardo, D. Escot Bocanegra, and S. F. Romero, "Limitations in the measurement of the shielding effectiveness of aeronautical multi-ply cfc laminates," in *2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pp. 662–667, 2019.

[179] “Guide to certification of aircraft in a High-Intensity Radiated Field (HIRF) environment, EUROCAE ED-107A,” 2010.

[180] J. Alvarez, L. Angulo, M. Bandinelli, H.-D. Brüns, M. Francavilla, S. Garcia, R. Guidi, G. Gutierrez, C. Jones, M. Kunze, J.-P. Martinaud, I. Munteanu, M. Panitz, J.-P. Parmantier, P. Pirinoli, Z. Řezníček, G. Salin, A. Schröder, P. Tobola, and F. Vipiana, “Hirf interaction with metallic aircrafts. a comparison between td and fd methods,” in *International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pp. 1–6, 2012.

[181] G. A. Rasek and M. Gabrišák, “Wire bundle currents for high intensity radiated fields (hirf) and indirect effects of lightning (iel) with focus on bulk current injection (bci) test,” in *Radioelektronika (RADIOELEKTRONIKA), 2011 21st International Conference*, pp. 1–10, IEEE, 2011.

[182] A. Schröder, G. A. Rasek, H. D. Brüns, Z. Řezníček, J. Kučera, S. E. Loos, and C. Schuster, “Analysis of high intensity radiated field coupling into aircraft using the method of moments,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, pp. 113–122, Feb 2014.

[183] G. A. Rasek, A. Schröder, P. Tobola, Z. Řezníček, S. E. Loos, T. Tischler, and H.-D. Brüns, “Hirf transfer function observations: Notes on results versus requirements and certification approach,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, no. 2, pp. 195–202, 2015.