KUNGL
TEKNISKA
HÖGSKOLAN

Time-Domain Methods for the Maxwell Equations

Ulf Andersson

Stockholm 2001

Doctoral Dissertation
Royal Institute of Technology
Department of Numerical Analysis and Computer Science

Akademisk avhandling som med tillstånd av Kungl Tekniska Högskolan framläg-ges till offentlig granskning för avläggande av teknisk doktorsexamen fredagen den 9 mars 2001 kl 9.00 i sal D3, Kungl Tekniska Högskolan, Lindstedtsvägen 5, bv, Stockholm.

ISBN 91-7283-043-3

TRITA-NA-0101
ISSN 0348-2952
ISRN KTH/NA/R--01/01--SE

Universitetsservice US AB, Stockholm 2001

Abstract

The most widespread time-domain method for the numerical simulation of the Maxwell equations is the finite-difference time-domain method (FD-TD). It has been widely used for electromagnetic simulation, for instance in radar cross section computations and electromagnetic compatibility investigations. The FD-TD method is second-order accurate and very efficient for simple geometries. A major drawback with the FD-TD method is its inability to accurately handle curved boundaries. Such boundaries are approximated with so-called staircasing to fit into the Cartesian FD-TD grid. Staircasing introduces errors that destroy the second-order accuracy of the FD-TD method.

We present three different methodologies to tackle the errors caused by staircasing. They are parallelization, hybridization with unstructured grids, and regularization of material interfaces.

By using parallel computers it is possible to lower the staircasing errors by using a grid with many cells. We examine the scale-up and speed-up properties of the FD-TD method and demonstrate that it can be used to solve gigantic problems. This is shown by a one-billion-cell computation of an aircraft.

We also present a new hybridization strategy. We hybridize FD-TD with methods for unstructured tetrahedral grids. On the unstructured grid we use either an explicit finite volume method or an implicit finite element method, depending one the size of the smallest tetrahedron in the unstructured grid. The implicit method is used on grids with tetrahedra that are much smaller than the hexahedra in the FD-TD grid. Otherwise the explicit method is used.

In two dimensions, our hybrid methods are second-order accurate and stable. This is demonstrated by extensive numerical experimentation.

In three dimensions, our hybrid methods have been successfully used on realistic geometries such as a generic aircraft model. The methods show super-linear convergence for a vacuum test case. However, they are not second-order accurate. This is shown to be caused by the interpolation when sending values from the FD-TD grid to the unstructured grid.

Our hybrid methods have been implemented in a code package that is used in an industrial environment.

The hybridization strategy is successful but can be expensive in terms of memory and arithmetic operations needed per cell in the grids. We present a new regularization procedure for material interfaces that restore second-order accuracy without adding any extra memory or arithmetic operations during the timestepping. By replacing the discontinuous material function with a properly chosen continuous function prior to the discretization, we can restore second-order accuracy. This is shown for a circular dielectric cylinder for the TM$_z$ polarization of the Maxwell equations.

ISBN 91-7283-043-3 • TRITA-NA-0101 • ISSN 0348-2952 • ISRN KTH/NA/R--01/01--SE

Contents

Acknowledgments

First I would like to express my gratitude to all the people involved with the GEMS project. Even though the project has delayed the appearance of this thesis, it has immensely increased the quality of the research herein. I am particularly grateful to my colleague Gunnar Ledfelt. I would also like to express my thanks to my other co-authors Erik Abenius, Fredrik Edelvik and Lasse Eriksson. I would like to thank Erik Engquist, Jonas Gustafsson, Lennart Hellström, Lars Lovius, Torleif Martin, Åke Rydell, Bo Strand and Anders Ålund for their contributions to the time domain codes in the GEMS project.

I am very grateful to my advisor Professor Björn Engquist. His vast knowledge and experience have been invaluable. He has also arranged several visits for me to the University of California (UCLA). I deeply appreciate all the stimulating discussions with Professor Jesper Oppelstrup. I want to thank all my colleagues at C2M2 for creating a nice working environment.

I would also like to thank my parents, my brother and my sister. Dad showed me the way by letting me choose my upper secondary school education freely, as long as it was either “Teknisk” or “Natur”.

Many thanks to all my friends who with incresing frequency have manifested their interest in my research by asking: “Will you finish soon?”

A special thanks to all those who took time to proofread parts of my thesis. I am particularly grateful to Faith Short, who has explained many intricate points of the language of Shakespeare.

This work was supported with computing resources by the Swedish Council for Planning and Coordination of Research (FRN) and Parallelldatorcentrum (PDC), Royal Institute of Technology. I am grateful to the wonderful staff at PDC for all their support. In particular I would like to thank Harald Barth, Mike Hammill, Lars Malinowski, Nils Smeds and Per Öster.

Financial support has been provided by NADA, PSCI, KTH (internationaliseringsmedel), Svenska Institutet, and Telefonaktiebolaget L M Ericssons stiftelse för främjandet av elektroteknisk forskning, and is gratefully acknowledged.

Chapter 1

Introduction

1.1 Computational electromagnetics

1.1.1 The Maxwell equations

The Maxwell equations were first formulated by James Clerk Maxwell. They are:

$$\begin{aligned}
\nabla \cdot \boldsymbol{D} &= \rho && \text{(Gauss' law)} \\
\nabla \cdot \boldsymbol{B} &= 0 && \text{(Gauss' law)} \\
\frac{\partial \boldsymbol{B}}{\partial t} &= -\nabla \times \boldsymbol{E} && \text{(Faraday's law)} \\
\frac{\partial \boldsymbol{D}}{\partial t} &= \nabla \times \boldsymbol{H} - \boldsymbol{J}_e && \text{(Ampere's law)}
\end{aligned} \tag{1.1}$$

where $\boldsymbol{E}$ is the electric field, $\boldsymbol{D}$ is the electric flux density, $\boldsymbol{H}$ is the magnetic field, $\boldsymbol{B}$ is the magnetic flux density, $\boldsymbol{J}_e$ is the electric current density and ρ is the charge density [Che89].

The Maxwell equations describe electromagnetic phenomena. This includes micro-, radio and radar waves. The Maxwell equations are discussed in more detail in Chapter 3. Faraday's and Ampère's laws constitute a first-order hyperbolic system of equations. The two Gauss' laws can be derived from Faraday's and Ampère's laws provided that the initial conditions fulfill the Gauss' laws.

These equations are linear and it may hence appear to be rather easy to solve them analytically. However, boundary and interface conditions make the Maxwell equations hard to solve analytically. They can be solved analytically only for a few very simple shapes such as a sphere or an infinite circular cylinder. Hence one has to rely on a mix of experiments and approximative and/or numerical methods. Numerical methods for the Maxwell equations are usually referred to as computational electromagnetics (CEM).

Experiments and CEM complement each other when developing a product. An advantage of numerical methods is that they make it possible to test a large number of different constructions without actually building them. CEM is also useful when experiments are difficult and/or dangerous to perform. Such an example is a lightning strike on an aircraft in flight.

The fast variations in the electromagnetic fields make it a challenge to construct numerical methods for the Maxwell equations.

1.1.2 Applications

There is a wide range of applications for CEM. Some of the more important ones are:

- electromagnetic compatibility (EMC),
- antenna analysis and synthesis,
- radar cross section (RCS) calculations,
- cellular phone–human body interaction,
- microwave ovens,
- target recognition and
- hybrid/monolithic microwave integrated circuits.

Many of these applications are impossible to model in every detail. For instance, the interior of a modern aircraft is filled with numerous wires and other small details that are impossible to resolve in a computation. Thus, when modeling a radar pulse striking an aircraft it is impossible to numerically compute the induced current in every cable. However, it might still be possible to accurately predict the radar cross section of the aircraft. Whether this is possible depends strongly on the "electrical size" of the aircraft. By "electrical size" we mean the relation between some appropriate length scale, for instance the length of the aircraft, and the wavelength of the radar wave.

A major task in industrial CEM is the creation of the computational grids. The objects of a calculation are usually described with CAD models. Quite often, the CAD files are broken (in the sense that the CAD surfaces are not connected together properly), which means that the geometry must be repaired. This thesis will not address this important aspect of industrial CEM: instead, we will assume that the computational grids exist.

We will also be rather brief on the important aspect of postprocessing the computational results. The answer of a computation is seldom a simple "YES" or "NO". In many cases, large efforts need to be spent on analyzing the results. This is often done by visualizing the data. The visualization of 3D electromagnetic fields is a non-trivial business, and is further discussed in Sections 4.8 and 6.6.3.

1.1.3 Numerical methods for the Maxwell equations

The Maxwell equations can be solved either in the time domain or the frequency domain. Furthermore, the numerical method can be applied either on the partial differential equation (PDE) formulation of the Maxwell equations in (1.1) or on a boundary integral formulation. Table 1.1 displays examples of methods with this classification.

	Time Domain	Frequency domain
PDE formulation	FD-TD	FEM
Integral formulation	MOT	MoM

Table 1.1. Classification of numerical methods for the Maxwell equations.

The abbreviations in Table 1.1:

- FD-TD = Finite-Difference Time-Domain
- MOT = Marching-On-in-Time
- FEM = Finite Element Method
- MoM = Method of Moments

Table 1.1 lists only the most commonly used method in each category. There are of course numerous other methods.

Time-domain methods can solve a problem for several frequencies in one single calculation and they can also follow the pulse evolution in time. Because this thesis concerns time-domain methods we will concentrate our discussion on these methods and only briefly comment on frequency-domain methods. The latter methods are of course best suited for applications where only a few frequencies are present.

Frequency-domain methods, integral formulation

Frequency-domain integral-formulation methods, such as MoM [Wan91], reduce the volumetric equations to surface equations and thus reduce the number of spatial dimensions of the problem by one. Another advantage is that after solving a particular problem for one angle of incidence, it is relatively easy to find the response for another angle of incidence. However, it is cumbersome to handle cases with varying material properties.

MoM results in a dense linear system of equations. Solving this system directly with Gaussian elimination has the complexity $O(N^3)$ if the size of the matrix is $N \times N$. Assuming that we keep the number of elements per wavelength constant, N increases proportionally to f^2, where f is the frequency. The work to solve the MoM system directly is therefore $O(f^6)$. One way to diminish this workload is to solve the system with iterative methods. Iterative methods are usually based on matrix-vector multiplication, which has a complexity of $O(N^2)$. The work to solve the MoM system iteratively is $O(f^4)$ if the iterative method converges nicely. An even better complexity can be achieved by Multipole methods. In this case the linear system of equations can be solved with $O(Nlog(N))$ arithmetic operations if a multilevel method is used [CRW93, SC95].

Another way to reduce the complexity of MoM is to use the so-called physical optics (PO) method. Here the unknowns on the surface are computed directly from the incident field. Interaction between different parts of the surface is hence neglected. This is a high frequency approximation, PO and MoM give identical results as the frequency tends to infinity. Using PO we can compute the unknowns on the surface in $O(N)$ arithmetic operations

Frequency-domain methods, PDE formulation

One PDE formulation of the Maxwell equations in the frequency domain is the vector Helmholtz equation. For the electric field it is

$$\nabla \times (\frac{1}{\mu} \nabla \times \boldsymbol{E}) = \omega^2 \epsilon \boldsymbol{E} \,, \tag{1.2}$$

where ω is the angular frequency and ϵ and μ are space-dependent material properties. The most common way to solve this is to use finite elements (FEM) [VCK98] because of their geometric flexibility. However, finite differences are also used [Lar00]. The widespread commercial code HFSS [HFS] uses FEM.

Time-domain methods, integral formulation

Time-domain methods for the integral formulation of the Maxwell equations have not been widely used. However, in the last few years there has been an increase in efforts on this subject. Most methods are so-called marching-on-in-time (MOT) methods. The complexity of original MOT methods is $O(N_t N_s^2)$, where N_t is the number of timesteps and N_s is the number of surface patches. This complexity can be improved by using so-called plane-wave time-domain (PWTD) [ESM98] methods.

PWTD methods have been created by adapting ideas from multipole methods described above. The complexity of the two-level PWTD is $O(N_t N_s^{4/3} log(N_s))$, and the complexity of the multilevel PWTD is $O(N_t N_s log(N_s))$.

Some advantages of integral equation methods as compared to PDE methods in the time domain are:

- They do not suffer from dispersion errors.
- They only discretize a surface.
- No absorbing boundary condition is needed.

A drawback of MOT methods is that they are prone to instability [RS90]. The issue has been studied in detail by Walker's group. They state that MOT schemes for solving magnetic field integral equations "can be stabilized for all practical purposes" using implicit timestepping methods [DWB97].

Time-domain methods, PDE formulation

In the time domain there are several possible techniques for intermediate frequencies, including finite differences (FD-TD) [Taf00], finite volumes (FV-TD) [SHM89], and finite elements (FE-TD) [SF90]. The advantages and disadvantages of FD-TD, FE-TD and FV-TD are thoroughly discussed in this thesis, in particular in Chapter 7. We will describe them briefly here.

In CEM, the acronym FD-TD refers to a finite difference approximation of Faraday's and Ampère's laws using second-order accurate central differences in time and space on a grid that is staggered in space and time. The grid is illustrated in Figure 1.1 by showing one cell of the grid. This method was introduced in 1966 by Yee [Yee66] and was further developed by Taflove in the 1970s. It is the most commonly used time-domain method. It is conceptually easy to grasp and very efficient for homogeneous domains. The major drawback is its inability to handle curved boundaries accurately. The FD-TD method is described in [Taf00]

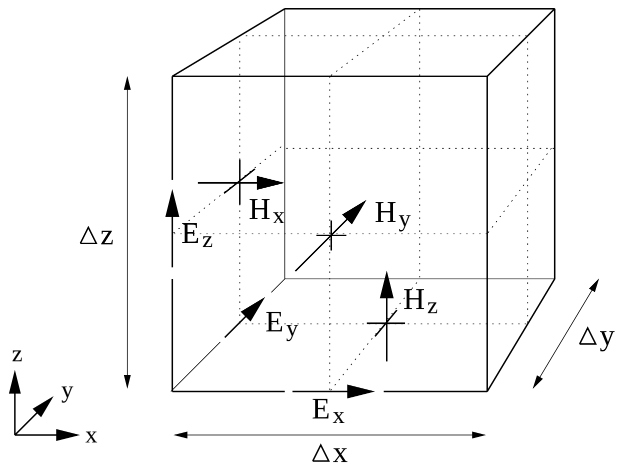


Figure 1.1. Positions of the electric and magnetic field vector components in a unit Yee cell.

and discussed in Chapter 4.

It is possible to construct FD-TD schemes on unstructured grids (see for instance Chapter 4 of [Taf98]). In this case it is very tricky to achieve a stable method. There are two other approaches available on unstructured grids, namely FV-TD and FE-TD.

Finite Volumes were introduced to CEM by Shankar [SHM89] by exporting methods from computational fluid dynamics (CFD). His early work used structured grids, but lately he has turned to unstructured grids. His main reason for doing so is the difficulty of creating a global body-conforming grid for realistic geometries, such as a complete aircraft. This work is also described in Chapter 4 of [Taf98].

Riley introduced another type of Finite Volume method [RT97]. His scheme was based on staggering the electric and magnetics fields. This work has been continued by Edelvik [Ede00], whose work with explicit finite volume solvers is a fundamental part of the time-domain hybrid codes in the GEMS project [GEM] (see Section 1.2 for a description of the GEMS project). The finite volume grid is illustrated in Figure 1.2.

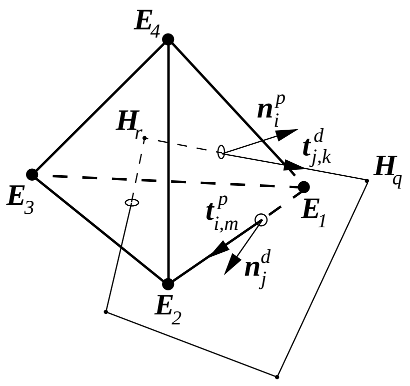


Figure 1.2. A cell in the primary grid and a dual face.

Another method that is well adapted for unstructured grids is the finite element time-domain method (FE-TD) [LLC97]. The finite element method is based on a variational formulation of the PDE in some suitable Hilbert space. Approximations to the solution are then sought in a finite-dimensional subspace.

A common approach for the Maxwell equations is to discretize space with tetrahedra (triangles in 2D) and use so-called "edge elements" [Ned80] as basis functions for the finite-dimensional subspace. The vector basis function for an edge e in 2D is plotted in Figure 1.3. (Even though only one triangle is shown in Figure 1.3, the basis function actually has support on both the triangles that has e as an edge.)

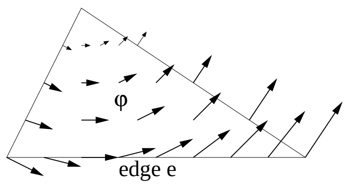


Figure 1.3. The vector basis function φ_e for edge e.

This vector basis function is designed to fit very well with the physics of the Maxwell equations. It enforces tangential continuity but allows normal discontinuity of the fields. Furthermore, it fulfills $\nabla \cdot \varphi_e = 0$, where φ_e is the basis function. This is in agreement with the two Gauss' laws.

A drawback to unstructured grid methods as compared to FD-TD is that they need more memory and floating point operations per unknown. Furthermore, the computer code for unstructured grid methods is usually slower than the code for FD-TD, measured in floating point operations per second. This is due to the indirect addressing needed for unstructured grids.

We think that the best approach is to combine FD-TD with unstructured grid methods into so-called hybrid methods. Unstructured grids are used near curved objects and around small geometrical details, while structured grids are used in the homogeneous parts of the computational domain. This combines the efficiency of structured grids with the geometric flexibility of unstructured grids. Wu and Itoh [WI95] were first to present a combination of the FD-TD method and an implicit FE-TD method. They have been followed by several others [MM98, KLI97, SDPP98, Yeu99, Ryl00, RB00, Ril01]. A combination of an explicit FV-TD solver and FD-TD was proposed by Riley and Turner [RT97] and has been further investigated and improved in [EL00, Ede00]. The hybrid concept is illustrated in Figure 1.4, which displays a hybrid grid for a dielectric circular cylinder. (This figure is a reproduction of Figure 1 in [WI95], and is used with the consent of the authors.)

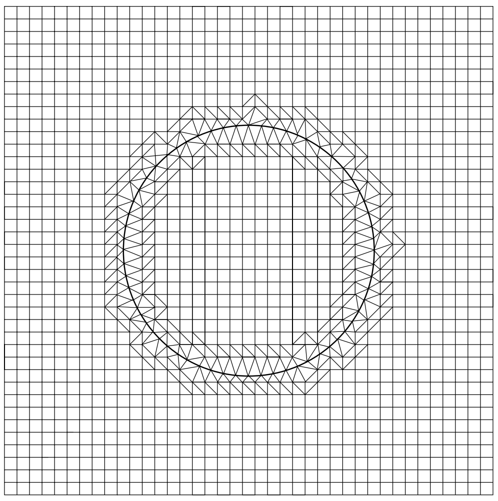


Figure 1.4. A hybrid grid in two dimensons.

One way to decrease the numerical errors is to use higher-order methods. For homogeneous domains this is rather straightforward. A fourth-order accurate FD-

TD method is easily realized. However, staircasing of boundaries and interfaces destroys accuracy. Computing the scattered field from a circular perfectly conducting cylinder would result in less than second-order accuracy, both for a second-order accurate discretization and a fourth-order accurate discretization. This is shown for the second-order accurate discretization in Chapter 8.

To avoid staircasing errors, we could again try to use an unstructured grid close to curved boundaries and interfaces. However, to get a fourth-order accurate scheme, we would need at least a third-order accurate implementation of the boundary condition. This is not easy to achieve. Furthermore, the interface between the structured and unstructured grids must be designed to support fourth-order accuracy and not cause numerical instability. This is also difficult to achieve.

It is our opinion that a fully fourth-order accurate method for industrial applications is not feasible in the near future. However, higher-order discretizations can still be useful. For instance, we could use a fourth-order accurate FD-TD scheme away from the transition region and smoothly revert to the Yee scheme close to the transition region. This would give a second-order scheme, but with smaller error than our present hybrid methods.

High-frequency methods

In both time-domain and frequency-domain methods for the numerical approximation of the Maxwell equations, one needs at least ten mesh points per wavelength for practical engineering accuracy. It follows that for moderately high frequencies a large number of mesh points is required to be able to resolve the problem.

For very high frequencies it becomes impossible to resolve the problem using time-domain methods. Here one has to use high-frequency methods, such as the geometrical theory of diffraction (GTD) [BK94] and uniform theory of diffraction (UTD) [KP74]. High-frequency methods are based on analytical approximations of the Maxwell equations.

1.2 GEMS

The Parallel and Scientific Computing Institute (PSCI) [PSC] is a center of excellence funded by an industrial consortium, the Swedish National Board for Industrial and Technical Development (NUTEK), KTH and Uppsala University. PSCI was created in 1995. One of the programs within PSCI is Computational Electromagnetics (CEM). From 1995 to 1998, the project "Large Scale FD-TD" [Lar] was conducted within the CEM program with the author as project leader. During this project, we developed a 3D FD-TD code, which we called `pscyee`.

In 1998, "Large Scale FD-TD" was succeeded by another PSCI project, the much more extensive General ElectroMagnetic Solvers (GEMS) project [GEM]. This was a Swedish three-year code development project that was supported by an extensive research program. A substantial part of the funding was supplied by the National Aeronautical Research Program (NFFP).

The main objective of the GEMS project was to develop a software suite for solving the Maxwell equations. This software suite aims to be state-of-the-art and to form a platform for future development by Swedish industry and academia. The code will be used in an industrial environment.

The core of the software suite is two hybrid codes, one for the time domain and one for the frequency domain. The time-domain code is a hybrid between FD-TD, explicit FV-TD and implicit FE-TD. The frequency-domain code is a hybrid between MoM, PO and GTD/UTD.

The industrial partners in GEMS are Ericsson Microwave Systems (EMW), Saab Ericsson Space (SES) and Ericsson Saab Avionics (Avionics). Code developers are PSCI, the Swedish Institute of Applied Mathematics (ITM) and the Swedish Defence Research Establishment (FOA). The industrial partners also take part in the code development.

1.3 Outline and main results

Background

The next chapter contains a vernacular description of my research. The two following chapters give background information on the research presented within this thesis. Chapter 3 covers the Maxwell equations and Chapter 4 addresses the Finite-Difference Time-Domain (FD-TD) method [Taf00].

Chapter 5 is a brief description of the GEMS time-domain codes. Chapters 6 through 9 contain the results of my research. The order of these chapters is chronological, though there has been considerable overlap.

Parallelization

Chapter 6 covers parallelization of the leap-frog update in the FD-TD method. Domain decomposition is used to distribute the computations on the nodes of a parallel machine, and communication is performed using the message passing interface (MPI) standard. Having p nodes of a parallel computer, we split the computational domain in p domains of almost equal size. The Cartesian topology facility of MPI is used to distribute these domains on the p nodes.

We show that perfect scale-up can be achieved on a parallel computer with distributed memory. On the other hand, perfect speed-up is usually not possible to obtain. The time to complete a time step on each node is proportional to $n_x n_y n_z$, while the time needed to communicate is proportional to $n_x n_y + n_x n_z + n_y n_z$, where $n_x \times n_y \times n_z$ is the problem size on each node. As this size decreases, the communication time will no longer be negligible compared with the computation time.

The results mentioned in the previous paragraph were achieved on an IBM SP. Similar results can be obtained of for instance a Cray T3E. An exception, "super-linear speed-up", occurs on computers where cache effects are dominant. For example, this happens on a cluster of Dec Alpha computers.

On the parallel shared-memory vector computer Cray J90, we demonstrate that autotasking gives approximately the same performance as the MPI implementation. We also show that on a Fujitsu vector computer, it is possible to achieve more that 50% of the peak performance. The performance on the Fujitsu vector computer is more dependent on the problem size than other computers. Having a large value on the number of cell in the x-direction (N_x) will give the best performance because it leads to long vector lengths.

We show that our parallel implementation can be used for gargantuan computations by performing a one-billion-cell computation on an aircraft. This computation was achieved with 125 nodes with 160 MHz RS/6000 processors of an IBM SP.

Hybrid time-domain methods

Chapters 7 and 8 cover a new technique for hybridization of the finite-difference time-domain (FD-TD) method with methods for unstructured grids. On the unstructured grids, we either use an implicit finite element (FE) method or an explicit finite volume (FV) method. The hybridization is performed by having a transition layer between the structured and unstructured grids, where structured and unstructured cells overlap. In 2D, this region is half a cell thick, and in 3D it is one cell thick. Chapter 7 contains 2D, and Chapter 8 contains 3D.

In Chapter 7 we show that both the FD-FE and FD-FV hybrid for the transverse magnetic Maxwell equations are second-order accurate. This is shown for five different cases: a perfect electric conducting circular cylinder, a perfect magnetic conducting circular cylinder, a dielectric circular cylinder ($\epsilon_r = 4$), a diamagnetic circular cylinder ($\mu_r = 4$) and vacuum. Stability is thoroughly studied by numerical tests. The FD-FV hybrid is stable for all test cases, provided that the stability condition is not violated. The FD-FE hybrid can be unstable when the Crank-Nicholson method is used for timestepping. We have found examples where this happens. In all these cases, stability could be restored by making the timestepping method slightly more implicit or by switching to the two stage backward difference formula (BDF-2) method. We also show that our hybrid methods perform well on a test case with a point source and a perfectly conducting wall with 45 degrees inclination. This case is very similar to one of the test cases in the classical paper by Cangellaris and Wright [CW91], which is the most frequent reference when the problems of staircasing are discussed.

In Chapter 8 we show that our hybrid method in 3D can be used to achieve good results on a generic aircraft model geometry and the NASA almond model problem by computing the radar cross section of these objects. The methods show super-linear convergence for a vacuum test case. However, they are not second-order accurate. This is shown to be caused by the interpolation of diagonal components, which is performed when sending values from FD-TD to the unstructured solvers. This interpolation could also be the cause of instabilities. Hence it might be better to use a pyramidal cell to interface between the tetrahedra in the unstructured grid and the hexahedra in the structured grid.

The GEMS time-domain code is briefly described in Chapter 5. This code package is unique. It is the only code where it is possible to have an unlimited number of unstructured regions and for each region choose whether to use a method with an explicit time integration scheme or a method with an implicit time integration scheme.

The transition layer may intersect with a perfectly conducting object and material interfaces. The code can handle frequency dispersive materials both in FD-TD and in the unstructured regions [ES00].

The flexibility of the GEMS time-domain code is further enhanced by having three different near-to-far-field transformations and several different absorbing boundary conditions (ABC). Among the ABCs, there is a uniaxial perfectly matched layer, which is able to handle frequency-dependent materials that are extended to infinity.

Port excitation and registration is also possible in the GEMS time-domain 3D code. Homogeneous ports in FD-TD and inhomogeneous ports in FE-TD are implemented. A 2D frequency-domain finite-element code is used to compute the mode solution in cases where it is not known analytically.

Modeling of inhomogeneous materials in FD-TD

Chapter 9 addresses the issue of how to model inhomogeneous materials in FD-TD. First we study the case where the material interface coincides with the cells of the FD-TD grid. We show that arithmetic mean values should be used for material coefficients when the field component is tangential to the interfaces, while harmonic mean values should be used for material coefficients when the field component is normal to the interface.

In Chapter 9 we also study how to model material interfaces that do not coincide with the FD-TD cells. A traditional FD-TD staircasing model does not yield second-order accuracy in such cases. We want to do this without changing the width of the updating stencil. To achieve this, we replace the discontinuous function for $\epsilon_r(\bar{x})$ for the transverse magnetic Maxwell equation, with a continuous function $\tilde{\epsilon}_r(\bar{x})$. These two functions only differ in a neighborhood of the material interface. This introduces errors, but is balanced by the fact that the errors from the discretization are smaller for a continuous material function. This procedure allows us to restore the second-order accuracy for a circular cylinder.

1.4 List of papers

This thesis is partly based on material from the following papers:

1. Ulf Andersson. Parallelization of a 3D FD-TD code for the Maxwell equations using MPI. In B. Kågström et al., editors, *Applied Parallel Computing, PARA'98*, Lecture Notes in Computer Science, No. 1541, pages 12–19, June 1998.

2. Ulf Andersson. Parallelization of a 3D FD-TD code for the Maxwell equations using MPI. In G. Kristensson, editor, *EMB 98 – Electromagnetic Computations for analysis and design of complex systems*, pages 94–101. SNRV, November 1998.

3. Ulf Andersson and Gunnar Ledfelt. Large scale FD-TD—A billion cells. In *15th Annual Review of Progress in Applied Computational Electromagnetics*, volume 1, pages 572–577, Monterey, CA, March 1999.

4. E. Abenius, U. Andersson, F. Edelvik, L. Eriksson, and G. Ledfelt. Hybrid time domain solvers for the Maxwell equations in 2D. Technical Report 00:01, PSCI, Parallel and Scientific Computing Institute, KTH, SE-100 44 Stockholm, Sweden, February 2000. Available at http://www.psci.kth.se/Activities/Report

5. Gunnar Ledfelt, Fredrik Edelvik and Ulf Andersson. Hybrid Time Domain Solver for the 3D Maxwell Equations. In U. Zander, editor, *ANTENN 00 – Nordic Antenna Symposium*, pages 57–62, Lund, Sweden, September 2000. FMV, SNRV.

1.5 Division of research

Chapters 3 and 4 were written with Gunnar Ledfelt. They also appear in his Ph.D. thesis [Led01].

The material in Chapters 2, 5, 6 and 9 is completely my own, with the exception of the visualization sections in Chapter 6.

The material presented in Chapters 7 is a collaborative effort. This chapter is based on Paper 4 in the list in Section 1.4. My main responsibility was to design and perform the numerical evaluation. Chapter 7 also appears in Gunnar Ledfelt's Ph.D. thesis [Led01].

Chapter 8 is also based on a collaborative effort. I was responsible for the design of the serial 3D FD-TD code. I also designed and performed the convergence tests and contributed to the other numerical evaluations appearing in Chapter 8.

Chapter 2

Popular Description in Swedish

2.1 Numeriska metoder för Maxwells ekvationer

Denna doktorsavhandling behandlar numeriska metoder för Maxwells ekvationer för de elektromagnetiska fälten. Många fenomen i vår värld kan beskrivas av dessa ekvationer. Exempel är bland annat radiovågor, radarvågor, ljus och mikrovågor. Numeriska metoder innebär att man räknar ut en ungefärlig/approximativ lösning till det givna problemet, vilket i mitt fall alltså beskrivs av Maxwells ekvationer. Eftersom sådana beräkningar innehåller ett stort antal aritmetiska operationer så låter man datorer utföra dem.

För några enstaka enkla fall så vet man den exakta lösningen till Maxwells ekvationer. En våg i vakuum rör sig rakt fram med ljusets hastighet. Om vågen skulle träffa en metallisk sfär så finns det en exakt lösning. Om vågen däremot stöter på ett flygplan blir det omöjligt att räkna ut en exakt lösning på grund av flygplanets komplicerade geometri. I detta fall finns två alternativ. Antingen räknar man ut en approximativ lösning eller så utför man mätningar. Det vanliga är dock att kombinera dessa två sätt.

För att kunna göra mätningar på ett flygplan måste man bygga det. I vissa fall räcker det med en modell av flygplanet. Om man under konstruktionsstadiet vill testa många olika variationer av ett flygplan blir det således ett stort antal modeller som måste byggas. Genom att istället använda numeriska metoder kan man undvika det. Dock krävs det att den numeriska metoden är tillräckligt bra för att man ska kunna använda den för att optimera fram en flygplanskonstruktion.

Tack vare allt snabbare och större datorer samt utveckling av de numeriska metoderna så blir det allt vanligare att använda dem, men för många komplicerade problem existerar det inte tillräckligt bra numeriska metoder.

Min forskning handlar om att utveckla bättre numeriska metoder. Jag arbetar inte direkt med tillämpningar som att konstruera bättre flygplan, mikrovågsugnar eller antenner. Min situation kan liknas vid den person som designar verktygen åt en snickare. Jag bygger inte huset, men jag måste vara insatt i hur det görs annars har jag ingen aning om vilka verktyg som behövs.

En fråga som ofta nämns i pressen och på TV gäller mobiltelefoners eventuella skadlighet. De typer av metoder som jag jobbar med kan användas till att approximativt beräkna hur mycket energi som absorberas i huvudet på en person som talar i mobiltelefon. Denna absorption leder till en temperaturhöjning i hjärnan. Frågan om denna värmeökning eller andra fysikaliska effekter är skadlig eller inte får dock analyseras av forskare med gediget medicinskt kunnande.

En annan tillämpning av numeriska metoder som alla dagligen kommer i kontakt med är väderprognoser. De bygger på en kombination av numeriska metoder och mätningar. Jag nämner denna tillämpning för att alla ska inse att det finns behov av bättre numeriska metoder.

En av de mest använda numeriska metoderna för Maxwells ekvationer går under benämningen FD-TD (Finite-Difference Time-Domain). Den bygger på att man räknar ut approximativa värden till de elektromagnetiska fälten i ett stort antal punkter som ligger jämt utspridda i beräkningsområdet. Om man vill placera till exempel ett flygplan i detta område så måste flygplanet modelleras så att det passar in i den fördelning av punkter som finns. Det leder till att flygplanet ser ut att vara byggt av Lego. Denna justering av flygplanets utseende kan leda till avsevärda felaktigheter i resultaten.

Mina bidrag till att förbättra de numeriska metoderna för Maxwells ekvationer finns beskrivna i kapitel 6 till 9. Kapitel 3 och 4 innehåller huvudsakligen bakgrundsmaterial.

I kapitel 6 visar jag hur man med hjälp av så kallade parallelldatorer kan ha ett extremt stort antal punkter i sina beräkningar. Som mest har jag använt drygt sex miljarder punkter, vilket kräver 22,4 Gbyte minne.

I kapitel 7 och 8 använder jag så kallade hybridmetoder för att undvika Lego problematiken. I närheten av flygplanet placeras punkterna så att de passar bättre ihop med flygplanets verkliga utseende. Att göra på detta sätt kräver mer minne och fler beräkningar för varje punkt. Därför använder jag detta enbart i närheten av flygplanet. Överallt annars använder jag FD-TD.

Om flygplanet är delvis byggt av till exempel glasfiber så innebär det att materialegenskaperna i Maxwells ekvationer är diskontinuerliga. Med diskontinuerlig menas att funktionen som beskriver parametrarna gör ett hopp. I kapitel 9 presenterar jag metoder där man ersätter de diskontinuerliga parametrarna med snarlika fast kontinuerliga funktioner innan man applicerar den numeriska metoden. Metoderna prövas i detta kapitel inte på komplicerade objekt såsom flygplan utan på cirkulära cylindrar i två rumsdimensioner. Att utveckla metoder genom att testa dem på enkla objekt i en eller två rumsdimensioner är en mycket vanlig metodik.

Chapter 3

The Maxwell Equations

3.1 The equations

This thesis deals with numerical approximations of electromagnetic phenomena. These are described by the Maxwell equations, see for instance page 323 in [Che89],

$$\begin{aligned}
\nabla \cdot \boldsymbol{D} &= \rho && \text{(Gauss' law)} \\
\nabla \cdot \boldsymbol{B} &= 0 && \text{(Gauss' law)} \\
\tfrac{\partial \boldsymbol{B}}{\partial t} &= -\nabla \times \boldsymbol{E} && \text{(Faraday's law)} \\
\tfrac{\partial \boldsymbol{D}}{\partial t} &= \nabla \times \boldsymbol{H} - \boldsymbol{J}_e && \text{(Ampere's law)}
\end{aligned} \tag{3.1}$$

where $\boldsymbol{E}(\boldsymbol{x},t)$ is the electric field $[V/m]$, $\boldsymbol{D}(\boldsymbol{x},t)$ is the electric flux density $[C/m^2]$, $\boldsymbol{H}(\boldsymbol{x},t)$ is the magnetic field $[A/m]$, $\boldsymbol{B}(\boldsymbol{x},t)$ is the magnetic flux density $[Wb/m^2]$, $\boldsymbol{J}_e(\boldsymbol{x},t)$ is the electric current density $[A/m^2]$ and $\rho(\boldsymbol{x},t)$ is the charge density $[C/m^3]$. The Maxwell equations are complemented by the equation of continuity,

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \boldsymbol{J}_e \ . \tag{3.2}$$

The two Gauss' laws can be derived from Ampère's law, Faraday's law and the equation of continuity by taking the divergence on Ampère's and Faraday's laws.

For linear, isotropic and non-dispersive materials we have

$$\boldsymbol{B} = \mu \boldsymbol{H} \quad \text{and} \quad \boldsymbol{D} = \epsilon \boldsymbol{E} \ . \tag{3.3}$$

Furthermore we allow for materials with isotropic, non-dispersive electric losses that attenuate $\boldsymbol{E}$ fields via conversion to heat energy. This yields

$$\boldsymbol{J}_e = \sigma \boldsymbol{E} \ . \tag{3.4}$$

Materials for which $\sigma = 0$ are referred to as lossless. Inserting these three relations in (3.1) yields

$$\begin{aligned}
\nabla \cdot (\epsilon \boldsymbol{E}) &= \rho && \text{(Gauss' law)} \\
\nabla \cdot (\mu \boldsymbol{H}) &= 0 && \text{(Gauss' law)} \\
\mu \frac{\partial \boldsymbol{H}}{\partial t} &= -\nabla \times \boldsymbol{E} && \text{(Faraday's law)} \\
\epsilon \frac{\partial \boldsymbol{E}}{\partial t} &= \nabla \times \boldsymbol{H} - \sigma \boldsymbol{E} && \text{(Ampere's law)}
\end{aligned} \tag{3.5}$$

where $\boldsymbol{E} = (E_x, E_y, E_z)$ is the electric field $[V/m]$, $\boldsymbol{H} = (H_x, H_y, H_z)$ is the magnetic field $[A/m]$, $\epsilon(\boldsymbol{x})$ is the electric permittivity $[F/m]$, $\mu(\boldsymbol{x})$ is the magnetic permeability $[H/m]$ and $\sigma(\boldsymbol{x})$ is the electric conductivity $[S/m]$. Writing them component by component, we get

$$\left\{ \begin{aligned}
\epsilon \frac{\partial E_x}{\partial t} &= \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x \ , \\
\epsilon \frac{\partial E_y}{\partial t} &= \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y \ , \\
\epsilon \frac{\partial E_z}{\partial t} &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \ , \\
\\
\mu \frac{\partial H_x}{\partial t} &= \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \sigma^* H_x \ , \\
\mu \frac{\partial H_y}{\partial t} &= \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \sigma^* H_y \ , \\
\mu \frac{\partial H_z}{\partial t} &= \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \sigma^* H_z \ .
\end{aligned} \right. \tag{3.6}$$

We have now introduced the equivalent magnetic loss $\sigma^*(\boldsymbol{x})$ $[\Omega/m]$, see Chapter 3 in Taflove [Taf95]. This increases the symmetry of the Maxwell equations though it is not compatible with Gauss' law for the magnetic flux density. We introduce it because our implementation of FD-TD has the capability to include this term.

Yet another way to write (3.6) is,

$$\boldsymbol{u}_t = A\boldsymbol{u}_x + B\boldsymbol{u}_y + C\boldsymbol{u}_z \ , \tag{3.7}$$

where $\boldsymbol{u} = (E_x \, E_y \, E_z \, H_x \, H_y \, H_z)^T$. All matrices $\xi_1 A + \xi_2 B + \xi_3 C$ for any vector ξ with $\xi_1^2 + \xi_2^2 + \xi_3^2 = 1$ have the same six eigenvalues. They are $-c$, $-c$, 0, 0, c and c where $c = 1/\sqrt{\mu\epsilon}$ is the speed of propagation for the electromagnetic wave. This means that we need exactly two boundary conditions at any given boundary.

The Maxwell equations is a hyperbolic system because all eigenvalues are real. system. See [GKO95] for the definition of hyperbolic.

3.2 Reduction to two dimensions

In two dimensions, (3.6) reduces to two independent set of equations, usually referred to as the transverse magnetic (TM) mode and the transverse electric (TE) mode. If we assume that there are no variations in the z-direction, we get the TM$_Z$ mode

$$\begin{cases} \mu\frac{\partial H_x}{\partial t} & = & -\frac{\partial E_z}{\partial y} - \sigma^* H_x \ , \\ \mu\frac{\partial H_y}{\partial t} & = & \frac{\partial E_z}{\partial x} - \sigma^* H_y \ , \\ \epsilon\frac{\partial E_z}{\partial t} & = & \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \ , \end{cases} \tag{3.8}$$

and the TE$_Z$ mode

$$\begin{cases} \epsilon\frac{\partial E_x}{\partial t} & = & \frac{\partial H_z}{\partial y} - \sigma E_x \ , \\ \epsilon\frac{\partial E_y}{\partial t} & = & -\frac{\partial H_z}{\partial x} - \sigma E_x \ , \\ \mu\frac{\partial H_z}{\partial t} & = & \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} - \sigma^* H_z \ . \end{cases} \tag{3.9}$$

These two modes are decoupled, i.e. they contain no common field component. They are completely independent for isotropic materials, and they can exist simultaneously with no mutual interaction.

3.3 Reduction to one dimension

If we further assume that the magnetic field in (3.8) has no variation in the y-direction, we get

$$\begin{cases} \mu\frac{\partial H_y}{\partial t} & = & \frac{\partial E_z}{\partial x} - \sigma^* H_y \ , \\ \epsilon\frac{\partial E_z}{\partial t} & = & \frac{\partial H_y}{\partial x} - \sigma E_z \ . \end{cases} \tag{3.10}$$

Similar formulas can be derived for other combinations of the fields.

3.4 Integral formulation

The Maxwell equations in (3.5) are given in a partial differential equation (PDE) formulation. It is also possible to cast them in an integral formulation. It can be derived from the PDE formulation: The two Gauss' laws are integrated over an arbitrary fixed control volume after which the divergence theorem is applied to these integrals. Faraday's and Ampère's laws are integrated over a control surface,

S, after which the Stokes theorem is applied to the integrals containing the curl operator. We get

$$\begin{aligned}
\oiint_S \epsilon \boldsymbol{E} \cdot d\hat{S} &= 0 && \text{(Gauss' law)} \\
\oiint_S \mu \boldsymbol{H} \cdot d\hat{S} &= 0 && \text{(Gauss' law)} \\
\frac{\partial}{\partial t} \iint_S \mu \boldsymbol{H} \cdot d\hat{S} &= -\oint_C \boldsymbol{E} \cdot d\hat{l} - \iint_S \sigma^* \boldsymbol{H} \cdot d\hat{S} && \text{(Faraday's law)} \\
\frac{\partial}{\partial t} \iint_S \epsilon \boldsymbol{E} \cdot d\hat{S} &= \oint_C \boldsymbol{H} \cdot d\hat{l} - \iint_S \sigma \boldsymbol{E} \cdot d\hat{S} && \text{(Ampere's law)}
\end{aligned} \tag{3.11}$$

where C is the contour that bounds the surface S. The surface S in the Gauss' laws is not the same as the S in the Faraday's and Ampère's laws. In Gauss' laws, it is the surface of the control volume. This integral formulation of the Maxwell equations is used to construct several of the numerical methods treated in this thesis. It is possible to get other integral formulations of the Ampère's and Faraday's laws, for instance by integrating them over a volume instead of a surface.

Formulas (3.8), (3.9) and (3.10) can also be cast in integral formulations in a similar manner.

3.5 The wave equation

If we take the time derivative of Ampère's law in (3.5) and assume that the material properties are time independent, we obtain

$$\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} = -\nabla \times \frac{1}{\mu} \nabla \times \boldsymbol{E} - \sigma \frac{\partial \boldsymbol{E}}{\partial t}\,. \tag{3.12}$$

For lossless homogeneous materials this reduces to

$$\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = c^2 \Delta \boldsymbol{E}\,, \tag{3.13}$$

where $c = 1/\sqrt{\mu\epsilon}$ is the speed of propagation for the electromagnetic wave. In a similar manner, we may show that

$$\frac{\partial^2 \boldsymbol{H}}{\partial t^2} = c^2 \Delta \boldsymbol{H} \tag{3.14}$$

for lossless homogeneous materials.

3.6 Material properties

In (3.6) we have four parameters. For vacuum they are $\mu \equiv \mu_0 = 4\pi \cdot 10^{-7}\,\mathrm{Vs/Am}$, $\epsilon \equiv \epsilon_0 \approx 10^{-9}/36\pi \approx 8.8541878 \cdot 10^{-12}\,\mathrm{As/Vm}$, $\sigma^* = 0\,\Omega/\mathrm{m}$ and $\sigma = 0\,\mathrm{S/m}$. The speed of light in vacuum is defined by $c_0 = 2.99792458 \cdot 10^8\,\mathrm{m/s} \approx 1/\sqrt{\mu_0\epsilon_0}$. For other materials it is customary to define their permeability and permittivity relative to those of vacuum, i.e. we have $\epsilon = \epsilon_r\epsilon_0$ and $\mu = \mu_r\mu_0$. The relative permittivities for some common materials are listed in Table 3.1. The data have been taken from Table B-3 in [Che89]. For most materials, ϵ_r and μ_r are frequency dependent. Materials for which we assume that ϵ_r and μ_r are independent of frequency are referred to as simple materials. Frequency-dependent materials will be briefly addressed in Chapter 4.12. The values listed in Table 3.1 are average low-frequency values at room temperature. Note that we always have $\epsilon_r \geq 1$. Most materials where $\mu_r \neq 1$ are metals with high conductivity. We treat these materials as perfect electric conductors.

Material	ϵ_r
Teflon	2.1
Rubber	2.3-4.0
Bakelite	5.0
Distilled Water	80

Table 3.1. Relative permittivities for some common materials.

At the interface between two lossless media (we have, see Table 7-3 on page 330 in [Che89])

$$\begin{aligned} \boldsymbol{n} \cdot (\boldsymbol{D}_1 - \boldsymbol{D}_2) &= 0\,, \\ \boldsymbol{n} \times (\boldsymbol{E}_1 - \boldsymbol{E}_2) &= 0\,, \\ \boldsymbol{n} \cdot (\boldsymbol{B}_1 - \boldsymbol{B}_2) &= 0\,, \\ \boldsymbol{n} \times (\boldsymbol{H}_1 - \boldsymbol{H}_2) &= 0\,, \end{aligned} \tag{3.15}$$

where the subscripts indicate which region the field belongs to, and n is the interface normal. Using the relations in (3.3), we get

$$\begin{aligned} \boldsymbol{n} \cdot (\epsilon_1\boldsymbol{E}_1 - \epsilon_2\boldsymbol{E}_2) &= 0\,, \\ \boldsymbol{n} \times (\boldsymbol{E}_1 - \boldsymbol{E}_2) &= 0\,, \\ \boldsymbol{n} \cdot (\mu_1\boldsymbol{H}_1 - \mu_2\boldsymbol{H}_2) &= 0\,, \\ \boldsymbol{n} \times (\boldsymbol{H}_1 - \boldsymbol{H}_2) &= 0\,. \end{aligned} \tag{3.16}$$

For perfect electric conductors (PEC) we have (compare with Table 7-4 on page 331 in [Che89])

$$\begin{aligned} \boldsymbol{n} \cdot \epsilon \boldsymbol{E} &= \rho_s \, , \\ \boldsymbol{n} \times \boldsymbol{E} &= 0 \, , \\ \boldsymbol{n} \cdot \boldsymbol{H} &= 0 \, , \\ \boldsymbol{n} \times \boldsymbol{H} &= \boldsymbol{J}_s \, , \end{aligned} \tag{3.17}$$

where ρ_s is the surface charge density $[C/m^2]$ and $\boldsymbol{J}_s$ is the surface current density $[A/m]$. Note that the normal $\boldsymbol{n}$ is pointing out from the PEC region. It may seem odd that we have six boundary conditions when we should only have two. However the first and fourth conditions are not true boundary conditions, because ρ_s and $\boldsymbol{J}_s$ are unknown, and the third condition can easily be shown to be a consequence of the second condition.

PECs are characterized by having no tangential electric field at the surface. This is a consequence of the term perfect conductor. If there were a tangential electric field it would drive an infinite surface current which is clearly unphysical. However, this does not imply that the surface current must be zero. In fact, if there is an external field there will always be surface currents since the magnetic field does only have tangential components at the PEC surface and the surface current is related to the tangential magnetic field through the fourth condition in (3.17).

Chapter 4

FD-TD

This Chapter was written prior to the publication of the second edition of Taflove's book on FD-TD [Taf00]. Hence all references are to the first edition [Taf95].

4.1 Introduction to FD-TD

The most commonly used time-domain method for solving the Maxwell equations is the Finite-Difference Time-Domain (FD-TD) method. It was introduced by Yee in 1966 [Yee66] and is sometimes referred to as the Yee scheme. The method was further developed and promoted by Taflove in the 1970s, and he also coined the acronym FD-TD.

Several books have been published dealing with the FD-TD scheme [KL93, Taf95, Taf98, IH98, Sul00]. The survey paper by Shlager and Schneider that appeared in [Taf98] illustrates the rapid growth in the use of FD-TD.

The FD-TD method has been attractive for industrial users since the early 1980s because the basic method is relatively simple to program and because the geometry handling is fairly straightforward. The method can also be efficiently implemented on vector computers which made it feasible to solve complex problems on the early supercomputers. As an example, in 1987 SAAB performed lightning analysis on the Swedish fighter aircraft Gripen on a grid with approximately $60 \times 30 \times 30$ cells.

4.2 Discretization used in FD-TD

The FD-TD scheme is an explicit finite difference scheme using central differences on a staggered Cartesian grid (both space and time), i.e. it is a leap-frog scheme. It is second-order accurate in both time and space. "Staggered" here indicates that the different electromagnetic components are not located at the same place (see Figures 4.1 and 4.2). Furthermore, the fields are not represented on the same time levels (see Figure 7.1).

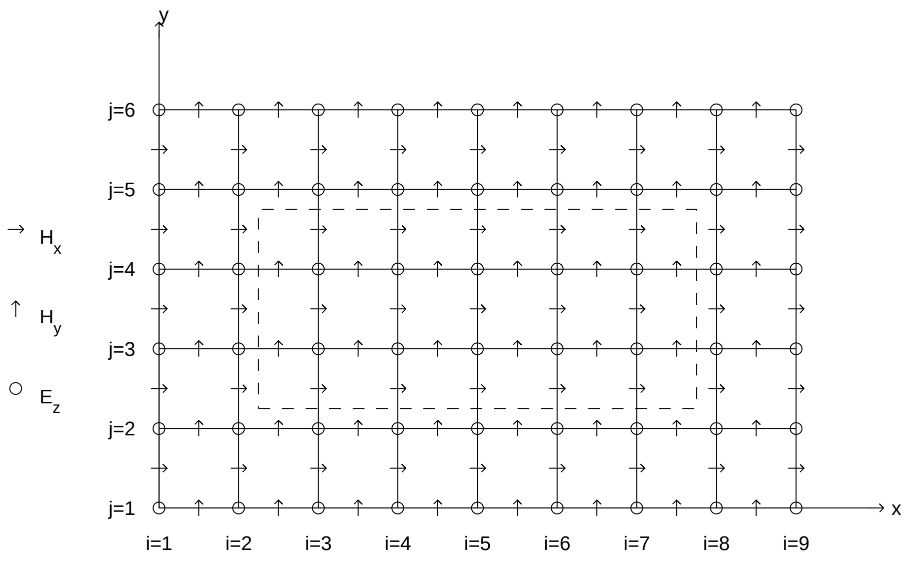


Figure 4.1. The FD-TD TM grid, with problem size $N_x = 8$ and $N_y = 5$. The Huygens' surfaces (dashed line) are placed in the second cell from the outer boundary.

Staggering the variables in the computational grid is a straightforward consequence of the nature of the Maxwell equations and the central finite differences. If the leap-frog scheme is applied on a grid that is not staggered, it would result in $2 \cdot 8$ uncoupled discrete equations. Time and memory are saved by only solving one of these. (memory savings: 8, time step savings: 2 per unknown => 16)

The main drawback of the FD-TD scheme is the inability to represent curved boundaries and small geometrical details. Curved objects must be modeled by staircasing, i.e. they must fit into the Cartesian grid and hence look like they were made of Lego blocks. As will be demonstrated in Chapter 7, staircasing destroys the second-order accuracy.

Figure 4.1 shows a grid for the 2D TM equations. The dashed line indicates the limit between the total field region and the scattered field region. Further explanation can be found in Section 4.7.

One cell of a 3D FD-TD grid is given in Figure 4.2. The magnetic field components are defined on the cell's faces, and the electric field components are defined on the cell's edges. This choice is arbitrary and does not affect the behavior of the scheme. For a computational domain with the number of cells (N_x, N_y, N_z), we

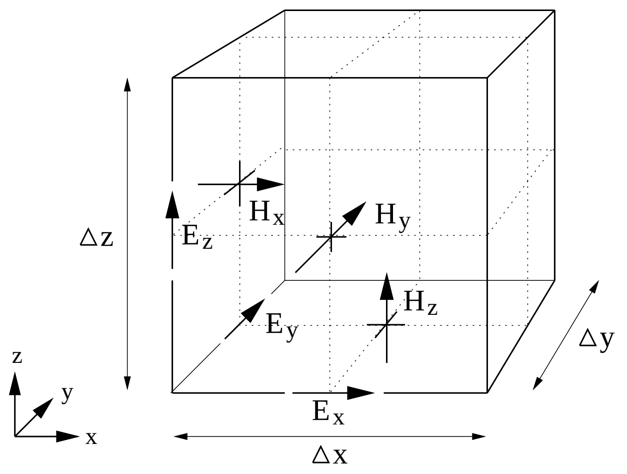


Figure 4.2. Positions of the electric and magnetic field vector components in a unit Yee cell.

define

$$
\begin{array}{llll}
E_x|^n_{i+\frac{1}{2},j,k}\ , & i=1,\ldots,N_x\ , & j=1,\ldots,N_y+1\ , & k=1,\ldots,N_z+1\ ,\\
E_y|^n_{i,j+\frac{1}{2},k}\ , & i=1,\ldots,N_x+1\ , & j=1,\ldots,N_y\ , & k=1,\ldots,N_z+1\ ,\\
E_z|^n_{i,j,k+\frac{1}{2}}\ , & i=1,\ldots,N_x+1\ , & j=1,\ldots,N_y+1\ , & k=1,\ldots,N_z\ ,\\
H_x|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}\ , & i=1,\ldots,N_x+1\ , & j=1,\ldots,N_y\ , & k=1,\ldots,N_z\ ,\\
H_y|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}\ , & i=1,\ldots,N_x\ , & j=1,\ldots,N_y+1\ , & k=1,\ldots,N_z\ ,\\
H_z|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}\ , & i=1,\ldots,N_x\ , & j=1,\ldots,N_y\ , & k=1,\ldots,N_z+1\ ,
\end{array}
\tag{4.1}
$$

where $n = 0,\ldots,N_t$ for all six components and N_t is the number of time steps taken. Initial values are needed for $\boldsymbol{H}^{-\frac{1}{2}}$ and $\boldsymbol{E}^0$. Note that our notation is slightly different from that used in [Taf95]. In our notation there is always a direct correspondence between the indexes and the physical location of a field component. For example,

$$H_x|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} \text{ is located at } ((i-1)\Delta x, (j-1/2)\Delta y, (k-1/2)\Delta z)\,, \tag{4.2}$$

at $t = (n-1/2)\Delta t$ where Δx, Δy and Δz are the spatial cell sizes and Δt is the time increment. The total spatial problem size is $N = N_xN_yN_z$. The storage space needed for this is approximately $24N$ byte for 32-bit precision and $48N$ byte for 64-bit precision.

4.3 The leap-frog scheme

In homogeneous materials with $\sigma = \sigma^* = 0$, the following formulas comprise the FD-TD updating stencils for the electromagnetic field components.

$$\begin{aligned} H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = H_x\big|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} &+ \frac{\Delta t}{\mu \Delta z}\left[E_y\big|^{n}_{i,j+\frac{1}{2},k+1} - E_y\big|^{n}_{i,j+\frac{1}{2},k}\right] \\ &- \frac{\Delta t}{\mu \Delta y}\left[E_z\big|^{n}_{i,j+1,k+\frac{1}{2}} - E_z\big|^{n}_{i,j,k+\frac{1}{2}}\right] \end{aligned} \tag{4.3}$$

$$\begin{aligned} H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} = H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} &+ \frac{\Delta t}{\mu \Delta x}\left[E_z\big|^{n}_{i+1,j,k+\frac{1}{2}} - E_z\big|^{n}_{i,j,k+\frac{1}{2}}\right] \\ &- \frac{\Delta t}{\mu \Delta z}\left[E_x\big|^{n}_{i+\frac{1}{2},j,k+1} - E_x\big|^{n}_{i+\frac{1}{2},j,k}\right] \end{aligned} \tag{4.4}$$

$$\begin{aligned} H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} = H_z\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} &+ \frac{\Delta t}{\mu \Delta y}\left[E_x\big|^{n}_{i+\frac{1}{2},j+1,k} - E_x\big|^{n}_{i+\frac{1}{2},j,k}\right] \\ &- \frac{\Delta t}{\mu \Delta x}\left[E_y\big|^{n}_{i+1,j+\frac{1}{2},k} - E_y\big|^{n}_{i,j+\frac{1}{2},k}\right] \end{aligned} \tag{4.5}$$

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},j,k} = E_x\big|^{n}_{i+\frac{1}{2},j,k} &- \frac{\Delta t}{\epsilon \Delta z}\left[H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right] \\ &+ \frac{\Delta t}{\epsilon \Delta y}\left[H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k}\right] \end{aligned} \tag{4.6}$$

$$\begin{aligned} E_y\big|^{n+1}_{i,j+\frac{1}{2},k} = E_y\big|^{n}_{i,j+\frac{1}{2},k} &- \frac{\Delta t}{\epsilon \Delta x}\left[H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H_z\big|^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2},k}\right] \\ &+ \frac{\Delta t}{\epsilon \Delta z}\left[H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} - H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k-\frac{1}{2}}\right] \end{aligned} \tag{4.7}$$

$$\begin{aligned} E_z\big|^{n+1}_{i,j,k+\frac{1}{2}} = E_z\big|^{n}_{i,j,k+\frac{1}{2}} &- \frac{\Delta t}{\epsilon \Delta y}\left[H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} - H_x\big|^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k+\frac{1}{2}}\right] \\ &+ \frac{\Delta t}{\epsilon \Delta x}\left[H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H_y\big|^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}}\right] \end{aligned} \tag{4.8}$$

For lossy materials the electric field update equations are modified, for example (4.6), is replaced by

$$\begin{aligned} E_x\big|^{n+1}_{i+\frac{1}{2},j,k} = \frac{1 - \frac{\sigma \Delta t}{2\epsilon}}{1 + \frac{\sigma \Delta t}{2\epsilon}} E_x\big|^{n}_{i+\frac{1}{2},j,k} &- \frac{\Delta t/\epsilon \Delta z}{1 + \frac{\sigma \Delta t}{2\epsilon}}\left[H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right] \\ &+ \frac{\Delta t/\epsilon \Delta y}{1 + \frac{\sigma \Delta t}{2\epsilon}}\left[H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H_z\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k}\right], \end{aligned} \tag{4.9}$$

where the conductivity term σE_x is discretized using the average of the electric field at time levels n and $n + 1$. Using only the value from time level n gives an unstable scheme, and using only the value from time level $n+1$ gives a noncentered scheme. For highly lossy media one could take into account the rapid exponential decrease of field strengths and introduce a scaling of the variables; this would yield the so-called exponential timestepping scheme. However, this scheme does not give any significant improvements [Pet97] and is thus not further discussed.

Comparing (4.6) and (4.9) we see that one extra arithmetic operation is needed for lossy material. For inhomogeneous materials, we replace ϵ in (4.9) with $\epsilon_{i+\frac{1}{2},j,k}$ and similarly for σ. Note that we need an ϵ-value for each of the three electric field component updates in a cell. These three ϵ-values will differ in the vicinity of a material interface. Chapter 9 gives a detailed analysis of how to calculate these discrete values.

4.4 Stability conditions

Because FD-TD is an explicit scheme, there is a limit on the time step Δt to ensure stability. It is given by:

$$\Delta t < \frac{1}{c\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}} \, , \tag{4.10}$$

where c is the wave propagation speed. We define the CFL number as

$$CFL = c\Delta t\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}} \, , \tag{4.11}$$

and may thus write the stability condition as

$$CFL < 1 \, . \tag{4.12}$$

CFL stands for *Courant-Friedrichs-Lewy* (see page 54 in [GKO95]).

4.5 Performance of the leap-frog update

The leap-frog update is the core of an FD-TD solver, and therefore it must be implemented as efficiently as possible. There are two major obstacles in getting an efficient implementation. The updating stencils of (4.3)–(4.8) consist of two multiplications and four additions/subtractions each. Most computers today are constructed to perform the same number of multiplications and additions in every clock cycle. In our case, this means that we can achieve at most 75% of the peak performance. The other main obstacle is the need for memory bandwidth. For instance, to compute (4.3) we need to fetch five field values from memory and store one field value to memory. Most computers cannot do this as quickly as they perform the calculations. The constant coefficients will reside in registers and need not be fetched from memory for every update.

It is possible to reduce the number of multiplications in (4.3) by one by scaling the fields with the cell size. Let $\tilde{E}_z = \Delta z E_z$, etc. We get

$$\tilde{H}_x|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = \tilde{H}_x|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta t}{\mu}\Big[\tilde{E}_y|^{n}_{i,j+\frac{1}{2},k+1} - \tilde{E}_y|^{n}_{i,j+\frac{1}{2},k} - \tilde{E}_z|^{n}_{i,j+1,k+\frac{1}{2}} + \tilde{E}_z|^{n}_{i,j,k+\frac{1}{2}} \Big]. \tag{4.13}$$

However, this reduction will lead to little or no gain in execution time on most modern computers because the number of additions still dominates. This is illustrated in Table 4.1.

We will now present some illustrating performance results for the leap-frog update. All tests are performed on the same “standard” problem. We set $N_x = N_y = N_z = 100$ and $N_t = 100$. We use a point source for excitation and the Mur first-order ABC [Mur81] as grid terminator. Timing is performed over the timestepping loop, i.e. we omit initialization and post processing. We also omit the first time step from the timing (timing is performed over 99 iterations), because it might contain initialization overhead. All calculations are performed in 64-bit precision.

Table 4.1 shows the effect on execution time of a reduction in the number of multiplications per component update by one. On the IBM processors, there are no gain in execution time. On the Sun there is some gain, but only about 4%. This should be compared to the 17% (compare (4.3) and (4.13)) decrease in the number of floating point operations.

Computer	Reduced code	Original code
IBM pwr3, 200 MHz	23.13 sec.	23.07 sec.
IBM pwr2, 160 MHz	22.21 sec.	22.24 sec.
Sun Ultra 1, 167 MHz	93.28 sec.	97.00 sec.

Table 4.1. Execution times for the leap-frog update for lossless homogeneous material. These test were performed in February 2000.

For lossless materials we have 36 arithmetic operations per cell. For lossy materials we have 42 arithmetic operations per cell if both σ and σ^* are nonzero. Obviously, lossy inhomogeneous materials increase the execution time. This effect is illustrated in Table 4.2. The number of floating point operations per iteration (Flop/iteration) includes the operations performed by the first-order Mur ABC, see Section 4.6.

Material	Lossless homogeneous	Lossy inhomogeneous
Flop/cell	36	42
Flop/iteration	36 000 900	41 941 200
Performance (Mflop/s)	120.01	93.31
Percentage of peak perf.	13.6	10.6
Time (s)	29.70	44.50

Table 4.2. Performance for homogeneous and inhomogeneous materials on an IBM pwr3 222 MHz processor. These test were performed in June 2000 and cannot be compared to the results in Table 4.1 because different processors and compiler versions were used.

4.6 Boundary conditions

Perfect electric conductors (PEC) are characterized by the absence of tangential electric field at the surface, as discussed in Section 3.6. A PEC must be described using a staircase approximation to fit into the FD-TD scheme. This staircase procedure is a major cause of errors in FD-TD calculations.

It is possible to model PECs by changing the coefficients in (4.3)–(4.5), but a more efficient implementation for homogeneous materials is to first update all the electric field components using (4.6)–(4.8) and then set all $\boldsymbol{E}$ fields on the surface of the object to zero.

Many applications involve geometries with unlimited surroundings. These situations are called open problems. In these cases it is necessary to limit the computational domain by introducing an artificial outer boundary. At this boundary we need to apply a boundary condition, and this condition should be designed to absorb outgoing waves. Hence we refer to it as an absorbing boundary condition (ABC). One could also think of this boundary as having the property of not reflecting any outgoing waves back into the computational domain and hence name it a nonreflecting boundary condition (NRBC).

The history of absorbing boundary conditions for the FD-TD scheme is carefully covered in Chapter 7 of [Taf95]. This chapter concludes with a description of the perfectly matched layer (PML) introduced by Berenger [Ber94] in 1994, which was a tremendous breakthrough in ABC methodology. The basic idea is to surround the computational domain with an absorbing layer. This concept had been tried before, but there were problems with reflections in the interface between the computational domain and the absorbing layer [HW83]. The key to the success of PML is that there are no reflections at this interface, at least not for the continuous problem. This is true for all frequencies and all angles of incidence.

One of the chapters in [Taf98] covers the further development of PML. The original formulation of PML is a weakly hyperbolic system [AG97], which might cause stability problems. This formulation is based on splitting the six field components into two parts each. Later formulations instead introduced a lossy anisotropic absorbing material [Ged96]. They are referred to as unsplit PML (U-PML).

The U-PML formulation relies on the fact that it is possible to derive matching conditions for lossy anisotropic (uniaxial) media, so that incident plane waves are purely transmitted. The reflectionless conditions for the permittivity and permeability in the uniaxial media are found to be:

$$\overline{\overline{\epsilon}} = \overline{\overline{\mu}} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a^{-1} \end{bmatrix} . \tag{4.14}$$

The parameter a is then chosen to be lossy. A natural choice in the frequency domain is

$$a = 1 + \frac{\sigma}{j\omega\epsilon_0} , \tag{4.15}$$

where ω is the frequency.

The attenuation of the PML depends on the size of the lossy parameter (σ), the depth of the absorbing layer and the angle of incidence of the wave. The value of σ should be as large as possible to improve absorption. However, this would result in a step discontinuity in σ in the transition between the interior region and the absorbing layer. In the discrete space, this leads to large reflections of the fields. The parameter σ is therefore chosen as a smoothly increasing function, starting from zero.

A systematic way to evaluate the performance of different ABCs was given in [MBTK88]. Their test consisted of using a point source in 2D and comparing the results with numerical results from a larger domain. We have used this test case on a number of different ABCs. The result is presented in Figure 4.3. All the different ABCs are described in Chapter 7 of [Taf95], except the U-PML scheme which is described in [Ged96]. The Mei Fang result presented in Figure 4.3 has been obtained by applying the Mei Fang procedure to the second-order Mur scheme. The notation U-PML X refers to a U-PML with a layer of X cells. The U-PML results are equivalent to what we would have obtained using PML. It is evident from the results in Figure 4.3 that it is possible to achieve much better absorption with U-PML/PML than with previously developed ABCs. Figure 4.3 is the same type of graph as Figure 5b in [MBTK88] and Figure 7.8 in [Taf95]. Note that formula (7.46) in [Taf95] contains a misprint. The factor should be 1/320 not 1/32 (see [MBTK88]).

The U-PML can be extended to treat frequency dispersive materials, as shown in Section 5.9 in [Taf98]. With U-PML it is possible to construct an arbitrarily good ABC by increasing the number of cells in the U-PML layer. However, there is an increase in cost when increasing the thickness of the layer.

A perfectly matched layer must be terminated at its outer boundary, and one possibility would be to use a classical nonreflecting boundary condition to terminate the U-PML layer. But this is seldom done because the extra cost of implementing and performing this is higher than simply terminating the outer boundary with a PEC condition. This is not as bad as it first might appear. A wave propagating from the inner of the domain will be attenuated exponentially during propagation

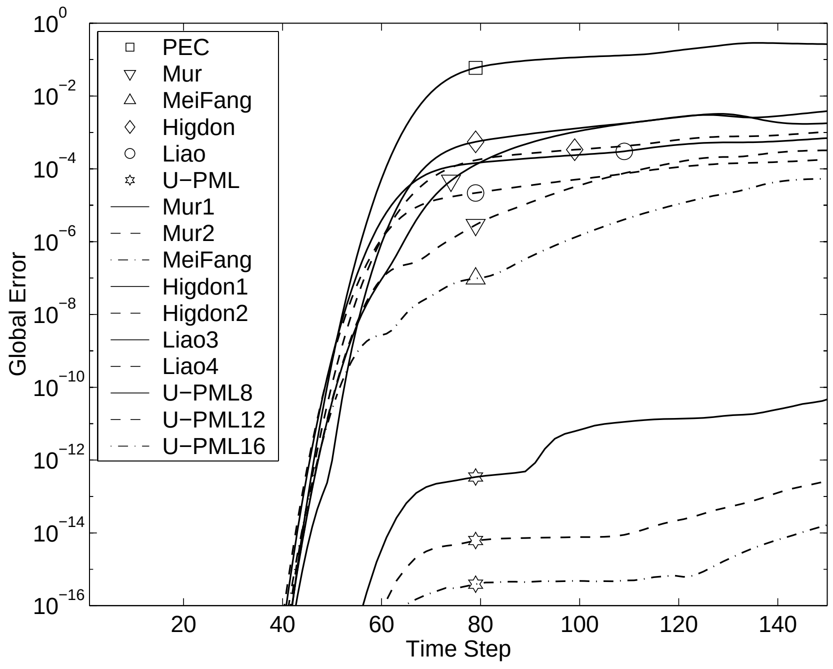


Figure 4.3. Comparison of different ABCs for the TM Maxwell equations.

through the U-PML, and when the wave is reflected in the outer boundary, it will be further attenuated on its way back to the inner domain. If this damping is not enough, we just add another cell layer of U-PML, and the effect is still better than using a classical nonreflecting boundary condition to terminate the U-PML layer.

The Gems time domain 3D code includes PML, U-PML for dispersive materials, and also the first-order Mur ABC [Mur81]. The Mur scheme is the application of the Engquist-Majda [EM77] ABC to the Maxwell equations.

Other approaches to ABC are still being explored in the search for a cheaper ABC, for example [GK98, Ram98]. We have not explored this area. One interesting approach is to use the plane wave time domain (PWTD) method (see Section 1.1.3) [SEAM00] as ABC. This would make it possible to put the ABC only a few cells from the scattering object. On the other hand, this is a global ABC. Traditionally, global ABCs have been considered too computationally expensive. This obstacle may be overcome by PWTD, but we are not quite there yet.

It is well known that it is possible to achieve very good ABCs by using integral formulations. However, if an almost perfect ABC consumes 90% of the available computer resources, it is better to spend some of these resources on a finer discretization.

4.7 Sources

There are many different ways to excite the fields depending on what we want to simulate. Excitations that we use include Huygens' surfaces to model plane waves, point sources to model dipoles, and current and voltage sources in thin wires. Waveguide excitations, which are a major part of the GEMS project, are briefly discussed in [Led01].

Incident plane waves are generated by Huygens' surfaces. They are carefully described in Chapter 6.5 of [Taf95]. However, he never calls them "Huygens' surfaces". Instead he has a more mathematical viewpoint and refers to it as "Total-Field/Scattered-Field Formulation".

A very simple excitation is to use a point source. It can for instance be used to model a dipole. This is done by adding a source term to one of the electric fields. For example,

$$E_x|^{n+1}_{i_s+\frac{1}{2},j_s,k_s} = E_x|^{n+1}_{i_s+\frac{1}{2},j_s,k_s} - \frac{\Delta t f(t_{n+\frac{1}{2}})}{\epsilon V}\,, \tag{4.16}$$

where $V = \Delta x \Delta y \Delta z$ is the cell volume. Because the discrete time derivative of E_x is centered around time level $n + 1/2$, we evaluate the source function f(t) at $t = (n + 1/2)\Delta t$. This is necessary in order to retain the second-order convergence.

We distinguish between hard sources and soft sources. A soft point source is characterized by adding a source term to the field equation as in (4.16). A hard point source is characterized by setting the field to a source term and hence overwriting the value given by the leap-frog update.

In order to avoid introducing high frequency components in the numerical solution, it is important to use a smooth source. The source must be zero at $t = 0$ when we start our simulation (unless we combine it with suitably chosen initial values). This introduces a discontinuity at $t = 0$, but this discontinuity can be kept at machine precision level by suitable choices of parameters for the source.

4.8 Visualization as debugging and validation tool

There are several reasons to visualize results produced by electromagnetic computations. We distinguish between three different purposes of visualization. The first purpose is the most obvious which is to present research to other scientists, funding agencies or the general public. The second purpose is perhaps the most thrilling, which is to learn more about the features of the Maxwell equations. This often stems from working with the results in order to present a good visualization. The third purpose is often regarded as unglamorous, but is nevertheless important. Given a certain expectation of the result, visualizations can be used for debugging and validation of the code. This is a very useful technique to find out which part of the code that is incorrect. In this section we will focus on visualization as debugging and validation tool.

There is an intrinsic problem in visualization of FD-TD data, which originates from staggering the electromagnetic components. Saving E_x, E_y and E_z to file for a given time step is straightforward, but doing visualizations based on these staggered components can be misleading. Simply taking a Cartesian slice through the data and plotting one component seldom poses any problem if the half-cell bias is not important. But plotting the magnitude of the electric field requires either some averaging to get the field at the corners of the FD-TD cell or acceptance for the half-cell bias of the components. The latter is equivalent to a onesided interpolation to the cell corner. The first approach blurs the effect of errors in a single component, and the second is more difficult to combine with exact geometric representations. However, bearing this in mind, debugging still benefits from studying the magnitude of staggered components.

A common techniques for debugging and validation is to compare obtained results with a reference solution pointwise, i.e. plot $(u_{\mathrm{code}} - u_{\mathrm{ref}})$ using slice planes or isosurfaces. Analytic solutions are in some cases available but often numerical results are the only option. Numerical results might come from other codes or obtained from the code prior to the feature examined.

Another emerging technology is virtual reality, which provides tools and display systems for an immersive exploration of data. This might be judged as only a hyped technology that is too complicated for anyone but experts to use. The truth is that used correctly, it enhances the possibility of sharing results with others, especially if a complicated geometry is included.

4.9 Parallelization

The core of the FD-TD scheme is relatively straightforward to parallelize. This issue is studied in detail in Chapter 6, and we give a brief summary here. Chapter 5 in [IH98] is partly devoted to parallelization of the FD-TD scheme. It also covers vectorization and other optimization issues.

Scale-up and speed-up are different measures of efficiency of a parallel code. We define them by:

Definition 4.1 (Scale-up). *The problem size, N, is increased linearly with the number of processors.*

Definition 4.2 (Speed-up). *The problem size, N, is kept fixed, independent of the number of processors.*

We will discuss parallelization of the leap-frog update. Adding PEC, the first-order Mur ABC, and a point source will not affect our discussion, but adding Huygens' surfaces and PML will.

On a parallel computer, it is easy to achieve perfect scale-up for the leap-frog update. This means that the time to execute a certain number of time steps is constant, when the number of processors (and the problem size) is increased. This is a rather nice result, because the need for more memory is one major reason for using parallel computers. Many applications are such that the time to complete them is acceptable if we can fit them into memory.

Consider the one-billion-cell computation in Section 6.6. The time to complete this calculation was only slightly more than one hour. Here the number of time steps was chosen so that the wave would sweep past the airplane once. If we were to use this calculation for practical purposes, we would have to increase the number of time steps to allow for reflections, surface waves, etc. to evolve. However, even if we increase the number of time steps by a factor of ten, the calculation could still be performed overnight.

Perfect speed-up is usually not possible to achieve for the Yee scheme. The main reason for this is the low number of arithmetic operations per cell and time step. For homogeneous materials with $\sigma = \sigma^* = 0$, we only perform 36 arithmetic operations per cell and time step. This can be compared to computational fluid dynamics, where a Navier-Stokes solver performs thousands of arithmetic operations per cell and time step.

The parallelization of a full FD-TD solver, including subcell models, near-to-far-field transformation, Huygens' surfaces, etc. is a much more complex problem. It gets even more complicated when the hybrid schemes described in Chapters 7 and 8 are parallelized. The tricky part is how to achieve a good load balancing. This is more or less automatic for the leap-frog update, since the same amount of work is performed in every cell.

4.10 Subcell models

In the numerical simulation of electromagnetic wave propagation, the existence of subgrid scale phenomena poses some difficulties. Subgrid scale phenomena refer to geometrical features that should influence the solution on the computational grid but have length scales shorter than the grid size. For some problems, such as narrow slots, thin material sheets, surface impedances and thin wires there are subcell models developed.

A thin-wire model permitting arbitrarily oriented wires is presented in [Led01]. Subcell models are thoroughly described in Chapter 10 in Taflove's book [Taf95].

4.11 Near-to-far-field transformations

Three different near-to-far-field transformations have been implemented within the GEMS project. The three transforms are the frequency-domain transform (FD), the time-domain transform (TD) and the continuous-wave transform (CW). These transforms have different applicabilities. The TD transform is suitable when the far field is desired over a range of frequencies, but only for a small number of directions. For the FD transform, the opposite is true. The CW transform is a special case of the FD transform, in which it is assumed that only one frequency is present in the solution. The use of this transform is somewhat limited, since one of the major reasons for using a time-domain method is the possibility of computing a large number of frequencies simultaneously.

The FD transform is equipped with dispersion compensation. This procedure is described in [Mar98]. It can yield significant improvements, especially for the forward scattering direction. The TD transform is described in [MP00].

All transforms probe the fields on a surface. The FD transform performs a discrete Fourier transform (DFT) on surface currents enclosing the object during the timestepping. After the timestepping the field is transformed to far field. For the CW transform, the DFT is only performed during the final period of the computation. The TD transform performs a near-to-far-field transformation in the time domain during each time step. After timestepping a fast Fourier transform (FFT) is performed at each far field direction. This gives us the far field for a broad spectrum of frequencies.

4.12 Frequency-dispersive materials

The Yee scheme is constructed from a non-dispersive formulation of the Maxwell equations and hence cannot be used for computation on frequency-dependent materials. However, it is possible to extend the Yee scheme to handle these materials (see Chapter 9 in [Taf98]). This thesis will not treat dispersive materials.

4.13 Divergence-free nature

If $\sigma = \rho = 0$ in (3.5), the divergence of $\boldsymbol{E}$ and $\boldsymbol{H}$ should be zero. A very nice property of the FD-TD method is that it preserves the divergence; i.e., if it is zero initially, it will stay zero. A proof of this is given in Section 3.6.9 of [Taf95]. The proof is presented for free space, but it is also valid for inhomogeneous materials. It is actually valid independent of how the discrete ϵ-values are chosen due to cancellation of terms in the proof.

Divergence is a continuous property. The proof is of course applied to a discrete approximation of the divergence. However, this approximation is the most natural way to approximate the divergence.

Chapter 5

The GEMS Time-Domain Codes

5.1 Introduction

The codes in the general electromagnetic solvers (GEMS) project are primarily written in Fortran 90. A minor part is written in `C`. Typically, `C` is used for various system-dependent routines.

The Concurrent Versions System (`CVS`) is used for version control, and we use the network Common Data Form (`netCDF`) for output.

Parallelization is performed using `MPI` and `OpenMP`. `MPI` is used to parallelize the structured part of the codes, including the boundary conditions. The unstructured parts are run on one node. The FV-TD code in 3D is parallelized for shared memory machines with `OpenMP`. Parallelization is further discussed in Section 6.7.

5.2 Basic principle

The codes are structured so that every part is performed by a separate module. For instance, there are modules for the leap-frog update, Huygens' surfaces, PML, U-PML, PECs, etc. Furthermore, the FV-TD and FE-TD parts are separate libraries.

The core of the code consists of the timestepping loop. Each timestep (ts) consists of two parts. First, the magnetic fields at $t = (ts - 1/2)\Delta t$ are updated and second, the electric fields at $t = ts\Delta t$ are updated. These two parts are similar in their basic principles. Hence, we will only describe one of the two parts. We choose the electric field update.

First, all interior electric field components are updated according to (4.6)–(4.8) if no material object is present. If material objects are present, the entire domain is handled as if it consists of material. This means that some unnecessary arithmetic operations are performed in the parts of the domain that are vacuum. The positive aspects of this approach is that we do not disrupt the loops with if-statements, checking whether this particular update is vacuum or material.

After the leap-frog update, the outer boundary is updated; and then a number of corrections are applied to the electric field components. PEC components are set to zero, source terms are added, subcell model corrections are applied, etc.

Next, information is exchanged with the unstructured codes. As will be described in Chapter 8, only electric field components are involved in this exchange in 3D.

Finally, the field is probed by near-to-far-field transforms and by the probes requested by the user. Output is also written to file if so requested by the user.

5.3 Storage

The relations between the spatial indexes given in (4.1) and the indexes used in the 3D code follow.

$$\begin{array}{lll}
H_x|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} & \text{are stored in} & \texttt{Hx(i,j,k)} \\
H_y|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} & \text{are stored in} & \texttt{Hy(i,j,k)} \\
H_x|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} & \text{are stored in} & \texttt{Hz(i,j,k)} \\
E_x|^{n}_{i+\frac{1}{2},j,k} & \text{are stored in} & \texttt{Ex(i,j,k)} \\
E_y|^{n}_{i,j+\frac{1}{2},k} & \text{are stored in} & \texttt{Ey(i,j,k)} \\
E_z|^{n}_{i,j,k+\frac{1}{2}} & \text{are stored in} & \texttt{Ez(i,j,k)}
\end{array} \tag{5.1}$$

These variables are reused for all time levels. The default is to use 64-bit precision for these and all other real variables. It can easily be changed to 32-bit precision at the expense of recompiling the entire code.

5.4 Extent of the code

The GEMS time-domain codes (2D and 3D) consist of approximately 160 000 lines. Roughly one third of this is code for the unstructured method, one third is serial FD-TD implementation, and one third is the parallel FD-TD implementation.

5.5 Portability

The codes are written to be portable. They only use standard Fortran 90 features. However, this is not enough to guarantee portability. Many, if not all, compilers have shortcomings, which cause them to create erroneous executable code. In particular, this happens quite often when compiler options for optimization (usually `-O3`) is used. A lot of work is spent on making sure that the code compiles on all computer architecture used within the GEMS project.

The codes are equipped with more than twenty test examples used for automatic validation. The codes are frequently compiled and validated on a number of different computer architectures. They include RISC-processor-based systems, both distributed-memory machines like the IBM SP2, the Cray T3E and clusters of workstations or personal computers, and shared-memory systems like the Silicon Graphics Origin series and the Sun Enterprise Server series, as well as vector-processor systems like the Fujitsu VX and the Cray YMP.

Chapter 6

Parallel Implementation

This chapter is based on Papers 1, 2 and 3 in the list in Chapter 1. Results for the buffered send facility in MPI (`MPI_BSEND`) have been added. More discussions of the results have also been added. Section 6.8 is completely new material.

6.1 Introduction

In this chapter we shall study the parallelization of the core of the Yee scheme, the leap-frog update. These studies have been performed using the parallel version of the FD-TD code `pscyee`. (See Section 1.2 for a description of `pscyee`.) The parallel version of `pscyee` was implemented using the Message Passing Interface (MPI) [MPI] standard. MPI was chosen to guarantee portability of the code. The major part of the implementation was performed on the IBM SP at the center for parallel computers (PDC) at KTH, but `pscyee` has also been tested on several other parallel computers such as a Cray J932, a Fujitsu VX/2 and a Dec Alpha cluster.

The parallel version of `pscyee` uses first-order Mur as absorbing boundary condition. Wave excitation is done with point sources or with Huygens' surfaces. The parallel version of `pscyee` can handle perfect electric conductors (PEC). All these features are described in Chapter 4.

A nice review of parallelization of FD-TD is given by in Chapter 5 of [IH98].

6.2 Terminology

With node we refer to an entity that runs a process. In some cases a node consists of one processor which has its own memory. In other cases it is a processor that shares memory with other processors.

We will use the definitions of speed-up and scale-up given in Section 4.9.

6.3 Serial performance

In Table 6.1, the performance of the sequential version of `pscyee` for several different parallel computers is given.

These values only represent the performance for large problems. For small problems we get an increase of performance to 115–135 Mflop/s for the Dec Alpha processor due to the four MByte secondary level cache, while other computers show a decrease in performance. A substantial effort has been put into optimizing the sequential code.

Computer	Performance of `pscyee`	Peak performance
IBM SP, 160 MHz	160–190	640
Cray J932	80–100	200
Fujitsu VX/2	1000–1350	2200
Dec Alpha	67–90	600

Table 6.1. Performance (Mflop/s) of the FD-TD code `pscyee` for large problem sizes.

6.4 Parallelization strategy

Consider (4.3), repeated here for convenience,

$$\begin{aligned} H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = H_x\big|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} &+ \frac{\Delta t}{\mu \Delta z}\left[E_y\big|^{n}_{i,j+\frac{1}{2},k+1} - E_y\big|^{n}_{i,j+\frac{1}{2},k}\right] \\ &- \frac{\Delta t}{\mu \Delta y}\left[E_z\big|^{n}_{i,j+1,k+\frac{1}{2}} - E_z\big|^{n}_{i,j,k+\frac{1}{2}}\right] . \end{aligned} \tag{6.1}$$

Implementing (6.1) with Fortran we get

```
Hx(i,j,k) = Hx(i,j,k) +                                         &
          (  (Ey(i,j,k+1)-Ey(i,j  ,k))*Cbdz +                   &
             (Ez(i,j,k  )-Ez(i,j+1,k))*Cbdy  ) ,
```

where `Cbdz` $= \Delta t/(\mu\Delta z)$ and `Cbdy` $= \Delta t/(\mu\Delta y)$. We see that the updating of one field component consists of four additions (or subtraction) and two multiplications. The code for updating the other five components is very similar. Hence, during one time step we need to perform 36 arithmetic operations per cell (24 additions and 12 multiplications).

Because all involved operations are local in space it is convenient to perform parallelization using domain decomposition. Figure 6.1 illustrates our parallelization strategy. For clarity, we illustrate with a 1D example using only six cells and two nodes. When node two calculates the Ex_4-value it needs to know the value of $\mathrm{Hz}_{3.5}$ which is stored in node one. Similarly, when node one calculates the $\mathrm{Hz}_{3.5}$-value it needs to know the value of Ex_4 which is stored in node two. This means

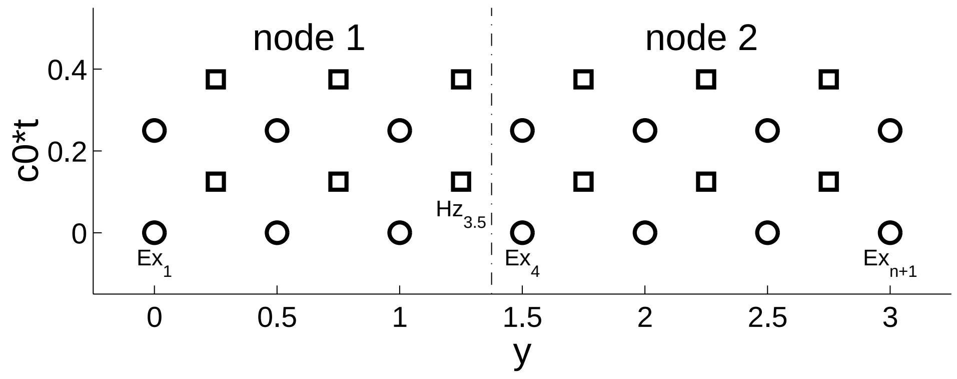


Figure 6.1. Illustration of our parallelization strategy. A 1D Yee grid with six cells (n=6) and $\Delta y = 0.5$ distributed on two nodes.

that during each time step two messages have to be sent, one in each direction. In 3D two of the six electromagnetic variables are included in these messages. Two magnetic field components are sent upwards and two electric field components are sent downwards. If every node houses a $100 \times 100 \times 100$ block, these messages will contain 20 000 floating point values each.

An alternative parallelization strategy would be to let Ex_4 be shared between the two nodes. This would mean that the $\text{Hz}_{3.5}$-value still needs to be sent from node one to node two. However, it is now the $\text{Hz}_{4.5}$-value that is sent from node two to node one. It is still one value that is sent from each node. The difference is that both these messages are now sent at the same time level, which might be more efficient if they can be sent simultaneously. On the other hand, the Ex_4-value must now be calculated on both nodes. Hence there is an increase in the number of arithmetic operations to perform. This strategy is used in [Ged95].

Even larger overlap is of course possible. This would mean that communication would not be needed at every time step. On the other hand, the messages would be larger. This kind of parallelization would be useful on architectures where the communication latency is large.

We have performed five different parallelization implementations using MPI. We denote them by:

- SSEND: where all nodes send first using synchronized blocking send and then receive. (With periodic boundary conditions this would lead to a deadlock. We are saved by the fact that one of the boundary blocks does not need to send any message. This implementation is not very efficient as will be demonstrated in Section 6.5.)

- ISEND: where all nodes send first using nonblocking send and then receive.

- SENDRECV: uses `MPI_SENDRECV` and thus lets the MPI implementation handle the order in which messages are sent and received.
- red-black: where every second node sends first using synchronized blocking send (`MPI_SSEND`) and then receives and vice versa for the other half of the nodes.
- BSEND: where all nodes send first using buffered send and then receive.

For an explanation of the MPI terminology used in this description we refer to [MPI].

6.5 Parallel performance

6.5.1 Scale-up on the IBM SP

The performance of `pscyee` on a 160 MHz node of the IBM SP is 179 Mflop/s for a problem size of $100 \times 100 \times 100$. One iteration takes approximately 0.2 seconds. The peak performance of this type of node is 640 Mflop/s. Hence, the performance of `pscyee` is 28% of the peak performance.

Figure 6.2 illustrates the performance of `pscyee` when the problem size is scaled up with the number of processors (p), so that it is $100 \times p \cdot 100 \times 100$. It displays the result for the five different MPI implementations described in Section 6.4. We can see that all but one of them achieve negligible communication time, i.e. the execution time is independent of the number of processors.

The performance model for the SSEND implementation is based on the assumption that the time needed to take a time step is given by:

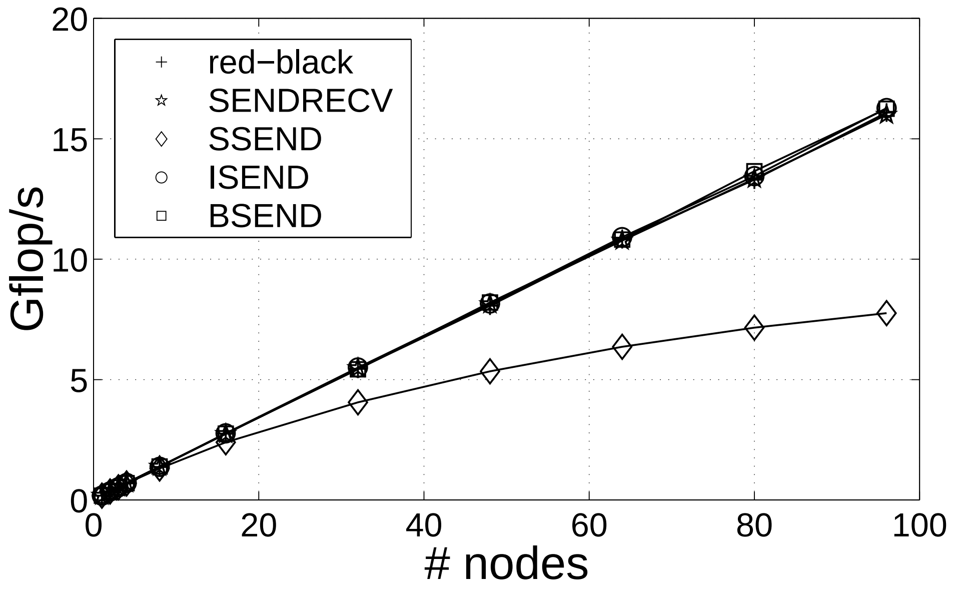
$$t = m(p-1)t_{40000} + t_{calc} \tag{6.2}$$

where p is the number of processors, m is the number of messages sent by each node, t_{40000} is the time it takes to send a message of 40 000 bytes and t_{calc} is the time it takes to perform the calculations on a $100 \times 100 \times 100$ block. In this case we have $m = 4$, $t_{40000} \approx 0.6\,\text{ms}$ and $t_{calc} \approx 0.2\,\text{s}$. (When these computations were performed we sent each field component in a separate message.)

The performance model for the red-black implementation (NOT drawn in Figure 6.2) is based on the assumption that the time for each time step is given by

$$t = 2mt_{40000} + t_{calc} \approx t_{calc} \approx 0.2\,\text{s}\,. \tag{6.3}$$

Notice that time is independent of the number of processors. This is in agreement with the measured performance displayed in Figure 6.2.



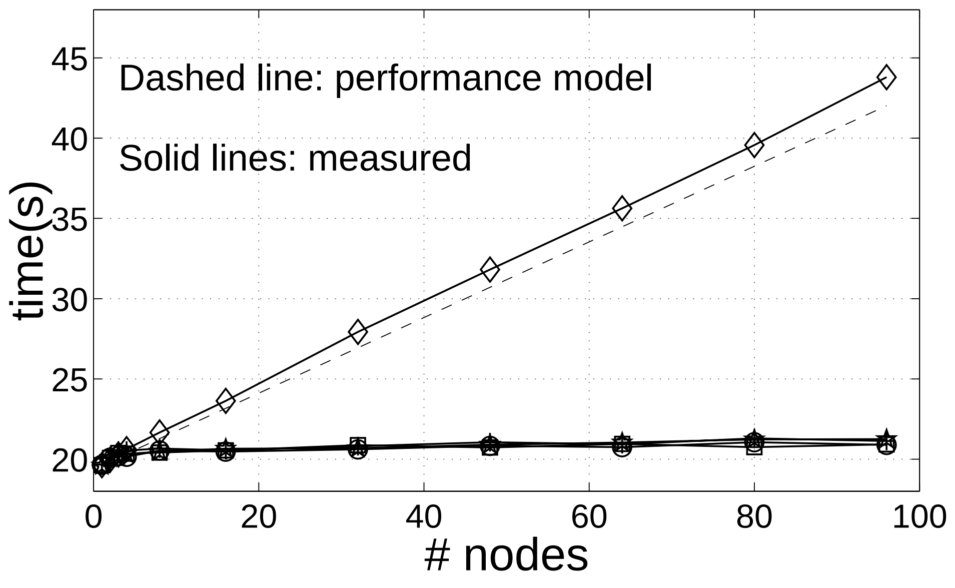


Figure 6.2. Code performance on a 160 MHz node of the IBM SP when the problem size is scaled with the number of processors (p) so that it is $100 \times p \cdot 100 \times 100$.

6.5.2 Speed-up on the IBM SP

Figure 6.3 displays performance results for a problem size of $100 \times 100 \times 100$ using 160 MHz processors of the IBM SP. It displays the best results achieved for a given implementation and a given number of processors. For a given number of processors (p), we have tested all possible domain decompositions that fulfills the constraint that all nodes should have equal sized blocks.

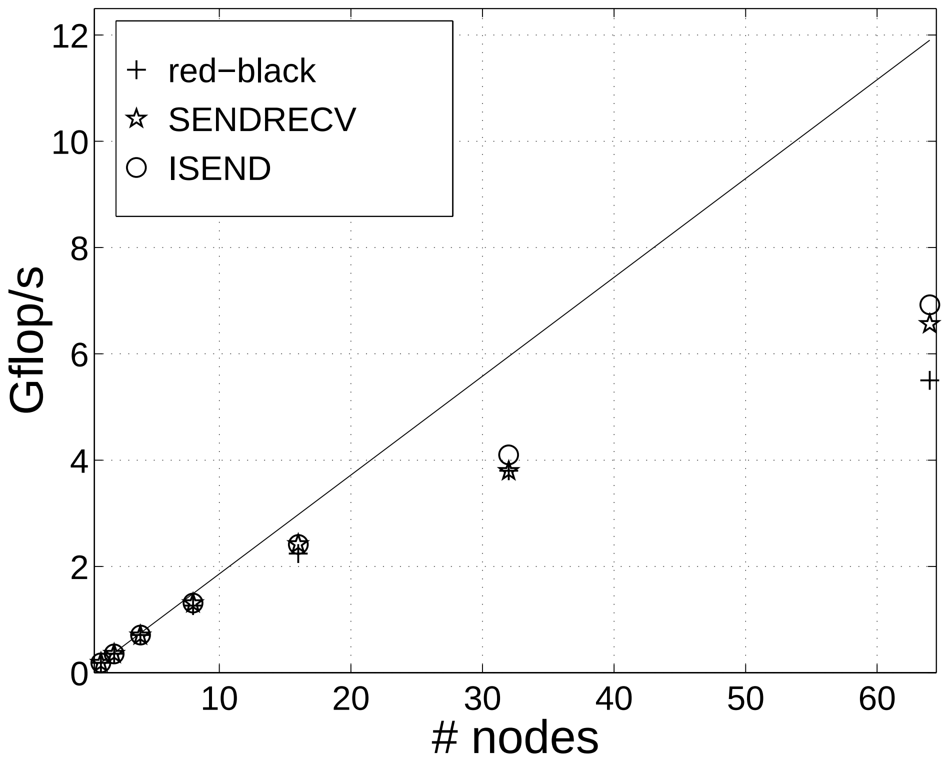


Figure 6.3. Code performance on a 160 MHz node of the IBM SP for a fixed problem size of $100 \times 100 \times 100$. The solid line represents ideal speed-up.

It is more difficult to create a performance model for this case. There are two major reasons for this:

1. The computing speed of the leap-frog updates changes with the problem size. As can be seen in Table 6.1, the speed varies between 160 Mflop/s and 190 Mflop/s on one node.

2. The time to copy to and from the temporary arrays that are sent and received depends on the axis along which the message is sent. Temporary arrays are needed because we want to send 2D slices of the 3D fields that do not lie contiguously in memory. It is more efficient to handle this copying explicitly than letting MPI take care of it. The time to perform the copying depends on the stride of the field values in the 2D slice. When the computational domain on each node becomes small, the time to perform this copying will no longer be negligible.

Let us examine how large effect the copying mentioned above have on the performance in Figure 6.3. We start by constructing an estimate for the maximum performance for the red-black implementation. We do this by comparing calculation time and communication time. For $p = 64$, each node houses a $25 \times 25 \times 25$ block. Measurements show that a time step takes about $t_{calc} = 3.09\,\text{ms}$ A node which has a block with no outer boundaries needs to send two messages and receive two messages per space dimension. (Two field variables are sent in each message.) The size of these messages is $25^2 * 4 * 2 = 5000$ bytes. Measurement show that sending a message of this size takes roughly $0.164\,\text{ms}$ meaning that the total communication time is $t_{com} = 12 * 0.164 = 1.97\,\text{ms}$. Thus neglecting the overhead of copying, we get a maximum possible speedup for $p = 64$ that is given by

$$\frac{t_{calc} + t_{com}}{t_1} \approx 38\,. \tag{6.4}$$

where $t_1 \approx 192\,\text{ms}$ is the time it takes to perform one time step of this computation when only one node is used. This estimate neglects the copying time mentioned above in item 2. Inspecting Figure 6.3 we can calculate the actual speed-up for 64 nodes for the red-black implementation. It is about $(5.5/11.8) * 64 \approx 30$ n. This difference between this value and the result in (6.4) is due to the copying. This conjecture has been proved by removing all copying from the code, which of course will make the results erroneous. This procedure gave a measured speed-up of 36 which is near enough the estimated maximum possible speed-up calculated in (6.4).

The analysis above assumes that each node can only be involved in one message at a time. This is in agreement with the behavior of the red-black implementation because it is based on synchronized blocking send. The SENDRECV and the ISEND implementations do not have this limitation. This is probably the main reason for them having a slightly better performance than the red-black implementation.

Figure 6.4 contains speed-up results for a larger problem size of $252 \times 252 \times 127$. Here, we have also included results for the BSEND implementation. Again, we see that the red-black implementation has lower performance than the other implementations.

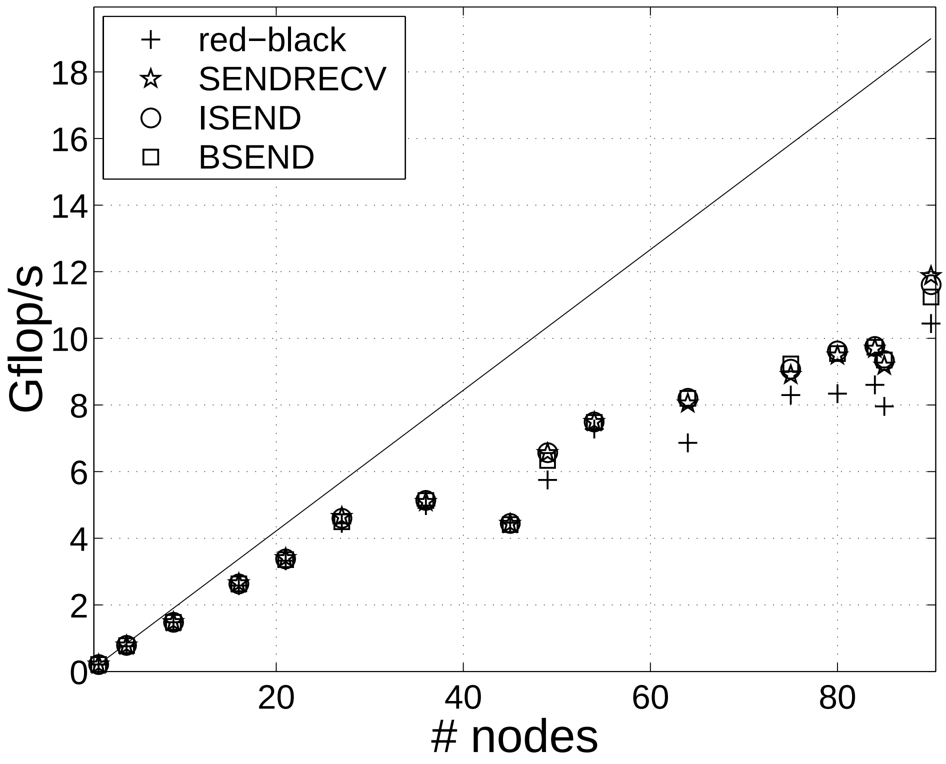


Figure 6.4. Code performance on a 160 MHz node of the IBM SP for a fixed problem size of $252 \times 252 \times 127$. The solid line represents ideal speed-up.

6.5.3 A note on automatic domain decomposition

Lets us consider the following issue. Given p homogeneous nodes of a parallel computer and a computational domain of size $N_x \times N_y \times N_z$, how shall this domain be decomposed into blocks in order to get the shortest execution time? We simplify the problem by demanding that all blocks are of the same size ($n_x \times n_y \times n_z$). If we had a function $E_{time}(n_x, n_y, n_z, p)$, we could find a minima for this function under the constraints $p_x n_x = N_x$, $p_y n_y = N_y$, $p_z n_z = N_z$ and $p_x p_y p_z = p$. However, such a function is hard to construct, due to the reasons mentioned in the enumerated list on page 45

It has been shown [LMG98] that attempts to model the performance of a parallel FD-TD code with polynomials like $E_{time} = k_0 + k_1 n_x n_y n_z + k_2 n_x n_y$ give estimates that in average differ with about 10% from the actual performance. The three terms in the formula given here represents the case when the computational domain is split in slices along the z-direction. Reference [LMG98] had restrictions on the choice of problem size. It was assumed that $N_z << N_x = N_y$ which is typical for microwave circuit analysis. However, we think that a more general case would at best give similar results. Hence with a deviation of 10% between model and actual performance, it is impossible to predict the optimal domain decomposition for a given problem size and a given number of nodes, unless detailed cache behavior analysis is performed. A possible way to lessen the cache effects on performance would be to use padding which is a technique where extra array elements are included in order to obtain an efficient memory access pattern.

6.5.4 Fujitsu VX/2 performance

We have also tried `pscyee` on a two node Fujitsu VX/2. On one node it is possible to achieve a performance of 1.35 Gflop/s. Using the MPI version of `pscyee` this can be doubled to 2.7 Gflop/s. The performance is however problem size dependent. The performance mentioned above has been achieved on extremely elongated problems sizes of $2045 \times 59 \times 59$ respectively $2045 \times 118 \times 59$. On more square like computational domains, for example $200 \times 200 \times 200$, the one node performance is about 1.0 Gflop/s.

6.5.5 Shared memory processors

The main parallelizing effort has been put into the MPI implementation but the code has been tried on several shared memory parallel computers sing automatic parallelization. This includes an SMP node of the IBM SP and a Cray J932.

The SMP nodes of the IBM SP that we used in this test have four 332 MHz processors capable of two floating point operations per clock cycle. This gives a peak performance of 2656 Mflop/s. The performance of `pscyee` is 74 Mflop/s using one processor and 220 Mflop/s using all four processors, i.e. a speedup of 3.0. The poor relation between peak performance and the performance of `pscyee` is due to the low memory bandwidth of these nodes. The compiler option `-qsmp=schedule=dynamic` was used. However, it gave only fractionally better performance than the other schedules.

Figure 6.5 displays the performance on a Cray J932. Automatic parallelization was performed using autotasking with the compiler option `-Otask3`. The problem size is now $128 \times 128 \times 128$ which fits well with the vector length of the Cray J932 which is 64. The one node performance on the Cray J932 is 100 Mflop/s for this problem size, which is exactly half the peak performance.

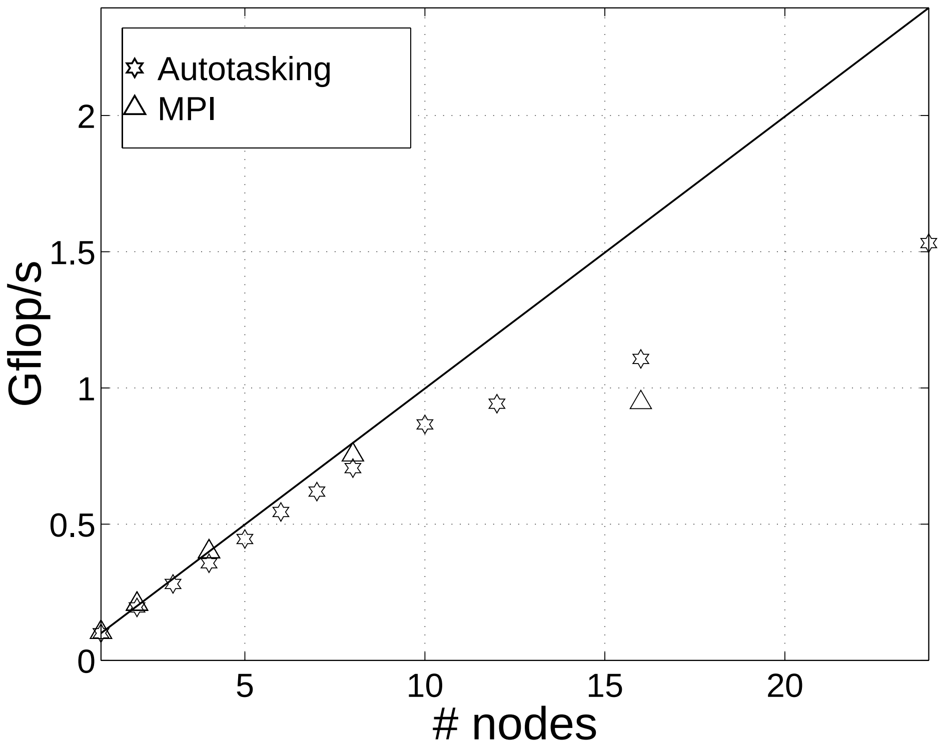


Figure 6.5. Code performance on the Cray J932 for a fixed problem size of $128 \times 128 \times 128$. The solid line represents ideal speed-up.

Figure 6.5 compares the performance of the MPI version with the performance of the autotasking. Here only the best MPI result is shown, which is usually achieved by the SENDRECV implementation. As can be seen they are rather similar demonstrating that automatic parallelization can achieve good performance on a shared memory machine.

6.5.6 Super-linear speed-up

Results for a DEC Alpha cluster with twelve processors are displayed in Figure 6.6. On this computer we have super-linear speed-up. This is due to the four MByte secondary level cache. The full problem of $100 \times 100 \times 100$ uses 36 Mbytes since 32-bit precision is used and nine values are stored for each cell (the three spatial coordinates are also stored in 3D arrays). Hence, when the computational domain is split into smaller parts we get a decrease in cache misses.

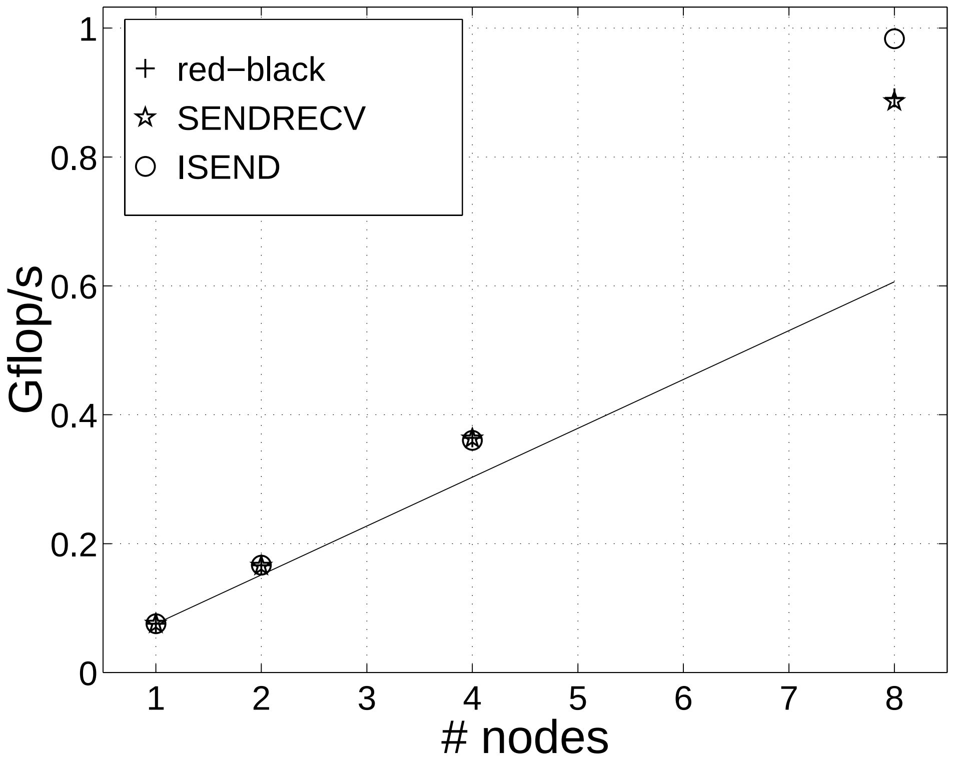


Figure 6.6. Code performance on the cluster of Dec Alpha servers for a fixed problem size of $100 \times 100 \times 100$. The solid line represents ideal speed-up. Results for three different MPI implementations are displayed.

The results for the red-black implementation in Figure 6.6 are hard to spot. The are in fact coinciding with the SENDRECV results. This suggests that the implementation of `MPI_SENDRECV` on the DEC Alpha cluster might be based on the red-black principle.

6.6 A one billion-cell computation

6.6.1 Introduction

We have used our parallel implementation of `pscyee` to perform a very large computation on a commercial airliner, a SAAB 2000. The main purpose of this computation was to demonstrate that it was possible to do a computation with that many cells. This computation was presented at the conference SuperComputing 98. To our knowledge, this is the first published FD-TD computation with more than one billion cells.

6.6.2 Technical details

For homogeneous materials we need only six floating point values per cell, the three electric and the three magnetic field components. Using 32-bit precision (four bytes) means that we need $4 * 6 * 10^9 = 22.4$ Gbyte memory for one billion cells. To get access to that much memory, we used 125 nodes with 160 MHz RS/6000 processors of the IBM SP at PDC, KTH. These nodes have 256 Mbyte memory each making a total of 31.25 Gbyte. Actually, a few of them have more than 256 Mbyte, but we do not want to use it because that would destroy the load balancing.

It is not possible to use all the RAM memory on a node because some of the memory is used by the operating system. Using too much memory will result in swapping which must be avoided because it has a drastic effect on the performance. Tests indicated that it was safe to use up to 200 Mbyte on each node and that one usually could use up to 220 Mbyte.

The object chosen for this computation was a SAAB 2000 aircraft. A Cartesian description of this aircraft was created from a CAD description using CADfix [CAD]. This was performed by Ericsson Saab Avionics. The file delivered from Ericsson Saab Avionics only contained one half of the airplane so we had to create the entire airplane by mirroring. Hence, we got an absolutely symmetric airplane. The resolution was 2.5 cm in all three dimensions which was a factor two per space dimension smaller than the resolution previously used at Ericsson Saab Avionics. We used a problem size of $1260 \times 1260 \times 635$ equaling 1 008 126 000 cells in total. The computational domain was split in $5 \times 5 \times 5$ blocks each with a size of $252 \times 252 \times 127$ cells.

The number of $\boldsymbol{E}$-fields on the surface of the SAAB 2000 was almost two millions and the number of surface quads was almost one million. The memory needed to store this information varied from node to node and was at most nine Mbytes.

The input file containing the PEC information for one half of the SAAB 2000 contained almost one million lines. Since all nodes read this it was very inefficient to use the standard file system, `afs`, which only gave a CPU activity of 2–3% while reading the file. Instead we used the parallel file system, `pfs`, which gave a CPU activity of about 20%. A more efficient strategy would probably be to let one node read the data, analyze it and then distribute it to the other nodes.

The performance of the core of the code is almost 25 Gflop/s. This figure excludes output. When the surface currents and two cutting planes were saved every 20th time step the performance dropped to about 20 Gflop/s. In this case, the time to complete one time step was 1.8 seconds. A total of 2500 time steps were taken and the total execution time, including initializations, was 86 minutes.

The first-order Mur ABC was used and excitation was performed with a point source in front of the airplane. When performing this computation, we used the `MPI_SENDRECV` implementation described in Section 6.4.

Figures 6.7 and 6.8 display a snap shot in time of the solution. Color versions of these figures can be found in Chapter 10.

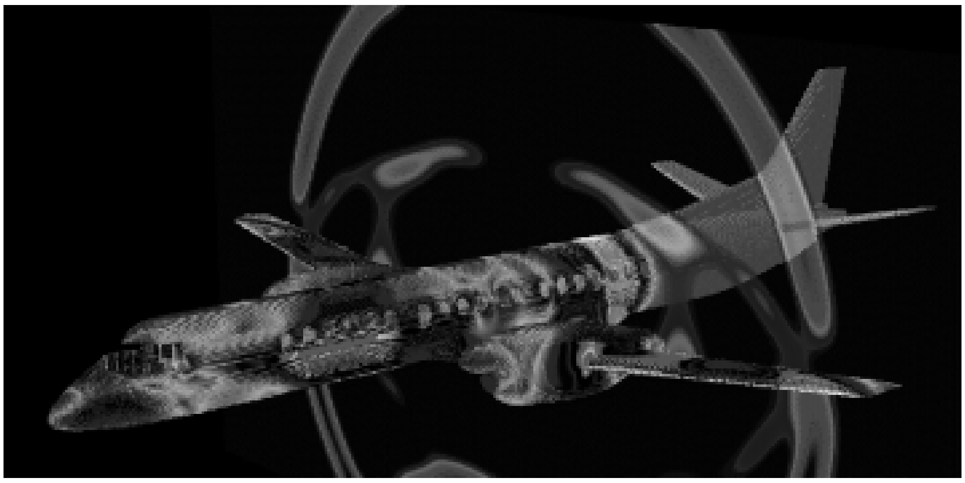


Figure 6.7. The surface currents after 1500 time steps on the SAAB 2000 aircraft. Also the magnitude of the ***H***-field is shown on a cutting plane across the wings perpendicular to the fuselage.

6.6.3 Visualization of large FD-TD data

Large FD-TD simulations do not only require powerful computers. They also put high demands on the post processors. For small and medium size problems you can save the entire electromagnetic field at each time step (or every n:th time step) if you have sufficiently large discs. After the simulation you can go through the data and visualize the features you are looking for and also find unexpected properties of your solution. This is not the case when you are solving large problems. Not only the disc space is limited but the I/O bandwidth is also an effective bottleneck. You have to decide a priori what field values you want to post process and save only them. Furthermore, you need a high end graphical system to visualize the multitude of polygons that constitute the objects in your simulation.

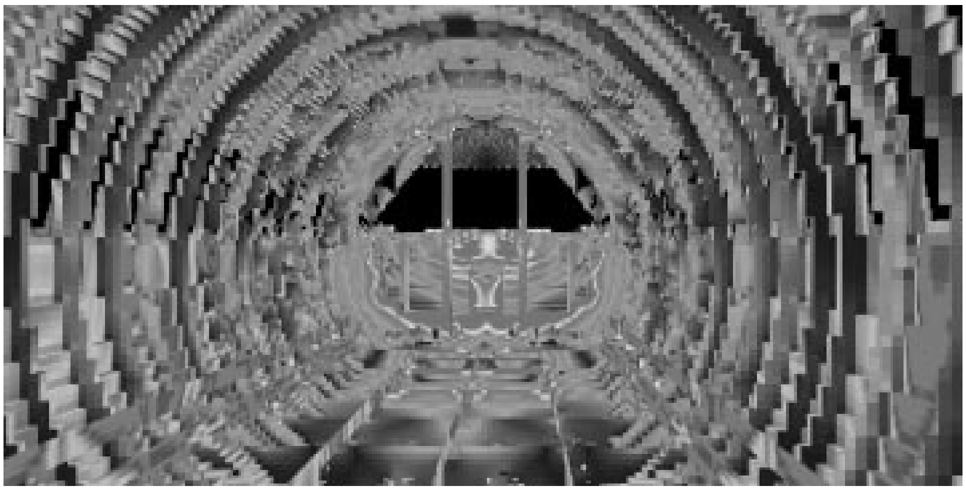


Figure 6.8. The interior of the SAAB 2000 aircraft. Surface currents are shown at the same time as in Figure 6.7.

A new technique has emerged during the last few years which is believed to make the understanding of scientific computing results easier. The concept is usually called "CAVE", CAVE Automated Virtual Environment, and consist of back-projection of images onto semitransparent surfaces. If several surfaces are put together you get a room where you are surrounded by the images. By adding a tracking system where your head position is tracked, stereo images can be produced and highly realistic 3D environments are perceived. With a tracking system for a hand held device you can also interact with this virtual reality.

This technique has obvious benefits: it is easy for several people to be in the room simultaneously and therefore look at images together. The users can thus interact with the virtual reality together and focus on interesting areas. Also, because users see their own hands and feet, for example, as part of the virtual world, they get a heightened sense of being inside that world.

Most of the existing CAVE-like environments have up to four projection surfaces; images are usually projected on three walls and the floor. Adding projection on the ceiling gives a fuller sense of being enclosed in the virtual world.

Projection on all six surfaces of a room allows users to turn around and look in all directions. Thus, their perception and experience are not limited, which is necessary for full immersion. Such a six-surface-system was inaugurated in October 1998 at the Center for Parallel Computers (PDC), KTH and several projects on visualizing CEM solutions have been started where the users will be able to navigate, for example inside an aircraft while lightning strikes. In this case one will directly see the field penetrating the openings of the fuselage and detect "hotspots" to avoid in the context of EMC.

However, the CAVE technology does not ease the urge of effective handling of the output from FD-TD solvers. Even though the computers serving CAVE environments often are high end graphical systems you still have to limit the data saved for post processing. In Figure 6.7 the surface currents are displayed on each 2.5×2.5 cm^2 square constituting the surface of the FD-TD object. Approximately one million quads are put to the visual system and clearly, most of them are not visible in the picture. Furthermore, perhaps one could utilize the concept "level of details" where smaller parts in the background are combined to fewer objects and thus lower the number of polygons to be rendered. For volumetric data semitransparent 3D texture mapping can be utilized. This volume rendering technique can be used to show the field inside the aircraft.

More details on how to interactively visualize FD-TD computations an a CAVE-like environment can be found in [Eng99].

6.7 Parallelization of a full FD-TD code

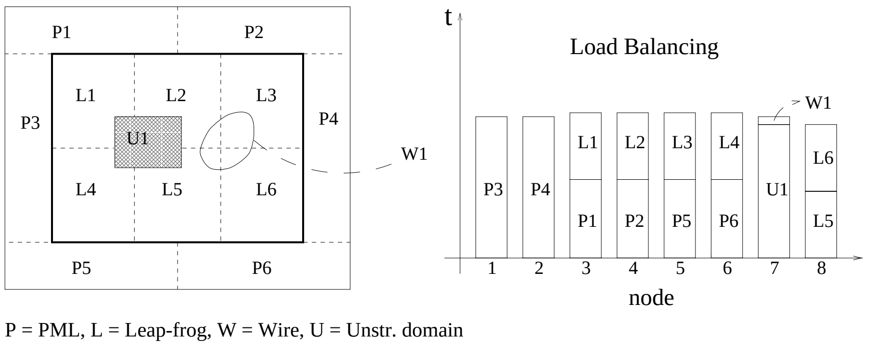


Figure 6.9. The load balancing for a computation including PML, an unstructured domain and an arbitrarily oriented thin wire.

So far we have concentrated on parallelization of the leap-frog update where an almost perfect load balancing is easily obtained. We have demonstrated that it is possible to achieve negligible communication time on the IBM SP when the problem size is increased with the number of processors. On the other hand, ideal speed-up cannot be achieved for a fixed problem size because the communication time will not be negligible when the problem size on each node decreases.

Small additions like the first-order Mur ABC, point sources and PEC materials do not complicate the load balancing. However, a full FD-TD code contains much more. Examples include Huygens' surfaces, frequency dispersive material, PML and a subcell model for thin wires. All these issues, except thin wires, complicate the load balancing. Within GEMS, a parallel multi-block out-of-core implementation of the hybrid methods in Chapter 8 is being implemented. The parallelization approach is illustrated in Figure 6.9.

6.8 Embarrassingly parallel computations

A very trivial parallelization occurs when one wants to run a code with a lot of different input data. Such a case occurs for the FD-TD method, when an angle sweep of monostatic data is desired. A new computation must be performed for each angle. All these computations are independent of each other and might hence be performed simultaneously if the computational resources are available.

Consider an object that is discretized with $150 \times 150 \times 600$ cells. Calculating in 64-bit precision this computation requires slightly more than one Gbyte of memory. This includes memory used by the near-to-far-field transform and a six cells thick PML layer. If the object in question is rotationally symmetric around the z-axis, it would be of interest to find the monostatic RCS for different values of the spherical coordinate θ. We have performed computations on an object like this, where the FD-TD grid was supplied by Ericsson Microwave Systems. The object was modeled as being a PEC. We used the TD near-to-far-field transform described in Section 4.11.

Each of these computations required 3000 time steps and took eight and a half hours to perform on an IBM pwr3 222 MHz processor. The IBM computer at PDC has eight so-called Nighthawk nodes. These nodes each have eight IBM pwr3 222 MHz processors and 4 Gbytes of memory, with the exception of one node that has 16 Gbytes of memory. Using all these nodes, it is possible to run 8+7*3=29 jobs in parallel. Ideally we would like to run one process per available processor, but memory limitations prevents us from running more than three processes on the 4-Gbyte nodes.

We performed a total of 91 computations, letting $\theta = 0, 1, ..90$ degrees. This was done over a weekend. Doing the same computations on the computational resources available at Ericsson Microwave Systems would, according to their estimate, have taken them approximately one month.

Chapter 7

Hybrid Methods in 2D

7.1 Introduction

In this section we present a detailed description and a thorough performance analysis of our hybrid methods in 2D. We consider the TM equations given in (3.8) with $\sigma = \sigma^* = 0$. We demonstrate that staircasing of a circular cylinder destroys the second-order accuracy of FD-TD and that second-order accuracy is recovered when using our hybrid methods. We also present numerical stability experiments.

This Chapter is based on Paper 4 in the list in Chapter 1. Results for discontinuous μ and ϵ have been added. There has also been some improvements in the notation, especially in the Finite Volume section. The description of the Finite Volume method is based on the the Licentiate thesis of Fredrik Edelvik [Ede00].

7.2 Finite-difference method

The FD-TD method is described in Chapter 4. The TM grid is illustrated in Figure 4.1. Here we give a brief additional description. We recall that the fields are staggered both in time and space. This means that when the electric field E_z is known on a time-level, the magnetic field components H_x and H_y can be explicitly calculated on the next half time-level using only the previous H_x and H_y values and the latest E_z values, see Figure 7.1. We use the U-PML method described in Section 4.6 and the profile for σ is the one suggested in [Ged96], i.e.

$$\sigma(\rho) = \frac{1}{3\pi d}(\frac{\rho}{d})^4 \,. \tag{7.1}$$

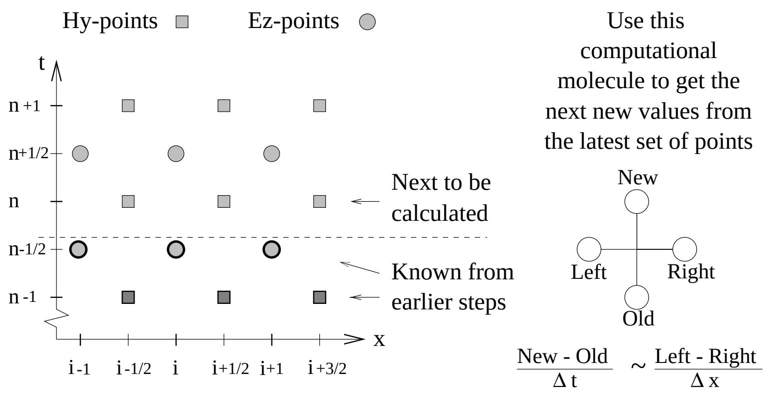

Hy-points
Ez-points
t
n +1
n+1/2
n
n-1/2
n -1
i -1
i -1/2
i
i+1/2
i+1
i +3/2
x
Next to be calculated
Known from earlier steps
Use this computational molecule to get the next new values from the latest set of points
New
Left
Right
Old
New - Old / Δ t ~ Left - Right / Δ x

Figure 7.1. The timestepping mechanism for the FD-TD method: only H_y and E_z are considered here.

7.3 Finite-element method

7.3.1 FE-TD formulation

The FE-TD formulation is based on the second-order differential equation obtained by eliminating either the $\boldsymbol{H}$- or $\boldsymbol{E}$-field in the Maxwell equations. In the 2D TM case with $\sigma = \sigma^* = 0$ the $\boldsymbol{E}$-field is eliminated to yield

$$\mu \frac{\partial^2 \boldsymbol{H}}{\partial t^2} + \nabla \times \frac{1}{\epsilon} \nabla \times \boldsymbol{H} = \boldsymbol{0} \,. \tag{7.2}$$

Together with initial values and boundary conditions this defines the problem to solve. As boundary conditions (BC) we use:

$$\boldsymbol{n} \times (\nabla \times \boldsymbol{H}) = \boldsymbol{0} \quad \text{on } \Gamma_e \text{ (PEC)} \,, \tag{7.3}$$

$$\boldsymbol{n} \times \boldsymbol{H} = \boldsymbol{0} \quad \text{on } \Gamma_h \text{ (PMC)} \,, \tag{7.4}$$

$$\boldsymbol{n} \times \boldsymbol{H} = \boldsymbol{n} \times \boldsymbol{H}_{fdtd}(t) \quad \text{on } \Gamma_t \,. \tag{7.5}$$

In the TM case the first type of BC (PEC) becomes a Neumann boundary condition. The second type models a PMC and becomes a Dirichlet boundary condition. The third type is also a Dirichlet boundary condition, but it is time dependent, and is used at the interface to the FD-TD domain.

For the hybridization we also integrate E_z in the transition layer between the FE-TD domain and FD-TD domain by using Ampère's law. These E_z components are required by the FD-TD scheme as boundary values, see Section 7.6. For visualization purposes E_z is actually integrated in the whole domain.

7.3.2 Spatial discretization

The weak form, or the Galerkin form, of our problem can be stated as: find $\boldsymbol{H} \in W$, where $W = H(curl, \Omega) = \big\{ v : v \in L^2(\Omega), \nabla \times v \in L^2(\Omega) \big\}$, such that

$$\begin{aligned} \int_\Omega \left(\mu \frac{\partial^2 \boldsymbol{H}}{\partial t^2} \cdot \boldsymbol{w} + \frac{1}{\epsilon} \nabla \times \boldsymbol{H} \cdot \nabla \times \boldsymbol{w} \right) d\Omega = \\ - \int_\Gamma \frac{1}{\epsilon} \boldsymbol{n} \times \nabla \times \boldsymbol{H} \cdot \boldsymbol{w} \, ds \ , \end{aligned} \tag{7.6}$$

for all $\boldsymbol{w} \in W$. As trial and test functions we have chosen "edge" or "Whitney" elements [Ned80]. These elements constitute the natural FEM analogue to the Yee scheme in that they essentially yield the same scheme on a Yee grid [Mon93]. They give a "physical" approximation in the sense that only tangential continuity across element edges is enforced, and not normal continuity. This is in agreement with the interface condition given in (3.16). A comparison between edge elements and standard node elements have been presented in [Mon91]. The edge elements have the advantage that the Gauss' laws are better fulfilled by the approximations, and it is easy to implement essential boundary condition.

To define these linear edge elements which are second-order accurate, consider the standard linear basis functions Φ_i for nodal-based finite elements, constructed such that $\Phi_i = 1$ in node number i and $\Phi_i = 0$ in all other nodes. Take edge e to be the edge on a triangular element joining node i and node j. The basis function for edge e is

$$\varphi_e = \Phi_i \nabla \Phi_j - \Phi_j \nabla \Phi_i \ , \tag{7.7}$$

and is illustrated in Figure 7.2.

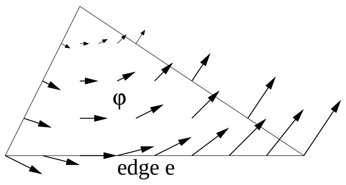


Figure 7.2. The basis function φ_e for edge e plotted over a triangle.

The basis function for edge e has the following properties:

- $\nabla \cdot \varphi_e = 0$.
- φ_e has constant tangential component ($= 1/length$) along edge e, which means that the tangential component is continuous around edge e. The normal component is discontinuous.
- φ_e has zero tangential component along the other edges.

The approximation $\boldsymbol{H}^h \in W^h$ of $\boldsymbol{H} \in W$ can then be written as $\boldsymbol{H}^h = \sum_e \alpha_e \varphi_e$, where α_e is an approximation of the $\boldsymbol{H}$ component tangential to edge e, multiplied by the length of edge e. The edge element method applied to (7.6) then becomes: solve the system of ordinary differential equations

$$\boldsymbol{M}\frac{d^2\alpha}{dt^2} + \boldsymbol{S}\alpha = \boldsymbol{f}^A \ , \tag{7.8}$$

where the vector α contains the unknown α_e:s. The matrix elements coupling edge k and edge l is:

$$(\boldsymbol{M})_{kl} \quad = \quad \int_\Omega \mu\varphi_k \cdot \varphi_l \, d\Omega \ , \tag{7.9}$$

$$(\boldsymbol{S})_{kl} \quad = \quad \int_\Omega \frac{1}{\epsilon} (\nabla \times \varphi_k) \cdot (\nabla \times \varphi_l) \ d\Omega \ , \tag{7.10}$$

and the right-hand side $\boldsymbol{f}^A$ contains the contributions from the boundary conditions, i.e. we have $\boldsymbol{f}^A = \boldsymbol{f}^D + \boldsymbol{f}^N$. The general form of the contributions from a time dependent Dirichlet boundary Γ_t is

$$(\boldsymbol{f})_k^D = -\sum_{l\in\Gamma_t}\left((\boldsymbol{M})_{kl}\frac{\partial^2\alpha_l}{\partial t^2} + (\boldsymbol{S})_{kl}\,\alpha_l\right) . \tag{7.11}$$

The PMC BC is just a simple special case of this, where these terms become zero. In the hybrid code the "boundary values" from the FD-TD domain will enter the FE-TD domain in this way. At a Neumann boundary Γ_e, $\boldsymbol{n} \times \nabla \times \boldsymbol{H}$ is to be specified, see (7.3). If this expression is a known function on the boundary then

$$(\boldsymbol{f})_k^N = -\int_{\Gamma_e} \frac{1}{\epsilon}\,\boldsymbol{n} \times \nabla \times \boldsymbol{H} \cdot \varphi_k \, ds \ , \tag{7.12}$$

has to be computed.

When formulating the local mass and stiffness matrices it is convenient to introduce the following notation. Define

$$\boldsymbol{l}_i = (x_{i-1} - x_{i+1}, y_{i-1} - y_{i+1})^T \ , \tag{7.13}$$

as the vector along edge i, with direction counterclockwise (see Figure 7.3). The index i in (7.13) assumes the values 1, 2 and 3 cyclically, so that if $i = 3$ then $i + 1 = 1$. Furthermore, let the the vector $\boldsymbol{d}$ be given by

$$\boldsymbol{d} = (d_1, d_2, d_3)^T \ , \tag{7.14}$$

where $d_i = 1$ if the local direction of edge vector l_i is the same as the globally defined direction, otherwise $d_i = -1$. Expressed in this notation the local three by three geometrical stiffness matrix $\boldsymbol{s}^g$ becomes

$$\boldsymbol{s}^g = \boldsymbol{d}\boldsymbol{d}^T / A \ , \tag{7.15}$$

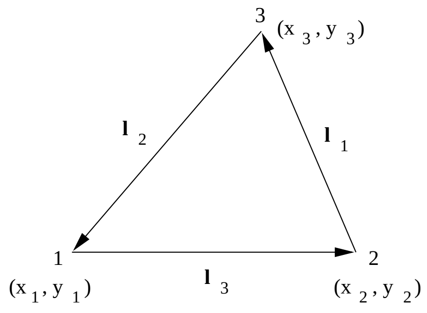


Figure 7.3. Triangular element with nodes and vectors along edges.

where A is the area of the triangular element, and the geometrical mass matrix $\boldsymbol{m}^g$ has the elements

$$\begin{aligned} m^g_{ii} &= (\boldsymbol{l}^T_{i+1}\boldsymbol{l}_{i+1} + \boldsymbol{l}^T_{i-1}\boldsymbol{l}_{i-1} - \boldsymbol{l}^T_{i+1}\boldsymbol{l}_{i-1})/(24A) \ , \\ m^g_{ij} &= -(\boldsymbol{l}^T_k\boldsymbol{l}_k + \boldsymbol{l}^T_i\boldsymbol{l}_j)/(24A \cdot d_i d_j), \ \ i \neq j \neq k \ , \end{aligned} \tag{7.16}$$

where i is cyclically defined as explained above. The material properties μ and ϵ are assumed to be constant within each triangle. The matrices $\boldsymbol{m}$ and $\boldsymbol{s}$ are thus given by scalars times geometric mass/stiffness matrices, i.e.

$$\boldsymbol{m} = \mu \cdot \boldsymbol{m}^g \ , \tag{7.17}$$

$$\boldsymbol{s} = \frac{1}{\epsilon} \cdot \boldsymbol{s}^g \ . \tag{7.18}$$

7.3.3 Time discretization

A major drawback of the edge element method is that it is difficult to obtain an explicit scheme on a general unstructured grid. For the first-order formulation of the Maxwell equations, it has been shown that masslumping can be justified only if all the triangles are acute [MP94]. However, the FE solver is meant to be used to resolve fine geometrical features and in that case an explicit time integrator cannot be used anyway due to the small time step required. Using (7.11) the system (7.8) may be written as

$$\begin{cases} (\boldsymbol{M} \ \boldsymbol{M}_D) \begin{pmatrix} \frac{d^2\alpha}{dt^2} \\ \frac{d^2\alpha_D}{dt^2} \end{pmatrix} + (\boldsymbol{S} \ \boldsymbol{S}_D) \begin{pmatrix} \alpha \\ \alpha_D \end{pmatrix} = \boldsymbol{f} \ , \\ \alpha(0) = \alpha_0 \ , \\ \frac{d\alpha(0)}{dt} = \boldsymbol{v}_0 \ , \end{cases} \tag{7.19}$$

where $\boldsymbol{M}$ and $\boldsymbol{M}_D$ are mass matrices, $\boldsymbol{S}$ and $\boldsymbol{S}_D$ are stiffness matrices from the discretization of the double-curl operator and α is the magnetic field vector, which we will solve for. The vector α_D contains the given magnetic field at the Dirichlet boundary and the matrices with index D describe the influence from this boundary. Note that the right-hand side now is $\boldsymbol{f} = \boldsymbol{f}^N$. To get a first-order system we introduce $\boldsymbol{v}$ as the time derivative of α and rewrite the system as

$$\left\{ \begin{array}{l} \begin{pmatrix} \frac{d\alpha}{dt} \\ \frac{d\alpha_D}{dt} \end{pmatrix} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{v}_D \end{pmatrix} , \\ (\boldsymbol{M}\ \boldsymbol{M}_D) \begin{pmatrix} \frac{d\boldsymbol{v}}{dt} \\ \frac{d\boldsymbol{v}_D}{dt} \end{pmatrix} + (\boldsymbol{S}\ \boldsymbol{S}_D) \begin{pmatrix} \alpha \\ \alpha_D \end{pmatrix} = \boldsymbol{f} \, . \end{array} \right. \tag{7.20}$$

For the time-discretization of this system we have implemented two methods, the two stage backward difference formula (BDF-2) and the θ-method. The θ-method yields

$$\frac{\alpha^{n+1} - \alpha^n}{\Delta t} = \theta \boldsymbol{v}^{n+1} + (1-\theta)\boldsymbol{v}^n \, , \tag{7.21}$$

$$\begin{aligned} &\boldsymbol{M}\frac{\boldsymbol{v}^{n+1} - \boldsymbol{v}^n}{\Delta t} + \boldsymbol{S}(\theta\alpha^{n+1} + (1-\theta)\alpha^n) = \\ &\theta \boldsymbol{f}^{n+1} + (1-\theta)\boldsymbol{f}^n - \boldsymbol{M}_D \frac{\boldsymbol{v}_D^{n+1} - \boldsymbol{v}_D^n}{\Delta t} - \boldsymbol{S}_D(\theta\alpha_D^{n+1} + (1-\theta)\alpha_D^n) \, , \end{aligned} \tag{7.22}$$

where

$$\boldsymbol{v}_D^{n+1} = \left(-(1-\theta)\boldsymbol{v}_D^n + \frac{\alpha_D^{n+1} - \alpha_D^n}{\Delta t} \right) / \theta \, , \tag{7.23}$$

and α^n corresponds to $\alpha(t_n)$, $\Delta t = t_{n+1} - t_n$ and $0 < \theta \leq 1$. Note that we use the θ-method not only for the time-integration but also for approximating the time-derivative $\boldsymbol{v}_D^{n+1}$ on the right-hand side of (7.22). The solution process in each time step begins by applying (7.23) to compute $\boldsymbol{v}_D^{n+1}$ given the boundary condition α_D^{n+1} and then solve the system (7.22), which is rewritten as

$$\begin{bmatrix} \boldsymbol{I}/\Delta t & -\theta \boldsymbol{I} \\ \theta \boldsymbol{S} & \boldsymbol{M}/\Delta t \end{bmatrix} \begin{bmatrix} \delta\alpha \\ \delta\boldsymbol{v} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{res}_\alpha \\ \boldsymbol{res}_v \end{bmatrix} , \tag{7.24}$$

where

$$\begin{aligned} \delta\alpha &= \alpha^{n+1} - \alpha^n \, , \\ \delta\boldsymbol{v} &= \boldsymbol{v}^{n+1} - \boldsymbol{v}^n \, , \end{aligned} \tag{7.25}$$

and the residuals on the right hand side are

$$\begin{aligned}
\boldsymbol{res}_\alpha &= -\boldsymbol{v}^n \ , \\
\boldsymbol{res}_v &= \boldsymbol{S}\alpha^n - \theta \boldsymbol{f}^{n+1} - (1-\theta)\boldsymbol{f}^n \\
&+ \boldsymbol{M}_D \tfrac{\boldsymbol{v}_D^{n+1} - \boldsymbol{v}_D^n}{\Delta t} + \boldsymbol{S}_D(\theta\alpha_D^{n+1} + (1-\theta)\alpha_D^n) \ ,
\end{aligned} \tag{7.26}$$

The solution of (7.24) utilizes the following block LU factorization of the Jacobian of the system

$$\begin{bmatrix} \boldsymbol{I}/\Delta t & -\theta\boldsymbol{I} \\ \theta\boldsymbol{S} & \boldsymbol{M}/\Delta t \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}/\Delta t & \boldsymbol{0} \\ \theta\boldsymbol{S} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\Delta t\theta\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} , \tag{7.27}$$

where the Schur complement matrix $\boldsymbol{J}$ is given by

$$\boldsymbol{J} = \boldsymbol{M}/\Delta t + \Delta t\theta^2 \boldsymbol{S} \ . \tag{7.28}$$

The solution can then be divided into the three steps

$$\begin{aligned}
\delta\alpha^p &= -\Delta t \boldsymbol{res}_\alpha \ , \\
\boldsymbol{J}\delta\boldsymbol{v} &= -\boldsymbol{res}_v - \Delta t\theta\boldsymbol{S}\delta\alpha^p \ , \\
\delta\alpha &= \delta\alpha^p + \Delta t\theta\delta\boldsymbol{v} \ .
\end{aligned} \tag{7.29}$$

The first two steps are the forward substitution in the block LU factorization while the third step is the back substitution.

This scheme is unconditionally stable for $1/2 \le \theta \le 1$. With $\theta = 1/2$ we obtain the only second-order θ-method, the midpoint or Crank-Nicholson method. Note that this method is equivalent to the method used by Hwang and Wu [HW99] on the second-order system. As an alternative we have implemented the BDF-2 method, which also is second-order accurate and unconditionally stable. Applied to

$$\frac{dx}{dt} = \boldsymbol{f}(x) \ , \tag{7.30}$$

the BDF-2 method, with constant time step, becomes

$$\frac{3x^{n+1} - 4x^n + x^{n-1}}{2\Delta t} = \boldsymbol{f}(x^{n+1}) \ . \tag{7.31}$$

The BDF-2 method has an error constant which is four times larger than the error constant of the Crank-Nicholson method, but it has shown better stability properties than the Crank-Nicholson method when used in the hybrid code. We note that the time-integration requires initial values for the “velocities”. This is similar to the requirement of initial data in the Yee scheme, and we simply set an initial state where all fields and their time derivatives are zero. Also note that BDF-2 requires values at two previous time steps, which for the initial values is equivalent to knowing also the initial “accelerations”. This would normally require some special treatment to start the integration, but since we assume zero initial values for the “accelerations” this poses no problem.

7.3.4 Workload and memory requirements

The workload per time step is of the order $4bN$ where b is the bandwidth of the system (7.24) after renumbering and N is the number of unknowns which is equal to the number of edges in the unstructured grid. Neglecting boundaries, the number of edges equal 1.5 times the number of triangles. The factorization takes $2b^2N$ arithmetical operations, but it is only performed once. The bandwidth is of course problem dependent. The memory requirements are bN floating point values. We use the reverse Cuthill-McKee [GPS76] ordering algorithm to reduce the bandwidth of the system (7.24).

The use of an iterative method would decrease the computational workload and the memory requirements. Iterative solvers will be further discussed in Chapter 8.

7.4 Finite-volume method

7.4.1 FV formulation

The Finite Volume (FV) solver is based on the following integral formulation of Ampère’s and Faraday’s laws:

$$\frac{\partial}{\partial t}\int_A \mu\,\boldsymbol{H}\,dA = -\oint_\Gamma \boldsymbol{n}\times\boldsymbol{E}\,dl\,, \tag{7.32}$$

$$\frac{\partial}{\partial t}\int_A \epsilon\,\boldsymbol{E}\,dA = \oint_\Gamma \boldsymbol{n}\times\boldsymbol{H}\,dl\,, \tag{7.33}$$

where A is an arbitrary area, Γ is the path that encloses A and $\boldsymbol{n}$ is the unit normal.

7.4.2 Spatial discretization

The integral formulations in (7.32) and (7.33) are discretized on a staggered grid by introducing a dual grid to the primary triangular grid. The magnetic components are situated at the nodes of the primary grid and the electric components, in the 2D TM case only E_z, are situated at the nodes of the dual grid. The dual grid is created at the preprocessing stage by defining dual nodes at the barycenters of the primary cells; see Section 7.4.5 for a detailed description.

In the 2D TM case integrating the magnetic and electric fields over each dual and primary cell gives the following integral form:

$$\frac{\partial}{\partial t}\int_{A_j^d} \tilde{\mu}_j^d\, \boldsymbol{H}\, dA = -\sum_k \int_{\Gamma_{j,k}^d} \boldsymbol{n}_{j,k}^d \times (E_z\,\hat{\boldsymbol{z}})\, dl\,, \tag{7.34}$$

$$\frac{\partial}{\partial t}\int_{A_i^p} \epsilon_i^p\, E_z\, dA = \sum_m \int_{\Gamma_{i,m}^p} \boldsymbol{n}_{i,m}^p \times \boldsymbol{H}\, dl\,, \tag{7.35}$$

where A_j^d is the area of dual cell j, Γ_j^d is the path that encloses A_j^d and $\boldsymbol{n}_{j,k}^d$ are the unit edge normals for the dual edges k in dual cell j. The variables belonging to the primary cell i are defined similarly.

All materials are defined relative to primary grid cells. For dielectric materials that is a natural definition. However, the magnetic permeability, μ, is associated with the dual grid cells. Therefore, μ requires averaging. The average permeability of dual cell j is computed as

$$\tilde{\mu}_j^d = \sum_q \mu_q^p A_{j,q}^d\,, \tag{7.36}$$

where $A_{j,q}^d$ is the part of the area A_j^d that is inside primary cell q. Performing the material averaging in this manner preserves the second-order accuracy of the solver for inhomogeneous materials.

The area integrals in (7.34) and (7.35) are evaluated by taking the average values of the fields multiplied by the areas of the respective cells. Simplifying the two integrands in the line integrals implies

$$\tilde{\mu}_j^d \frac{\partial}{\partial t}\boldsymbol{H}_j \quad = \tfrac{1}{A_j^d}\sum_k \int_{\Gamma_{j,k}^d} \tilde{E}_z|_{j,k}\, \boldsymbol{t}_{j,k}^d\, dl\,, \tag{7.37}$$

$$\epsilon_i^p \frac{\partial}{\partial t} E_z|_i \quad = \tfrac{1}{A_i^p}\sum_m \int_{\Gamma_{i,m}^p} \tilde{\boldsymbol{H}} \cdot \boldsymbol{t}_{i,m}^p\, dl\,, \tag{7.38}$$

where $\boldsymbol{t}_{j,k}^d$ are unit vectors for the dual edges k in dual cell j and $\boldsymbol{t}_{i,m}^p$ are unit vectors for the edges m in primary cell i. The line integral in (7.37) is evaluated by assuming that the electric field is piecewise linear along the dual edges. Hence, $\tilde{E}_z|_{j,k}$ is computed by taking the arithmetic mean value of the electric field at the two nodes defining the dual edge, $\boldsymbol{t}_{j,k}^d$. However, we cannot use the same approach when calculating the integral in (7.38) since that does not guarantee that the divergence is preserved on a local cell level. This has been found to be very critical if spurious modes in the numerical solution are to be suppressed. The divergence is preserved if we incorporate an “FD-TD”-correction along the edges in the primary grid (see Section 7.4.4 for a proof). Following Riley et al. [RT97], the magnetic field projected along the primary edge $\boldsymbol{t}^p_{i,m}$ is evaluated as

$$\begin{aligned}\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p_{i,m} &= \boldsymbol{H} \cdot \boldsymbol{n}^d_{j,k} \left(\boldsymbol{n}^d_{j,k} \cdot \boldsymbol{t}^p_{i,m} \right) \\ &\quad + \frac{1}{2} \left[(\boldsymbol{H}_j + \boldsymbol{H}_r) - \left((\boldsymbol{H}_j + \boldsymbol{H}_r) \cdot \boldsymbol{n}^d_{j,k} \right) \boldsymbol{n}^d_{j,k} \right] \cdot \boldsymbol{t}^p_{i,m} \,,\end{aligned} \tag{7.39}$$

where $\boldsymbol{H}_j$ and $\boldsymbol{H}_r$ are the magnetic field at the two nodes defining the primary edge and $\boldsymbol{H} \cdot \boldsymbol{n}^d_{j,k}$ is the FD-TD component in the direction $\boldsymbol{n}^d_{j,k}$ orthogonal to the dual edge $\boldsymbol{t}^d_{j,k}$, which crosses the primary edge $\boldsymbol{t}^p_{i,m}$, see Figure 7.4. The FD-TD component is updated according to

$$\frac{\partial\, \boldsymbol{H} \cdot \boldsymbol{n}^d_{j,k}}{\partial t} = \frac{E_z|_q - E_z|_i}{\Delta^d_{j,k}\, \tilde{\mu}^d_{j,k}} \,, \tag{7.40}$$

where $\Delta^d_{j,k}$ is the length and $\tilde{\mu}^d_{j,k}$ is the average permeability of the dual edge $\boldsymbol{t}^d_{j,k}$. The average permeability, $\tilde{\mu}^d_{j,k}$, is approximated as

$$\tilde{\mu}^d_{j,k} = \mu^p_q \, \Delta_q + \mu^p_i \left(\Delta^d_{j,k} - \Delta_q \right) \,, \tag{7.41}$$

where i and q are the two primary cells sharing dual edge $\boldsymbol{t}^d_{j,k}$, Δ_q is the part of the length of the dual edge that is inside primary cell q, see Figure 7.4.

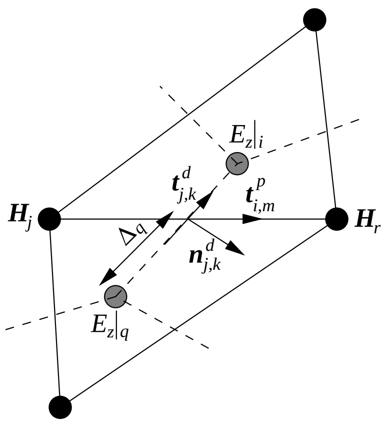


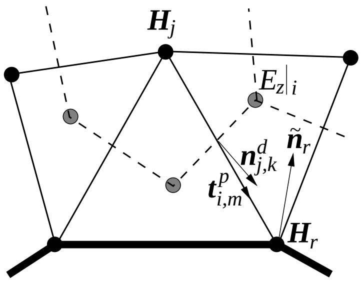
Figure 7.4. The magnetic field along $\boldsymbol{t}^p_{i,m}$ is corrected using the FD-TD value in direction $\boldsymbol{n}^d_{j,k}$.

Taking a closer look at (7.39) we see that if the primary and dual edges are orthogonal, the vectors $\boldsymbol{n}^d_{j,k}$ and $\boldsymbol{t}^p_{i,m}$ are parallel and the second part of (7.39) vanishes. Hence, the name “FD-TD”-correction is somewhat misleading since that term is actually the important one. The magnetic node values are only used to give a better approximation of the edge-projected field in the case when $\boldsymbol{n}^d_{j,k}$ and $\boldsymbol{t}^p_{i,m}$ do not align.

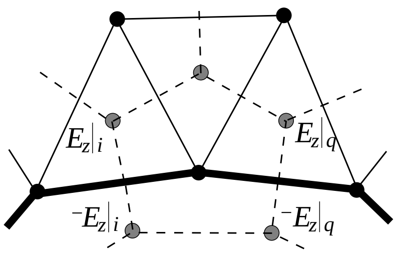
The boundary condition for a perfect magnetic conductor (PMC) gives us that the tangential component of the magnetic field at the object is zero. A complication occurs whenever the computation of $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p_{i,m}$ does not reduce to the FD-TD term, $\boldsymbol{H} \cdot \boldsymbol{n}^d_{j,k}$, where $\boldsymbol{t}^p_{i,m}$ denotes a primary edge with one node on the boundary. When that is not the case the following alternative is used (see Figure 7.5(a)):

$$\begin{aligned}\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p_{i,m} &= \boldsymbol{H} \cdot \boldsymbol{n}^d_{j,k} \left(\boldsymbol{n}^d_{j,k} \cdot \boldsymbol{t}^p_{i,m}\right) \\ &\quad + \frac{1}{2} \left[\boldsymbol{H}_j + \left(\boldsymbol{H}_j \cdot \tilde{\boldsymbol{n}}_r\right) \tilde{\boldsymbol{n}}_r - \left(\boldsymbol{H}_j + \left(\boldsymbol{H}_j \cdot \tilde{\boldsymbol{n}}_r\right) \tilde{\boldsymbol{n}}_r\right) \cdot \boldsymbol{n}^d_{j,k}\right] \cdot \boldsymbol{t}^p_{i,m} ,\end{aligned} \tag{7.42}$$

where $\tilde{\boldsymbol{n}}_r$ denotes an average normal at the boundary node. This normal is defined by taking the average direction of the two boundary edges including the boundary node and then taking the cross product with that direction and $\hat{z}$.



(a) PMC boundary



(b) PEC boundary

Figure 7.5. Primary and dual cells at PMC and PEC boundaries. The boundaries are illustrated by thick lines. The PEC boundary condition is enforced using the method of images.

For a perfect electric conductor (PEC) the tangential electric components, in the TM case only E_z, should equal zero at the boundary. However, the E_z components are not situated on the boundary. The boundary condition is enforced by setting the electric field inside the conductor equal to the value of the electric field directly outside the conductor with a change of sign, see Figure 7.5(b). These uniquely determined virtual image nodes are identified when the dual grid is constructed in the preprocessing phase, see Section 7.4.5.

7.4.3 Time discretization

We use a third-order staggered Adams–Bashforth scheme (ABS3) [GFD00],

$$\begin{aligned} \boldsymbol{H}_j^{n+\frac{1}{2}} &= \boldsymbol{H}_j^{n-\frac{1}{2}} \\ &+ \frac{\Delta t}{\tilde{\mu}_j^d A_j^d} \sum_k \left(\frac{25}{24} \tilde{E}_z|_{j,k}^n - \frac{1}{12} \tilde{E}_z|_{j,k}^{n-1} + \frac{1}{24} \tilde{E}_z|_{j,k}^{n-2} \right) \boldsymbol{t}_{j,k}^d \Delta_{j,k}^d \,, \end{aligned} \tag{7.43}$$

$$\begin{aligned} E_z|_i^{n+1} = E_z|_i^n + \frac{\Delta t}{\epsilon_i^p A_i^p} \sum_m \Bigg(\frac{25}{24} \tilde{\boldsymbol{H}} \cdot \boldsymbol{t}_{i,m}^p |^{n+\frac{1}{2}} - \\ \frac{1}{12} \tilde{\boldsymbol{H}} \cdot \boldsymbol{t}_{i,m}^p |^{n-\frac{1}{2}} + \frac{1}{24} \tilde{\boldsymbol{H}} \cdot \boldsymbol{t}_{i,m}^p |^{n-\frac{3}{2}} \Bigg) \Delta_{i,m}^p \,, \end{aligned} \tag{7.44}$$

$$\boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d |^{n+\frac{1}{2}} = \boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d |^{n-\frac{1}{2}} + \Delta t \, \mathcal{F} \left(\frac{25}{24} E_z^n - \frac{1}{12} E_z^{n-1} + \frac{1}{24} E_z^{n-2} \right) \,, \tag{7.45}$$

where $\mathcal{F}$ is an operator taking care of the update of $\boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d$ according to (7.40). Since ABS3 is a staggered time integrator the time-coupling with FD-TD is straightforward, see Section 7.6. Furthermore, its stability properties are superior compared to the traditionally used leap-frog scheme, see Section 7.4.7.

7.4.4 Preservation of divergence

The finite volume solver is based on an integral formulation of Ampère's and Faraday's laws. However, the Maxwell equations also include the Gauss' laws that have to be satisfied in order to ensure a physical solution. For lossless materials, the divergence of the magnetic flux density vector, $\boldsymbol{B}$, as well as that of the electric flux density vector, $\boldsymbol{D}$, should equal zero.

Proposition 7.1. ***For the 2D TM$_z$ Maxwell equations,* $\nabla \cdot \boldsymbol{D} = 0$ *is automatically satisfied.***

Proof The proof is trivial because $\boldsymbol{E}$ only has a z component in the TM case and the last term is zero because we do not have any variation in the z-direction in 2D,

$$\nabla \cdot \boldsymbol{D} = \nabla \cdot \epsilon \boldsymbol{E} = \frac{\partial \, \epsilon E_x}{\partial x} + \frac{\partial \, \epsilon E_y}{\partial y} + \frac{\partial \, \epsilon E_z}{\partial z} = 0 \,.$$

$\square$

Proposition 7.2. ***The FV-solver preserves $\nabla \cdot \boldsymbol{B}$ on local cell level to machine precision.***

Proof For dual cell j we have

$$\frac{\partial}{\partial t}\int_{A_j^d} \nabla \cdot \boldsymbol{B}\, dA = \frac{\partial}{\partial t}\int_{A_j^d} \nabla \cdot \mu\, \boldsymbol{H}\, dA = \frac{\partial}{\partial t}\oint_{\Gamma_j^d} \mu\, \boldsymbol{H} \cdot \boldsymbol{n}_j^d\, dl\,,$$

where Gauss' theorem is used to get the last equality. Splitting the integral path into k segments gives us

$$\frac{\partial}{\partial t}\sum_k \int_{\Gamma_{j,k}^d} \mu\, \boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d\, dl\,.$$

So far we are still using continuous variables. If we think of the segments as being the dual edges and also assume that $\boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d$ and μ are constant along each edge we get

$$\frac{\partial}{\partial t}\sum_k \tilde{\mu}_{j,k}^d\, \boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d \int_{\Gamma_{j,k}^d} dl\; = \frac{\partial}{\partial t}\sum_k \tilde{\mu}_{j,k}^d\, \boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d\, \Delta_{j,k}^d\,,$$

where $\Delta_{j,k}^d$ is the edge length. The permeability is time independent and we can swap the sum and time derivative operator to obtain

$$\sum_k \tilde{\mu}_{j,k}^d\, \Delta_{j,k}^d\, \frac{\partial \boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d}{\partial t}\,.$$

Finally we can use (7.40) which gives us

$$\sum_k \left(E_z|_q - E_z|_i\right) = 0\,,$$

where the last expression equals zero because in the sum over k, each electric node value occurs twice with opposite signs.

Thus, the time derivative of the divergence is equal to zero and hence the divergence is preserved to machine precision in each cell. □

Note that if $\boldsymbol{H} \cdot \boldsymbol{n}_{j,k}^d$ had been computed as a projection of the arithmetic mean value of the two magnetic node values (see Figure 7.4), instead of being updated using (7.40), the divergence would in general have differed from zero.

We would also like to point out that the proof is independent of how $\tilde{\mu}_{j,k}^d$ is calculated. However, if we use (7.41) in (7.40) and (7.36) in (7.37) we obtain a second-order accurate FV solver.

7.4.5 Creating the dual grid

Perhaps the most demanding task in implementing the FV solver is the construction of the dual grid from the primary grid. This is done in the preprocessing phase by joining barycenters of the neighboring cells. Hence, the dual grid is uniquely determined by the primary grid.

Starting with the inner edges, i.e. edges not situated on a boundary, each inner edge in the primary grid is crossed by a dual edge, see Figure 7.6. These dual edges belong to the dual elements surrounding the start- and stop-nodes of the respective primary edges.

At a PEC boundary, as already mentioned, the electric field nodes are mirrored in the boundary. The dual edges crossing the boundary edges are assigned to the dual elements surrounding the start- and stop-nodes of the boundary edges. The dual edges lying inside the PEC object are constructed by joining the image nodes together and they are assigned to the dual elements that hold both the dual nodes of these edges.

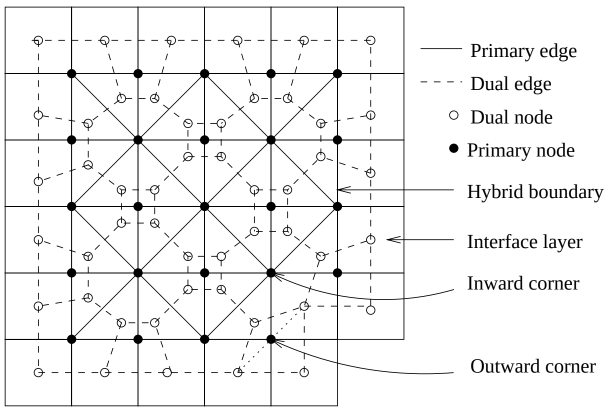


Figure 7.6. Construction of the dual grid from the primary grid.

Note that if we have a very thin PEC object it is possible that two dual elements lying on opposite sides of the boundary will overlap each other geometrically. This does not pose any problems because the image nodes are only virtual and the electric fields in these nodes are not updated, but assigned values from the field in their corresponding node.

The hybrid boundary is slightly more complicated. The nodes lying on a hybrid interface should be updated as usual, i.e. they have dual cells surrounding them, see Section 7.6. The dual edges that cross the edges lying on the interface boundary belong as usual to the dual elements surrounding the start- and stop-nodes of the boundary edges, see Figure 7.6. Generating the outermost dual edges in the interface layer is the trickiest part. Mainly because the interface is allowed to be staircased to minimize the number of unstructured cells. Hence, we cannot, as in the PEC case, create the outermost dual edges by simply joining the dual nodes lying in the interface layer. We have to take care of the convex corners, which are always four more than the concave corners. We know that we have an convex corner when the midpoint of the dual edge joining two consecutive dual nodes, moving counterclockwise in the interface layer, is the same point as the corner node in the primary grid; see the dotted line in bottom right corner of Figure 7.6. If that is the case we construct a new dual node and remove the dual edge going through the corner. Instead we generate the two dual edges going to and from the new node.

When all dual nodes and dual edges are constructed, the edges in each dual element are sorted such that they are traversed counterclockwise for each dual element. The area of a dual element is calculated by dividing it into triangles and summing the contributions from each triangle.

7.4.6 Workload and memory requirements

To implement the solver efficiently we have chosen to compute as much as possible initially. Hence, the update of the fields is accomplished using matrices acting on the respective vectors. Due to the fact that the matrices will be sparse but without structure we store them in compress sparse row format. This format is a very memory efficient format for sparse matrices and gives fast access to the matrix elements. After the matrices have been created, all grid variables can be written to disk. Hence, this approach is much more efficient than recomputing the expressions needed to update the field variables at every time step or using indirect addressing in several levels.

To be able to obtain a theoretical estimate of the memory requirements and number of arithmetic operations of the solver we have to make a few assumptions. First of all we neglect boundary conditions. The following is assumed about the grid:

- There are n triangles.
- There are three edges in each triangle and each edge is shared by two triangles. $\Rightarrow$ There are $1.5\,n$ edges.
- There are three nodes in each triangle and each node belongs to six triangles. $\Rightarrow$ There are $0.5\,n$ nodes and six dual edges per dual cell.

The field variables that we need to store are E_z, $\boldsymbol{H} \cdot \boldsymbol{n}^d$, $\boldsymbol{H}$ and $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$. Using ABS3, E_z and $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$ are stored at three time levels, whereas $\boldsymbol{H} \cdot \boldsymbol{n}^d$ and $\boldsymbol{H}$ only are stored at the latest time level.

Variable	Floating point numbers
E_z	$3 \cdot n$
$\boldsymbol{H} \cdot \boldsymbol{n}^d$	$1.5\,n$
$\boldsymbol{H}$	$2 \cdot 0.5\,n$
$\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$	$3 \cdot 1.5\,n$
$\sum$	$10\,n$

Table 7.1. Memory requirements to store the field variables.

The memory requirements for the matrix operators used to update the field variables can be calculated from (7.37)–(7.40).

Variable	Floating point numbers	Integers
E_z	$n \cdot 3 = 3\,n$	$3\,n + n = 4\,n$
$\boldsymbol{H} \cdot \boldsymbol{n}^d$	$1.5\,n \cdot 2 = 3\,n$	$3\,n + 1.5\,n = 4.5\,n$
$\boldsymbol{H}$	$n \cdot 6 = 6\,n$	$6\,n + n = 7\,n$
$\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$	$1.5\,n \cdot 5 = 7.5\,n$	$7.5\,n + 1.5\,n = 9\,n$
$\sum$	$19.5\,n$	$24.5\,n$

Table 7.2. Memory requirements to store the matrices.

Hence, using the results in Tables 7.1 and 7.2 the total memory requirements for the FV solver is 29.5 floating point numbers and 24.5 integers per cell. Using 64-bit precision for the floating point numbers and 32-bit precision for the integers this means that approximately 334 bytes per cell are needed.

The number of arithmetic operations used by the solver are calculated from (7.37)–(7.40) and (7.43)–(7.45).

Variable	Arithmetic operations
E_z	$n \cdot 11 = 11\,n$
$\boldsymbol{H} \cdot \boldsymbol{n}^d$	$1.5\,n \cdot 4 = 6\,n$
$\boldsymbol{H}$	$n \cdot 12 = 12\,n$
$\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$	$1.5\,n \cdot 14 = 21\,n$
$\tilde{E}_z$	$n \cdot 5 = 5\,n$
$\sum$	$55\,n$

Table 7.3. Number of arithmetic operations per cell.

Hence, from Table 7.3 we conclude that the FV solver performs approximately 55 arithmetic operations per cell for each time step. This should be compared with FD-TD, which needs eleven arithmetic operations per cell and 36 bytes of memory. Hence, the FV solver is a factor 5 more expensive in terms of arithmetic operation per cell and a factor of nine in terms of memory per cell compared to FD-TD.

Note that in the above calculations we have not used the fact that when the primary and dual edges are orthogonal the second part of (7.39) vanishes and the $\boldsymbol{H}$ values are not needed. A case where this will happen is the equilateral grid. An implementation of the solver should of course take advantage of this and the memory requirements and workload will go down considerably. On an equilateral grid, or any other grid where orthogonality occurs, the memory requirements are 16.5 floating point numbers and 10 integers per cell. The total memory requirement is in this case 172 bytes using the same precision. The number of arithmetic operations goes down to 31 per cell for such a grid. Thus, it is possible to reduce the memory requirements and the number of arithmetic operations by approximately a factor of two for the special case when the grids are mutually orthogonal.

7.4.7 Stability analysis

We will now investigate the stability properties of our FV scheme. The stability region for ABS3 for the scalar test equation $u' = \lambda u$ is given in Figure 7.7. The

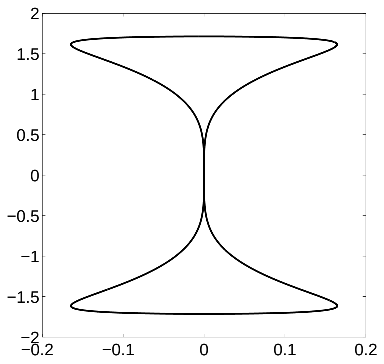


Figure 7.7. Stability region for ABS3.

scheme is stable for λ between $\pm 12/7$ along the imaginary axis compared to the leap-frog scheme, which is stable between ± 2. That implies that we have to use a shorter time step for ABS3. However, the main disadvantage with the leap-frog scheme is that it is only stable on the imaginary axis and becomes unstable as soon as we have eigenvalues with a nonzero real part, which we are likely to have on unstructured grids and when boundaries are taken into account.

By introducing operators $\mathcal{A}$ and $\mathcal{B}$ that take care of the spatial discretization we are able to write the semi-discrete problem on matrix form as

$$\frac{\partial}{\partial t}\begin{pmatrix}\hat{\boldsymbol{H}} \\ E_z\end{pmatrix} = \begin{pmatrix}\boldsymbol{0} & \mathcal{A} \\ \mathcal{B} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\hat{\boldsymbol{H}} \\ E_z\end{pmatrix}, \tag{7.46}$$

where $\hat{\boldsymbol{H}} = \begin{pmatrix}\boldsymbol{H} & \boldsymbol{H}\cdot\boldsymbol{n}^d\end{pmatrix}$.

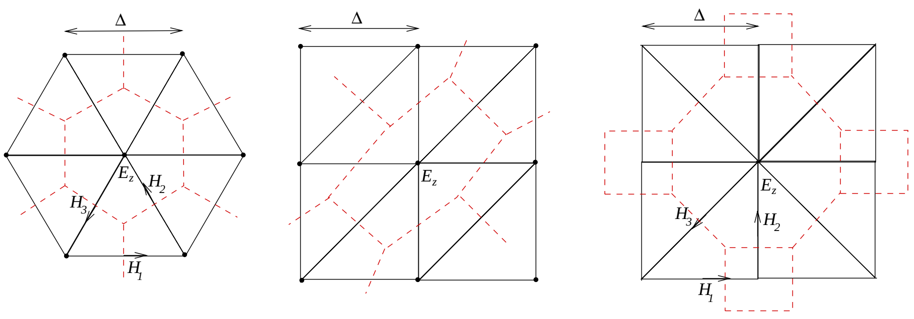


Figure 7.8. Three different uniform triangular grids, the equilateral grid (left), the one-directional grid (middle) and the diamond grid (right). The respective dual grids are indicated by dashed lines.

A detailed Fourier analysis of the stability and dispersion properties on the three types of unstructured grids in Figure 7.8 is given in the Licentiate thesis of Fredrik Edelvik [Ede00]. This analysis shows that the stability condition for leap-frog timestepping is given by

$$c\Delta t_{LF} \leq \frac{\Delta}{\sqrt{2}} \cdot \frac{\Delta_{min}}{\Delta}\,, \tag{7.47}$$

where the first part in the right-hand side is the stability condition on Cartesian grids and Δ_{min} equals the shortest edge length in the primary and dual grids. The stability condition for ABS3 timestepping is given by

$$c\Delta t_{ABS3} \leq \frac{6}{7}\frac{\Delta}{\sqrt{2}} \cdot \frac{\Delta_{min}}{\Delta}\,. \tag{7.48}$$

To analyze the eigenvalues for a general unstructured grid including its boundaries we can no longer use Fourier analysis. Instead, let

$$\boldsymbol{z}^n = \begin{pmatrix} E_z|^n & \hat{\boldsymbol{H}}^{n-\frac{1}{2}} & E_z|^{n-1} & \hat{\boldsymbol{H}}^{n-\frac{3}{2}} & E_z|^{n-2} \end{pmatrix}^T\,, \tag{7.49}$$

and after some straightforward rearrangements we are able to write (7.43)–(7.45) on matrix form as $\boldsymbol{z}^{n+1} = \mathcal{P}(\mathcal{A},\mathcal{B})\boldsymbol{z}^n$, where

$$\mathcal{P} = \begin{pmatrix} \mathcal{I} + \frac{625}{576}\Delta t^2\mathcal{BA} & \frac{23}{24}\Delta t\mathcal{B} & -\frac{25}{288}\Delta t^2\mathcal{BA} & \frac{1}{24}\Delta t\mathcal{B} & \frac{25}{576}\Delta t^2\mathcal{BA} \\ \frac{25}{24}\Delta t\mathcal{A} & \mathcal{I} & -\frac{1}{12}\Delta t\mathcal{A} & \mathbf{0} & \frac{1}{24}\Delta t\mathcal{A} \\ \mathcal{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{I} & \mathbf{0} & \mathbf{0} \end{pmatrix}\,, \tag{7.50}$$

where $\mathcal{A}$ and $\mathcal{B}$ are defined as in (7.46). Analyzing the eigenvalues of the companion matrix $\mathcal{P}$, for the grid shown in Figure 7.9, reveals that if we choose the time step for ABS3 in the same manner as above we get the eigenvalue spectrum shown in Figure 7.9, where all eigenvalues are within the unit circle. If we use the leap-frog scheme with the same time step, the largest eigenvalue is of the order 1.0003. Hence the leap-frog scheme is unstable even for a time step well within the stability limit along the imaginary axis. The eigenvalues close to the origin in Figure 7.9 are the "parasitic" roots of the ABS3 multi-step scheme. However, since these roots are all close or equal to zero they are quickly damped away.

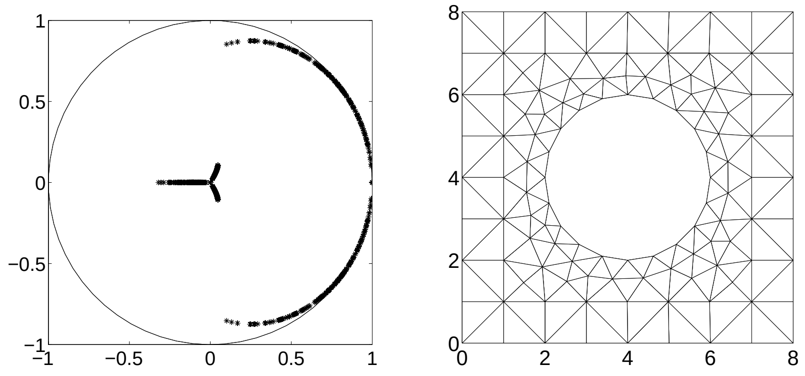


Figure 7.9. Eigenvalues of the companion matrix P using ABS3 for a PMC cylinder scattering case. The primary grid around the cylinder is shown to the right.

The values of the elements of $\mathcal{P}$ are taken directly from the code. Hence, we do not need analytical definitions of these elements to perform the eigenvalue analysis.

7.5 Grid requirements

The cell size of the structured Cartesian grid is chosen by balancing the computational cost of using smaller cells and the inability to capture the physics by using too large cells. When it comes to unstructured grids the cell size does not have to be constant throughout the whole grid. Smaller cells are used where fine geometrical details need to be resolved and larger cells are used as much as possible to save computer resources. If the cell size in the Cartesian grid is based on resolving characteristic wave lengths, then the unstructured cells must not be larger than the structured cells.

The local wave propagation speed for waves propagating through a discrete grid depends on the local grid size. If the grid size changes abruptly this will inevitably give spurious reflections. Even if the cell size is changed in a smooth way there will be some reflections although they can be made small. A few fundamental observations about unstructured grids can be made:

1. The rate of change in grid size must be controlled, see Figure 7.10a.
2. The cells should be as close to equilateral as possible in homogeneous domains, see Figure 7.10b.
3. At material interfaces the cell size could be changed abruptly by a factor of $\sqrt{\epsilon_1/\epsilon_2}$ (for a dielectric transition). This is due to the change in wave impedance for the physical problem, see Figure 7.10c.

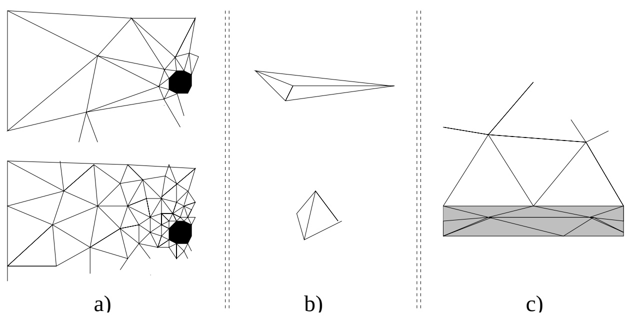


Figure 7.10. a) Cell size changes must be reasonable. b) Equilateral cells are preferable. c) Cell sizes are only allowed to change abruptly at material interfaces.

Some comments about the three points above are necessary:

1. One would of course like to have a very smooth transition from large to small cells and vice versa, but this means that many unstructured cells are required. For performance reasons we would like to have as few unstructured cells as possible and hence a well balanced compromise must be found between these two contradictory desires. Moreover, a too rapid transition in the FV case leads to dual cells that are ill-balanced, i.e. a primary node will be surrounded by a dual cell whose center is far from the primary node.
2. The reasons for the desire for equilateral triangles is that skewed triangles augment the local numerical error. This is expressed as a stronger anisotropy of the local wave propagation speed.
3. Abrupt change of the cell size at material interfaces is possible due to the fact that the number of cells per wave length is conserved in the normal direction. This must of course be balanced against the second observation above. However, in the tangential direction there is not much one can do since the nodes on the material interface are shared between the cells on both sides.

The unstructured grid should be as free as possible from global anisotropy, which means that the orientation of the individual cells should be as random as possible or counteract each others' directivity as much as possible. This desire comes from the fact that the local numerical wave propagation speed is not constant for all angles of propagation. See Figure 7.11 for illustrations.

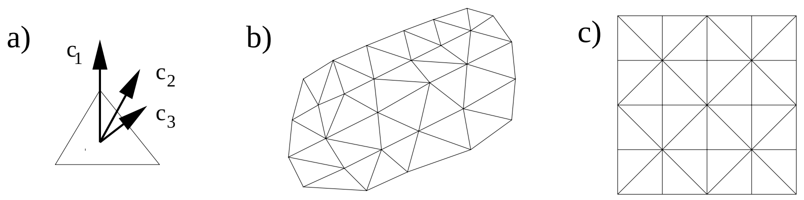


Figure 7.11. a) The local wave propagation speed c depends on the propagation direction ($c_1 \neq c_2 \neq c_3$). b) Global directivity might amplify global anisotropy. c) Global anisotropy might be reduced by locally counteracting anisotropies.

Coupling unstructured grids to structured Cartesian grids is described in Section 7.6. In this case the unstructured grids must have a transition layer of split rectangles which coincide with the Cartesian cells. An example of a transition layer is shown in the left part of the unstructured grid in Figure 7.12.

7.6 Hybridization

It is very important that an electromagnetic solver for realistic industrial applications can handle complex geometries without compromising the efficiency more than necessary. We believe that the best way to achieve this is to combine structured grids with unstructured grids thereby getting a hybrid method.

Our strategy of using hybrid techniques between unstructured grids and structured grids is based on the following observations:

- The FD-TD method is an extremely efficient second-order method for homogeneous materials, with respect to time and memory consumption.
- The Cartesian grid handles general geometries poorly, and fine details cannot be taken into account without special tricks.
- An unstructured grid can be fitted to general geometries and small details can be resolved.
- Unstructured grids lead to more elaborate memory accessing, and the number of operations per cell or wave length is higher than in the classical FD-TD method.

Coupling of structured grids and unstructured grids strives to utilize their advantages without suffering too much from their disadvantages. For unstructured grids we distinguish between two cases:

1. The cell sizes in the unstructured domain are of the same size as in the Cartesian grid. Typically this is the case when moderately curved boundaries are meshed.
2. The cell sizes in the unstructured domain vary from the same size as in the Cartesian grid to very small cells. The small cells are used close to fine geometrical details in order to capture the physical influence of these details on the global solution.

We distinguish between two types of time integrators: explicit and implicit time integrators. The principal difference is:

$$\mathbf{u}^{n+1} = \mathbf{B}u^n + \mathbf{f} \quad \text{(explicit)}, \tag{7.51}$$

$$\mathbf{K}u^{n+1} = \mathbf{B}u^n + \mathbf{f} \quad \text{(implicit)}, \tag{7.52}$$

where $\mathbf{u}^n$ is the vector of discrete unknowns at time level n, $\mathbf{u}^{n+1}$ is the vector of discrete unknowns at time level $n+1$, $\mathbf{B}$ and $\mathbf{K}$ are (non-diagonal) matrices (for finite elements a combination of the stiffness- and mass-matrix) and $\mathbf{f}$ is a source term vector.

For explicit timestepping algorithms the time step is limited by the smallest cell size. This is generally not the case for implicit timestepping algorithms, where the time step usually can be chosen arbitrarily regarding stability. On the other hand, explicitly marching on in time is trivial since this is only a straightforward matrix-vector multiplication whereas implicitly marching on in time gives a system of linear equations to solve for each time step. However, the mass matrix in (7.52) is time independent and hence an LU factorization can be done in the initialization phase. But implicit algorithms cannot, in general, be used for large unstructured grids since the computational cost of doing an LU factorization does not scale linearly with the number of cells. An option to circumvent this is to us iterative methods to solve (7.52).

For large unstructured grids where some cells are small, explicit and implicit solvers will be very computer demanding if used separately. In that case one could try to couple an explicit solver (used for the larger cells) with an implicit solver (used for the smaller cells). This is outlined in Section 7.6.3.

There is an appropriate note to make here. Using implicit timestepping means that the time step does not have to be in accordance with the space step. Hence the high spatial frequencies that can be (locally) supported by regions of small cells cannot be propagated by the timestepping mechanism because larger time steps only support temporal frequencies up to a limit of

$$max(f_{\text{supp}}) = \frac{1}{2 * \Delta t} \ll \frac{c}{2 * \Delta},$$

where Δ is the smallest cell size. A justified question is then why bother to resolve small details? The answer is that small geometric details do have an impact on the lower part of the (temporal) frequency spectrum part of the solution. This is easily realized by thinking of a thin infinite large PEC plate with a small hole. Even though the hole might be smaller than a wave length, a wave impinging the plate on one side will result in fields excited into the other side.

7.6.1 FD-FE

Coupling between the FD-TD solver for structured grids and the implicit FE solver for unstructured grids (described in Section 7.3) is shown in Figure 7.12 and Figure 7.13.

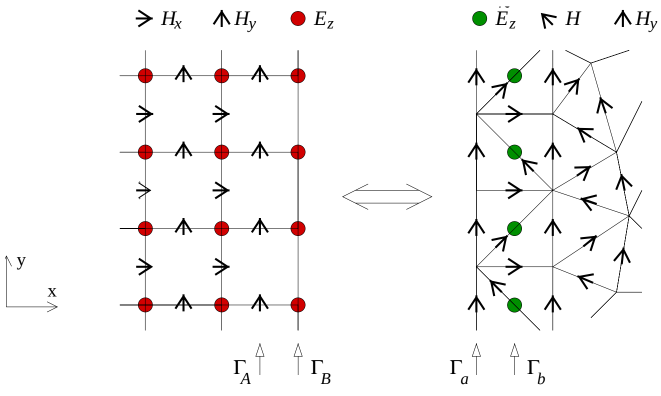


Figure 7.12. Coupling between structured FD-TD and unstructured FE. The locations of the H_y components on Γ_A and Γ_a coincide. The locations of the E_z components on Γ_B and Γ_b also coincide.

1. The H_x and H_y components in the structured grid (to the left in Figure 7.12) are updated using the standard FD-TD method once E_z on Γ_B is given (Γ_B is the FD-TD boundary at B). When H_x and H_y are calculated the H_y components along Γ_A are sent to Γ_a in the FE grid.

2. The E_z components in the structured domain are updated using the standard FD-TD method.

3. The FE solver calculates the magnetic field implicitly in the unstructured domain using the values at Γ_a as boundary conditions.

4. By utilizing a discrete ∇-operation around the auxiliary $\tilde{E}_z$ variables, $\tilde{E}_z$ can be time stepped according to $\tilde{E}_z^{n+1} = \tilde{E}_z^n + \frac{\Delta t}{\epsilon}\nabla \times \mathbf{H}$ at Γ_b.

With the grid we use at the interface the updating of $\tilde{E}_z$ becomes identical to the standard FD-TD method and when $\tilde{E}_z$ is known on Γ_b these variables are copied to Γ_B for the next time step.

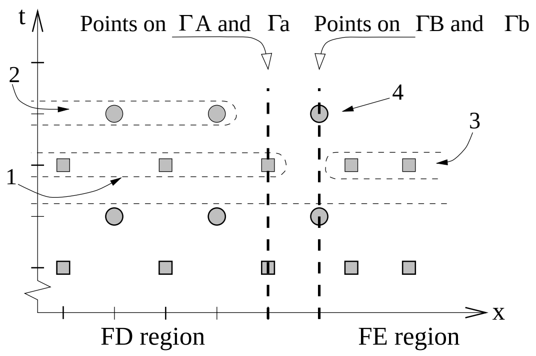


Figure 7.13. The timestepping mechanism for the FD-FE hybrid. Compare with Figure 7.1.

7.6.2 FD-FV

Coupling between the FD-TD solver for structured grids and the FV solver for unstructured grids (described in Section 7.4) is very similar to the procedure described in the previous section. However, one main difference is important to point out. The FD-TD solver sends E_z components to the FV solver and receives magnetic components from the FV solver, see Figure 7.14.

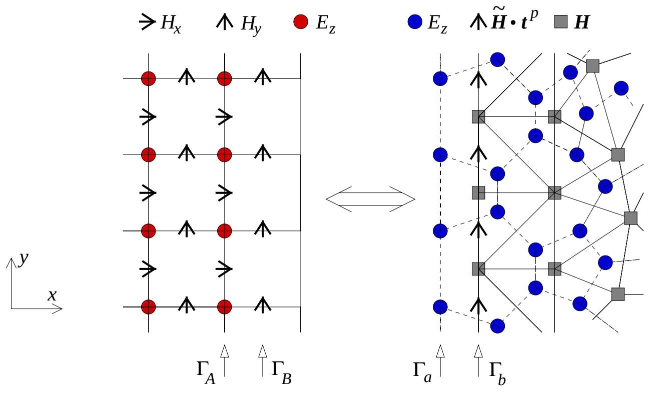


Figure 7.14. Coupling between structured FD-TD and unstructured FV. The locations of the E_z components on Γ_A and Γ_a coincide. The locations of the H_y components on Γ_B and Γ_b also coincide. (On Γ_b, the H_y components are denoted $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$ to signify that they are edge values.)

In Figure 7.14 only the magnetic edge values along Γ_b are drawn (denoted $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$). They are collocated with the H_y components along Γ_B in the FD-TD region and hence a straightforward sending of magnetic components from the FV domain at Γ_b to Γ_B in the FD-TD domain can be performed.

7.6.3 FE-FV

A coupling between the two unstructured solvers is rendered possible because of the differences in sending to and receiving from the two unstructured solvers when coupled with the structured FD-TD solver. The FE solver sends E_z components to the FV solver and receives magnetic components from the FV solver, see Figure 7.15.

In this case one can use the explicit solver for the larger cells and the implicit solver where smaller cells otherwise would limit the time step.

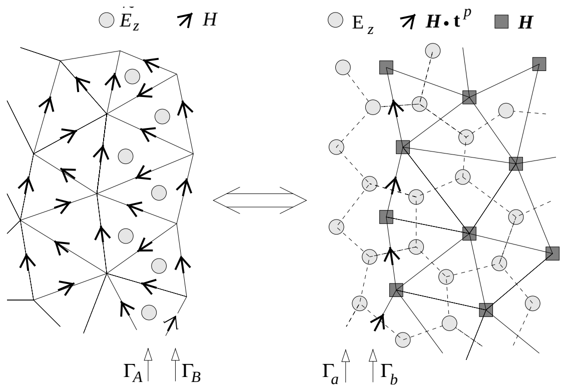


Figure 7.15. Coupling between unstructured FE and unstructured FV. The locations of the E_z components on Γ_A and Γ_a coincide. The locations of the $\boldsymbol{H}$ components on Γ_B and Γ_b also coincide. (On Γ_b, the $\boldsymbol{H}$ components are denoted $\tilde{\boldsymbol{H}} \cdot \boldsymbol{t}^p$ to signify that they are edge values.)

Our hybrid code is currently limited to coupling between FD-TD and one of the two unstructured solvers which means that the coupling of the two unstructured solvers has not been verified experimentally. There are no reasons to believe that there would be any accuracy problems but one cannot say anything about stability without numerical experiments.

7.7 Stability

A critical aspect of every numerical method is stability All three schemes described in Sections 7.2–7.4 are stable as long as the CFL limits in (4.12) and (7.48) are not violated. However, this does not guarantee stability for the hybrid schemes since coupling of two stable schemes might result in an unstable scheme.

The complexity of the hybrid scheme makes it very difficult and cumbersome to perform a theoretical stability analysis. Instead we perform a numerical study. The entire hybrid scheme can be written as

$$\boldsymbol{u}^{n+1} = \boldsymbol{A}\boldsymbol{u}^n + \boldsymbol{f}^n \,, \tag{7.53}$$

where $\boldsymbol{f}^n$ represent the source terms. The vector $\boldsymbol{u}$ contains the unknown values of both $\boldsymbol{E}$ and $\boldsymbol{H}$ fields and includes unknowns from several time levels from both the structured domain and the unstructured domain. Furthermore $\boldsymbol{u}^0$ is given by initial values. Instability in the sense of exponential growth is generated if any eigenvalue of A is larger than one. One possible way to find the eigenvalues of $\boldsymbol{A}$ would be to use the technique employed for the FV solver in Section 7.4.7. However, it would be very cumbersome to find the explicit values for all the elements of $\boldsymbol{A}$ in the hybrid case. An alternative technique is to examine the dominant eigenvalue of $\boldsymbol{A}$ by running the code for a very large number of time steps. We have utilized this procedure which in numerical linear algebra is known as the power method.

7.7.1 Details

We performed stability tests for several hybrid grids, including those three grids used in the convergence study for the circular cylinder, see Section 7.8. In all cases we tested both PEC and PMC boundary conditions.

We ran the code for ten million time steps. The time to complete these computations on an IBM Power 3 processor ranged from a few hours up to 24 hours depending on the size of the grid. As ABC we used a twelve cell thick U-PML layer with the profile given in (7.1). A plane wave was continuously fed into the computational domain using Huygens' surfaces. The actual shape of the pulse is not so important. It is crucial that all frequencies supported by the grid are present in the calculation, but roundoff errors will introduce them even if they are not present initially. A clever choice of excitation may lead to a quicker discovery of instability, but instability will finally show up no matter what pulse is used.

7.7.2 FD-FE

We did encounter stability problems for the FD-FE hybrid for some grids when the Crank-Nicholson scheme was used for time discretization. The grids used in Section 7.8 were all stable, with one exception. The finest grid was unstable when PMC BC was used. Furthermore, another grid, which is very similar to the coarse grid used for the circular cylinder, showed instabilities for both PEC and PMC.

Stability problems with FD-FE hybrids are not unique to our approach. It has been noted by other researchers in this area. A remedy has been suggested by Hwang and Wu [HW99]. They used a temporal filtering technique to stabilize their scheme. However, this approach reduces the order of convergence to one.

Our technique to stabilize the unstable cases is to increase the value of θ in the timestepping scheme, see (7.29). Thus, we increase the stability region and introduce dissipation for purely imaginary eigenvalues of the spatial discretization matrix. However, $\theta = 0.5$ is the only value that gives a second-order time integration scheme. Table 7.4 summarizes our results for the three grids that were unstable using the θ-method with $\theta = 0.5$, i.e. Crank-Nicholson.

θ	coarse grid, PEC	coarse grid, PMC	fine grid, PMC
0.50001	Unstable	Unstable	Unstable
0.5001	Unstable	Stable	Stable
0.501	Stable	Stable	Stable

Table 7.4. Result of stability tests for the FD-FE hybrid.

For all three grids it was enough to increase θ to 0.501. This change in θ is so small that the impact on accuracy is negligible even though it formally reduces the order of convergence (in time) to one.

We have not encountered any instabilities when using the BDF-2 timestepping scheme. The drawback of this scheme is that it has a time integration error approximately four times larger than the Crank-Nicholson scheme.

7.7.3 FD-FV

All grids tested were stable for the FD-FV hybrid as long as the time step was selected properly. The time step we used was a factor of $\sqrt{2}/4$ smaller than the stability limit for the FD-TD scheme, i.e. we chose

$$\Delta t = \frac{\sqrt{2}}{4c_0\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}}}. \tag{7.54}$$

Using a time step twice as large proved to be unstable. This is not surprising since it violates the stability condition in (7.48). The grids used in these stability tests all have a shortest edge length of slightly less than half the cell size of the structured grid.

7.8 Convergence

The order of convergence is at least two in both time and space for all three schemes described in Sections 7.2–7.4. The hybridization techniques presented in Section 7.6 are designed to preserve this property. We have validated this by doing calculations on five different cases:

- vacuum,
- a circular PMC cylinder,
- a circular PEC cylinder,
- a circular cylinder with $\epsilon_r = 4$, and
- a circular cylinder with $\mu_r = 4$.

Note that the duality of the Maxwell equations means that PMC boundary condition for the TM mode is equivalent to PEC boundary condition for the TE mode. Hence a PMC cylinder is an interesting test case even though there are no PMC materials in the real world. We compare with FD-TD solutions which use staircase approximations of the circular cylinder. We demonstrate that when the circular cylinder is present, the FD-TD method does not show second-order convergence but the hybrid methods do. This holds for all four test cases with a circular cylinder. In all cases, we have used Huygens' surfaces to create a plane wave, with a Gaussian shape given by

$$f(t) = e^{-(t-t_0)^2/t_w^2}, \tag{7.55}$$

with $t_0 = 20/c_0$ and $t_w = t_0/6$. The angle of incidence was given by $\boldsymbol{k} = (1, 0)$. As ABC we used a twelve cell thick U-PML layer with the profile given in (7.1).

7.8.1 Modeling of circular cylinders in FD-TD

As explained in Section 4.6, we have to model the circular cylinder using a staircased approximation when we use the FD-TD method. The PEC and PMC boundary conditions are implemented by zeroing components on the boundary of the staircase representation of the circular cylinder. However, we still have to choose an algorithm for finding this representation of the circular cylinder.

In the PEC case, a simple way would be to zero all E_z components that are located inside the circular cylinder. However, this would lead to a staircase representation that always has an area smaller than the true area of the circular cylinder. Instead, we have chosen to zero all E_z components that are located inside a virtual radius of the true radius plus half the cell size. (We use the same resolution in both spatial dimensions.) Numerical tests have confirmed our belief that this procedure gives better results.

PMC boundary condition is implemented by zeroing the four surrounding $\boldsymbol{H}$ components of every E_z component that are located inside the circular cylinder, see Figure 4.1.

A computational cell is considered to be a part of the cylinder if its center lies inside the cylinder. An E_z component is located at the corners of four cells, see Figure 4.1. The ϵ value needed for the update of this E_z component is taken as the arithmetic mean value of the ϵ values in these four cells. The H_x and H_y components are located on the sides of the cells. The μ values needed for the update of these components are taken as the harmonic mean value of μ in the two cells sharing this side. The reason for using a harmonic mean value rather than an arithmetic is the fact that these components are normal to the interface. This issue will be discussed in detail in Chapter 9.

7.8.2 The coarse grids

Figure 7.16 displays the unstructured grids for a circular cylinder with a radius of two meters. These 8×8 meter grids were created using `Femlab` [Fem] and inserted

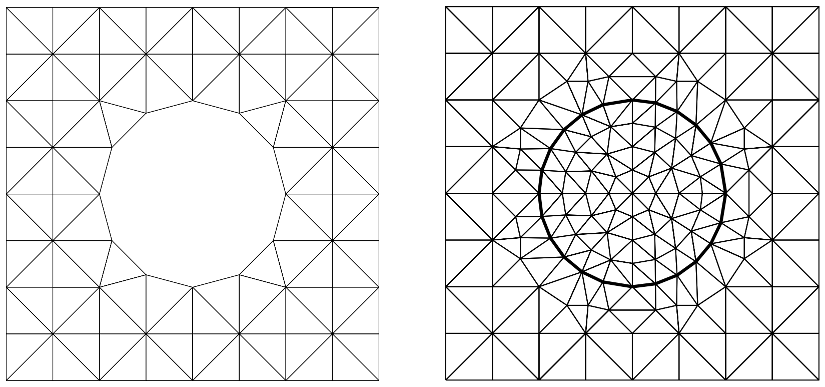

(a) PEC/PMC (b) Material

Figure 7.16. The coarsest unstructured grids.

in the center of a structured FD-TD grid with 40×40 cells of the size $\Delta = 1\,\mathrm{m}$ so that the center of the cylinder was located at (x,y)=(19.5, 20.5). The coarsest unstructured grid for the vacuum calculations consists of eight by eight squares that have been split into two triangles each.

In all FD-TD calculations we have used $CFL = 1/\sqrt{2}$ for the circular cylinder. The same time step was used for the FD-FE hybrid, while the time step for the FD-FV hybrid was a factor of two smaller for the PEC and PMC cylinders and a factor of four smaller for the material cylinders. For $\Delta x = \Delta y = \Delta = 1\,\mathrm{m}$ this gives $\Delta t \approx 1.67\,\mathrm{ns}$, $\Delta t \approx 0.83\,\mathrm{ns}$ and $\Delta t \approx 0.42\,\mathrm{ns}$. The FD-FV time step has not been chosen as the largest possible value. It has deliberately been chosen so that the FD-FE time step is a multiple of the FD-FV time step. This allows us to avoid temporal interpolation when probing the solutions. In the vacuum case all schemes have used $CFL = 1/\sqrt{8}$. We have used Crank-Nicholson for the FE timestepping.

The error has been measured in those E_z components in the FD-TD grid that lie closest to the unstructured grid. There are 36 such components, namely the outer components in the set E_z(17:24,18:25). By choosing to measure the error in the FD-TD grid we avoid spatial interpolation since all refined grids will have E_z components in these locations. The error is defined as the difference between the computational solution and a reference solution. The reference solutions for the circular cylinders have been obtained by highly resolved FD-TD calculations. We used $N_x = N_y = 10240$ for the PEC and PMC cylinders and $N_x = N_y = 5120$ for the material cylinders. The errors in these reference solutions have been estimated by comparing them with solutions where the problem size was a factor of two smaller in each dimension. The results are given in Table 7.5.

	Error estimate
Vacuum	0
PEC	$1.3 \cdot 10^{-4}$
PMC	$1.6 \cdot 10^{-4}$
$\epsilon_r = 4$	$2.8 \cdot 10^{-5}$
$\mu_r = 4$	$3.8 \cdot 10^{-4}$

Table 7.5. Estimates of the maximum errors in the reference solutions.

For vacuum, we have used the analytical solution as the reference solution.

The Huygens' surfaces were placed two meters from the outer boundary. In this case it means that they were placed in the third cell. When we refine the grid we will keep the physical location of the Huygens' surfaces at two meters from the boundary.

7.8.3 Grid refinement

The unstructured grids in Figure 7.16 consist of two parts: the interior 6×6 meter part and a one cell thick transition layer. When we refine the grids, the interior part of the unstructured grids are refined by splitting all triangles into four triangles. This is done in such a way that all edges are split into two edges. This was done using a slight adjustment in the grid refinement procedure of `Femlab` [Fem]. When splitting one right angle triangle into four new triangles, `Femlab` splits the triangles as in the left part of Figure 7.17. As mentioned in Section 7.5 we prefer the methodology illustrated in the right part of Figure 7.17.

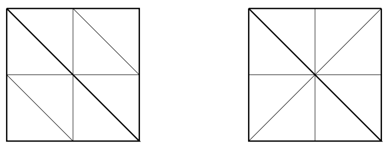


Figure 7.17. Two different grid refinement methodologies.

After the refinement of the interior part, we add a one cell thick transition layer outside the interior part. This means that the area of the unstructured grid shrinks with the refinement, see Table 7.6.

The nature of the hybrid scheme and the refinement procedure described above make it necessary to move the circular cylinder when a grid is refined with a factor of two. The lower left corner of the interior part of the unstructured grid must coincide with the center of a cell in the structured grid, i.e. a location where there is no field component. When we refine the grid with a factor of two, a new E_z component will appear in the center of the coarse grid cell and hence, we may not have the lower left corner of the interior part there. Hence, we have to move the unstructured grid. The location of the center of the circular cylinder, (xcyl, ycyl), is specified in Table 7.6. A separate reference solution was calculated for each location of the circular cylinder. The relocation of the cylinder could have been avoided by using a refinement factor of three, but we considered this to be a too rapid refinement.

Grid	Coarse	Medium	Fine
$N_x = N_y$	40	80	160
Δ (m)	1	0.5	0.25
side of unstr. domain (m)	8	7	6.5
PEC, area (m^2) (FD-TD)	9	11.25	11.56
PMC, area (m^2) (FD-TD)	12	13	13
xcyl (m)	19.5	19.25	19.125
ycyl (m)	20.5	20.25	20.125
number of triangles	100	280	904
number of edges	172	460	1432
shortest dual edge	0.4714	0.2357	0.1179
Bandwidth (FE)	14	19	36

Table 7.6. Parameters for the circular cylinder test case. (PEC and PMC)

Grid refinement of the semi-structured grid used for the vacuum test case was performed in a similar manner. However, in this case we need only one reference solution since there is no object in the unstructured grid.

7.8.4 Results

The results for the convergence study are shown in Figures 7.18–7.22. The plots display the mean value of the absolute error in the 36 measure points as a function of time. The peak of the Gaussian pulse passes the circular cylinder at $t \approx 125$ ns.

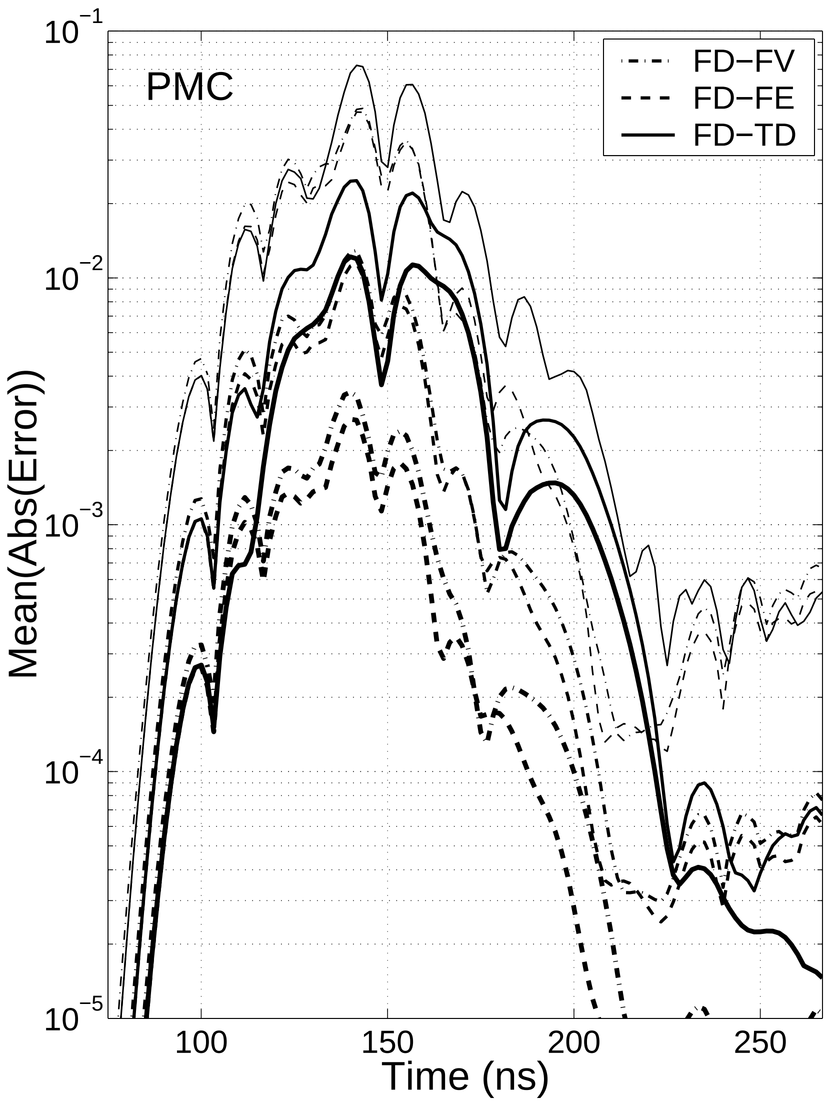


Figure 7.18. Errors for three refinement levels and three methods for the circular PMC cylinder.

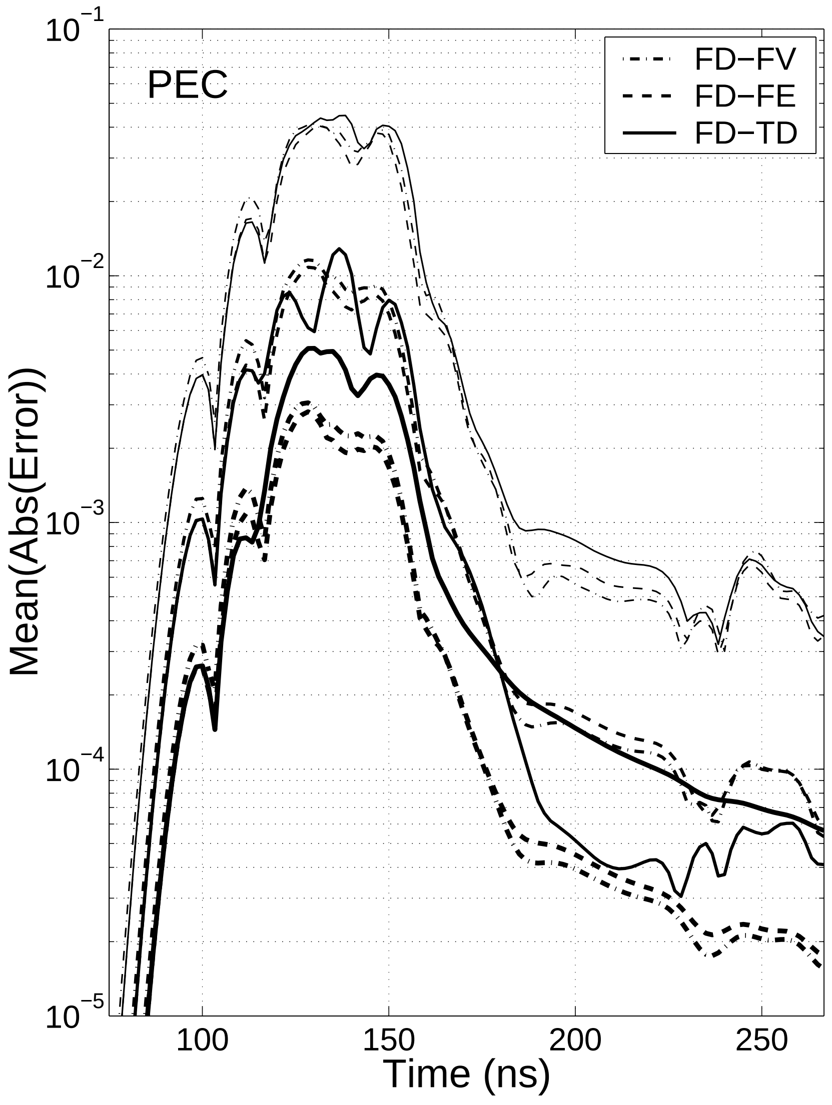


Figure 7.19. Errors for three refinement levels and three methods for the circular PEC cylinder.

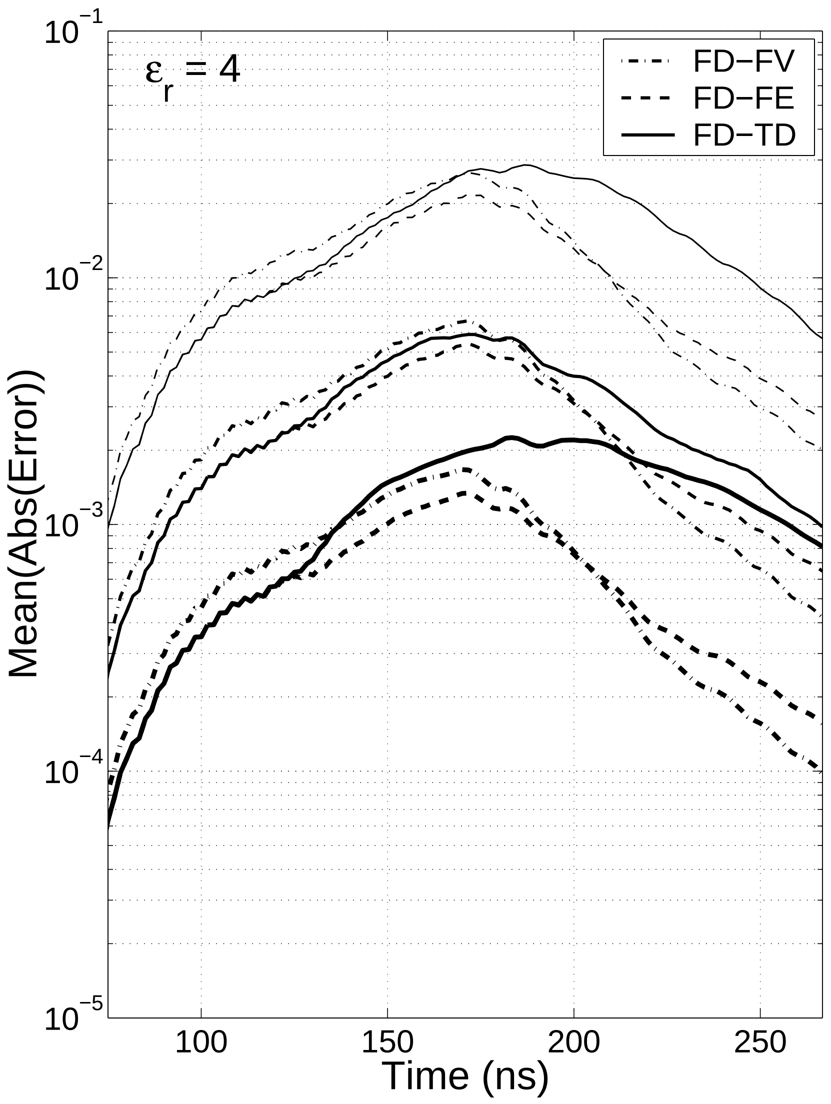


Figure 7.20. Errors for three refinement levels and three methods for the circular cylinder with $\epsilon_r = 4$.

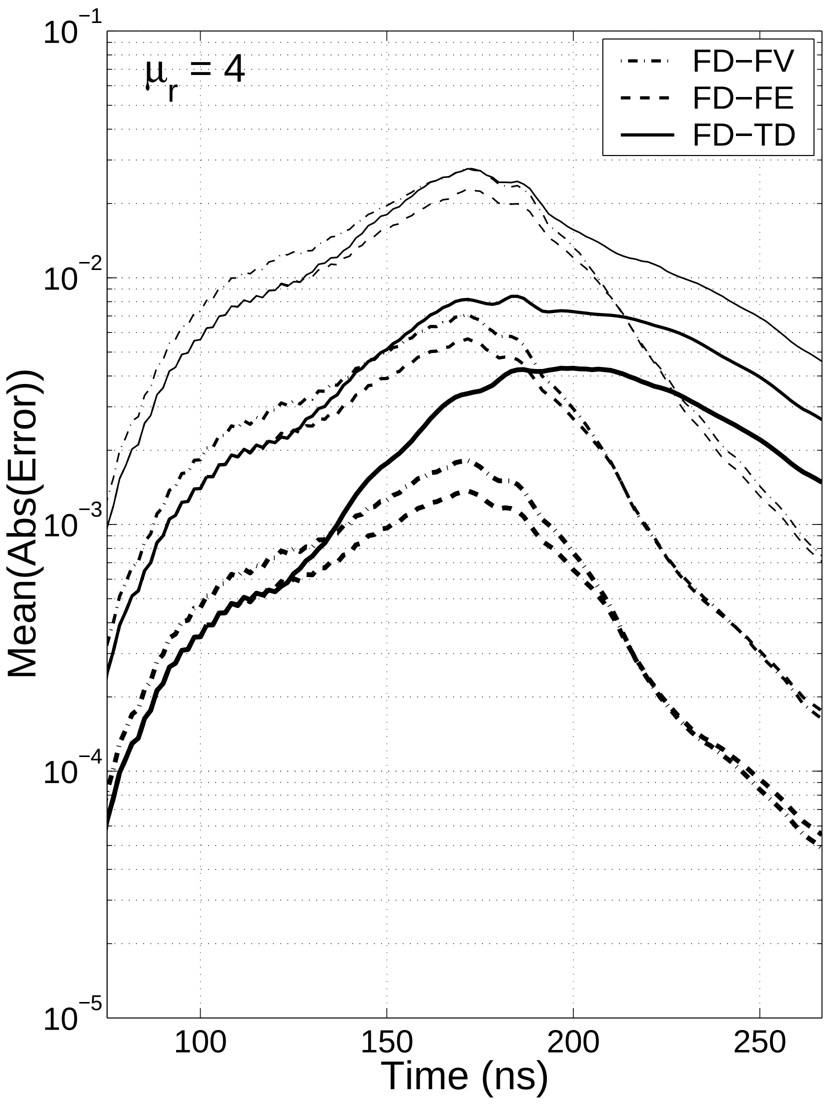


Figure 7.21. Errors for three refinement levels and three methods for the circular cylinder with $\mu_r = 4$.

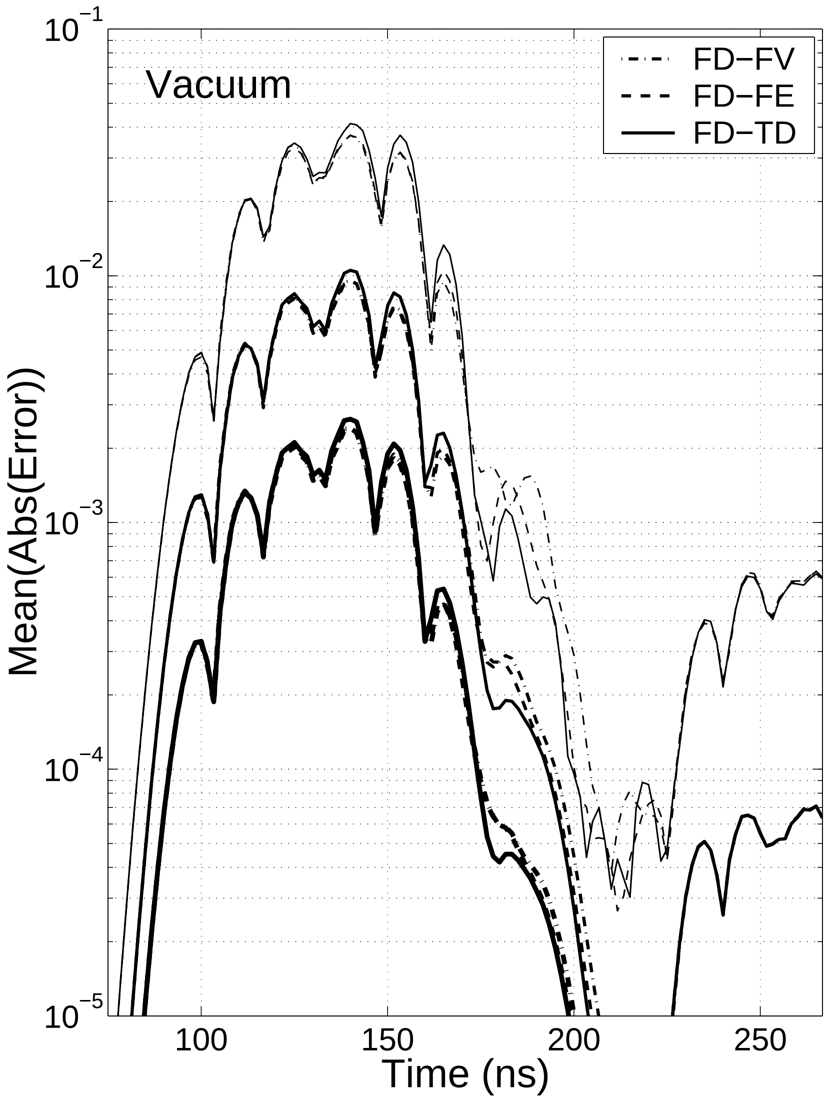


Figure 7.22. Errors for three refinement levels and three methods for vacuum.

We note that in the beginning, the error is larger for the FD-FV hybrid. This is caused by the shorter time step which gives a larger error in the FD-TD scheme. This effect is not present in Figure 7.22 because we have used the same time step for all methods in this particular test case.

For PEC and PMC, we note that around 110–115 ns, the error in the FD-TD solution is slightly smaller than the errors in the hybrid solutions. This is most likely due to reflections in the transition layer between the structured grid and the unstructured grid. However, once the errors in the geometrical representation of the circular cylinder affects the solution, we clearly see that the hybrid schemes are superior.

The increase in error that can be seen in Figure 7.22 for $t > 220$ ns is caused by the Huygens' surfaces. When the Gaussian pulse reaches the upper Huygens' surface a Gaussian with opposite sign is generated which is intended to zero out the approaching Gaussian and thus giving a zero scattered field. Due to the dispersion error there is a mismatch leading to a nonzero scattered field. This error is however absorbed by the ABC. The error mentioned above is an error component generated at the upper Huygens' surface and traveling in the opposite direction. This effect is present also in the other four cases, but is harder to spot since errors in the scattered field are also present.

The order of convergence has been estimated by calculating the L_1 norm of the errors in Figures 7.18–7.22. The result is given in Table 7.7. In all cases we have second-order convergence for the hybrid schemes. The FD-TD scheme however, only has second-order convergence for vacuum. Note that we have used a numerical reference solution for the circular cylinder cases. This affects the convergence estimates in Table 7.7. The last digit could be slightly altered if we use a better reference solution. However, this does not alter the main conclusion that we have second-order convergence for the hybrid schemes.

We would also like to point out that the results in Figures 7.18–7.22 prove that there are no large reflections at the grid interfaces. Small reflections are unavoidable, but as long as these reflection errors do not dominate the other error sources, they are acceptable.

	FD-TD		FD-FE		FD-FV	
	coar	fine	coar	fine	coar	fine
Vacuum	2.08	2.02	2.08	2.02	2.06	2.02
PEC	2.23	1.06	2.01	1.98	1.99	1.98
PMC	1.33	0.97	2.13	2.08	2.01	1.91
$\epsilon_r = 4$	1.50	1.10	2.03	2.00	2.03	2.00
$\mu_r = 4$	1.54	1.16	2.06	2.01	2.04	1.97

Table 7.7. Estimates of the order of convergence for the three methods. The values in the columns labeled coar have been obtained by comparing errors on the coarse and medium grids and the values in the columns labeled fine have been obtained by comparing errors on the medium and fine grids.

7.9 PMC wall

A classical paper demonstrating the errors caused by staircasing is that by Cangellaris and Wright [CW91]. They study waves propagating along PEC walls in 2D and conclude that the TE modes suffer dispersion due to the staircase approximation of a PEC wall, while the TM modes do not.

One of their test cases involved a line source close to a PEC wall. The field was probed at a distance of 100 cells from the line source. We make a similar test here. Since we are working with the TM equations, we use the PMC boundary condition. The duality of the Maxwell equations makes this equivalent to the PEC boundary condition for the TE equations, which was shown by Cangellaris and Wright [CW91] to be the "bad" case.

7.9.1 Details of the numerical setup

The computational domain is a square, i.e. we have $N = N_x = N_y$ and $\Delta = \Delta x = \Delta y$. The value of N is chosen such that effects from the ABC or from the edges of the PMC wall do not reach the probing points during our probing window, i.e. N depends on the number of time steps taken. We use $\Delta = 0.1\,\mathrm{m}$ and $CFL = 0.75$. This gives us $\Delta t \approx 0.176\,\mathrm{ns}$.

The line source is applied as a current source, i.e.

$$E_z|_{i,j}^{n+1} = E_z|_{i,j}^{n+1} + \frac{\Delta t}{\epsilon \Delta^2} e^{-((n+\frac{1}{2})\Delta t - t_0)^2/t_w^2}\,, \tag{7.56}$$

where $t_0 = 10.56\,\mathrm{ns}$ and $t_w = t_0/6 = 1.76\,\mathrm{ns}$. The weighting with $\Delta t/\Delta^2$ is necessary to ensure convergence since the discrete current source is equivalent to an integral over the cell area which is Δ^2. The Δt comes from the time discretization.

In the same way as in [CW91] we made two different FD-TD calculations: one where the PMC wall aligns with the FD-TD grid and one where the wall is tilted 45 degrees compared to the grid axes. We will refer to the first case as "para" and the second as "diag".

In the "para" test case we placed the line source at E_z(N/2-50,8) and the PMC wall at H_x(:,2). Hence, the distance from the wall to the line source was $h = 0.65\,\mathrm{m}$. The probing point was placed in E_z(N/2+50,8), i.e. it lies at the same distance from the wall as the line source. The distance between the line source and the probe point were $d = 10.0\,\mathrm{m}$.

In the "diag" test case we placed the line source at E_z(N/2-54,N/2-46) and the PMC wall was modeled by zeroing H_y(k,k), k=1,..,N, and H_x(k+1,k), k=1,..,N-1 in each time step. If we regard the wall as being placed along the line y=x-Δ/2, then the distance from the wall to the line source is $h = 0.60\,\mathrm{m}$. The probing point was located at E_z(N/2+17,N/2+25). Hence, the line source and the probe point were separated by a distance of $d = 10.04\,\mathrm{m}$.

To obliterate the error introduced by the staircase approximation of the tilted wall, we introduced an unstructured grid close to the wall, see Figure 7.23. The line source and the probing point are located as in the "diag" case. The distance h from the PMC wall to the source point and probe point was $h = 4.5 \cdot \sqrt{2}\Delta \approx 0.636\,\mathrm{m}$. Due to the one half cell overlap between the structured and unstructured grid, it is not possible to align the wall along y=x-Δ/2. In Figure 7.23 we only display a small piece of the unstructured grid. In all calculations we have used a large enough grid to insure that effects from the edges of the PMC wall do not influence the solution at our probing point.

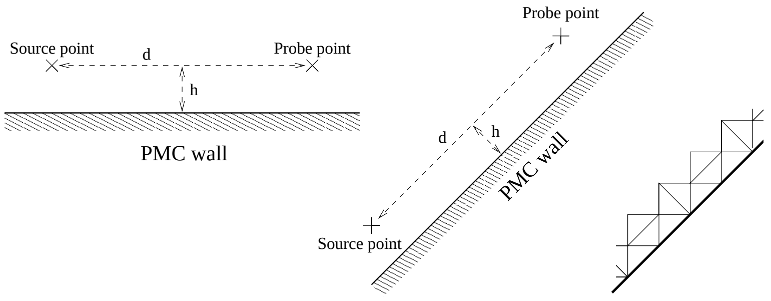


Figure 7.23. The horizontal wall is shown to the left and the tilted wall in the middle. The unstructured grid used for the tilted wall is shown to the right.

The FV hybrid was not stable for $CFL = 0.75$. Instead we used $CFL = 0.375$, doubled the number of time steps and probed the field every other time step.

7.9.2 Results

Figure 7.24 presents the result of this test case. The analytical solution is given by

$$u(t) = \sum_{i=1}^{2} \int_{-\infty}^{t} q(\tau) G(\rho_i, t-\tau) d\tau \,, \tag{7.57}$$

where ρ_1 is the distance between the current source and the probing point, ρ_2 is the distance between the image point of the current source and the probing point,

$$q(\tau) = \frac{1}{\epsilon_0} \frac{\partial}{\partial \tau} e^{-(\tau-t_0)^2/t_w^2} \,, \tag{7.58}$$

and the Greens function is given by

$$G(\rho, t) = \frac{H(tc_0 - \rho)}{2\pi c_0 \sqrt{t^2 - \rho^2/c_0^2}} \,, \tag{7.59}$$

where H(t) is the Heaviside step function. Note that ρ_1 and ρ_2 are slightly different for "para", "diag" and the unstructured solutions. Hence we calculate different analytical solutions for these cases. The analytical solution displayed in Figure 7.24 is for the "para" case.

Comparing the errors in Figure 7.24 we clearly see that the hybrid schemes outperform the "diag" case. We also note that the errors of the two hybrid schemes are very similar. Furthermore, the hybrid solutions are actually slightly better than the "para" solution. This may be due to the fact that the error in the FD-TD scheme is smaller for waves traveling diagonally than for waves traveling along a Cartesian axis, see Chapter 5 in [Taf95].

We also tested two contour path modeling schemes. The first scheme is the one described in Chapter 10.6 of [Taf95]. It is labeled "Taf" in Figure 7.24. This scheme was found to be unstable, but the instability did not significantly affect the solution in the probing point during our probing window. The instability was generated at the wall. Finally, we also implemented the scheme of Dey and Mittra [DM97]. This scheme was found to be stable for this test case.

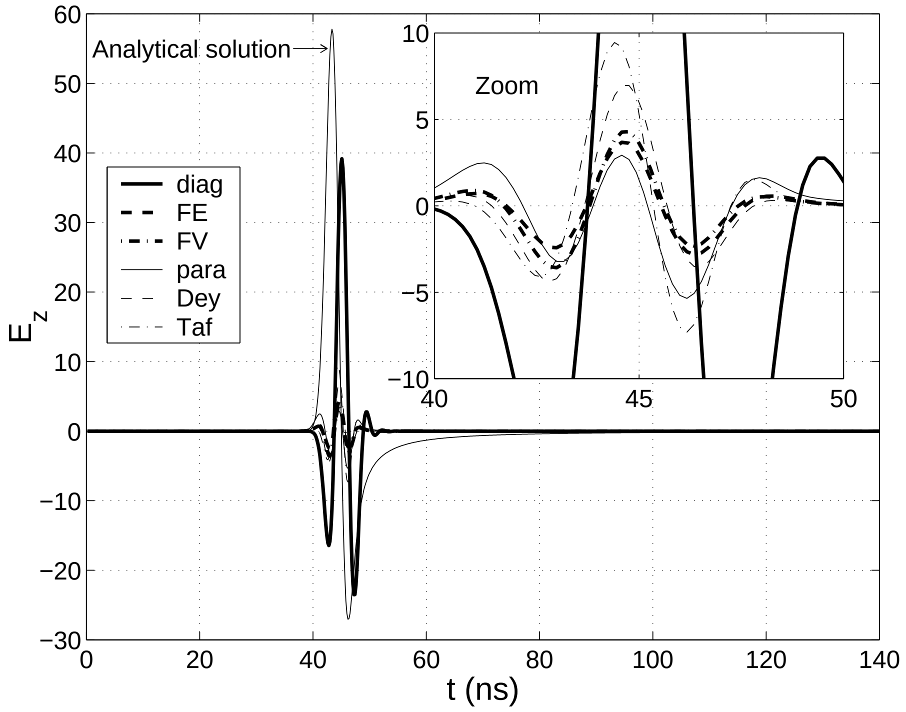


Figure 7.24. The analytical solution in the probing point and the errors of the six schemes.

We note that these contour path modeling schemes give much better results than the "diag" solution, but they are not quite as good as our hybrid schemes. Furthermore, even though the contour path modeling schemes were rather easy to implement for this very special 2D case, we feel that it would be a very complex and cumbersome task to make a general implementation in 3D.

7.10 Conclusion

We have introduced a new general hybrid approach for solving the Maxwell equations in the time domain. By combining the efficiency of the classical FD-TD method with the flexibility of solvers for unstructured grids we obtain a very favorable compromise between efficiency and accuracy. Flexibility is further enhanced by using two solvers on the unstructured grids, one explicit FV solver and one implicit FE solver. Note that the key words here are implicit and explicit and not FE and FV. An explicit solver is much faster per time step than an implicit solver. On the other hand, we have unconditional stability for the implicit solver while the explicit solver must obey a stability limit where the maximum time step is proportional to the shortest edge in the unstructured grid. Furthermore, the accuracy in the FD-TD scheme decreases when the time step decreases, since the spatial and temporal errors are of opposite sign. Hence, the proper choice of solver for an unstructured part in the hybrid depends on the cell sizes in the unstructured grid.

We chose not to put any extra effort into optimizing the unstructured 2D solvers. Hence, we feel that it might be misleading to make any measurements of the efficiency in 2D. Furthermore, we feel that this question is much more relevant in 3D.

Chapter 8

Hybrid Methods in 3D

8.1 Introduction

The hybridization methods in three dimensions are constructed from the same basic principles as in two dimensions. Unstructured grids are used in the vicinity of curved boundaries and small details. A structured grid is used in homogeneous parts of the computational domains. We use FD-TD on the structured grid and either an explicit finite volume (FV) solver or an implicit finite element (FE) solver on the unstructured grids. It is possible to have more than one unstructured grid. The FE solver is intended for grids containing tetrahedra that are small compared to the cell size in the structured grid. On the other hand, the FV solver is intended for unstructured grids, where the cells are of the same size as in FD-TD.

The FD-TD method is described in Chapter 4 and in [Taf00]. The FV and FE solvers are briefly described in the following sections.

8.2 Finite-volume method

The FV solver is based on the integral formulation of Faraday's and Ampère's laws given in (3.11). They are repeated here for convenience:

$$\frac{\partial}{\partial t}\iint_S \mu\, \boldsymbol{H}\cdot\boldsymbol{n}\, dS = -\oint_C \boldsymbol{E}\cdot\boldsymbol{dl}\,, \tag{8.1}$$

$$\frac{\partial}{\partial t}\iint_S \epsilon\, \boldsymbol{E}\cdot\boldsymbol{n}\, dS = \oint_C \boldsymbol{H}\cdot\boldsymbol{dl} - \iint_S \sigma\, \boldsymbol{E}\cdot\boldsymbol{n}\, dS\,, \tag{8.2}$$

where S is an arbitrary area, C is the path that encloses S and $\boldsymbol{n}$ is a unit normal.

The FV method in 3D is similar to the 2D method. It is also based on an integral formulation and employs dual grids. The primary grid consists of tetrahedra. One cell of this grid is shown in Figure 8.1 together with a dual face. In this case the dual face is planar. This is not always the case. In Figure 8.1 there are four corners for the dual face. The number of corners in a dual face equals the number of tetrahedra that share the primary edge associated with the face.

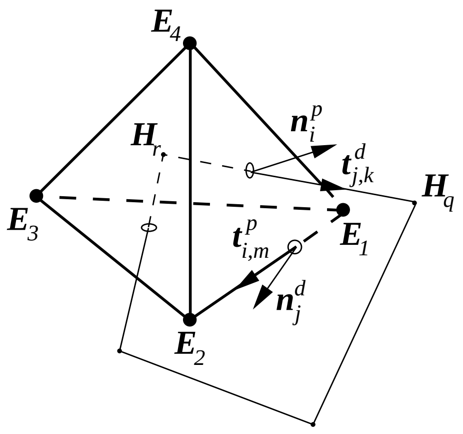


Figure 8.1. A cell in the primary grid and a dual face. Superscript p indicates primary and d indicates dual.

There are some differences between the 2D and 3D FV method. One difference is that the integral formulations are different. The 2D formulation corresponds to what we would get if we integrate the PDE formulation over a volume (V) and then use Gauss' theorem. For instance, Faraday's law would then become

$$\frac{\partial}{\partial t} \iiint_V \mu \boldsymbol{H} \cdot dV = - \oiint_S \boldsymbol{E} \times dS \,, \tag{8.3}$$

where S is the surface that encloses the volume V.

Another difference is that nodes of the dual grid are as often as possible placed in the center of the sphere that circumscribes each tetrahedron. This insures that the primary and dual grids are orthogonal, i.e. the edges of the primary grid ($\boldsymbol{t}_{j,k}^d$) are normals of the faces of the dual grid and vice versa. However, a circumcenter might be located outside its tetrahedron. In this case the dual node is put at the barycenter of the tetrahedra. Using circumcenters is only a sufficient but not a necessary condition for getting an orthogonal dual grid.

The 3D solver is edge based rather than node based. The main unknowns are the fields normal to the respective faces, i.e. they are $\boldsymbol{E} \cdot \boldsymbol{n}_j^d$ and $\boldsymbol{H} \cdot \boldsymbol{n}_i^p$. They are updated by circulating the edge values, i.e. $\boldsymbol{E} \cdot \boldsymbol{n}_j^d$ is updated using $\boldsymbol{H} \cdot \boldsymbol{t}_{j,k}^d$ and similarly for $\boldsymbol{H} \cdot \boldsymbol{n}_i^p$.

On orthogonal grids, we only need $\boldsymbol{E} \cdot \boldsymbol{n}_j^d$ and $\boldsymbol{H} \cdot \boldsymbol{n}_i^p$ because they coincide with the edge values ($\boldsymbol{t}_{j,k}^d \equiv \boldsymbol{n}_i^p$ and $\boldsymbol{t}_{i,m}^p \equiv \boldsymbol{n}_j^d$). For grids that are not orthogonal we need to calculate the edges values from the fields normal to the faces: $\boldsymbol{E} \cdot \boldsymbol{n}_j^d$ and $\boldsymbol{H} \cdot \boldsymbol{n}_i^p$. Simply projecting these fields would not be accurate enough. Instead, node values, e.g. $\boldsymbol{H}_r$ in Figure 8.1, are calculated using a least square fit of the field values normal to the faces. At a dual node, $\boldsymbol{H}_r$ is always calculated from four values since a tetrahedron has four faces. At primary nodes, the number of faces varies, but the same principle applies. Each edge value is then calculated by projecting the field normal to the face to the edge and correct this value using the two node values, see formulas (4.5) and (4.6) in the Licentiate thesis of Fredrik Edelvik [Ede00]. Nodes values are only used where the grids are not orthogonal. On such grids, it is sometimes necessary to damp instability by applying a filter on the node values [Ede00].

The FV solver is thoroughly described in [Ede00], where its dispersion and stability properties are also addressed.

8.3 Finite-element method

The Finite-Element Time-Domain (FE-TD) method in 3D is very similar to the 2D method described in Section 7.3.1. Again, we solve a second-order differential equation. However, in 2D we used the vector wave equation for the magnetic field. Here we use the vector wave equation for the electric field. It is, for $\sigma = \sigma^* = 0$,

$$\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \nabla \times \frac{1}{\mu} \nabla \times \boldsymbol{E} = \boldsymbol{0} \, . \tag{8.4}$$

The weak formulation is now: find $\boldsymbol{E} \in W$ such that

$$\int_\Omega \left(\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} \cdot \boldsymbol{w} + \frac{1}{\mu} \nabla \times \boldsymbol{E} \cdot \nabla \times \boldsymbol{w} \right) d\Omega = - \int_\Gamma \frac{1}{\mu} \boldsymbol{n} \times \nabla \times \boldsymbol{E} \cdot \boldsymbol{w} \, ds \, , \tag{8.5}$$

for all $\boldsymbol{w} \in W$, where $W = H(curl, \Omega) = \{ v : v \in L^2(\Omega), \nabla \times v \in L^2(\Omega) \}$. We use "edge" or "Whitney" tetrahedral elements [Ned80]. Since we are solving for the $\boldsymbol{E}$ field we get a Dirichlet boundary condition for PEC and a Neumann boundary condition for PMC. We use the same timestepping as in 2D, see Section 7.3.3 for a description. We also use the same procedure to assemble and solve the system of linear equations.

Our unknowns can be seen as edge values because on an edge there is only one basis function that has a nonzero tangential component.

8.4 Hybridization

Our hybridization technique in 3D requires the unstructured grids to have a transition layer consisting of semi-structured cells. This means that the outermost unstructured cells are obtained by splitting a hexahedra into five tetrahedra (the hexahedra is of the same size as the FD-TD cell). This outermost layer of cells is called the transition layer because the unstructured grid and the structured grid overlap here (see Figure 8.2). Hence, a number of components coincide in space and time for the solvers in the transition layer and it is these components that are involved in the actual communication between the solvers.

A major difference compared with the hybridization in two dimensions is that in three dimensions only $\boldsymbol{E}$ fields are involved in the communication between the grids. This is illustrated in Figure 8.2. The communication to and from the unstructured grids are now done at the same time. This means that we only communicate once every timestep. This fits very well with the multiblock strategy used by the GEMS multiblock time-domain code where communication between different FD-TD blocks are done once every time step and involve only $\boldsymbol{E}$ fields.

Another difference compared with 2D is that we need to interpolate when we supply values to the unstructured grids. As will be shown in Section 8.5.3, this is a considerable drawback.

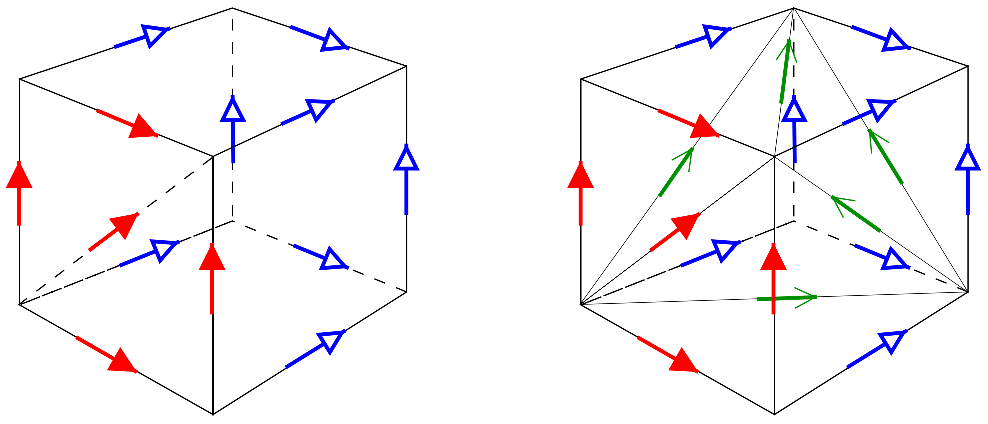


Figure 8.2. The transition layer: To the left we have an FD-TD cell and to the right a cluster of five tetrahedra forming a hexahedron. The FD-TD cell and the hexahedra coincides in space.

Based on Figure 8.2 the hybridization technique can be described as follows:

`Left part of Figure 8.2,` `the structured cell`		`Right part of Figure 8.2,` `the unstructured cells`
The four Cartesian components to the left (filled) are sent to the corresponding components in the unstructured region. The diagonal component (also filled) is calculated by interpolation of the four filled components.	$\Longrightarrow$	The five filled components are received from the structured region and are treated by the unstructured solver as an inhomogeneous Dirichlet boundary condition.
$\Downarrow$		$\Downarrow$
(A full time step is taken)		*(A full time step is taken)*
$\Downarrow$		$\Downarrow$
The remaining eight Cartesian components (hollow) are received from the corresponding components in the unstructured region.	$\Longleftarrow$	When the components are updated in the unstructured region the remaining eight Cartesian components (hollow) are sent to the structured solver.

The interpolation to the filled diagonal component is done in two steps. Assuming that this diagonal component lies in an x-plane, we start by calculating mean values of the two E_z components and the two E_y components. This gives us an $\boldsymbol{E}$ field value at the diagonal that has no x component, but its direction does not coincide with the direction of the diagonal edge. Therefore, we project the $\boldsymbol{E}$ field onto the diagonal. The remaining five diagonal components (ordinary arrows) in the right part of Figure 8.2 are not involved in the hybridization.

An advantage with the transition layer is that the primary and dual grids in the FV solver are orthogonal in the transition layer even though barycenters are used in the four smaller tetrahedra. If the primary and dual grids were not orthogonal in the transition layer, we would have to supply the FV solver with node values in the transition regions.

8.5 Results

8.5.1 Rund

`Rund` is a generic aircraft model geometry which has been designed by the Swedish Defence Research Establishment (FOA). (The Swedish word “rund” means round.) The hybrid mesh for `Rund` was generated at Ericsson Saab Avionics using CADfix [CAD]. Figure 8.3 shows `Rund` and part of the unstructured grid. (A color version of the same figure can be found on page 137.) `Rund` is approximately one meter long, one meter between the wing tips, and half a meter high. The computational domain is discretized using approximately 150 000 tetrahedra and 500 000 FD-TD cells. In the FD-TD grid we have $\Delta = \Delta x = \Delta y = \Delta z = 1\,\mathrm{cm}$. In the unstructured grid, we use the explicit FV solver because the sizes of the tetrahedra are of the same order as the cells in the FD-TD grid. Furthermore, the large number of tetrahedra makes it impossible to use the implicit FE solver based on a direct solver for the system of linear equations. We set $CFL = 0.2$.

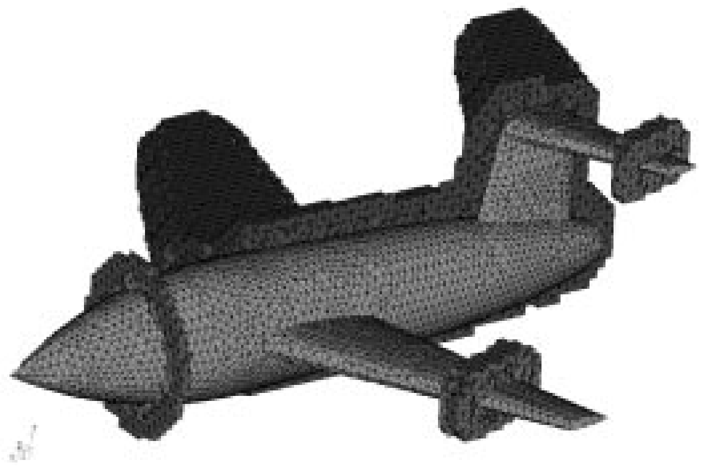


Figure 8.3. Part of the unstructured grid around the aircraft model geometry `Rund`. The structured grid continues outside the shown unstructured cells. The interface between the grids is staircased to minimize the number of unstructured cells.

We let a Gaussian wave strike `Rund` head on. We perform two computations, one with horizontal polarization and one with vertical polarization. We use the FD transform described in Section 4.11 to calculate the bistatic RCS. The results for 1.5 GHz are shown in Figures 8.4 and 8.5. At 1.5 GHz we have twenty cells per wavelength when $\Delta = 1\,\mathrm{cm}$. MoM solutions are used as reference solutions. These solutions have been calculated with the GEMS frequency-domain code.

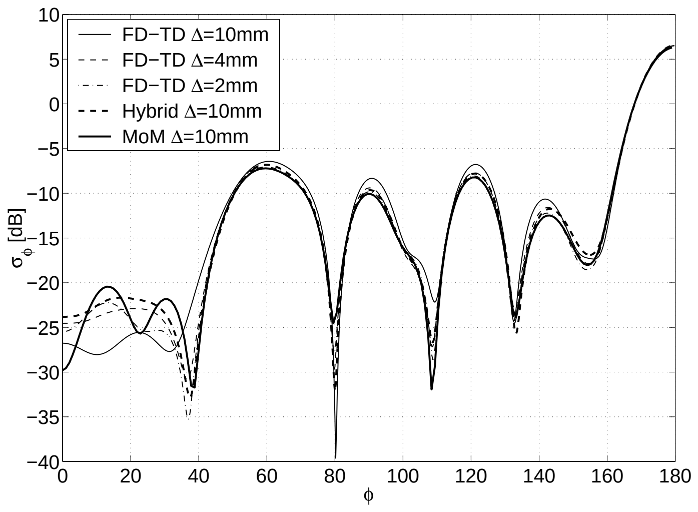


Figure 8.4. Comparison of bistatic RCS between the FD-FV hybrid, FD-TD and MoM at 1.5 GHz for horizontal polarization.

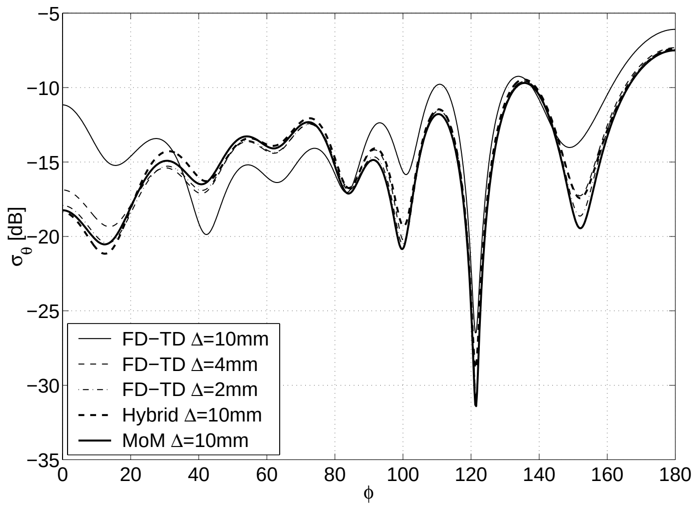


Figure 8.5. Comparison of bistatic RCS between the FD-FV hybrid, FD-TD and MoM at 1.5 GHz for vertical polarization.

We see that except for the 10 mm FD-TD results, all solutions are rather good. The largest deviations between the different solutions appears around the monostatic angle, $\phi = 0$. The staircasing in FD-TD makes it difficult to compute head-on monostatic RCS.

The amount of memory used for the hybrid is of course much more than for the 10 mm FD-TD computation. The same holds for the execution time. The hybrid uses slightly less resources than the 4 mm FD-TD computation. Hence it is more interesting to compare the accuracy between these two computations. The results are rather similar. The major difference is that the hybrid results are better close to the monostatic angle for vertical polarization. For horizontal polarization, none of the computation give accurate results close to the monostatic angle. The reason for this is not known. It might be due to the larger dynamic range for horizontal polarization. The monostatic RCS for horizontal polarization is 40 dB lower than the largest bistatic RCS value. For vertical polarization, this difference is only 15 dB. Further testing is needed to establish the cause of this discrepancy. However, in order to do a more refined hybrid computation we need a parallel FV solver. This is under development.

In Chapter 10, color plots of the surface currents on `Rund`are shown in Figures 10.2 and 10.3

8.5.2 The NASA almond

A common test problem for electromagnetic methods is the NASA almond. It is defined in for instance [DSWB86]. Figure 8.6 shows the surface grid on the NASA Almond.

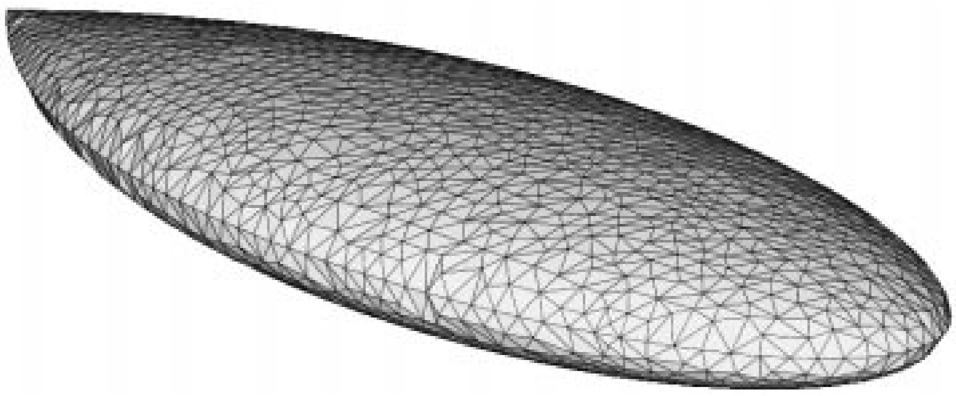


Figure 8.6. The NASA Almond.

The computations have been performed at Ericsson Saab Avionics using the FD-FV hybrid. They also generated the grids. The monostatic RCS for nineteen different angles of incidence are shown in Figure 8.7. They agree very well with a converged MoM solution.

In order to see how large the reflections from the grid interface are, Ericsson Saab Avionics also filled the interior of the NASA almond with tetrahedra and performed a vacuum computation. In Figure 8.7 these results are labeled `FV-TD`

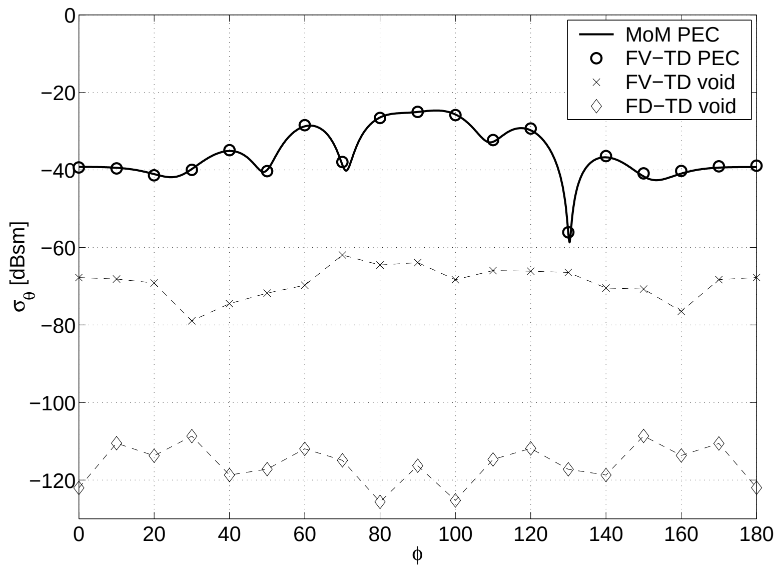


Figure 8.7. Monostatic RCS for the NASA Almond at 3 GHz.

`void`. The errors are -60 to -80 dB. This can be compared with the errors for pure FD-TD which are -110 to -125 dB.

The frequency-domain transform with dispersion compensation (see Section 4.11) was used to generate the far field. The compensation is based on the dispersion relation for FD-TD. Hence the higher error for the FD-FV hybrid might not only be due to reflections in the transition layer, but also partially caused by the discrepancy between the FD-TD dispersion relation and the actual dispersion in the FV method. In order to investigate the effects of the dispersion compensation, we also did the computations with $\phi = 0$. We did computations with and without dispersion compensation. The results are given in Table 8.1.

Dispersion compensation	Yes	No
FV-TD PEC	-39.2	-39.0
FV-TD void	-67.7	-63.9
FD-TD void	-122.0	-74.9

Table 8.1. Monostatic RCS (σ_θ) for $\phi = 180$.

The PEC RCS results are almost unchanged, while there is a drastic increase in the FD-TD void RCS and a small increase in the FV-TD void RCS. It is clear that the dispersion compensation plays a major roll in the excellent FD-TD void result. We can also conclude that our hybridization method does not introduce reflections that destroys the accuracy of the monostatic RCS for the NASA almond.

8.5.3 Convergence studies

Procedure

We will estimate the order of convergence primarily for vacuum test cases. Because we have analytical solutions for these cases, we only need two numerical solutions: one coarse grid solution and one fine grid solution. In the structured grid we will always use $\Delta = \Delta x = \Delta y = \Delta z$. The value of Δ for the coarse grid will always be an integer factor larger than in the fine grid. We call this factor the refinement factor (rf). The CFL number is defined as in (4.11), i.e. it is always related to the stability limit in the FD-TD method.

When refining a grid we keep the CFL value fixed. We also hold the location of the Huygens' surfaces fixed. A highly absorbing ABC is used, usually a twelve cells thick PML layer, to ensure that reflections from the outer boundary do not noticeably affect our convergence estimates.

We probe one or more electric field component every time step on the coarsest grid. On the fine grid we probe every rf time step. We always probe electric field components because their values are represented at $t = n\Delta t$ where $n = 0, \ldots, N_t$. Magnetic field values, on the other hand, have their values represented at $t = (n - 1/2)\Delta t$. Hence, by only probing electric field components we can use $rf = 2$ without introducing a need for temporal interpolation.

We calculate the convergence estimate p_{max} as

$$p_{max} = \frac{ln(max(abs(err_c))) - ln(max(abs(err_f)))}{ln(rf)}. \tag{8.6}$$

where err_c and err_f are vectors containing the errors in a probed point (or the l_2 error of some suitable part of the computational domain). The indexes c and f stands for coarse and fine. The convergence estimate p_{mean} is calculated equivalently as

$$p_{mean} = \frac{ln(mean(abs(err_c))) - ln(mean(abs(err_f)))}{ln(rf)}. \tag{8.7}$$

First test case

First we study the convergence properties of the unstructured solvers for vacuum. This is done using semi-structured unstructured grids, where each cube has been split into five tetrahedra (as in the right part of Figure 8.2). We excite the unstructured solvers by supplying analytical boundary conditions, i.e. a time dependent Dirichlet boundary condition. This is done in the same way as the FD-TD solver sends values to the unstructured solvers. Hence, it is only the the four (filled) Cartesian components in the left part of Figure 8.2 that are given boundary values. The filled diagonal component is calculated by interpolating the four Cartesian components.

The incident field is defined by

$$\boldsymbol{E}^{inc}(t,\boldsymbol{x}) = \boldsymbol{E}_p f(t - \boldsymbol{k}\cdot(\boldsymbol{x}-\boldsymbol{X}_0)/c_0)\,, \tag{8.8}$$

and

$$\boldsymbol{H}^{inc}(t,\boldsymbol{x}) = \frac{\boldsymbol{k}\times\boldsymbol{E}_p}{Z_0} f(t - \boldsymbol{k}\cdot(\boldsymbol{x}-\boldsymbol{X}_0)/c_0)\,, \tag{8.9}$$

where $f(t)$ is a Gaussian, defined as

$$f(t) = e^{-(t-t_0)^2/t_w^2}\,, \tag{8.10}$$

with $t_0 = 40/c_0$ and $t_w = t_0/6$. The angle of incidence is given by $\boldsymbol{k} = (1,0,0)$, the electric polarization is given by $\boldsymbol{E}_p = (0,0,1)$ and we set $X_0 = (2,0,0)$.

The size of the unstructured grids is 6 m × 6 m × 6 m. The coarsest grid is made from cubes with $\Delta = 1$ m. Hence, it contains 1080 tetrahedra. When we refine the grids we use $rf = 2$ and we perform two refinements. For FE we use $CFL = \sqrt{3}/2$ and for FV we use $CFL = \sqrt{3}/4$. We take 160 time steps for FE on the coarsest grid. For FE timestepping we use the Crank-Nicholson method.

We probe the z component of the electric field at $(x,y,z) = (12,10,10)$ in the FD-TD grid. There is no E_z component at this location. Instead we take a mean value of the two E_z components in $(12,10,10+\Delta z/2)$ and $(12,10,10-\Delta z/2)$. These two values are in fact equal due to the symmetry around $z = 10$. We make sure that the components we probe in the FD-TD grid are components that receive values from the unstructured grid. This means that the back border between the unstructured grid and the FD-TD grid is placed at $x = 12 + \Delta$.

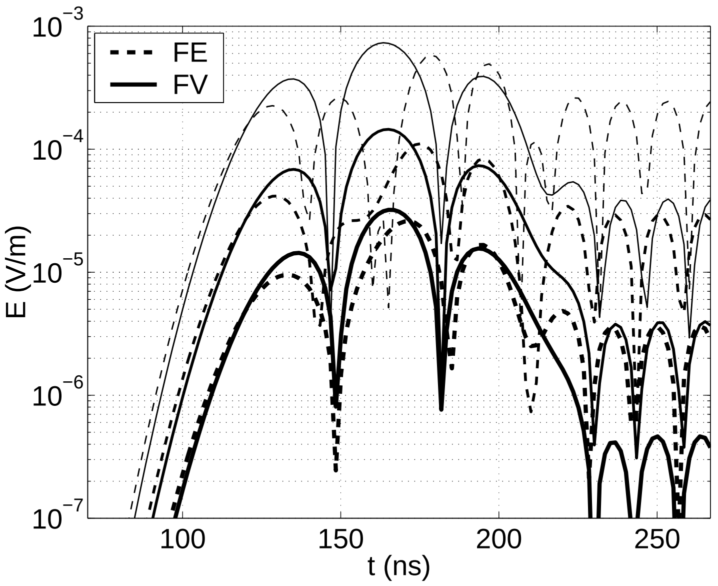


Figure 8.8. Errors for three refinement levels and two methods.

Solver	FE		FV	
Estimate	p_{mean}	p_{max}	p_{mean}	p_{max}
All three grids	2.61	2.48	2.45	2.38
Two coarsest grids	2.50	2.40	2.41	2.34
Two finest grids	2.18	2.09	2.25	2.18

Table 8.2. Estimates of the order of convergence for the two unstructured grid methods.

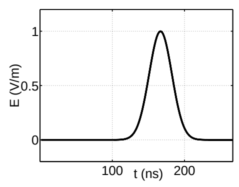


Figure 8.9. The analytical solution.

Table 8.2 displays the convergence estimates for this test. Clearly, we have second-order convergence. The fact that all the estimates are above two might be an effect of having the probing point so close to the boundary where exact values are given. In fact, they get closer to the boundary when we refine the grids. This procedure was chosen for our first test, because it was easier to probe the FD-TD grid than the unstructured grids.

The errors in the probing point are displayed in Figure 8.8. We see that the FE solver has slightly smaller error in the beginning. On the other hand, at the end when the analytical solution (see Figure 8.9) approaches zero, we have significantly smaller error for the FV solver.

Second test case

In this test we probe the interior of the unstructured grids. The unstructured solvers are still fed with analytical solutions from the FD-TD solver. We use the same Gaussian excitation as in the previous test. We set $CFL = \sqrt{3}/4$ for both solvers and take 320 time steps on the coarse grid. Again, we use the Crank-Nicholson method for the FE timestepping.

Because the electric components in the grids are located at the center of each edge, we need to use an odd value for rf. If we use an even value for rf, then all the midpoints of the edges in the coarse grid would become nodes in the fine grid. Here, we will use $rf = 3$ and $rf = 5$. Furthermore, we use an odd number of cells (cubes) in the z-direction an even number of cells in the x- and y-directions. This will ensure that we have a z-edge in the center of the unstructured grids.

The size of the unstructured grids is $4\,\text{m} \times 4\,\text{m} \times 3\,\text{m}$. The coarsest grid is made from cubes with $\Delta = 1\,\text{m}$. Our first probe is located at the centermost z-edge in the unstructured grid. The second probe is located with an offset of $(0.5, -0.5, 0)$ to the first probe.

Table 8.3 contains the convergence estimates for the two probes. We have almost exactly second-order accuracy on semi-structured grids for the unstructured solver.

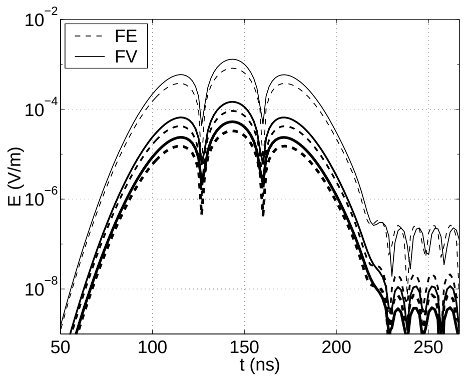
The errors in the centermost edge are displayed in Figure 8.10(a). Once again, the FE solver has slightly smaller error in the beginning but has a larger error in the end. However, this time the differences are smaller. This is due to the fact that

	rf	FE		FV	
Edge		1	2	1	2
p_{mean}	3	2.00	2.00	1.99	1.99
p_{max}	3	1.99	1.99	1.99	1.99
p_{mean}	5	2.00	2.00	2.00	2.00
p_{max}	5	1.99	1.99	1.99	1.99

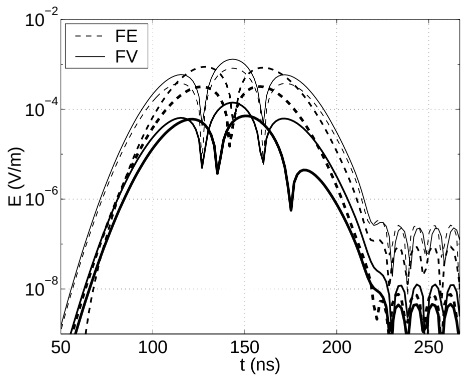
Table 8.3. Estimates of the order of convergence for the two unstructured grid methods. Edge 1 is the centermost edge.

we have the same CFL number for both solvers. The errors in the other probed edge are very similar to the result displayed in Figure 8.10(a).

This test, as well as the test in the previous section, was performed on very well behaved grids. To investigate how the accuracy of the solvers change when the grid quality decreases, we add random disturbances to the refined grids. All nodes with the same x- and y-values as the centermost z-edge, and all nodes in the transition layer are kept fixed. All other nodes are moved with uniformly distributed random numbers. The maximum change in a nodes x-, y- and z-values was 10% of Δ. The coarse grid was unchanged. The errors are displayed in Figure 8.10(b). The node filter in the FV solver was not applied. Comparing Figures 8.10(a) and 8.10(b) we see that the FE solver is more sensitive to these grid disturbances than the FV solver.



(a) Semi-structured grids



(b) Refined grids disturbed

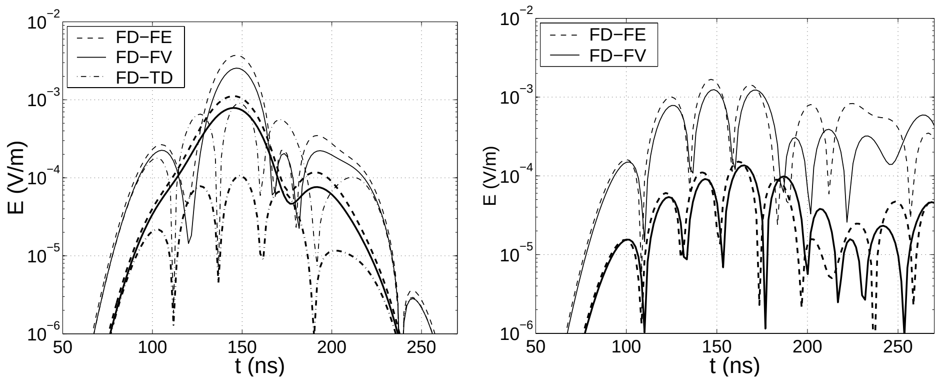
Figure 8.10. Errors for three refinement levels and two solvers.

Third test case

We will now study the hybrid code. We use the two coarsest unstructured grids from the previous test case. They are embedded in an FD-TD grid which is 14 m $\times$ 14 m $\times$ 13 m.

We use the same incident field as in the previous test. It is generated by Huygens' surfaces which are located two meters from the outer boundary. For FD-TD and the FD-FE hybrid we use $CFL = \sqrt{3}/2$ and for the FD-FV hybrid we use $CFL = \sqrt{3}/4$. On the coarsest grid, we take 640 time steps for the FD-FV hybrid and 320 time steps for the two other solvers.

The absolute value of the errors for E_z at $(x, y, z) = (4, 7, 6.5)$ are shown in Figure 8.11(a). We see that they are significantly larger for the hybrid methods.



(a) Standard hybrid

(b) Analytic diagonal replacement

Figure 8.11. Errors for two refinement levels at $(x, y, z) = (4, 7, 6.5)$.

We probe the solution in all E_z components (on the coarse grid) that lie one meter or one and a half meter from the outer boundary. We calculate the l_2 error and use these time dependent vectors to estimate the convergence. Estimates are given in Table 8.4. Clearly, we have super-linear convergence. However, we do not have second-order convergence.

FD-TD		FD-FE		FD-FV	
p_{mean}	p_{max}	p_{mean}	p_{max}	p_{mean}	p_{max}
1.96	1.98	1.25	1.11	1.15	1.04

Table 8.4. Estimates of the order of convergence for the two hybrid methods and the FD-TD method.

The convergence estimates for the two hybrid methods are rather similar. This fact and the fact that we have, in our previous test cases, shown that the FV-TD and FE-TD solvers are second-order accurate on these grids in stand alone mode, strongly indicates that our hybridization technique destroys the second-order accuracy. The likely culprit is the interpolation of diagonal values. The error in the interpolation itself is proportional to Δ^2. However the interpolated value is then used in what is basically a difference approximation. If this happened in only one point, it would not destroy the global accuracy. However, because it happens in every diagonal on the outer surface of the transition region, it does affect the order of accuracy globally.

To validate our hypothesis that the interpolation of diagonal values are causing the deteriorated accuracy, we performed two tests. In the first test we replaced the diagonal values with analytical values. The absolute value of the errors in the probe at $(x, y, z) = (4, 7, 6.5)$ are shown in Figure 8.11(b). Convergence estimates are given in Table 8.5.

FD-FE		FD-FV	
p_{mean}	p_{max}	p_{mean}	p_{max}
1.93	1.96	2.11	1.88

Table 8.5. Estimates of the order of convergence for the two hybrid methods when diagonal values are substituted by analytical values.

In the second test we used higher order interpolation. Four values were used instead of two when performing the interpolation of the diagonal components. This procedure was unstable. However, it seemed to be second-order accurate for the brief time span that was unaffected by the instability.

These tests clearly demonstrate that it is the interpolation of diagonal values that destroys the second-order accuracy.

8.6 Stability

The separate solvers

The implicit FE-TD solver is unconditionally stable [LS95]. The FD-TD solver is also stable as long as the CFL condition in (4.10) is not violated. The explicit FV-TD solver is stable on orthogonal grids [Ede00] as long as the time step is chosen properly. On general grids a node filter is applied to suppress the amplitude of the highest frequency components [Ede00]. In practice, for scattering problems, this filter improves the stability without loosing accuracy in the solution. However, in terms of stability, cavity problems are in general much more demanding than scattering problems due to longer simulations that could require millions of timesteps. Methods to improve the stability of the FV-TD solver for resonant cavities are currently under investigation.

The hybrid solvers

We have demonstrated in Sections 8.5.1 and 8.5.2 that our FD-FV hybrid solver produces good results for scattering problems. As demonstrated in [Led01], this is also the case for the FD-FE hybrid solver.

For highly resonant cavities our hybrid solvers become unstable. This happens even if the unstructured grid is orthogonal in which case all stand alone solvers are stable. Hence, these instabilities are caused by the hybridization. In most cases, the instabilities do not emerge until after several tens of thousands of iterations. Stability tests for our hybrid solvers for cavity problems are further discussed in [Led01].

8.7 Conclusions

Most results in this chapter have been generated with the FD-FV hybrid solver. This is because the FE solver uses a direct solver for the system of linear equations, and hence quickly becomes too expensive to use. An obvious solution to this dilemma is to use an iterative solver instead. Iterative solvers have been successfully used for FD-FE hybrids by other researches [WI97, Ryl00, Ril01].

We have shown that our FD-FV hybrid solve produces good results for two scattering problems: a generic aircraft model geometry and the NASA almond model problem. More scattering results may be found in [EL00, ES00, Ede00, Led01]. These results prove that the FD-FV hybrid is a successful way to overcome the staircasing errors of the FD-TD method. However, the FD-FV hybrid is not fully second-order accurate. This is most evident on vacuum test cases (no scattering object). Through extensive numerical testing, we have shown that the deterioration in order of accuracy is caused by the interpolation of the diagonal components in the transition layer. This interpolation is performed when the FD-TD solver sends values to the FV solver. A possible way to avoid the interpolation is to use pyramidal elements to connect the tetrahedral grid with the hexahedral grid.

The vacuum test case results for the FD-FE hybrid gave results that were very similar to the results of the FD-FV hybrid. Again, second-order accuracy is destroyed by the interpolation of the diagonal components. An FD-FE hybrid method using pyramidal elements to connect the tetrahedral grid with the hexahedral grid has recently been introduced [Ryl00, RB00, Ril01]. In [Ryl00] it is shown that this method gives second-order accuracy for scatters with a smooth boundary. This hybrid method is stated to be stable, even though no rigorous proof has been given.

Chapter 9

Modeling of Inhomogeneous Materials in FD-TD

9.1 Introduction

9.1.1 Motivation

As was demonstrated in Chapter 7, severe errors can be caused by the staircasing of a material interface and second-order accuracy will no longer be maintained. In Chapter 7 we showed that this can be circumvented by using unstructured grids at the material interface. Here we will examine a much cheaper recipe which we will refer to as regularization. We will replace the discontinuous material function with a continuous function. Hence, we do not need to increase the width of the updating stencil. We only change the coefficients in it. The ultimate goal of this procedure is to make hybridization unnecessary for material interfaces.

We will also study the case when the material interface coincides with the field components in the Yee grid. The issue here is how to choose the discrete values for the material properties at the interface. We will begin with this case.

9.2 Interfaces coinciding with the grid

9.2.1 Background

As described in Section 4.3, it is possible to model inhomogeneous materials by space dependent material coefficients. In this section we study the case were the material interface coincides with the Yee cell interfaces. We will not address the more complicated case of frequency-dependent materials, but will be content with studying simple lossless materials.

There is a fundamental physical difference between the normal and tangential components of the electric field ($\boldsymbol{E}$) and and the magnetic field ($\boldsymbol{H}$) at a material interface. The tangential components are always continuous, while the normal components are discontinuous if the related material parameter (ϵ or μ) is discontinuous (cf. (3.16)).

9.2.2 Interface conditions

The question we address here is: How shall the discrete value of a material coefficient be calculated at a material interface?

Interface condition 9.1. *If the updated component is normal to the interface, harmonic average should be used to calculate the material coefficient:*

$$\frac{2}{\xi_N} = \frac{1}{\xi_1} + \frac{1}{\xi_2}\,, \tag{9.1}$$

Interface condition 9.2. *If the updated component is tangential to the interface, arithmetic average should be used to calculate the material coefficient:*

$$\xi_T = \frac{\xi_1 + \xi_2}{2}\,, \tag{9.2}$$

The subscripts T and N are used to indicate that coefficients related to tangential field components respectively normal field components.

We will proceed by first giving some theoretical justification to these interface conditions and then verify that they give second-order accuracy by numerical experiments.

9.2.3 Our implementation

In the 3D GEMS [GEM] time-domain code it is assumed that each cell consists of one material. It follows (see Figure 4.2) that electric field components on a material interface are always tangential to the material interface, while magnetic field components are always normal to the interface. Because the electric field components are located at the edges of the cells, there may be as much as four different materials sharing this component. Magnetic field components are located on the faces of the cells. Hence there cannot be more that two materials sharing a magnetic field component.

In 2D, we study the TM$_z$ polarization of the Maxwell equations. Here the E_z components are always tangential, while the H_x and H_y components may be either tangential or normal to the interface. If we introduce the same limitation as in 3D, that a cell may consist of only one material, we again get that the magnetic components are normal to the interface.

9.2.4 Related work

The issue of how to set the coefficients at material interfaces is neglected in the major FD-TD references [Taf00] and [KL93]. A careful study can be found in [He97], where it is shown that an arithmetic mean of the material coefficient should be used if the updated component is tangential to the interface. The theory is described in the next section. Our work started out in order to verify this statement. However, their results turned out to be valid only for tangential field components.

Independently of our work, Hirono et al. [HSL+00] studied the case with a material interface for the TE$_y$ polarization of the Maxwell equations. The interface aligns with a Cartesian axis, but does not coincide with the components of the FD-TD grid. They derived a theoretical formula for the reflection coefficient and also gave conditions for how to choose the nearby values of ϵ to get second-order accuracy. For the special cases when the interface coincides with either a tangential or a normal electric field component, their formulas are equivalent to the formulas given in Section 9.2.2. In [HSL+00], the authors state that they are the first to theoretically prove the second-order accuracy of this averaging procedure.

The formulas in Section 9.2.2 can also be found in [LM99] where they are given without any motivation or derivation. The main issue of [LM99] is to find a method to treat arbitrarily shaped interfaces. They do this by introducing an effective dielectric tensor for the TE$_z$ polarization of the Maxwell equations. Their method is similar to methods for frequency dispersive materials. Near the interface, D_x and D_y components are calculated from the H_z components and then the E_x and E_y components are then calculated from the D_x and D_y components. Their method gives good results at the expense of introducing extra equations near the interface.

Analytical reflection conditions for the TE$_z$ polarization of the Maxwell equations for a dielectric interface can also be found in [AWV+99]. However, in this paper they do not use any averaged value of ϵ. This work has been expanded in [Mar01] to treat interfaces with lossy dielectric materials.

9.2.5 Tangential components by surface integration

Here we study the update of tangential field components. We have already stated that the arithmetic mean must be used for the material coefficient. This can be shown by studying the integral formulation of the Yee scheme. Consider Ampère's law in (3.11). If we let $\sigma = 0$ it becomes

$$\epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \, . \tag{9.3}$$

Let S be the rectangular surface defined by the four $\boldsymbol{H}$ field components in the update of E_z (see Figure 9.1 and (4.7)). If we integrate (9.3) over S and use Stokes' theorem we get

$$\frac{\partial}{\partial t} \iint\limits_S \epsilon E_z dS = \oint\limits_C \boldsymbol{H} \cdot d\hat{l} \, . \tag{9.4}$$

If we assume the magnetic field components to be constant along each of the four sides of S and E_z to be constant on S, we find that for homogeneous materials ($\epsilon_I = \epsilon_{II}$)) this is equivalent to the FD-TD formulation. An advantage with this integral formulation is that it gives us a clear indication on how to choose a discrete value for ϵ for inhomogeneous materials. Because we assume E_z to be constant on S we get

$$\frac{\partial E_z}{\partial t} \iint\limits_S \epsilon dS = \oint\limits_C \boldsymbol{H} \cdot d\hat{l} \, . \tag{9.5}$$

In the case illustrated in the left part of Figure 9.1, the integral in the left hand sides becomes $\Delta x \Delta y (\epsilon_I + \epsilon_{II})/2$. For the more general case in the right part of Figure 9.1, this integral becomes $\Delta x \Delta y (\epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4)/4$. Hence, our conclusion is that *an arithmetic average should be used for* ϵ.

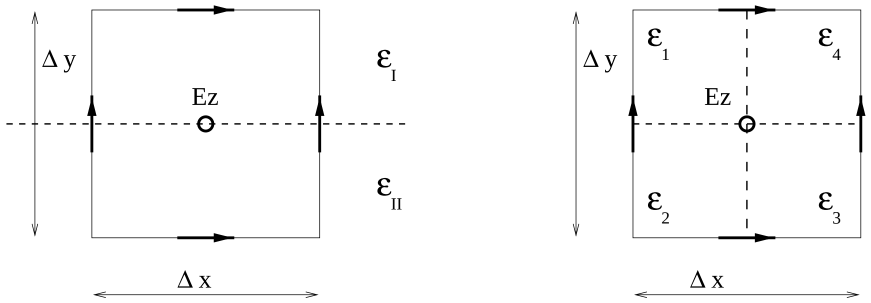


Figure 9.1. A tangential field component (E_z) at a material interface. Arrows signifies magnetic field components (H_x and H_y).

However this approach cannot be extended to components normal to the interface. An equivalent integral formulation for a normal field component gives us a surface integral, where the surface coincides with the interface. Hence, there is no well defined value of μ available if we try to integrate μ over this surface.

9.2.6 Tangential components by volume integration

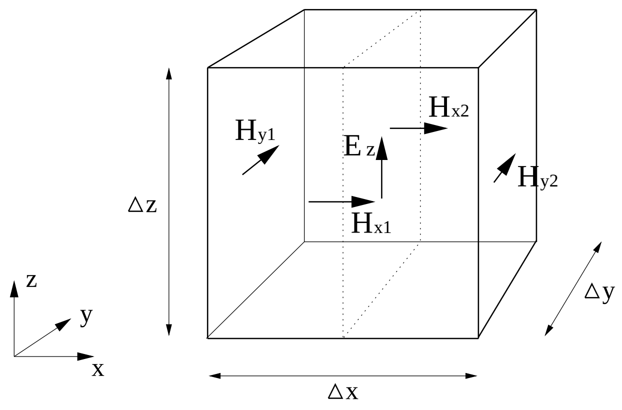


Figure 9.2. A brick with an E_z component in the center The E_z component is tangential to the material interface.

Consider the brick in Figure 9.2 and assume that it is split into two parts by a material interface. This interface is parallel with the yz-plane. If we integrate (9.3) over this brick we get

$$\frac{\partial}{\partial t}\iiint_V \epsilon E_z dV = \iiint_V \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right)dV = \oiint_S (H_y n_x - H_x n_y)dS\,, \tag{9.6}$$

where we have used Gauss' theorem. All the field components involved in (9.6) are continuous across the interface, but ϵ is discontinuous. If we assume E_z to be constant in V, the left hand side becomes

$$\frac{\partial E_z}{\partial t}\iiint_V \epsilon dV = \Delta x \Delta y \Delta z \frac{(\epsilon_I + \epsilon_{II})}{2}\frac{\partial E_z}{\partial t}\,. \tag{9.7}$$

The right hand side of (9.6) becomes

$$\iint_{S_1} H_{y2} dS_1 - \iint_{S_2} H_{y1} dS_2 - \iint_{S_3} H_{x2} dS_3 + \iint_{S_4} H_{x1} dS_4\,, \tag{9.8}$$

where S_i are sides of the brick. If we assume that the H_y components are constant on S_1 and S_2, and that the H_x components are constant on S_3 and S_4, we can easily evaluate the integrals in (9.8). We get

$$(H_{y2} - H_{y1})\Delta y \Delta z - (H_{x2} - H_{x1})\Delta x \Delta z\,. \tag{9.9}$$

Hence, we again reach the conclusion that *an arithmetic average should be used for* ϵ. The benefit of the volume integration procedure is that it is possible to extend it to normal field components.

9.2.7 Normal components by volume integration

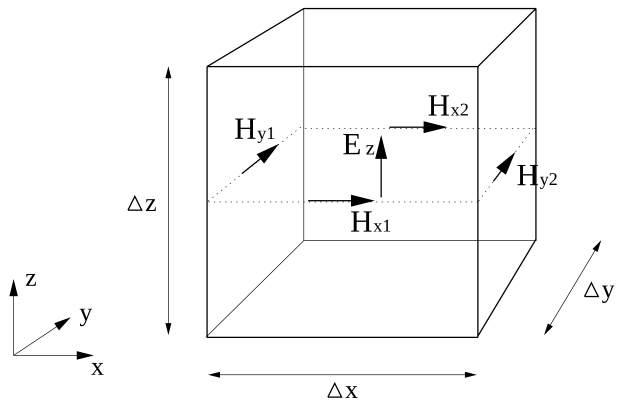


Figure 9.3. A brick with an E_z component in the center. The E_z component is normal to the material interface.

Again, we consider the brick in Figure 9.2. This time we let the material interface be the plane defined by the magnetic field components (see Figure 9.3). Hence, E_z is now a normal field component located at the interface. Integrating (9.3) over the brick and using Gauss' theorem we get

$$\frac{\partial}{\partial t}\iiint\limits_V \epsilon E_z dV = \oiint\limits_S (H_y n_x - H_x n_y) dS\,. \tag{9.10}$$

Here, we get into trouble. Both factors in the integral in the left hand side are discontinuous at the material interface. This makes the integral difficult to handle. To avoid this we divide (9.3) with ϵ before integrating it. We get

$$\frac{\partial}{\partial t}\iiint\limits_V E_z dV = \iiint\limits_V \frac{1}{\epsilon}(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y})dV\,. \tag{9.11}$$

Now we can handle the left hand side. Assume that E_z is constant on either side of the interface and takes the value E^I and E^{II} in these two regions. The left hand side now becomes

$$\Delta x \Delta y \Delta z \frac{1}{2}\frac{\partial(E^I + E^{II})}{\partial t}\,. \tag{9.12}$$

It follows that it is natural to let E_z represent the arithmetic mean value of E^I and E^{II}! Hence, we let

$$E_z = \frac{E^I + E^{II}}{2}\,. \tag{9.13}$$

The right hand side of (9.11) is a little trickier to handle. The appearance of the $1/\epsilon$ factor prohibits us from directly applying the Gauss' theorem. We solve this dilemma by splitting the right hand side into two parts. We get

$$\iiint\limits_{V_I} \frac{1}{\epsilon}(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y})dV_I + \iiint\limits_{V_{II}} \frac{1}{\epsilon}(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y})dV_{II}\,, \tag{9.14}$$

where V_I is the lower half of the brick in Figure 9.3, and $V = V_I \cup V_{II}$. Because ϵ is constant in V_I and constant in V_{II}, we get

$$\frac{1}{\epsilon}\iiint\limits_{V_I} (\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y})dV_I + \frac{1}{\epsilon}\iiint\limits_{V_{II}} (\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y})dV_{II}\,. \tag{9.15}$$

Applying Gauss' theorem gives us

$$\frac{1}{\epsilon_I}\iint\limits_{S_I}(H_y n_x - H_x n_y)dS_I + \frac{1}{\epsilon_{II}}\iint\limits_{S_{II}}(H_y n_x - H_x n_y)dS_{II}\,. \tag{9.16}$$

Now we assume that the magnetic field components in Figure 9.3 are constant on their respective surface. We get

$$\frac{\Delta z}{2}(\frac{1}{\epsilon_I} + \frac{1}{\epsilon_{II}})((H_{y2} - H_{y1})\Delta y - (H_{x2} - H_{x1})\Delta x)\,. \tag{9.17}$$

Comparing this with the left hand side in (9.12), we see that we have again reached the standard FD-TD update equation. Our conclusion is: *the discrete ϵ-value should be calculated as a harmonic mean value* if the updated electric field component is normal to the interface.

In our implementation, an electric field component can never be normal to a material interface (see Section 9.2.3). However, the derivation in this section is directly applicable on normal magnetic field components for discontinuous μ.

Materials with $\mu_r \neq 1$ are rare and furthermore usually modeled as perfect electric conductors. Hence it may seem rather unimportant to study how to handle discontinuous μ. However, the difference in how to take an average for ϵ_r and μ_r are not dependent on the physical parameters themselves, but on the positioning of field components in the Yee cell. The choice of where to locate the six electromagnetic field components in the Yee cell in Figure 4.2 is arbitrary. It could just as well be the other way around with magnetic field components at the edges and electric field components at the faces of the cell.

9.2.8 Numerical verification in two dimensions

We performed a numerical experiment in two dimensions to establish which kind of mean value that should be used for the material coefficient when a normal component is updated. (In our case, the material coefficient is μ_r.) Three natural options to test are: arithmetic, geometric and harmonic mean values.

It is not possible to perform this experiment in one dimension since no component in 1D is normal to the interface. We used the TM$_z$ equations in 2D for this experiment. The setup is illustrated in Figure 9.4. On the coarsest grid level we

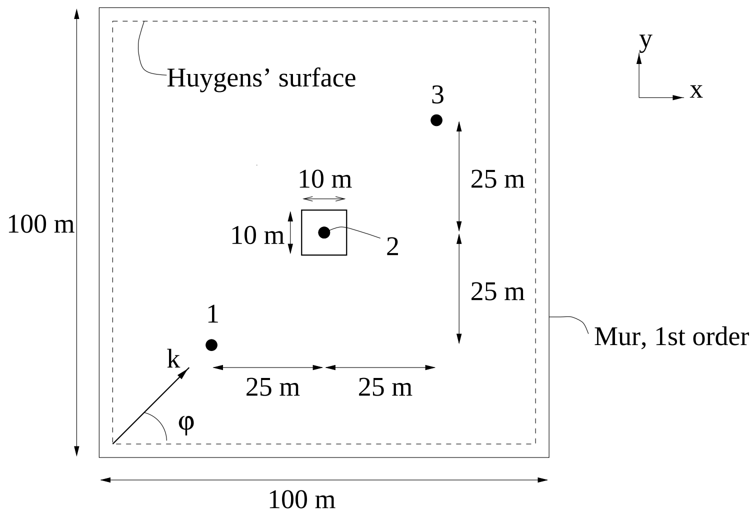


Figure 9.4. Setup of numerical experiment in 2D. The square in the center contains a material different from vacuum. Point 2 is located in the center of the computational domain and in the center of the square.

have 100×100 cells. We take 500 time steps using $CFL = 1/\sqrt{2}$. The shape of the plane wave is a Gaussian

$$f(t) = e^{-(t-t_0)^2/t_w^2}\,, \tag{9.18}$$

with $t_0 = s_0/c_0$ and $t_w = t_0/6$ where $s_0 = 40\,\text{m}$. The solution was probed in the three different points labeled 1 to 3 in Figure 9.4 and these time dependent vectors were used to estimate the order of convergence. The result is presented in Table 9.1. The convergence estimates in this table have been calculated using

$$p_{max} = \frac{ln(\frac{max(abs(Ez_{2h}-Ez_h))}{max(abs(Ez_{4h}-Ez_{2h}))})}{ln(rf)}\,, \tag{9.19}$$

and equivalently for p_{mean}. The vectors Ez_h, Ez_{2h} and Ez_{4h} contains the probed electric field from one of the points. The grid refinement factor, rf, was two in these experiments.

Type of mean value	μ_r at interface	p_{max}			p_{mean}		
		1	2	3	1	2	3
Arithmetic	2.5	1.72	2.04	1.53	1.56	1.94	1.44
Geometric	2	1.93	2.08	1.75	1.74	2.06	1.59
Harmonic	1.6	2.15	2.14	2.03	2.10	2.15	1.95

Table 9.1. Convergence estimates for a diamagnetic square with $\mu_r = 4$ in 2D. Estimates have been calculated for three different spatial locations using (9.19).

It is obvious from the results presented in Table 9.1 that a harmonic mean value is the proper choice.

9.2.9 Numerical verification in three dimensions

To further verify the propositions in Sections 9.2.2 we have performed experiments on a dielectric cube in 3D. When an electric field component on an edge of the cube is updated, we set our discrete ϵ-value to be $\epsilon_0(\epsilon_r/4+3/4)$ because only one fourth of S (see Section 9.2.5) lies inside the cube and $\epsilon_r \equiv 1$ in vacuum. A magnetic field component can never be located on the edge of the cube. The setup of this numerical experiment is very similar to the 2D experiment illustrated in Figure 9.4. On the coarsest level we have $N_x = N_y = N_z = 20$ and $\Delta x = \Delta y = \Delta z = 1\,\mathrm{m}$. We use $CFL = \sqrt{3}/2$ and take 160 time steps. A twelve cells thick PML layer is used as absorbing boundary condition. We use the same Gaussian pulse as in the 2D experiment. In this case it travels in the positive x-direction and the electric field is polarized in the z-direction. The region containing a simple material is a cube with a side of six meters situated in the center at the computational domain. All three sample points are located at $y = z = 10$. Their x-values are: $x = 5$, $x = 10$ and $x = 15$. Again, we use a refinement factor (rf) of two. We perform two refinements and estimate the order of convergence using (9.19).

Estimate	p_{max}			p_{mean}		
Point	1	2	3	1	2	3
$\epsilon_r = \mu_r = 1$	1.96	1.99	2.06	1.99	1.98	2.00
$\epsilon_r = 5$	2.07	1.86	1.86	2.06	2.00	1.90
$\mu_r = 4$	1.91	2.04	2.07	1.78	2.05	1.93

Table 9.2. Convergence estimates for a cube in 3D.

The results are given in Table 9.2. In order to verify that our procedure to estimate the order of convergence is correct, we have included the vacuum case in Table 9.2. The results in Table 9.2 clearly indicate that we have second-order accuracy for all cases.

9.2.10 Subcell model interpretation.

Consider the situation depicted in Figure 9.5. The H_z components are normal to the interface and hence discontinuous because μ is discontinuous. A natural question to ask is whether our discrete H_z component at the interface represents the value in vacuum, the value in the material or an average thereof? When we update an E_y component which is located at the interface (see the left part of Figure 9.5), we consider the H_z component value to represent a constant value along a line of length Δz. This is in contradiction of the discontinuity of the H_z component.

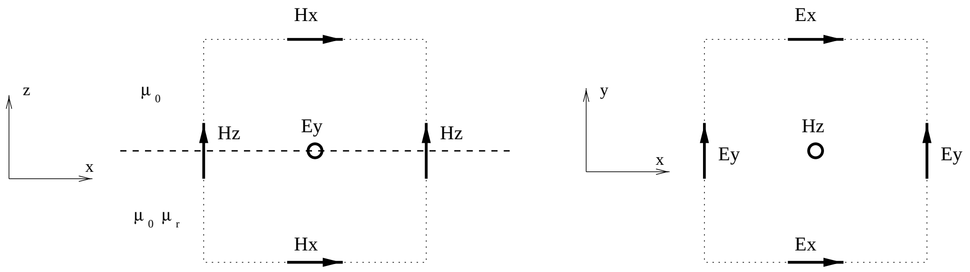


Figure 9.5. A material interface with discontinuous μ. The interface coincides with the components in the Yee grid.

One way to get around this dilemma is to introduce two discrete representations of H_z at the interface. We let H_{zV} represent the value in vacuum and H_{zM} the value in the material. Let these two variables be updated using

$$\mu_0 \frac{H_{zV}^{n+\frac{1}{2}} - H_{zV}^{n-\frac{1}{2}}}{\Delta t} = (\nabla \times E)^n \qquad \text{and} \qquad \mu_r \mu_0 \frac{H_{zM}^{n+\frac{1}{2}} - H_{zM}^{n-\frac{1}{2}}}{\Delta t} = (\nabla \times E)^n \,. \quad (9.20)$$

The update of an E_y component in (4.7) can be written as

$$\begin{aligned} \epsilon \Delta x \Delta z \left(\frac{E_y|_{i,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n}}{\Delta t} \right) &= \left[H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right] \Delta z \\ &+ \left[H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right] \Delta x \,. \end{aligned} \quad (9.21)$$

This is the formulation we get if we use an integral formulation derivation of the FD-TD scheme, see Section 3.6.8 in [Taf00] for a more complete description. In the right hand side of (9.21) we have two terms of the type $\Delta z H_z$. If the E_y component is located at an interface we replace these terms with $(H_{zV} + H_{zM})\Delta z/2$. If we isolate $H_{zV}^{n+1/2}$ and $H_{zM}^{n+1/2}$ in (9.20), and then add these two equations we get

$$\frac{1}{2}(H_{zV}^{n+\frac{1}{2}} + H_{zM}^{n+\frac{1}{2}}) = \frac{1}{2}(H_{zV}^{n-\frac{1}{2}} + H_{zM}^{n-\frac{1}{2}}) + (1 + \frac{1}{\mu_r})\frac{\Delta t}{\mu_0}(\nabla \times E)^n \,. \quad (9.22)$$

If we now let $H_z = \frac{1}{2}(H_{zV} + H_{zM})$ we see that this subcell model is equivalent with using a harmonic mean value for μ.

9.3 Arbitrary interfaces

9.3.1 Introduction

We now turn our attention to the more general case of interfaces that do not coincide with the field components of the Yee grid. We showed in Chapter 7 that staircasing of a curved material interface destroys the second-order accuracy of the FD-TD method. We will here show that it is possible to restore second-order accuracy for a circular cylinder with $\epsilon_r > 1$ without increasing the workload during the timestepping procedure.

9.3.2 Alternative methods

It has been showed [DDH99, DHD01, TDH00] that it is possible to get a second-order accurate scheme by widening the stencil. The drawback of this approach is that it will be very cumbersome to implement in three dimensions. A nice feature with this method is that stability is carefully studied. A proof of stability is given in [DDH99] for the one dimensional case.

Another possible approach would be to use subgridding close to the curved boundaries. However in order to get second-order accuracy, it would be necessary to have cell sizes in the subgridded domain that are proportional to the square of the cell size of the FD-TD grid.

We have developed a technique based on wavelet projection [AELR99]. This technique is able to catch the influence of finer geometric details on a coarse grid. This was demonstrated for the Helmholtz equation in [AELR99]. However, the same technique can be extended to create subcell models for the Maxwell equations.

9.3.3 Our method

Our method consists of two steps. First, we replace the discontinuous material function, with a continuous function. We refer to this a regularization. Second, we discretize with the standard FD-TD method. Both these steps introduce errors. The error introduced by a finite difference approximation is considerably smaller if the coefficient in the partial differential equation is continuous. Hence, by introducing the error in step 1, we decrease the error in step 2. Our method is based on balancing these errors. It will lead to only minor changes in the implementation of an FD-TD code and stability will be automatically retained.

9.3.4 Regularization

We will here exemplify with the TM_z polarization of the Maxwell equations and discontinuous ϵ.

Regularization means that we replace the discontinuous function for $\epsilon_r(\bar{x})$ with a continuous function, which we will refer to as $\tilde{\epsilon}_r(\bar{x})$. The functions $\epsilon_r(\bar{x})$ and $\tilde{\epsilon}_r(\bar{x})$

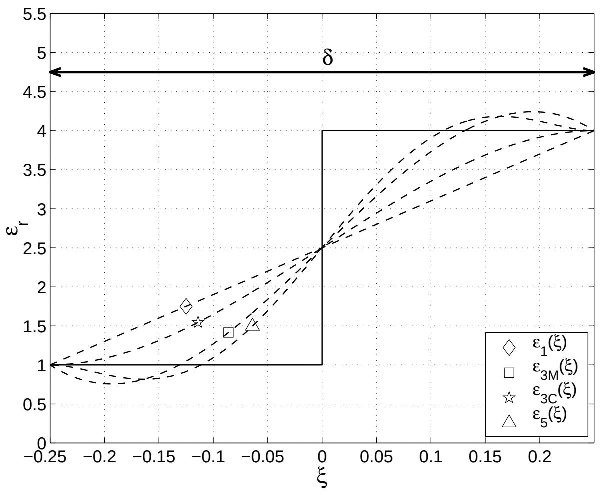


Figure 9.6. Several different possibilities for the transition region function $\tilde{\epsilon}_r(\xi)$.

only differ in a neighborhood of the interface, which we call the transition region. The width of this region is δ. This is illustrated in Figure 9.6 where ξ represents the distance to the material interface. The different $\tilde{\epsilon}_r(\xi)$ functions in Figure 9.6 will be described in Section 9.3.6 on page 128.

We have two choices to make when we regularize:

1. How thick should the transition region be? In other words, what should the value of δ/Δ be? (Δ is the grid spacing.)

2. How shall we design $\tilde{\epsilon}_r(\xi)$?

The second item above is addressed in Section 9.3.6.

9.3.5 Statement of result

We will show that the regularization works well for a wide range of different transition functions and transition region thicknesses.

Result 9.1. *If the transition region is thin, we achieve the best result with a linear transition function.*

Result 9.2. *If the transition region is thick, we achieve the best result with a transition functions that have the two first moments equal to zero.*

A thin transition region has a thickness of approximately one Δ (the cell size), while a thick transition region has the thickness of several cell sizes. Moments are defined in Definition 9.1 on page 128.

9.3.6 Numerical experiments in one dimension

We have performed a number of experiments in one dimension. For these studies we will use waves traveling in the y-direction with the electric field in the x-direction.

The distance between the actual interface and the location of a field component is referred to as grid offset and is denoted by θ.

As our first experiment we study the case where $\delta = \Delta y$, i.e. the thickness of the transition region is equal to the spatial discretization. We set $\epsilon_r = 4$ and let a Gaussian wave strike the material interface. On the coarsest grid level we have $N = 85$, $\Delta y = 0.5$ and take 160 time steps. We use $CFL = 0.5$, which gives us $\Delta t \approx 83.39\,\text{ns}$. The Gaussian pulse is defined by

$$f(t) = e^{-(t-t_0)^2/t_w^2}\,, \tag{9.23}$$

with $t_0 = s_0/c_0$ and $t_w = t_0/6$ where $s_0 = 10\,\text{m}$. The Huygens' surface that generates the Gaussian pulse is placed at $y = 20\,\text{m}$. The E_x component at $y = 20$ is defined as belonging to the scattered field region. Figure 9.7 displays the analytical solution and the coarsest numerical solution at $t \approx 100\,\text{ns}$. Here the interface between vacuum and the material is placed at $y = 30.25\,\text{m}$. This coincides with an H_z component. In this case the offset is $\theta = \Delta y/2 = 0.25\,\text{m}$ since we define the offset relative to the closest E_x component in vacuum.

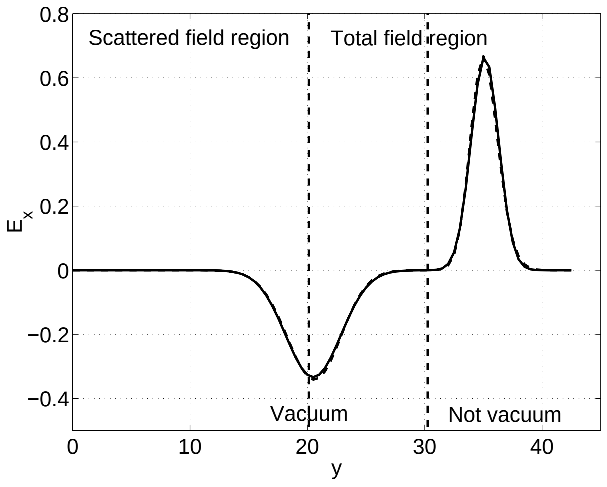


Figure 9.7. Analytical (solid) and numerical (dashed) solutions at $t \approx 100\,\text{ns}$.

Next we let θ sweep from zero to Δy in steps of $\Delta y/100$. For each θ we calculate the mean error of the E_x components at $t \approx 100\,\text{ns}$. The result is displayed in the left part of Figure 9.8. Here we perform two refinements, using a refinement factor of three. The location of the interface is kept fixed during the refinements, while the thickness of the transition region is scaled with the cell size, i.e. it is $\delta = \Delta y$. For each θ, we have calculated the order of convergence using the formula

$$p = \frac{ln(error_{3h}) - ln(error_h)}{ln(3)}, \tag{9.24}$$

where $error_{3h}$ is the mean error of E_x of the coarser solution and $error_h$ is the mean error of E_x of the finer solution in those points that also exist in the coarse grid. Since we have solutions on three grid refinement levels we get two convergence estimate for each θ. These estimates are displayed in the right part of Figure 9.8. We see that we have essentially second-order convergence independently of θ.

Having a transition region with thickness $\delta = \Delta y$ and using a piecewise linear continuous function for $\tilde{\epsilon}_r(y)$ is equivalent to calculating discrete ϵ_r values by integrating $\epsilon_r(y)$ over the surface S defined in Section 9.2.5. In one dimension, S is a segment of length Δy. It is nice to see that the same procedure that gives second-order accuracy for coinciding material interfaces works here to.

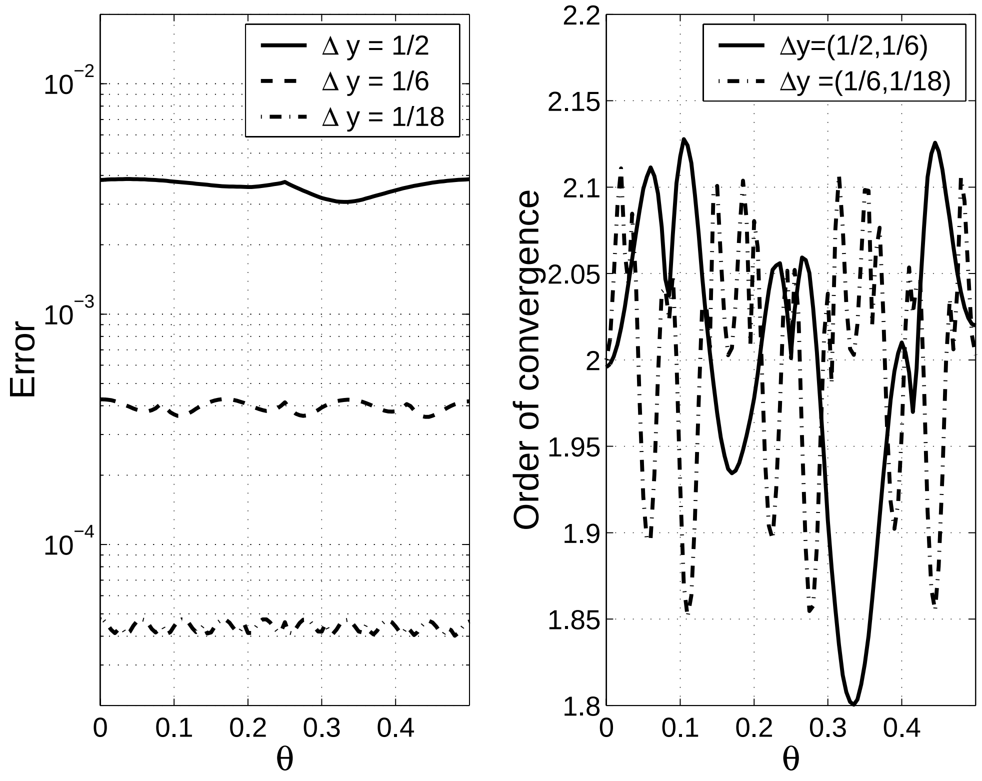


Figure 9.8. Convergence estimates as a function of grid offset for the linear transition function when $\delta = \Delta y$.

What we really need is a transition function $\tilde{\epsilon}_r(\xi)$ which yield second-order convergence over an interval in the relative transition layer thickness δ/Δ. In higher dimension we will apply finite-difference approximations along the Cartesian axes, while the normal of the interface may point in any direction. This means that the thickness as seen by the finite-difference approximation depends on the angle between the normal and the Cartesian axis. Furthermore, we will get varying grid offsets.

We will again study the same case, but we will now let δ vary from zero to $8\Delta y$. For each fixed value of δ, we let θ sweep from zero to Δy in steps of $\Delta y/100$. A calculation likes this takes several days to complete.

Figure 9.9 displays the result. The left part shows the maximum error for all different values of θ. The right part shows the mean value of the convergence estimates. Hence for $\delta/\Delta y = 1$ we get the results in the left part of Figure 9.9 by taking the maximum value over θ of the three curves in the left part of Figure 9.8. The results in the right part of Figure 9.9 for $\delta/\Delta y = 1$ are obtained by taking the mean values of the right part of Figure 9.8. We see that the error is minimized

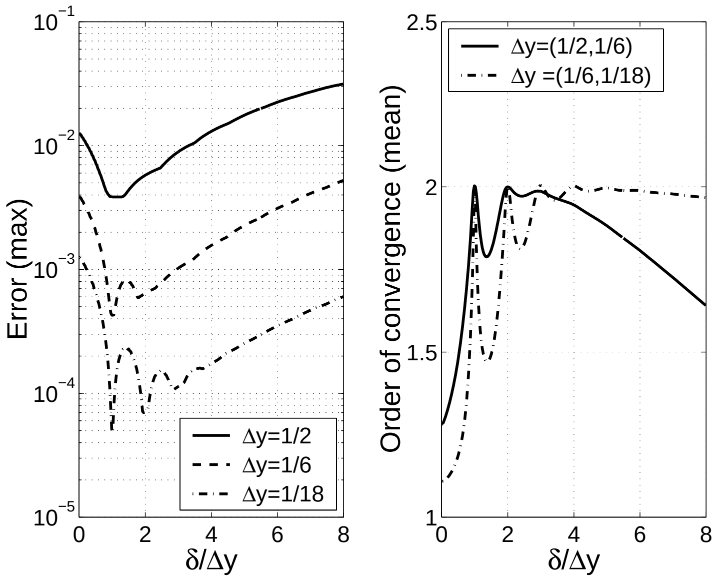


Figure 9.9. Errors and convergence estimates for the linear transition function.

when $\delta/\Delta y$ equals one, two or three and that we get second-order convergence in these cases. However, in between these integers we have considerably larger errors and for $\delta/\Delta y > 3$ the errors increases with $\delta/\Delta y$. What we would like to have is rather flat curves both for the error and the convergence estimate. To achieve this, we will try to use a different transition function.

To construct new transition functions we need a definition:

Definition 9.1. *The moments M_n are defined by*

$$M_n \equiv \int_{-\infty}^{\infty} (\epsilon_r(\xi) - \tilde{\epsilon}_r(\xi))\xi^n d\xi\,. \tag{9.25}$$

Since $\epsilon_r(\xi) = \tilde{\epsilon}_r(\xi)$ if $|\xi| > \delta/2$ we get

$$M_n = \int_{-\delta/2}^{\delta/2} (\epsilon_r(\xi) - \tilde{\epsilon}_r(\xi))\xi^n d\xi\,. \tag{9.26}$$

Some possible conditions that a transition function may fulfill are:

1. continuity at $\xi = \pm\delta/2$,
2. continuity of the derivative at $\xi = \pm\delta/2$ and
3. two vanishing moments, i.e. $M_0 = M_1 = 0$.

A motivation to why it should be favorable to fulfill condition 3 is given is Section 9.3.9.

In total, we have designed four alternative transition functions using the conditions above. They all fulfill condition 1. For $|\xi| < \delta/2$, they are:

- $\tilde{\epsilon}_5(\xi)= \frac{1}{2}(\epsilon_r + 1) + \frac{45}{16}(\epsilon_r - 1)\frac{\xi}{\delta} - \frac{25}{2}(\frac{\xi}{\delta})^3(\epsilon_r - 1) + 21(\frac{\xi}{\delta})^5(\epsilon_r - 1)\,.$
  This polynomial fulfills all three conditions.
- $\tilde{\epsilon}_{3C}(\xi)= \frac{1}{2}(\epsilon_r + 1) + \frac{3}{2}(\epsilon_r - 1)\frac{\xi}{\delta} - 2(\frac{\xi}{\delta})^3(\epsilon_r - 1)\,.$
  This polynomial fulfills condition 2, but not condition 3.
- $\tilde{\epsilon}_{3M}(\xi)= \frac{1}{2}(\epsilon_r + 1) + \frac{9}{4}(\epsilon_r - 1)\frac{\xi}{\delta} - 5(\frac{\xi}{\delta})^3(\epsilon_r - 1)\,.$
  This polynomial fulfills condition 3, but not condition 2.
- $\tilde{\epsilon}_1(\xi)= \frac{1}{2}(\epsilon_r + 1) + (\epsilon_r - 1)\frac{\xi}{\delta}\,.$
  This polynomial fulfills neither of the conditions 2 and 3. It does fulfill $M_0 = 0$ but not $M_1 = 0$. This is the linear transition function we have used so far.

For $\xi \geq \delta/2$ we have $\tilde{\epsilon}_r(\xi)= \epsilon_r$ and for $\xi \leq -\delta/2$ we have $\tilde{\epsilon}_r(\xi)= 1$ for all transition functions. The four functions in the list above are displayed in Figure 9.6. Result for $\tilde{\epsilon}_1(\xi)$ are displayed in Figure 9.9. Figures 9.10–9.12 show the result for the other three transition functions.

One important point to make is that this test was initially designed for $0 \leq \delta/\Delta \leq 4$. At the final time when we measure the error, we do not want to have any fields still lingering in the transition region. However, with $\delta/\Delta > 4$ this happens on the coarse grid. This is the reason for the increase in the convergence estimates in Figures 9.11 and 9.12 for $\delta/\Delta > 4$.

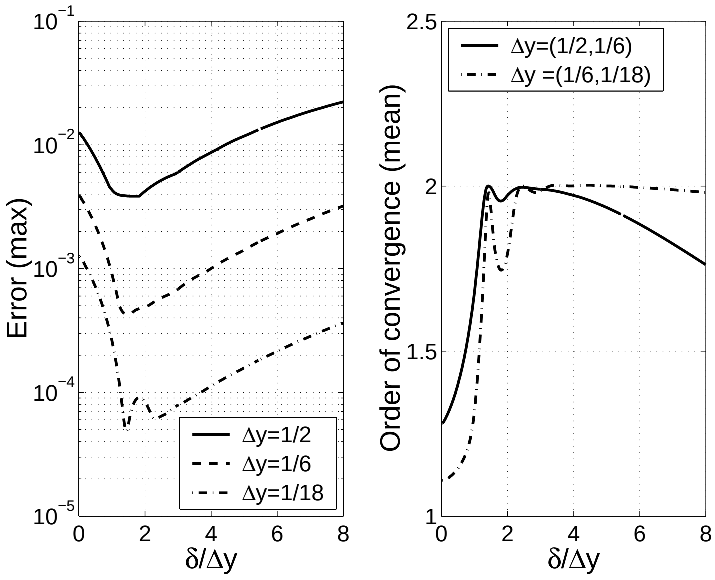


Figure 9.10. Errors and convergence estimates for the transition function $\tilde{\epsilon}_{3C}(\xi)$.

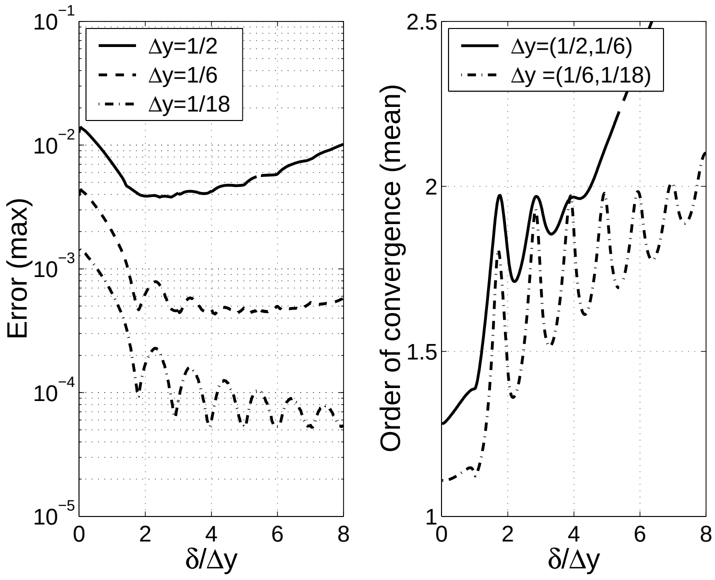


Figure 9.11. Errors and convergence estimates for the transition function $\tilde{\epsilon}_{3M}(\xi)$.

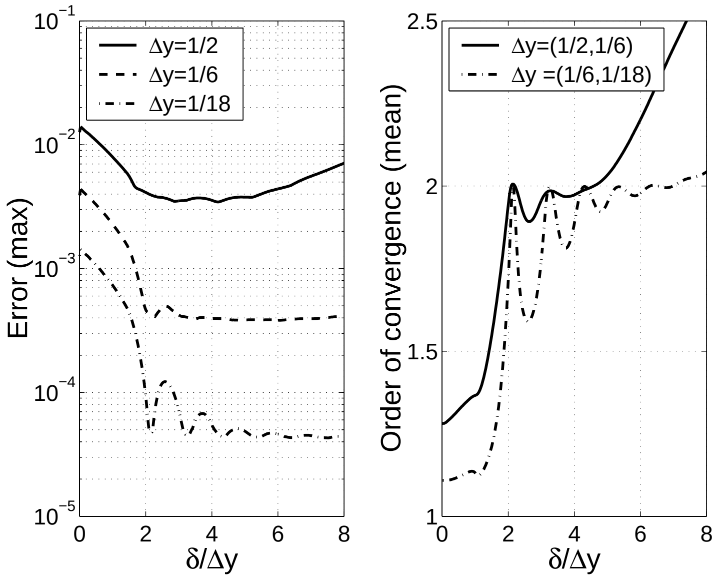


Figure 9.12. Errors and convergence estimates for the transition function $\tilde{\epsilon}_5(\xi)$.

A distinguishing difference between the results in Figures 9.11 and 9.12 and the results in Figures 9.9 and 9.10 is that the error is essentially constant for the two transition functions that fulfill condition 3 as δ/Δ increases from three to eight. This is exactly the feature we want, when we use transition functions in higher dimensions. (The increase in error for the coarsest solution in Figures 9.11 and 9.12 is due to the effect described in the previous paragraph.)

9.3.7 Numerical experiments in two dimensions

In order to verify the validity of our approach, we repeat one of the numerical experiments in Chapter 7, namely the circular dielectric cylinder with $\epsilon_r = 4$. We again use the transition functions defined on page 128.

The distance from the interface, ξ, is calculated along a line originating from the center of the circular cylinder. We set $\Delta = \Delta x = \Delta y = 1\,\text{m}$ on the coarsest grid and let the circular cylinder have a radius of two meters. This means that the largest possible value for δ/Δ is four. When refining the grid, we keep δ/Δ constant. Figure 9.13 displays the results for the linear transition function $\tilde{\epsilon}_1(\xi)$ for different values of δ/Δ. We note that the errors are larger for $\delta/\Delta = 4$ than for $\delta/\Delta = 1$ or 2. This is in accordance with the one dimensional results in Figure 9.9. We further note that the result for $\delta/\Delta = 0$ is worse than the FD-TD result in Figure 7.20. Hence the procedure to model the dielectric cylinder that was described in Section 7.8.4 is better than simply setting either $\epsilon_r = 1$ or $\epsilon_r = 4$ depending on whether the E_z component is located inside or outside the cylinder.

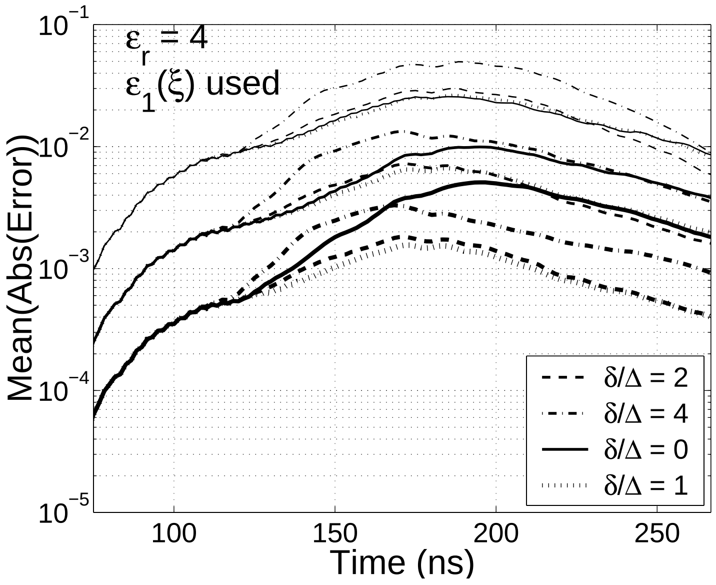


Figure 9.13. Errors for three refinement levels for the linear transition function $\tilde{\epsilon}_1(\xi)$ for four different values of δ/Δ.

Table 9.3 contains estimates of the order of convergence. These estimates have been calculated using the same procedure as in Section 7.8.4. We observe that we get second-order convergence for all values of δ/Δ, except $\delta/\Delta = 0$.

δ/Δ	1		2		4		0	
	coar	fine	coar	fine	coar	fine	coar	fine
Convergence estimate	2.05	2.12	2.09	2.00	1.89	2.07	1.50	1.10

Table 9.3. Estimates of the order of convergence for the linear transition function $\tilde{\epsilon}_1(\xi)$. The values in the columns labeled coar have been obtained by comparing errors on the coarse and medium grids and the values in the columns labeled fine have been obtained by comparing errors on the medium and fine grids.

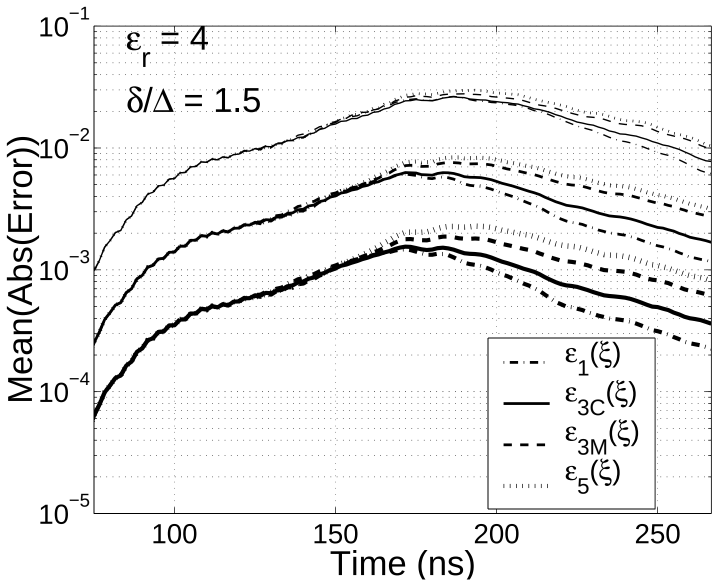


Figure 9.14. Errors for three refinement levels for four different transition functions.

Next we try different transition functions for a fixed value of δ/Δ. We choose $\delta/\Delta = 1.5$ and $\delta/\Delta = 4$. The results are displayed in Figures 9.15 and 9.14. In Figure 9.15, the transition functions $\tilde{\epsilon}_5(\xi)$ and $\tilde{\epsilon}_{3M}(\xi)$ give the best results, while in Figure 9.14 it is the transition function $\tilde{\epsilon}_1(\xi)$ and $\tilde{\epsilon}_{3C}(\xi)$ that give the best results. This is in accordance with the results in one dimension. Again, we see that having $M_0 = 0$ and $M_1 = 0$ is important when δ/Δ is large.

Transition function	$\tilde{\epsilon}_5(\xi)$		$\tilde{\epsilon}_{3M}(\xi)$		$\tilde{\epsilon}_{3C}(\xi)$		$\tilde{\epsilon}_1(\xi)$	
	coar	fine	coar	fine	coar	fine	coar	fine
Convergence estimate	2.08	1.93	2.14	1.97	2.02	2.09	1.89	2.07

Table 9.4. Estimates of the order of convergence for different transition functions when $\delta/\Delta = 4$. The values in the columns labeled coar have been obtained by comparing errors on the coarse and medium grids and the values in the columns labeled fine have been obtained by comparing errors on the medium and fine grids.

Transition function	$\tilde{\epsilon}_5(\xi)$		$\tilde{\epsilon}_{3M}(\xi)$		$\tilde{\epsilon}_{3C}(\xi)$		$\tilde{\epsilon}_1(\xi)$	
	coar	fine	coar	fine	coar	fine	coar	fine
Convergence estimate	1.88	1.92	1.94	2.05	2.11	2.07	2.22	2.11

Table 9.5. Estimates of the order of convergence for different transition functions when $\delta/\Delta = 1.5$. The values in the columns labeled coar have been obtained by comparing errors on the coarse and medium grids and the values in the columns labeled fine have been obtained by comparing errors on the medium and fine grids.

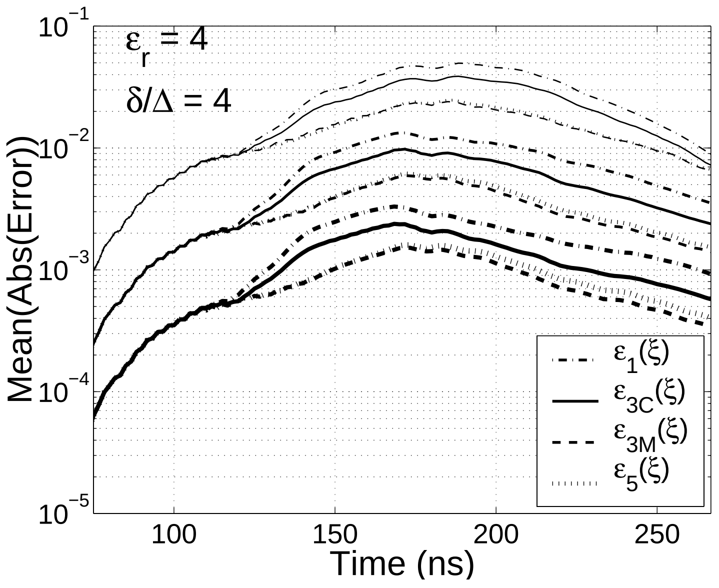


Figure 9.15. Errors for three refinement levels for four different transition functions.

Tables 9.4 and 9.5 contains estimates of the order of convergence. We see that all four transition functions essentially give second-order accuracy for both $\delta/\Delta = 1.5$ and $\delta/\Delta = 4$.

9.3.8 Extension to discontinuous μ and to 3D

We will now discuss how this procedure can be extended to inhomogeneous μ for the TM$_z$ polarization of the Maxwell equations. There is a fundamental difference between this case and discontinuous ϵ. In, the latter case all field components are continuous at the interface. With discontinuous μ we will have a discontinuous normal component of the magnetic field and a continuous tangential component of the magnetic field (cf. (3.16)).

We showed in Section 9.2 that normal and tangential components should be treated differently at an interface that coincides with the Yee grid. This results carries over to this situation. For tangential magnetic field components (H_t) we want to regularize μ, while for normal magnetic field components (H_n) we want to regularize $1/\mu$. However, we do not update H_t and H_n in the FD-TD method, we update H_x and H_y.

Two of the three TM$_z$ equations are, with $\sigma^* = 0$,

$$\begin{cases} \mu\frac{\partial H_x}{\partial t} &= -\frac{\partial E_z}{\partial y}\ , \\ \mu\frac{\partial H_y}{\partial t} &= \frac{\partial E_z}{\partial x}\ . \end{cases} \tag{9.27}$$

Consider the interface in Figure 9.16. If we rewrite (9.27) in tangential and normal

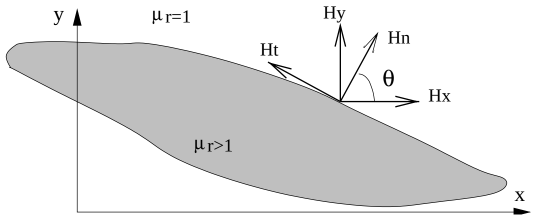


Figure 9.16. A discontinuity in μ.

components we get

$$\begin{cases} \mu\frac{\partial H_n}{\partial t} &= -c\frac{\partial E_z}{\partial y} + s\frac{\partial E_z}{\partial x}\ , \\ \mu\frac{\partial H_t}{\partial t} &= s\frac{\partial E_z}{\partial y} + c\frac{\partial E_z}{\partial x}\ . \end{cases} \tag{9.28}$$

where $c = \cos\theta$, $s = \sin\theta$, $H_n = cH_x + sH_y$ and $H_t = -sH_x + cH_y$. In (9.28) we regularize by replacing the discontinuous μ with regularized functions μ_H and μ_A and get

$$\begin{cases} \mu_H\frac{\partial(cH_x+sH_y)}{\partial t} &= -c\frac{\partial E_z}{\partial y} + s\frac{\partial E_z}{\partial x}\ , \\ \mu_A\frac{\partial(-sH_x+cH_y)}{\partial t} &= s\frac{\partial E_z}{\partial y} + c\frac{\partial E_z}{\partial x}\ . \end{cases} \tag{9.29}$$

If we rearrange this we get

$$\begin{cases} \mu_H\mu_A\frac{\partial H_x}{\partial t} &= (-\mu_A c^2 - \mu_H s^2)\frac{\partial E_z}{\partial y} + (\mu_A cs - \mu_H cs)\frac{\partial E_z}{\partial x}\ , \\ \mu_H\mu_A\frac{\partial H_y}{\partial t} &= (-\mu_A cs + \mu_H cs)\frac{\partial E_z}{\partial y} + (\mu_A c^2 + \mu_H s^2)\frac{\partial E_z}{\partial x}\ . \end{cases} \tag{9.30}$$

Comparing (9.30) with (9.27) we see that our procedure has introduced one new term in each equation. If we had used a non-staggered grid, this would not have been any problem. However, we are using the grid in Figure 4.1. When we update H_x it is straightforward to calculate $\partial Ez/\partial y$, but not $\partial Ez/\partial x$. One possible way to calculate $\partial Ez/\partial x$ at an H_x component would be to take the arithmetic average of $\partial Ez/\partial x$ at the four closest H_y components.

Once we have extended our regularization procedure to handle discontinuous field components in two dimension, there will be no principal obstacle left to overcome in order to extend it to three dimensions. Hence, if the approach outlined in the previous paragraph is successful, we will get a very cheap method for treating material interfaces in FD-TD.

9.3.9 Why moments?

We will here give justification to the use of condition 3 on page 128 when we constructed the transition functions. In other words, why is it important to have the two first moments zero? We will do this analysis in one dimension.

Consider (3.12) with $\sigma = 0$ and constant μ. In one dimension we get

$$\epsilon \frac{\partial^2 E_z}{\partial t^2} = \frac{1}{\mu} \frac{\partial^2 E_z}{\partial x^2} \, . \tag{9.31}$$

The left hand side in (9.31) is discontinuous but bounded. Hence $\partial^2 Ez/\partial x^2$ is also discontinuous and bounded. It follows that $\partial Ez/\partial x$ and E_z are continuous.

Now consider (3.10). With a time harmonic ansatz, the second equation in (3.10) becomes

$$i\omega\epsilon(x)\hat{E}_z(x) = \frac{\partial \hat{H}_y(x)}{\partial x} \, . \tag{9.32}$$

Integrating this, assuming $\hat{H}_y(0) = 0$, we get

$$\hat{H}_y(x) = \int_0^x i\omega\epsilon(\xi)\hat{E}_z(\xi)d\xi \, . \tag{9.33}$$

Here $\epsilon(\xi)$ is the piecewise constant function displayed in Figure 9.6. If we replace $\epsilon(\xi)$ with $\tilde{\epsilon}(\xi)$ in (9.33) we get

$$\tilde{H}_y(x) = \int_0^x i\omega\tilde{\epsilon}(\xi)\tilde{E}_z(\xi)d\xi \, . \tag{9.34}$$

We want $\tilde{H}_y(x) \approx \hat{H}_y(x)$ and $\tilde{E}_z(x) \approx \hat{E}_z(x)$. Subtracting (9.34) from (9.33) gives us

$$\begin{aligned}\hat{H}_y(x) - \tilde{H}_y(x) &= i\omega \int_0^x (\epsilon(\xi)\hat{E}_z(\xi) - \tilde{\epsilon}(\xi)\tilde{E}_z(\xi))d\xi \\ &= \int_0^x (\epsilon(\xi) - \tilde{\epsilon}(\xi))\hat{E}_z(\xi) + \int_0^x \tilde{\epsilon}(\xi)(\hat{E}_z(\xi) - \tilde{E}_z(\xi))d\xi \, .\end{aligned} \tag{9.35}$$

If we want the left hand side and the second term in the right hand side to be small, we need to make the first term in the right hand side small. If we expand $\hat{E}_z(\xi)$ around $\xi = 0$ the integral in the first term of the right hand side becomes

$$\begin{aligned}&\int_0^x (\epsilon(\xi) - \tilde{\epsilon}(\xi))(\hat{E}_z(0) + \xi\frac{\partial \hat{E}_z(0)}{\partial x} + O(\xi^2))d\xi \approx \\ &\hat{E}_z(0)\int_0^x (\epsilon(\xi) - \tilde{\epsilon}(\xi))d\xi + \frac{\partial \hat{E}_z(0)}{\partial x}\int_0^x (\epsilon(\xi) - \tilde{\epsilon}(\xi))\xi d\xi \, ,\end{aligned} \tag{9.36}$$

which equals zero if $M_0 = 0$ and $M_1 = 0$.

Higher moments have been used in other computational areas [Tor00, BM95]. However, since $\partial^2 Ez/\partial x^2$ is discontinuous it is not possible to add one more term in the expansion of $\hat{E}_z(\xi)$. It is thus doubtful if higher order moments would be an improvement in our context.

Chapter 10

Color Electromagnetics

The acronym CEM is sometimes said to stand for Color Electromagnetics. We here present some pretty color plots produced from the computations performed in the previous sections.

10.1 Rund

This section contains three color plots of the generic aircraft `Rund` used in Chapter 8. Figure 10.1 is a color version of Figure 8.3 on page 102. Figures 10.2 and 10.3 display surface currents.

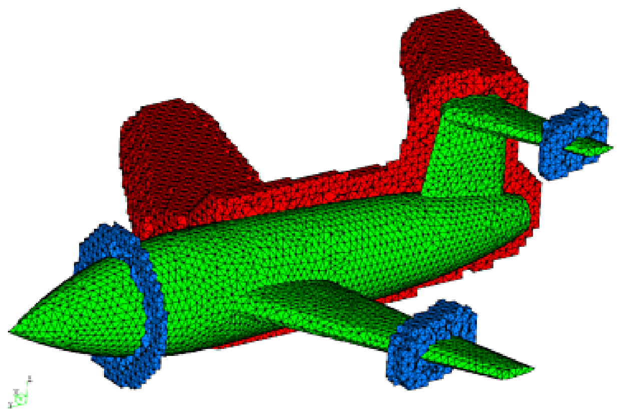


Figure 10.1. Part of the unstructured grid on and around the generic aircraft geometry `Rund`. The green surface is triangulated and the unstructured volume grid consists of tetrahedra. Parts of this grid are shown in red and blue. The structured grid continues outside the shown unstructured cells.

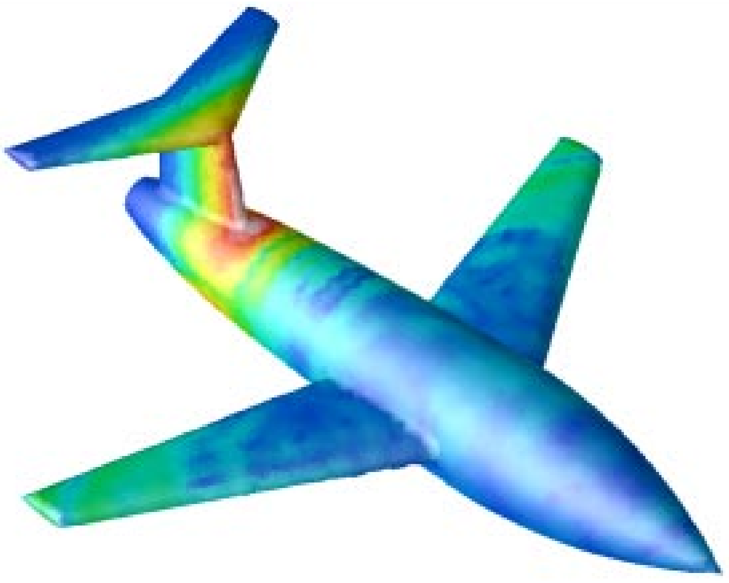


Figure 10.2. Surface currents on `Rund` for vertical polarization.

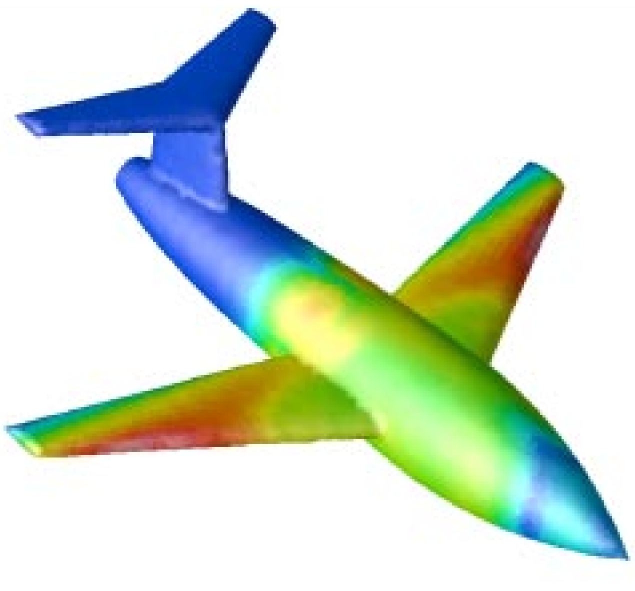


Figure 10.3. Surface currents on `Rund` for horizontal polarization.

10.2 Saab 2000

Here we present two color plots of the one-billion-cell computation in Chapter 6.

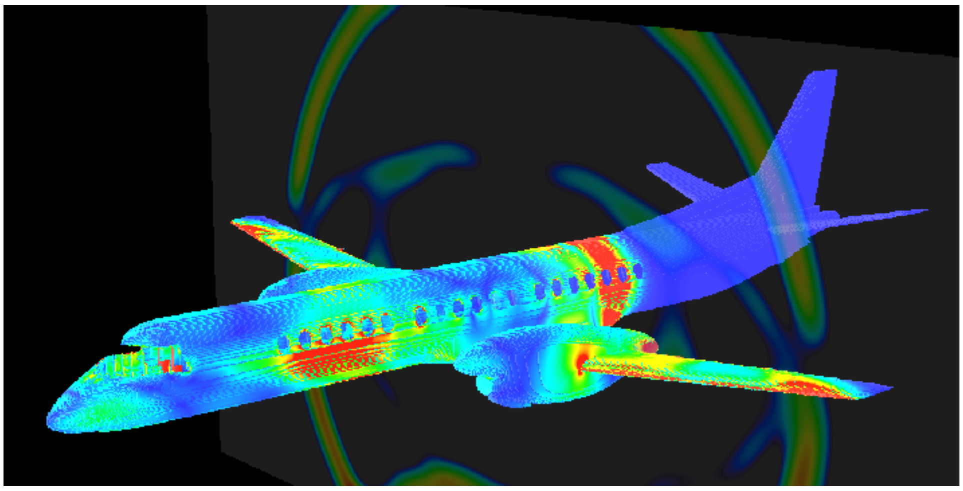


Figure 10.4. Surface currents on the exterior of the Saab 2000 aircraft 125 ns (1500 time steps) after a lightning stroke the nose. Also the magnitude of the magnetic field is shown on a cutting plane across the wings perpendicular to the fuselage.

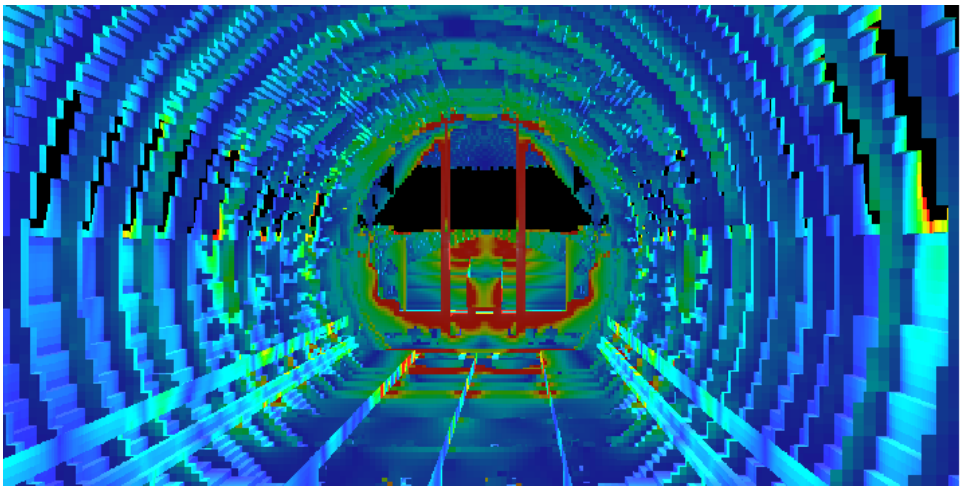


Figure 10.5. The interior of the SAAB 2000 aircraft. Surface currents are shown at the same time as in Figure 10.4. The view is from center of aircraft towards the cockpit. High surface currents are seen on the door pillar and the sill.

Appendix A

Abbreviations used in this thesis

ABC = Absorbing Boundary Condition

ABS3 = third-order Staggered Adams–Bashforth scheme

a.o. = arithmetic operation(s)

BC = Boundary Condition

BDF-2 = two stage Backward Difference Formula

CAD = Computer Aided Design

CEM = Computational ElectroMagnetics

CFD = Computational Fluid Dynamics

CFL = Courant-Friedrich-Levy (stability limit/condition)

CVS = Concurrent Versions System

CW = Continuous-Wave (near-to-far-field transform)

DFT = Discrete Fourier Transform

EMC = ElectroMagnetic Compatibility

FD-TD = Finite-Difference Time-Domain method/solver
(FD-TD is also referred to as the Yee scheme.)

FD-FE = Finite-Difference Finite-Element hybrid method/solver

FD-FV = Finite-Difference Finite-Volume hybrid method/solver

FE-FV = Finite-Element Finite-Volume hybrid method/solver

FE-TD or FE = Finite-Element Time-Domain method/solver

FEM = Finite Element Method (in this thesis we use FEM to refer to Finite-Elements the frequency domain. FE-TD or FE are used to refer to Finite-Elements the time domain.)

FFT = Fast Fourier Transform

FV-TD or FV = Finite-Volume Time-Domain method/solver

GEMS = General ElectroMagnetic Solvers

GTD = Geometric Theory of Diffraction

MoM = Method of Moments

MOT = Marching-On-in-Time

MPI = Message Passing Interface

NetCDF = network Common Data Form

PDC = Center for parallel computers (ParallellDator Centrum in Swedish)

PDE = Partial Differential Equation

PEC = Perfect Electric Conductor

PMC = Perfect Magnetic Conductor

PML = Perfectly Matched Layer

PO = Physical Optics

PSCI = Parallel and Scientific Computing Institute

PWTD = Plane-Wave Time-Domain

RCS = Radar Cross Section

SMP = Shared Memory Processor

TD = Time Domain

TE = Transverse Electric (mode of the Maxwell equations)

TM = Transverse Magnetic (mode of the Maxwell equations)

U-PML = Uniaxial Perfectly Matched Layer

UTD = Uniform Theory of Diffraction

Bibliography

[AELR99] U. Andersson, B. Engquist, G. Ledfelt, and O. Runborg. A contribution to wavelet-based subgrid modeling. *Applied and Computational Harmonic Analysis*, 7:151–164, 1999.

[AG97] S. Abarbanel and D. Gottlieb. A mathematical analysis of the PML method. *J. Comput. Phys.*, 134:357–363, 1997.

[AWV$^+$99] A. Akyurtlu, D. H. Werner, V. Veremey, D. J. Steich, and K. Aydin. Staircasing errors in FDTD at an air-dielectric interface. *IEEE Microwave Guided Wave Lett.*, 9(11):444–446, November 1999.

[Ber94] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(1):185–200, 1994.

[BK94] V. A. Borovikov and B. Y. Kinber. *Geometrical Theory of Diffraction*. IEE, UK, 1994.

[BM95] J. T. Beale and A. Majda. Vortex methods II: Higher order accuracy in two and three dimensions. *Mathematics of Computation*, 39:29–52, 1995.

[CAD] CADfix. `http://www.cadfix.com/`.

[Che89] D. K. Cheng. *Field and Wave Electromagnetics*. Addison Wesley, second edition, 1989.

[CRW93] R. Coifman, V. Rokhlin, and S. Wandzura. The fast multipole method for the wave equation: A pedestrian approach. *IEEE Trans. Antennas Propagat.*, 35(7), 1993.

[CW91] A. C. Cangellaris and D. B. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in FDTD simulations of electromagnetic phenomena. *IEEE Trans. Antennas Propagat.*, 39(10):1518–1525, October 1991.

[DDH99] A. Ditkowski, K. Dridi, and J. S. Hesthaven. Convergent Cartesian grid methods for Maxwells equations in complex geometries. *J. Comput. Phys.*, 1999. Submitted.

[DHD01] K. H. Dridi, J. S. Hesthaven, and A. Ditkowski. Staircase free finite-difference time-domain formulation for general materials in complex geometries. *IEEE Trans. Antennas Propagat.*, 2001. Accepted.

[DM97] S. Dey and R. Mittra. A locally conformal finite-difference time-domain (FDTD) algorithm for modeling three-dimensional perfectly conducting objects. *IEEE Microwave Guided Wave Lett.*, 7(9):273–275, September 1997.

[DSWB86] A. Dominek, H. Shamanski, R. Wood, and R. Barger. A useful test body. In *Proc. Antenna Measurement Techniques Assoc.*, volume 24, September 1986.

[DWB97] S. Dodson, S. P. Walker, and M. J. Bluck. Implicitness and stability of time domain integral equation scattering analysis. *Appl. Comput. Electromagn. Soc. J.*, 13(3):291–301, May 1997.

[Ede00] F. Edelvik. *Finite Volume Solvers for the Maxwell Equations in Time Domain.* Licentiate thesis No. 2000-005, Department of Information Technology, Uppsala University, October 2000.

[EL00] F. Edelvik and G. Ledfelt. Explicit hybrid time domain solver for the Maxwell equations in 3D. *J. Sci. Comput.*, 15(1), 2000.

[EM77] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comput.*, 31(139):629–651, 1977.

[Eng99] E. Engquist. Steering and visualization of electromagnetic simulations using globus. In B. Engquist, L. Johnsson, M. Hamill, and F. Short, editors, *Simulation and visualization on the grid*, Lecture Notes in Computational Science and Engineering. Parallelldatorcentrum, KTH, Springer, December 1999.

[ES00] F. Edelvik and B. Strand. Frequency dispersive materials for 3D hybrid solvers in time domain. In *Antenn 00, Nordic Antenna Symposium, Lund, Sweden*, pages 63–68, September 2000.

[ESM98] A. A. Ergin, B. Shanker, and E. Michielssen. Fast evaluation of three-dimensional transient wave fields using diagonal translation operators. *J. Comput. Phys.*, 146(1):157–180, 1998.

[Fem] Femlab, a Matlab toolbox. http://www.comsol.se/.

[Ged95] S. D. Gedney. Finite-difference time-domain analysis of microwave circuit devices on high performance vector/parallel computers. *IEEE Trans. Microwave Theory Tech.*, 43(10):2510–2514, October 1995.

[Ged96] S. D. Gedney. An anisotropic PML absorbing media for the FDTD simulation of fields in lossy and dispersive media. *Electromagnetics*, 16(4):399–415, 1996.

[GEM] General Electromagnetic Solvers (GEMS). http://www.psci.kth.se/Programs/GEMS/.

[GFD00] M. Ghrist, B. Fornberg, and T. A. Driscoll. Staggered time integrators for wave equations. *SIAM J. Num. Anal.*, 38(3):718–741, 2000.

[GK98] M. J. Grote and J. B. Keller. Nonreflecting boundary conditions for Maxwell's equations. *J. Comput. Phys.*, 139:327–342, 1998.

[GKO95] B. Gustafsson, H.-O. Kreiss, and J. Oliger. *Time Dependent Problems and Difference Methods*. Wiley-Interscience, 1995.

[GPS76] N. Gibbs, W. Poole, and P. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM J. Num. Anal.*, 13(2):236–250, April 1976.

[He97] L. He. FDTD - Advances in subsampling method, UPML and higher-order boundary conditions. Master's thesis, University of Kentucky, 1997.

[HFS] High frequency structure simulator (HFSS). http://www.ansoft.com.

[HSL$^+$00] T. Hirono, Y. Shibata, W. W. Lui, S. Seki, and Y. Yoshikuni. The second-order condition for the dielectric interface orthogonal to the Yee-latice axis in the FDTD scheme. *IEEE Microwave Guided Wave Lett.*, 10(9):359–361, September 2000.

[HW83] R. Holland and J. W. Williams. Total-field versus scattered-field finite-difference codes: A comparative assessment. *IEEE Trans. Nuclear Science*, NS-30(6):4583–4588, December 1983.

[HW99] C.-T. Hwang and R.-B. Wu. Treating late-time instability of hybrid finite-element/finite-difference time-domain method. *IEEE Trans. Antennas Propagat.*, 47(2):227–232, February 1999.

[IH98] T. Itoh and B. Houshmand, editors. *Time-Domain Methods for Microwave Structures: Analysis and Design*. IEEE Press, Piscataway, NJ, 1998.

[KL93] K. S. Kunz and R. J. Luebbers. *The Finite Difference Time Domain Method for Electromagnetics*. CRC Press, Boca Raton, FL, 1993.

[KLI97] D. Koh, H.-B. Lee, and T. Itoh. A hybrid full-wave analysis of via-hole grounds using finite-difference and finite-element time-domain methods. *IEEE Trans. Microwave Theory Tech.*, 45(12):2217–2223, 1997.

[KP74] R. G. Kouyoumjian and P. H. Pathak. A uniform theory of diffraction for an edge in a perfectly conducting surface. *Proc. IEEE*, 62(11):1448–1461, 1974.

[Lar] Large Scale FD-TD, a PSCI project. http://www.nada.kth.se/~ulfa/CEM.html.

[Lar00] E. Larsson. *Domain Decomposition and Preconditioned Iterative Methods for the Helmholtz Equation.* PhD thesis, Uppsala University, 2000.

[Led01] G. Ledfelt. *Hybrid Time-Domain Methods and Wire Models for Computational Electromagnetics.* TRITA-NA-0106, KTH, 2001.

[LLC97] J.-F. Lee, R. Lee, and A. Cangellaris. Time-domain finite-element methods. *IEEE Trans. Antennas Propagat.*, 45(3):430–442, March 1997.

[LM99] J.-Y Lee and N.-H. Myung. Locally tensor conformal FDTD method for modeling arbitrary dielectric surfaces. *Microwave Opt. Technol. Lett.*, 23(4):245–249, 1999.

[LMG98] J. E. Lumpp, Jr., S. K. Mazumdar, and S. D. Gedney. Performance modeling of the finite-difference time-domain method on parallel systems. *Applied Computational Electromagnetics Society Journal*, 13(2):147–159, 1998.

[LS95] J.-F. Lee and Z. Sacks. Whitney elements time domain (WETD) methods. *IEEE Trans. Magnet.*, 31(3):1325–1329, 1995.

[Mar98] T. Martin. An improved near- to far-zone transformation for the finite-difference time-domain method. *IEEE Trans. Antennas Propagat.*, 46(9):1263–1271, September 1998.

[Mar01] T. Martin. *Broadband Electromagnetic Scattering and Shielding Analysis Using the Finite Difference Time Domain Method.* PhD thesis, Department of Physics and Measurement Technology, Linköpings universitet, Sweden, March 2001.

[MBTK88] T. G. Moore, J. G. Blaschak, A. Taflove, and G. A. Kriegsmann. Theory and application of radiation boundary operators. *IEEE Trans. Antennas Propagat.*, 36(12):1797–1812, Decemeber 1988.

[MM98] A. Monorchio and R. Mittra. A hybrid finite-element finite-difference time-domain (FE/FDTD) technique for solving complex electromagnetic problems. *IEEE Microwave Guided Wave Lett.*, 8(2):93–95, 1998.

[Mon91] P. Monk. A comparison of finite element methods for the time dependent Maxwell equations. *Mathematical and Numerical Aspects of Wave Propagation Phenomena, G. Cohen et al. (eds), SIAM*, pages 80–88, 1991.

[Mon93] P. Monk. An analysis of Nedelec's method for the spatial discretization of Maxwell's equations. *J. Comput. Appl. Math.*, 47:101–121, 1993.

[MP94] P. Monk and K. Parrott. A dispersion analysis of finite element methods for Maxwell's equations. *SIAM J. Sci. Comput.*, 15(4):916–937, July 1994.

[MP00] T. Martin and L. Pettersson. Dispersion compensation for Huygens' sources and far-zone transformation in FDTD. *IEEE Trans. Antennas Propagat.*, 48(4):494–501, April 2000.

[MPI] The message passing interface (MPI) standard. `http://www-unix.mcs.anl.gov/mpi/`.

[Mur81] G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Trans. Electromagn. Compat.*, EMC-23(4):377–382, November 1981.

[Ned80] J. C. Nedelec. Mixed finite elements in R3. *Num. Math.*, 35(9):315–341, September 1980.

[Pet97] P. G. Petropoulos. Analysis of exponential time-differencing for FDTD in lossy dielectrics. *IEEE Trans. Antennas Propagat.*, 45(6):1054–1057, June 1997.

[PSC] Parallel and Scientific Computing Institute (PSCI). `http://www.psci.kth.se/`.

[Ram98] O. M. Ramahi. The concurrent complementary operators method for FDTD mesh truncation. *IEEE Trans. Antennas Propagat.*, 46(10):1475–1482, October 1998.

[RB00] T. Rylander and A. Bondeson. Stable FEM-FDTD hybrid method for Maxwell's equations. *Comput. Phys. Comm.*, 125:75–82, March 2000.

[Ril01] D. J. Riley. Transient finite-elements for computational electromagnetics: Hybridization with finite differences, modeling thin wires and thin slots, and parallel processing. In *17th Annual Review of Progress in Applied Computational Electromagnetics*, Monterey, CA, March 2001.

[RS90] B. P. Rynne and P. D. Smith. Stability of time marching algorithms for the electric field integral equations. *J. Electromagn. Waves Applicat.*, 12:1181–1205, 1990.

[RT97] D. J. Riley and C. D. Turner. VOLMAX: A solid-model-based, transient volumetric Maxwell solver using hybrid grids. *IEEE Antennas Propagat. Magazine*, 39(1):20–33, 1997.

[Ryl00] T. Rylander. *The Application of Edge Elements in Electromagnetics.* Licentiate thesis, Chalmers University of Technology, 2000.

[SC95] J. M. Song and W. C. Chew. Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering. *Microwave Opt. Technol. Lett.*, 10(14), 1995.

[SDPP98] S. Selleri, J. Y. Dauvignac, G. Pelosi, and C. Pichot. Comparison between FDTD and hybrid FDTD-FETD applied to scattering and antenna problems. *Microwave Opt. Technol. Lett.*, 18(4):247–250, 1998.

[SEAM00] B. Shanker, A. A. Ergin, K. Aygun, and E. Michielssen. Analysis of transient electromagnetic scattering phenomena using a two-level plane wave time-domain algorithm. *IEEE Trans. Antennas Propagat.*, 48(4), April 2000.

[SF90] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers 2nd Ed.* Cambridge University Press, Cambridge, 1990.

[SHM89] V. Shankar, W. Hall, and H. Mohammadian. A CFD-based finite-volume procedure for computational electromagnetics - interdisciplinary applications of CFD methods. *AIAA*, pages 551–564, 1989.

[Sul00] D. M. Sullivan. *Electromagnetic Simulation Using the FDTD Method.* IEEE Press, New York, 2000.

[Taf95] A. Taflove. *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House, Boston, MA, first edition, 1995.

[Taf98] A. Taflove, editor. *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House, Boston, MA, 1998.

[Taf00] A. Taflove. *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House, Boston, MA, second edition, 2000.

[TDH00] C. H. Teng, A. Ditkowski, and J. S. Hesthaven. Modeling dielectric interfaces in the FDTD-method: A comparative study. *IEEE Trans. Antennas Propagat.*, 2000. Submitted.

[Tor00] A.-K. Tornberg. *Interface Tracking Methods with Application to Multiphase Flows.* TRITA-NA-0010, KTH, 2000.

[VCK98] J. L. Volakis, A. Chatterjee, and L. C. Kempel. *Finite Element Method for Electromagnetics, Antennas, Microwave Circuits and Scattering Applications*. IEEE Press, 1998.

[Wan91] J. J. H. Wang. *Generalized Moment Methods in Electromagnetics, Formulation and Computer Solution of Integral Equations*. John Wiley & Sons Inc, 1991.

[WI95] R.-B. Wu and T. Itoh. Hybridizing FD-TD analysis with unconditionally stable FEM for objects of curved boundary. In *IEEE MTT-S Int. Microwave Symp.*, volume 2, pages 833–836, Orlando, FL, 1995.

[WI97] R.-B. Wu and T. Itoh. Hybrid finite-difference time-domain modeling of curved surfaces using tetrahedral edge elements. *IEEE Trans. Antennas Propagat.*, 45(8):1302–1309, 1997.

[Yee66] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propagat.*, 14(3):302–307, March 1966.

[Yeu99] M. S. Yeung. Application of the hybrid FDTD-FETD method to dispersive materials. *Microwave Opt. Technol. Lett.*, 23(4):238–242, 1999.

March 14, 2001

Errata for: Ulf Andersson, *Time-Domain Methods for the Maxwell Equations.* Doctoral Dissertation, Department of Numerical Analysis and Computer Science, Royal Institute of Technology. February 2001.

page 16: last line: erase the word system

page 26: In (4.13) $\frac{\Delta t}{\mu}$ should be replaced with $\frac{\Delta x \Delta t}{\Delta y \Delta z \mu}$.

page 55: After (7.1) add: where d is the thickness of the PML layer and ρ is the distance to the interface between PML and the computational domain.

page 59: In (7.16) add $k \neq i$

page 63: In (7.36) the right hand side should be divided by A_j^d.

page 64: The left hand side in (7.41) should be $\Delta_{jk}^d \tilde{\mu}$.

page 105: Two lines above Table 8.1: $\phi = 0$ should be $\phi = 180$

page 119: In (9.15), the first occurrence of ϵ should be ϵ_I, and the second should be ϵ_{II}.

page 127: On line 10, likes should be like.

page 130: third line from the end: Section 7.8.4 should be Section 7.8.1.

page 108: seven lines from the bottom: $(0.5, -0.5, 0)$ should be $(1, -1, 0)$.

page 110: second line below Figure 8.11: half meter from the outer boundary should be half meter from the outer boundary of the transition layer.