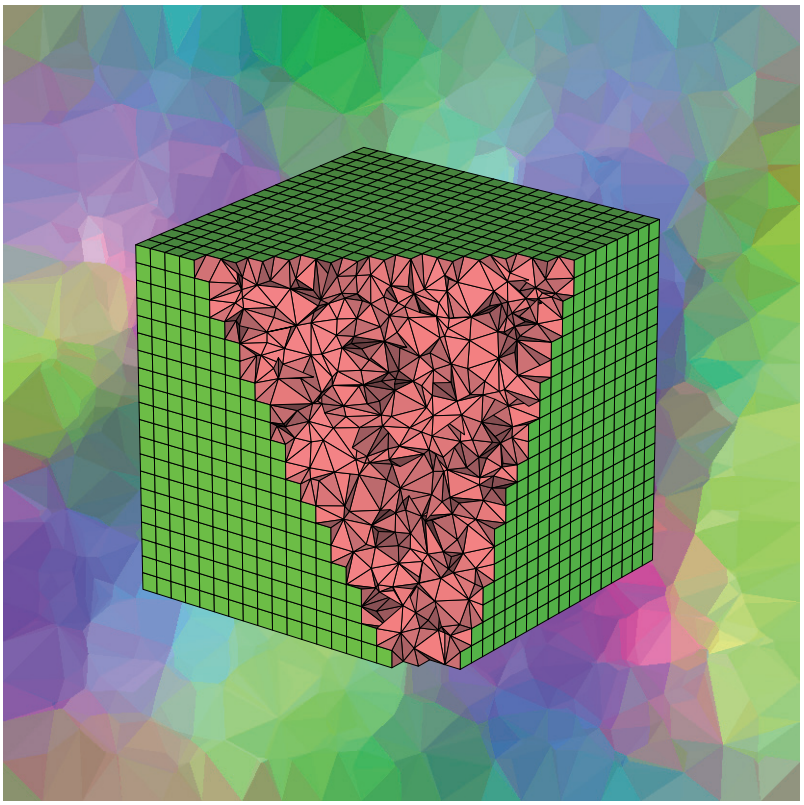
JYVÄSKYLÄ STUDIES IN COMPUTING
200

Jukka Räbinä

On a Numerical Solution of the Maxwell Equations by Discrete Exterior Calculus



JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 200

Jukka Räbinä

On a Numerical Solution of the Maxwell Equations by Discrete Exterior Calculus

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 1
marraskuun 26. päivänä 2014 kello 14.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 1, on November 26, 2014 at 14 o'clock.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2014

On a Numerical Solution of the Maxwell Equations by Discrete Exterior Calculus

JYVÄSKYLÄ STUDIES IN COMPUTING 200

Jukka Räbinä

On a Numerical Solution of the Maxwell Equations by Discrete Exterior Calculus

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2014

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-5951-7
ISBN 978-951-39-5951-7 (PDF)

ISBN 978-951-39-5950-0 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2014

ABSTRACT

Räbinä, Jukka
On a numerical solution of the Maxwell equations by discrete exterior calculus
Jyväskylä: University of Jyväskylä, 2014, 142 p.
(Jyväskylä Studies in Computing
ISSN 1456-5390; 200)
ISBN 978-951-39-5950-0 (nid.)
ISBN 978-951-39-5951-7 (PDF)
Finnish summary
Diss.

This study considers a numerical solution method based on discrete exterior calculus (DEC). The thesis concentrates on electromagnetic waves, meaning that the mathematical model is given by the Maxwell equations. The DEC offers a spatial discretization for the three-dimensional Maxwell problems. By applying the leapfrog style time discretization, we obtain a time-dependent simulation scheme, where the wave propagation can be tracked forward-in-time.

We customize the DEC framework for harmonic wave problems. The harmonic leapfrog equations produce an exact time discretization scheme for time-harmonic problems. The spatial correction is carried out by modifying the Hodge operator, which is the key factor in the DEC discretization. Using the spatially harmonic assumption, the harmonic Hodge operator is derived by minimizing the discretization error by the least squares method. The numerical experiments show significant improvement of the simulation efficiency compared to the Yee scheme. Further, we improve the time discretization by introducing a new non-uniform leapfrog scheme, where the time step size can be varied inside the domain. The energy conservation properties are verified by numerical experiments. The non-uniform leapfrog method offers a significant improvement of the simulation efficiency compared to the uniform leapfrog method.

Alternative iteration methods are introduced to solve time-periodic problems. The alternative methods are based on the controllability approach, where the quadratic cost function is minimized by preconditioned conjugate gradient algorithm. The controllability method increases the speed of convergence especially with tasks, where the wave is trapped inside the domain. We compare the DEC implementation to the well known scattering simulation method called discrete-dipole approximation (DDA). The comparison shows that a simplified and optimized DEC implementation could be a very competitive method for solving scattering problems.

Keywords: discrete exterior calculus, electromagnetism, the Maxwell equations, mesh generation, Voronoi diagram, discrete Hodge, harmonic wave, leapfrog, non-uniform time stepping, exact controllability, scattering.

Author

Jukka Räbinä
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Tuomo Rossi
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Professor Dirk Pauly
Faculty of Mathematics
University of Duisburg-Essen, Campus Essen
Germany

Doctor Sanna Mönkölä
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Oliver Rheinbach
Department of Mathematical Information Technology
Technical University Bergakademie Freiberg
Germany

Professor Keijo Ruotsalainen
Department of Electrical Engineering
University of Oulu
Finland

Opponent

Professor Lauri Kettunen
Institute of Electromagnetics
Tampere University of Technology
Finland

ACKNOWLEDGEMENTS

This study has been a long term project. The work has included a lot of trial and error combined with a mix of satisfaction and frustration. My working tasks as a researcher have varied mostly between reading, writing, coding and thinking. Several times the project felt like a lonely attempt to do science. Thus, it is remarkable that the thesis is now complete and it is time to thank all the people who have made this possible. Tuomo Rossi is definitely the most important person behind the project. Tuomo is the one who suggested the topic in the first place and since that he seemed to have endless trust on my research. I am very thankful for his positive comments and for those many occasions when he visited me and asked how the project was progressing. My project would have ended several years ago without his understanding and encouragement.

I am greatly indebted to my other two supervisors. I would like to thank Sanna Mönkölä for her great advice during the last two years. She was the key factor in motivating me to get started with the scientific writing. I also kindly thank Dirk Pauly for his good guidance and memorable moments at Essen. Please continue your excellent work.

I want to acknowledge the University of Jyväskylä and the Academy of Finland (grants 259925 and 260076) for funding my PhD studies. I also want to express my thanks to the scattering group from the University of Helsinki. Especially, I am thankful for the great researcher Antti Penttilä, who allowed me to use his DDA results in this thesis.

Lastly, I would like thank my family and friends. I would not be anywhere without my mother, Raila, who always says I am smarter that she is. In my mind, her kindness is the greatest act of wisdom. I want to express my thanks to all my friends, especially Maiju and Nanna, for their positive attitude on this academic adventure. Last but not least, I thank my love, Johanna, for being there and offering her great advice, encouragement, kindness and love.

Jyväskylä, October 2014

Jukka Räbinä

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Simulation is a natural way of predicting an outcome real-world process, and it is important from many aspects. In psychology, the threat simulation theory predicates that the human brain performs virtual threat simulations unconsciously while dreaming. The biological function of dreaming is then to simulate threatening events in order to rehearse the perception of threats and to learn to avoid them. Almost everyone has participated in an evacuation due to a fire alarm simulation, which is an example of a scheduled disaster preparedness simulation to test and rehearse the actions to be taken in case of fire.

Computers have become important for simulating schemes in many contexts, since computer programs can repeat controlled events safely and at low a cost. Education and training simulations are currently used by driving schools, and flight simulators are employed to train aircraft pilots. The simulations can include realistic models of even catastrophic scenarios, which could not be rehearsed in practice. The simulations also have an entertainment purpose because video games are based on computer simulations, as are the special effects in movies. In industrial engineering the computer simulations are applied throughout the entire engineering process, including design, testing, safety engineering, and performance optimization. The scientific modeling is applied to both natural systems and human systems to gain insight into their functioning. Weather forecasts and climate change prognoses are outcomes of scientific modeling using observed input values.

The general purpose of computer simulations is to imitate the real-world phenomena by the outcome of a computer program. The process of numerical simulation is illustrated in Figure 1. To model the real-world phenomenon numerically, a mathematical model is first required. This model represents the key characteristics of the selected physical or abstract process. The mathematical model is solved using numerical solution methods, which are implemented in a computer program. The development of computing resources has been fast during the last few decades due to the needs of home entertainment and more realistic game environments. Based on these observations, Moore's law predicts that the processing speed and the memory capacity of computers doubles ap-

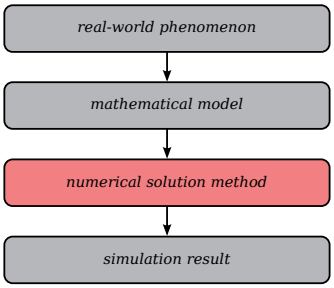


FIGURE 1 The four-step procedure of numerical simulations: The real-world phenomenon is first described as a mathematical model, which is then solved by a numerical solution method to obtain the simulation result.

proximately every two years. Problems that were computationally demanding a few decades ago can be solved significantly faster with today's computers. Thus, it is possible to improve the accuracy of the numerical simulations or solve more demanding problems. The increase of computer resources also provides opportunities to use new numerical solution methods. To take advantage of the increased computing resources, it is important that the size of the problem increases at the same pace as the available CPUs and memory.

This thesis concentrates on numerical solution methods from an implementation point of view. The considered numerical solution method is called the discrete exterior calculus (DEC), which is a naturally driven framework for solving differential equations. The thesis concentrates on electromagnetic waves and the mathematical model is given by the Maxwell equations. The convenient physical phenomena and corresponding simulation results are considered in simplified simulation examples. Mainly, the numerical experiments concentrate on scattering problems, but there is also glimpse of wave guide problems. The solution method can be applied in antenna design, light and radar scattering problems, electromagnetic brain studies, design of nano-structures and many other applications.

In this chapter, we begin with an introduction for the DEC framework. The mathematical model for electromagnetic wave equations is described by the Maxwell equations. The Maxwell equations are presented in Section 1.1 using the classical vector field formulation. In Section 1.2, the same equations are presented using differential forms. A short introduction for the DEC framework is given in Section 1.3, and a review of the essential literature is presented in Section 1.4.

A more detailed description for the DEC framework is given in the following chapters. The mesh formulation is provided in Chapter 2, and the spatial discretization of the Maxwell equations is described in Chapter 3. The framework is described in the bounded region and some of the most common boundary conditions and field formulations are introduced in Chapter 4. A time-dependent solution method is obtained by applying the leapfrog style time discretization method, which is discussed in Chapter 5. In physical applications, the discrete solution must usually be transformed into a different formulation and a few of

these transformation methods are introduced in Chapter 6. The rest of the chapters describe the main results of this thesis. In Chapter 7, we derive a non-uniform leapfrog time discretization scheme, where time step size can be varied inside the domain. In Chapter 8, we reformulate the essential terms of the spatial discretization using the assumption of harmonic solution. Controlled iteration strategies for time-periodic problems are discussed in Chapter 9. The resulting numerical solution method is verified by comparing it to another solution method in Chapter 10.

1.1 The classical Maxwell equations

The mathematical model for electromagnetic waves is described by the Maxwell equations (Cessenat, 1996). Even though the DEC framework is associated with differential forms, we describe the Maxwell equations first in classical vector field formulation. The derivation of the DEC framework is carried out mainly using the classical formulation because, in general, the vector field presentation might be better known by the readers. The traditional expression includes the vector fields **E**, **H**, **D** and **B**, which represent electric and magnetic fields and electric and magnetic flux densities, respectively. In this thesis, the variables in bold letters denote vectors in three-dimensional space $\mathbb{R}^3$. The classical presentation for symmetric Maxwell equations is written as

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} - \mathbf{J}^*, \tag{1}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \tag{2}$$

$$\nabla \cdot \mathbf{B} = \rho^*, \tag{3}$$

$$\nabla \cdot \mathbf{D} = \rho. \tag{4}$$

The vector fields $\mathbf{J}$ and $\mathbf{J}^*$ represent electric and magnetic current densities, and the symbols ρ and ρ^* are electric and magnetic charges, respectively. The relations between the vector fields are described by the constitutive relations. The constitutive relations are written as

$$\begin{aligned} \mathbf{D} &= \epsilon \mathbf{E}, \\ \mathbf{B} &= \mu \mathbf{H}, \end{aligned} \tag{5}$$

where ϵ and μ are either real valued functions or 3x3-tensors representing the permittivity and permeability of material, respectively. Absorbing materials are defined by the electric and magnetic conductivities σ and σ^*, which are also represented as 3x3-tensors. The equations for current densities inside an absorbing material are

$$\begin{aligned} \mathbf{J} &= \sigma \mathbf{E}, \\ \mathbf{J}^* &= \sigma^* \mathbf{H}. \end{aligned} \tag{6}$$

The classical Maxwell equations can also be expressed in integral formulation. The integral formulation is obtained by applying the Stokes' theorem to the classical Maxwell equations. Denoting an arbitrary surface by S and an arbitrary volume by V, the integral presentation of the Maxwell equations is written as

$$\begin{aligned}
\oint_{\partial S} \mathbf{E} \cdot d\mathbf{l} &= -\frac{\partial}{\partial t} \iint_S \mathbf{B} \cdot d\mathbf{a} - \iint_S \mathbf{J}^* \cdot d\mathbf{a}, \\
\oint_{\partial S} \mathbf{H} \cdot d\mathbf{l} &= \frac{\partial}{\partial t} \iint_S \mathbf{D} \cdot d\mathbf{a} + \iint_S \mathbf{J} \cdot d\mathbf{a}, \\
\oiint_{\partial V} \mathbf{B} \cdot d\mathbf{a} &= \iiint_V \rho^* dv, \\
\oiint_{\partial V} \mathbf{D} \cdot d\mathbf{a} &= \iiint_V \rho dv.
\end{aligned} \tag{7}$$

1.2 The Maxwell equations in terms of differential forms

The Maxwell equations have a nice expression in differential form calculus. In this expression, the vector fields are replaced by objects called differential forms. The differential forms are tailor made objects for integration over manifolds. More formally, a differential k-form is an expression that can be integrated over k-manifolds. Inspired by the integral formulation of the Maxwell equations, we begin by replacing the electric and magnetic fields $\mathbf{E}$ and $\mathbf{H}$ with one-forms $\tilde{E}$ and $\tilde{H}$. The electric and magnetic flux densities $\mathbf{D}$ and $\mathbf{B}$ are replaced by two-forms $\tilde{D}$ and $\tilde{B}$. In this thesis, the tilde symbol above the letter implies a differential form presentation. The differential forms have the coordinate expressions

$$\begin{aligned}
\tilde{E} &= E_x dx + E_y dy + E_z dz, \\
\tilde{H} &= H_x dx + H_y dy + H_z dz, \\
\tilde{D} &= D_x dy \wedge dz + D_y dz \wedge dx + D_z dx \wedge dy, \\
\tilde{B} &= B_x dy \wedge dz + B_y dz \wedge dx + B_z dx \wedge dy,
\end{aligned}$$

where $\mathbf{E} = (E_x, E_y, E_z)$, $\mathbf{H} = (H_x, H_y, H_z)$, $\mathbf{D} = (D_x, D_y, D_z)$ and $\mathbf{B} = (B_x, B_y, B_z)$ in Euclidean coordinates. The symbol $\wedge$ represents the wedge product, which is the product in an exterior algebra, mapping p- and q-forms into a $(p+q)$-form (for details, see Cartan, 2006; Flanders, 1989). We also replace the current densities and charges by two-forms $\tilde{J}$ and $\tilde{J}^*$ and three-forms $\tilde{\rho}$ and $\tilde{\rho}^*$, respectively, and get

$$\begin{aligned}
\tilde{J} &= J_x dy \wedge dz + J_y dz \wedge dx + J_z dx \wedge dy, \\
\tilde{J}^* &= J_x^* dy \wedge dz + J_y^* dz \wedge dx + J_z^* dx \wedge dy, \\
\tilde{\rho} &= \rho \; dx \wedge dy \wedge dz, \\
\tilde{\rho}^* &= \rho^* dx \wedge dy \wedge dz.
\end{aligned}$$

Now, the symmetric Maxwell equations (1)–(4) are written by differential forms as

$$d\tilde{E} = -\partial_t \tilde{B} - \tilde{J}^*, \tag{8}$$
$$d\tilde{H} = \partial_t \tilde{D} + \tilde{J}, \tag{9}$$
$$d\tilde{B} = \tilde{\rho}^*, \tag{10}$$
$$d\tilde{D} = \tilde{\rho}. \tag{11}$$

The symbol d represents the exterior derivative, which satisfies the generalized Stokes' theorem

$$\int_c d\tilde{\alpha} = \int_{\partial c} \tilde{\alpha}.$$

Here, c is an oriented k-dimensional manifold with boundary and ∂c is the boundary of c. The exterior derivatives of one-form and two-form imply the curl operator ($\nabla\times$) and divergence operator ($\nabla\cdot$), respectively. Applying the Stokes' theorem in Equations (8)–(11) gives the integral presentation for the Maxwell equations. By denoting a two-manifold by S and a three-manifold by V, we write

$$\begin{aligned}
\int_{\partial S} \tilde{E} &= -\partial_t \int_S \tilde{B} - \int_S \tilde{J}^*, \\
\int_{\partial S} \tilde{H} &= \partial_t \int_S \tilde{D} + \int_S \tilde{J}, \\
\int_{\partial V} \tilde{B} &= \int_V \tilde{\rho}^*, \\
\int_{\partial V} \tilde{D} &= \int_V \tilde{\rho}.
\end{aligned} \tag{12}$$

In differential calculus, the constitutive relations are defined by the concept of Hodge operator $\star$, which maps a differential k-form $\tilde{\alpha}$ to a differential $(n-k)$-form $\star\tilde{\alpha}$. Here, n is the dimension of the calculus; with Maxwell equations the dimension is $n = 3$. The constitutive relations are expressed using terms $\star\epsilon$ and $\star\mu$, as

$$\tilde{D} = \star\epsilon\tilde{E},$$
$$\tilde{B} = \star\mu\tilde{H}.$$

The equations for current densities are similarly presented by the terms $\star\sigma$ and $\star\sigma^*$, as

$$\tilde{J} = \star\sigma\tilde{E},$$
$$\tilde{J}^* = \star\sigma^*\tilde{H}.$$

Using these relations, the Maxwell equations can be expressed without variables $\tilde{D}$ and $\tilde{B}$. Next, we show that the Maxwell equations can be also expressed without the last two equations (10) and (11), if these equations are satisfied during the initial stage. The continuity of charge implies the following relation between current densities and charges

$$\partial_t\tilde{\rho}^* + d\tilde{J}^* = 0,$$
$$\partial_t\tilde{\rho} + d\tilde{J} = 0.$$

Taking the exterior derivatives from both sides of the Faraday and the Ampére equations (8) and (9) and applying the continuity of charge gives the relations

$$\partial_t d\tilde{B} = \partial_t \tilde{\rho}^*,$$
$$\partial_t d\tilde{D} = \partial_t \tilde{\rho}.$$

These are equal to the conservation of the Gauss laws (10) and (11) through the time integration. Thus, if the initial conditions obey the Gauss laws, then the full Maxwell system is included in the first two equations. We apply the constitutive relations and the equations for the current densities into the Ampére and Faraday equations and write the Maxwell equations as an initial value problem

$$\begin{aligned} d\tilde{E} &= -\star\mu\partial_t\tilde{H} - \star\sigma^*\tilde{H}, \\ d\tilde{H} &= \star\epsilon\partial_t\tilde{E} + \star\sigma\tilde{E}, \\ \tilde{E}(0) &= \tilde{E}_0, \qquad \tilde{H}(0) = \tilde{H}_0. \end{aligned}$$

1.3 Discrete exterior calculus in a nutshell

The discrete exterior calculus (DEC) provides the properties and calculus of differential forms in a natural way at the discretization stage. In this section, we give a short introduction to DEC. We apply the calculus on a regular grid and obtain the classical Yee scheme (Yee, 1966). This section is intended to be an introduction. For a more detailed description, see Chapter 3.

We assume regular grid **a)**, which consists of small cubes tiled next to each other (see Figure 2). A square surface, which is a cross section of neighboring cubes, is called a face. Each face is listed in a set $\{\mathcal{F}_i\}$. A line segment, which is a cross section of neighboring faces is called an edge, and the edge elements are listed in set $\{\mathcal{E}_j\}$. Each face and edge element is assigned an orientation. We introduce an incidence matrix d_1, which is defined by the relative orientations of the neighboring edges and faces. The incidence matrix has a non-zero value $(d_1)_{i,j} \neq 0$ only, if the edge $\mathcal{E}_j$ is included in the boundary of the face $\mathcal{F}_i$. The non-zero entries of the incidence matrix have either the value -1 or the value +1, depending on the relative orientation of the elements. If the edge element $\mathcal{E}_j$ is oriented in a counter-clockwise manner with respect to the face element $\mathcal{F}_i$, the matrix value is positive $(d_1)_{i,j} = 1$. Otherwise, it is negative $(d_1)_{i,j} = -1$.

We express the vector fields **E** and **B** as a discrete one-form E and discrete two-form B, respectively. The discrete one-form is a column vector, which has an element E_j assigned to each edge element $\mathcal{E}_j$. Similarly, the elements B_i of discrete two-form are assigned to each face element $\mathcal{F}_i$. More formally,

$$E_j := \int_{\mathcal{E}_j} \tilde{E} = \int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l},$$
$$B_i := \int_{\mathcal{F}_i} \tilde{B} = \int_{\mathcal{F}_i} \mathbf{B} \cdot d\mathbf{a}.$$

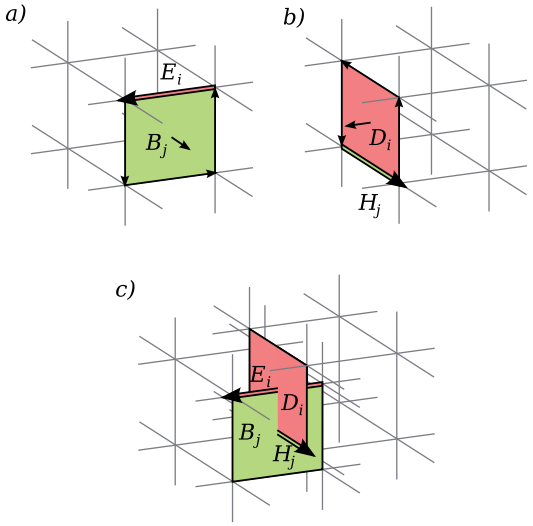


FIGURE 2 The Faraday law and the Ampére law are discretized on the separate grids **a)** and **b)**. The dual relation of the grids **c)** gives a natural estimate for the constitutive relations.

Based upon the integral formulation of the Maxwell equations (7) or (12), we get the discrete version of the Faraday law. Assuming $J^* = 0$, that is

$$\partial_t B_i = -\sum_{j=1}^{n} (d_1)_{i,j} E_j. \tag{13}$$

The Ampére law is discretized on another regular grid **b)**, as illustrated in Figure 2. This grid has an edge $\mathcal{E}_i^*$ corresponding to each face $\mathcal{F}_i$ of the original grid **a)**. Similarly, **b)** has a face $\mathcal{F}_j^*$ corresponding to each edge $\mathcal{E}_j$ of **a)**. The grid **b)** is called a dual of grid **a)**. The elements of the dual grid have a similar orientation to the corresponding primal elements. Then the incidence matrix of grid **b)** is actually the transpose d_1^T. We apply a discrete one-form H and discrete two-form D on the grid **b)** by

$$H_i := \int_{\mathcal{E}_i^*} \tilde{H} = \int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l},$$
$$D_j := \int_{\mathcal{F}_j^*} \tilde{D} = \int_{\mathcal{F}_j^*} \mathbf{D} \cdot d\mathbf{a}.$$

The discrete version of the Ampére law is obtained simply by applying the integral formulation of the Maxwell equations into discrete forms. If the density current is zero $J = 0$, we have

$$\partial_t D_j = \sum_{i=1}^{m} (d_1)_{i,j} H_i. \tag{14}$$

The grids **a)** and **b)** form a primal-dual pair **c)**, as illustrated in Figure 2. Each two-form value D_j or B_i can now be expressed by the corresponding one-form value E_j or H_i using the geometric properties of the corresponding elements

and the constitutive relations (5). Assuming the constant and real valued material parameters $\epsilon, \mu \in \mathbb{R}$, we get approximations

$$\frac{D_j}{|\mathcal{F}_j^*|} \approx \epsilon \frac{E_j}{|\mathcal{E}_j|}, \qquad \frac{B_i}{|\mathcal{F}_i|} \approx \mu \frac{H_i}{|\mathcal{E}_i^*|}.$$

The constitutive relations are now written in form $D = \star\epsilon E$ and $B = \star\mu H$, where the discrete Hodge operators $\star\epsilon$ and $\star\mu$ are diagonal matrices, with components

$$\star\epsilon_{j,j} = \epsilon \frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|}, \qquad \star\mu_{i,i} = \mu \frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|}.$$

We apply the constitutive relations with the discrete Faraday law (13) and discrete Ampére law (14). Then we get

$$\begin{aligned} \partial_t E &= (\star\epsilon)^{-1} d_1^T H, \\ \partial_t H &= -(\star\mu)^{-1} d_1 E. \end{aligned}$$

The two-forms B and D are now eliminated from the DEC framework, and the whole equations run on one-forms E and H. The leapfrog style time discretization is applied to obtain a forward-in-time simulation scheme. After defining electric field values at the half-integer time instances $E^{k-1/2}$ and the magnetic field values at integer time instances H^k, we obtain a leapfrog time discretization

$$\begin{aligned} E^{k+1/2} - E^{k-1/2} &= \Delta t (\star\epsilon)^{-1} d_1^T H^k, \\ H^{k+1} - H^k &= -\Delta t (\star\mu)^{-1} d_1 E^{k+1/2}. \end{aligned}$$

Starting from the given initial values $E^{-1/2}$ and H^0, this defines an iterative simulation method, where the propagation of electromagnetic waves is simulated numerically in a very natural manner. The method is stable, if the time step size Δt is small enough (Taflove and Brodwin, 1975). The presented model excludes the boundary conditions, which are needed to model physical problems.

1.4 History and previous work

The Finite-Difference Time-Domain (FDTD) method is one of the most successful numerical methods in the field of computational electromagnetics. The basic FDTD space grid and time-stepping algorithm trace back to the work of Yee (1966). Thus, the FDTD framework is sometimes called the Yee scheme. The acronym FDTD was originated by Taflove (1980). The FDTD method was first time applied with the wave scattering problems by Umashankar and Taflove (1982; 1983). In those studies, the near and far fields were computed for time-harmonic waves in two- and three-dimensional structures. Since the 1990's FDTD techniques have emerged as a primary means to model electromagnetic wave

interactions with material structures, and many scientific and engineering problems are included (Yee and Chen, 1997; Taflove and Hagness, 2005). The original FDTD algorithm is based on a regular space grid. The algorithm is easy to understand, and it has an exceptionally simple implementation for a full-wave solver. However, the structured grid design has its disadvantages, especially when approximating curved boundaries (Madsen, 1995). Applying the FDTD to irregular grids makes the implementation more complicated.

Several numerical algorithms, which allow the usage of unstructured grids, have been studied. For example, the finite element method (FEM) (Nédélec, 1980, 1986, 2001; Volakis et al., 1998; Jin, 2002) and the finite volume method (FVM) (Chung and Engquist, 2005) are competitive alternatives for solving electromagnetic problems numerically in unstructured grids. Another method is the finite integration technique (FIT), which was originated by Weiland (1977). The FIT provides a discrete reformulation of the Maxwell equations using their integral formulations. The resulting discretization scheme is applicable for computing, and it allows one to simulate electromagnetic problems with complex geometries. The FIT is based on the usage of integral balances and thus allows one to prove the stability and conservation properties of the discrete system. This enables the development of long-term stable numerical time integration schemes (Clemens and Weiland, 2001; Marklein, 2002).

In the past 15 years, many researchers have been working on discretizing the Maxwell equations in terms of differential forms. The language of differential forms and concepts of algebraic topology allow one to separate those equations, which are metric-free, arising from topology, from those, which are metric-dependent (Teixeira and Chew, 1999; Tonti, 1999; Tarhasaari et al., 1999). The discretization is then carried out using the concept of discrete differential forms. Despite the different background, the new formulation closely resembles the discrete formulation of the FIT, which has already been established for more than 35 years. In computational electromagnetics, this discretization method offers a simple generalization of the FDTD to unstructured complexes. Using the new approach, the Yee-like schemes were extended to unstructured grids for the first time by Bossavit and Kettunen (1999, 2000). In the literature, the pioneering work of computational electromagnetism in this research area includes Bossavit (1998, 2003), Hiptmair (2001, 2002), Mattiussi (2000), Gross and Kotiuga (2004) and Tonti (2001, 2002). The term discrete exterior calculus (DEC) was introduced in the PhD thesis of Hirani (2003). Since then, the term has appeared in several papers (Desbrun et al., 2005, 2008; Stern et al., 2009; Hirani et al., 2012).

The DEC framework is based on using both primal and dual meshes in the discretization scheme. The error sources are packed into the relation of the primal and dual elements, and this relation is described by the concept of the discrete Hodge operator. The preferred relation of primal and dual elements is developed during the pioneering work on the DEC method. In the beginning, many of the authors used a barycentric dual relation to determine the Hodge operator (Bossavit and Kettunen, 1999; Tarhasaari et al., 1999). Hiptmair (2002) was the first to use circumcentric dual instead of barycentric construction. The use of cir-

cumcentric dual was later shown to be essential for the theory of discrete exterior calculus (Desbrun et al., 2005). Circumcentric dual elements are orthogonal to corresponding primal elements, which is a highly desirable property. Then, the discrete Hodge star can be constructed as a diagonal matrix, which increases efficiency and gives tighter error bounds for the result. Since the Hodge operator is an essential element to ensure the accuracy of the DEC method, the development of an accurate Hodge star is still undergoing active research (Desbrun et al., 2008). The studies in this area have also moved in the direction of mesh generation and generalization of the dual construction (Hirani et al., 2012). A specific mesh optimization method is developed to improve the mesh quality in terms of the Hodge operator (Mullen et al., 2011). The resulting Hodge optimized triangulation (HOT) exploits the weighted circumcentric duality.

There are not many public software titles available on discrete exterior calculus. Perhaps the most advanced software is PyDEC (Bell and Hirani, 2012), which was developed by Bell and Hirani. PyDEC is a very general implementation, and it facilitates inquiry into both physical problems on manifolds as well as purely topological problems on abstract complexes. Both tetrahedral triangulations and regular cubic complexes are used. The DEC offers a wide degree of freedom to apply the method with different kind of meshes. Bossavit (2005) generalized the calculus for piecewise smooth manifolds, which provides more choice for the mesh generation. The DEC framework is also applied for fully unstructured spacetime by Stern et al. (2009). The construction of a fully unstructured spacetime mesh offers a great challenge since a four-dimensional Delaunay triangulation is needed.

The discretization of exterior calculus can be applied in many applications. In this thesis, we consider only the Maxwell equations and the electromagnetic wave problems. The acoustic, elastic and electromagnetic waves can be written under the same class of differential form equations, as discussed in previous studies (Picard, 2009; Picard and Freymond, 2012). Thus, the results of this thesis have wide scale of applications.

2 MESH FORMULATION

The Maxwell equations are presented in four-dimensional space, where the first three dimensions are devoted to spatial coordinates x, y, z, and the last dimension is reserved for the time coordinate t. The standard procedure is to separate the time dimension and consider the Maxwell system in a three-dimensional domain $\Omega \subset \mathbb{R}^3$. In numerical methods, the domain is traditionally discretized by a three-dimensional mesh, though the idea of entirely four-dimensional unstructured discretization is discussed by Stern et al. (2009).

Depending on the simulation method, the mesh is usually built using certain structural elements. For example, the classical Yee scheme relies on rectangular elements. The rectangular elements have their advantages, but there are also disadvantages, for example, in the approximation of curved or oblique boundaries. A system based on a regular grid is also anisotropic, having different properties with different orientations. The discrete differential forms offer a tool to relax the Yee-like scheme to work on more general meshes. Based upon the pioneering work of Bossavit and Kettunen, the Yee-like scheme can be fitted for tetrahedral elements (1999), for staggered cellular elements (2000) and for piecewise smooth manifolds (2005). The downside of the discretization method is the demand for an appropriate dual mesh, which is sometimes difficult to guarantee.

In this chapter, we describe a formulation for a three-dimensional mesh and its appropriate dual mesh. Our mesh formulation section starts by recalling the notion of manifolds in Section 2.1. Then we introduce a manifold-like cell complex with an orientation and the dual complex (see Sections 2.2, 2.3 and 2.4). In Section 2.5, the proposed formulation for mesh and dual mesh is presented as a special case of the complexes, where the body cell elements obey the polyhedral shapes. In Sections 2.6 and 2.7, we discuss the mesh generation and the optimization of the mesh. At the end of this chapter, we introduce several structured space filling grids, which can be applied and mixed together to obtain structured meshes for the DEC framework.

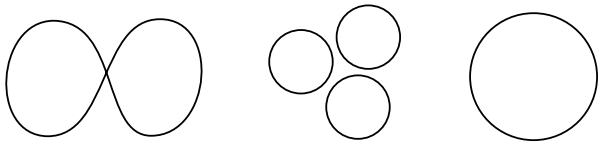


FIGURE 3 The figure 8 is not a manifold, but the group of circles is. The circle on the right is an example of a connected manifold.

2.1 Manifold

The tailor-made mathematical structure for differential forms is the differentiable n-dimensional manifold with boundary. In this section, we briefly recall the concept of manifolds. For further reading, see de Rham (1984); Munkres (1991); Lee (2009). An n-**manifold** (manifold of dimension n) is a topological space that is locally similar enough to the n-dimensional Euclidean space. In other words, each point of an n-manifold has a neighborhood that is homeomorphic to the open n-ball

$$B^n := \left\{ (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n} x_i^2 < 1 \right\}.$$

In the three-dimensional Euclidean space $\mathbb{R}^3$, we consider only manifolds of dimensions $n \in \{0, 1, 2, 3\}$. A zero-manifold is just a discrete space. The class of one-manifolds include lines and circles, but a figure eight is not a manifold due to its crossing point (see Figure 3). The two-manifolds and three-manifolds correspond to surfaces and volumes, respectively. In general, a manifold does not have to be connected. Thus, we recall the term **connected manifold**, meaning a connected space, which is also a manifold.

A **differentiable manifold** is a manifold, which is, near to each point, similar enough to a linear space. The concept of differentiable manifold is important, since it allows one to perform differential calculus. A **manifold with boundary** is a space containing both interior points and boundary points. Each **interior point** has a neighborhood homeomorphic to the open n-ball B^n. Every **boundary point** has a neighborhood homeomorphic to the open half n-ball

$$B^n_+ := \left\{ (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : \sum_{i=1}^{n} x_i^2 < 1 \text{ and } x_1 \geq 0 \right\}.$$

The homeomorphism must send each boundary point to a point with $x_1 = 0$.

2.2 Cell complex

The spatial discretization is based on the decomposition of the domain Ω into a cell complex (see Munkres, 1984; Hatcher, 2002). We start with the fundamental

object of the cell complex, which is called a **cell**. A general definition of a cell is based on connected differentiable manifolds with boundary. We give the following recursive definition for the p-cell, which is a cell of dimension p.

- A zero-cell is a single point in space.
- A one-cell is a differentiable path between two zero-cells.
- Inductively, a p-cell is a connected differentiable p-manifold with boundary, where the boundary is constructed by a finite set of $(p-1)$-cells.

The p-cell is denoted by c^p. A cell c^{p-1}, which is on the boundary of c^p, is called a **boundary cell** of c^p. Vice versa, c^p is called a **parent cell** of c^{p-1}. In $\mathbb{R}^3$, we consider zero-cells, one-cells, two-cells and three-cells, which are called **nodes**, **edges**, **faces** and **bodies**, respectively. We denote a p-volume of a cell by $|c^p|$, which corresponds to length, area and volume of one-, two- and three-cells, respectively. Examples of cells in $\mathbb{R}^3$ are illustrated in Figure 4.

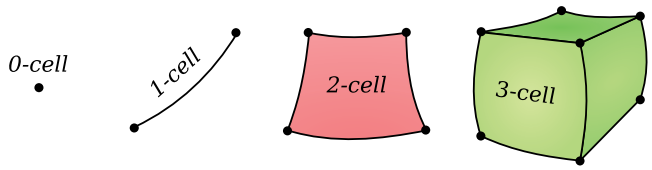


FIGURE 4 Examples of p-cells, $p \in \{0, 1, 2, 3\}$. The terms **node**, **edge**, **face** and **body** are used for these elements, respectively.

The definition of a cell requires that a cell has a boundary. Thus, a circle, which has no boundary points, is not an edge by the definition. Similarly, a sphere surface is not face because the sphere boundary has no boundary edges. Still, the sphere boundary can be constructed by two half-spherical faces, and the circle can be constructed by two half-circle edges. Because a boundary of a p-cell c^p is a $(p-1)$-manifold without a boundary, the cell c^p must have at least two boundary cells to complete the boundary.

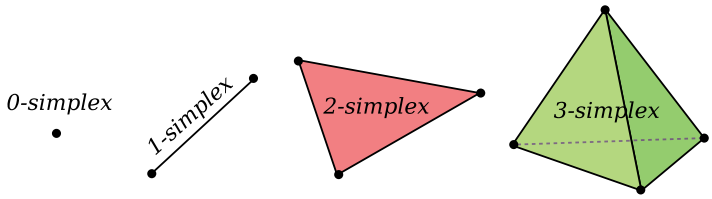


FIGURE 5 Examples of p-simplices, $p \in \{0, 1, 2, 3\}$.

One usually likes to operate on simplified cell elements to ease the calculus. Therefore, we present the idea of linear and convex cells. We define the **linear cell** as a cell, which is a linear manifold with linear boundary cells. The cell, which is a convex manifold, is called a **convex cell**. A **simplicial cell**, also called a simplex, is a special case of a linear and a convex cell. The p-simplex can be defined as a convex hull of linearly independent $p+1$ vertices. A simplicial edge

is basically a line segment between two boundary nodes. A simplicial face is a triangle, having exactly three boundary edges. A simplicial body is a tetrahedron with exactly four boundary faces. An example of each p-simplex is illustrated in Figure 5.

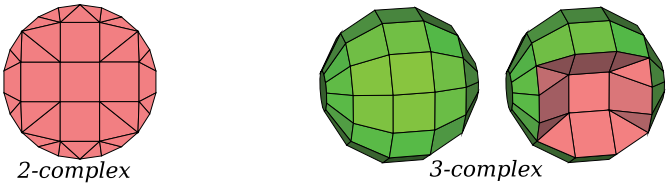


FIGURE 6 A two-complex and a three-complex are illustrated on the left and right, respectively. The existence of interior cells of the three-complex are shown by the cross-section on the very right.

The cell complex, or **complex**, is a certain collection of cells. Basically, the complex is used to decompose the domain Ω by a honeycomb of cells, where duplicate cells do not exist. Examples of complexes are illustrated in Figure 6. A k-complex, denoted by $\mathcal{K}$, is a complex of dimension k, and it is a finite set of cells such that

- The largest dimension of cells is k.
- Every boundary cell of a cell in $\mathcal{K}$ is also a cell in $\mathcal{K}$.
- The intersection of two cells in $\mathcal{K}$ is either a cell in $\mathcal{K}$ or it is the empty set.
- The intersection of two p-cells in $\mathcal{K}$ is not a p-cell.

The **underlying space** of a complex is the union of its cells, and it is denoted by $|\mathcal{K}|$. In general, the underlying space is not necessarily a manifold. A complex, where $|\mathcal{K}|$ is a manifold or a manifold with boundary, is called a **manifold-like complex**. Every cell of $\mathcal{K}$ that is included on the boundary of $|\mathcal{K}|$ is called **boundary cell** of $\mathcal{K}$. Otherwise, the cell is called the **interior cell** of $\mathcal{K}$.

In $\mathbb{R}^3$, the maximum dimension of a complex is three. A three-complex includes cells of four different dimensions. In this thesis, we denote the sets of nodes, edges, faces and bodies by $\mathcal{N}$, $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{B}$, respectively.

2.3 Oriented complex

In this section, we introduce the concept of orientation for the cells and the cell complexes. The concept orientation is necessary for the discretization of the Maxwell equations, and the oriented complex is a base for the mesh formulation. The orientation of a complex can be defined in several manners. The orientation is defined for an n-dimensional simplicial complex by Hirani (2003). Along the same lines, we explain the orientation for three-dimensional cell complex, assigning an orientation to edge and face elements.

A cell that has an orientation is called an **oriented cell**. The definition of the orientation is associated with the order of boundary cells. More precisely, if $\tau_0, \ldots, \tau_k$ are the boundary cells of a p-cell c^p, then the oriented p-cell c^p is the same cell together with a fixed order of $\tau_0, \ldots, \tau_k$. A node (zero-cell) has no boundary cells, which means by this definition that the node has only one possible orientation. An edge (one-cell) has two possible orientations, since the two boundary nodes can be ordered in two possible ways. We illustrate the orientation of an edge by an orientation vector pointing from the first node to the second, as shown on the left-hand side of Figure 7.

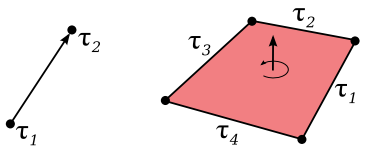


FIGURE 7 Oriented one-cell and oriented two-cell.

The boundary edges of a face element can be ordered in at least two possible ways because the face has at minimum two boundary edges. The number of edges can be larger, in which case the number of permutations is larger. We limit the number of orientations by accepting only those permutations that form a circulation around the face. In other words, the consecutive boundary edges in the ordered boundary list need to have a common boundary node element. By merging the congruent circulations (i.e. the circularions that rotate in the same direction), we reduce the number of orientations into two separate orientations. We illustrate the orientation of a face by a counter-clockwise orientation vector, as shown on the right-hand side of Figure 7. All the possible orientations of a body element are merged into one equivalence class. Thus, the order of the boundary face list has no meaning in practice.

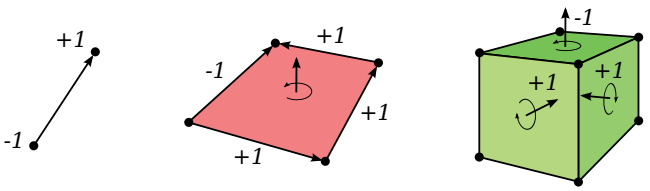


FIGURE 8 A boundary cell is said to have either a positive or negative induced orientation with respect to the parent cell. The positive orientation is denoted by +1 and the negative by -1.

The reason for defining the orientation of a cell is in the concept of **induced orientation**. The induced orientation declares the orientation of a boundary cell τ with respect to the parent cell c^p. The induced orientation is said to be either positive or negative. The induced orientation is positive (denoted by +1), if

- c^1: the edge orientation vector points at the boundary node τ.
- c^2: the boundary edge τ is oriented in a counter clockwise direction of c^2.

– c^3: the boundary face τ is oriented in inward direction of the body c^3.

Otherwise, the induced direction is negative (denoted by -1). Examples of induced orientations are illustrated in Figure 8.

An **oriented complex** is a complex where each cell is an oriented cell. Next, we recall an essential object of an oriented complex $\mathcal{K}$, called the **incidence matrix**. An incidence p-matrix, denoted by d_p, is a matrix that assigns an induced orientation for each pair of a $(p+1)$-cell and a p-cell. For the pair $\{c_i^{p+1}, c_j^p\}$, which is not a parent-boundary pair, the incidence matrix value is zero $(d_p)_{i,j} = 0$. The incidence matrix has non-zero values only if the c_j^p is a boundary cell of c_i^{p+1}. The non-zero entries of the incidence matrix have either the value -1 or the value +1, depending on the induced orientation of the cells. In a three-complex, the three incidence matrices d_0, d_1 and d_2 are qualified. The incidence zero-matrix d_0 is a map from nodes $\mathcal{N}$ to edges $\mathcal{E}$. Similarly, the incidence one-matrix d_1 is a map from edges $\mathcal{E}$ to faces $\mathcal{F}$, and the incidence two-matrix d_2 is a map from faces to $\mathcal{F}$ to bodies $\mathcal{B}$. Later in Chapter 3, these incidence matrices represent the discrete exterior derivatives in the formulation of the discrete exterior calculus.

2.4 Dual complex

The DEC framework requires the concept of **dual complex** associated with the (primal) cell complex. In the literature, there are at least two significantly different approaches to define the dual complex. The first approach is presented in Hirani (2003). In this approach, every p-cell is decomposed into a separate dual partition by the center points of the cells. This approach requires **well-centered cells**, where the center point of each primal cell has to be inside the cell.

The other approach, which is also applied in this thesis, is presented in Hiptmair (2002) and Bossavit (2005). This approach does not require a well-centered mesh. Still, every center point must lie inside the underlying space $|\mathcal{K}|$. Let c^p be an interior p-cell of a k-complex. We define the corresponding **dual cell** of c^p as a $(k-p)$-cell and denote it by $*c^p$. It is important to notice that $*c^p$ is not a p-cell but a $(k-p)$-cell, as illustrated in Figure 9. The following two conditions hold for the dual cells:

– If c^{p-1} is a boundary cell of c^p, then the dual cell $*c^{p-1}$ is a parent of $*c^p$.
– If c^{p+1} is a parent for c^p, then the dual cell $*c^{p+1}$ is a boundary for $*c^p$.

The dual cell of an interior cell is a cell similar to the primal cell, which is defined in Section 2.2. The dual cell of a boundary cell is slightly different. On the boundary, the dual cell is not closed because the boundary of the dual cell is incomplete. The boundary can be completed by cutting the cell inside the underlying space $|\mathcal{K}|$. The additional boundary cell of a dual cell $*c^p$ is called a **boundary dual cell**, and it can be defined as a dual cell of c^p in a $(k-1)$-dimensional boundary

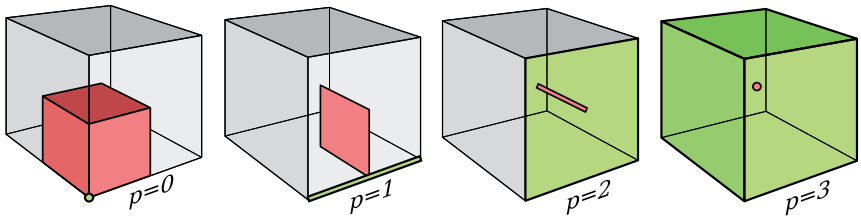


FIGURE 9 Primal p-cells (green) and corresponding dual $(3-p)$-cells (red) in a cubic three-complex.

complex. Thus, the boundary dual cell of c^p is a $(k-1-p)$-cell. We denote the boundary dual cell by $*^{\partial}c^p$.

A **dual complex** is a finite set of dual cells, similarly as the primal complex is a finite set of primal cells. Dual complex of a primal k-complex $\mathcal{K}$ is also a k-complex, and we denote it by $*\mathcal{K}$. The underlying space of a dual complex equals the underlying space of the corresponding primal complex i.e. $|*\mathcal{K}| = |\mathcal{K}|$. A two-complex and the corresponding dual two-complex are illustrated in Figure 10.

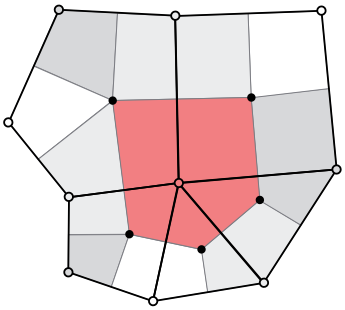


FIGURE 10 A two-complex of five two-cells is partitioned into an orthogonal dual complex. Each primal node corresponds to a dual face, as illustrated by the matching colors. Each primal edge corresponds to a dual edge, and each primal face corresponds to a dual node.

A dual complex of a three-complex includes cells of four different dimensions. We denote the nodes, edges, faces and bodies of the dual mesh by sets of cells $\mathcal{N}^*$, $\mathcal{E}^*$, $\mathcal{F}^*$ and $\mathcal{B}^*$, respectively. The cells are ordered in a similar manner to the primal cells, and we have relations

$$\mathcal{N}_i^* := *\mathcal{B}_i, \qquad \mathcal{E}_i^* := *\mathcal{F}_i, \qquad \mathcal{F}_i^* := *\mathcal{E}_i, \qquad \mathcal{B}_i^* := *\mathcal{N}_i.$$

The orientation of a dual complex follows from the primal complex orientation. Using a similar orientation for each corresponding primal and dual cell pair, the incidence matrix of the dual complex is just the transpose of the primal incidence matrix (Bossavit, 2005). Denoting the incidence p-matrix of a dual k-complex by $*d_p$, we write the relation between primal and dual incidence matrices as $*d_p = d_{k-p-1}^T$. In a three-complex, this means $*d_0 = d_2^T$, $*d_1 = d_1^T$ and $*d_2 = d_0^T$.

2.5 Mesh and dual mesh

The term mesh is often used to describe the discretization of the domain Ω in numerical methods. In this section, we finalize a mesh formulation for the DEC framework. The oriented complex gives a very general presentation for a mesh. Basically, a mesh could be defined as an oriented manifold-like k-complex. Then, for example, objects with curved boundaries could be discretized accurately. It is important to recognize this possibility, but for simplicity, we restrict to using only polyhedral elements. Thus, we define a k-**mesh** (a mesh of dimension k) as an oriented manifold-like k-complex with linear and convex cells.

In a computer implementation, the mesh is stored as lists of p-cells. The three-mesh then includes four lists: nodes $\mathcal{N}$, edges $\mathcal{E}$, faces $\mathcal{F}$ and bodies $\mathcal{B}$. The cells are stored in a recursive manner. A node is stored as a vector $v \in \mathbb{R}^3$. The other cells are stored topologically according to the list of their boundary cells. Specifically, an edge includes a list of exactly two boundary nodes. A face includes at least three boundary edges, and a body includes at least four boundary faces. The positions of the nodes and the topology declare a unique linear complex, which is the mesh.

A **dual mesh** is a dual complex of a primal mesh. Since the primal mesh is an oriented and manifold-like complex, the dual mesh is an oriented and manifold-like dual complex. The linearity and the convexity are not self-evident properties for the dual cells. The dual node positions must be selected carefully to obtain these properties. In the DEC framework, the orthogonality between primal and dual cells is a desired property. The orthogonality means that each vector of a primal cell is orthogonal to every vector of the corresponding dual cell. Using the circumcenters (or weighted circumcenters) of the primal bodies as the dual node positions, all these properties can be accomplished. Next, we recall the notion of circumcenter.

We assume a linear and convex p-cell c^p lying on a p-dimensional affine space A with extreme points $v_1, \ldots, v_n$. If there is a position $x \in A$, which is in equal distances to all nodes $v_1, \ldots, v_n$, then the position x is called the **circumcenter** of c^p. There exists always a circumcenter for an simplicial complex. The existence of the circumcenter for cellular cells depends on the node positions. For example, a rectangle has a circumcenter, which is actually same as the average point of the nodes (called the barycenter). A cell that has a circumcenter is called a **circumcentric cell**. Two examples of circumcentric cells are shown in Figure 11.

The circumcenter is not necessarily inside the cell i.e. $x \notin c^p$. In certain instances, the circumcenter can be far outside the cell boundaries and far from the barycenter of the cell. The **weighted circumcenter** is a more flexible alternative for the circumcenter. Each node is now assigned a pair that includes a position vector $v_i \in \mathbb{R}^3$ and a weight $w_i \in \mathbb{R}$. The weighted circumcenter is then a position x with equal distance $\|x - v_i\|^2 - w_i$ to each extreme point of c^p. This distance, where $\|\cdot\|$ stands for the Euclidean distance, is sometimes referred to as the Laguerre distance (Mullen et al., 2011). A cell, which has a weighted circumcenter,

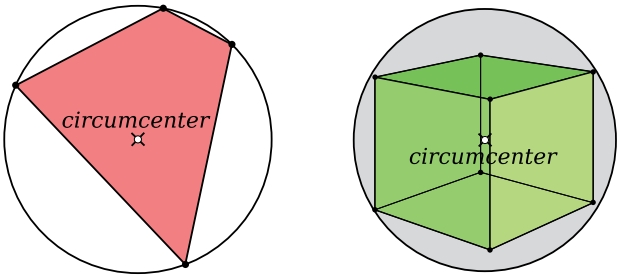


FIGURE 11 A circumcentric 2-cell and a circumcentric 3-cell are illustrated on the left and right, respectively.

is called a **weighted circumcentric cell**. A (weighted) **circumcentric dual mesh** is a linear dual mesh, where the dual node positions are selected as the (weighted) circumcenters of the primal body elements. The circumcentric duality sets several requirements for the primal mesh. Thus, the generation of primal and dual elements cannot be separated. We call (weighted) **circumcentric mesh** a mesh, which has a (weighted) circumcentric dual mesh.

2.6 Generation of a circumcentric mesh

A (weighted) circumcentric mesh and its dual mesh can be constructed by building the (weighted) Voronoi diagram from a set of (weighted) primal nodes. The generation of a Voronoi diagram, which is named after the (1908) work of Georgy Voronoi, is a standard procedure in geometry (see Aurenhammer and Klein, 2000; de Berg et al., 2000). Algorithms for the Voronoi diagrams are available for generally normed n-dimensional spaces (Reem, 2009). In the field of mesh generation, Voronoi diagrams are often applied to generate Delaunay triangulations (see Delaunay, 1934; Maur, 2002). For more details of the relation between Voronoi diagrams and Delaunay meshes, see O'Rourke (1998).

In the mesh generation procedure, which is based on the Voronoi diagram, the generation of primal and dual meshes are inverted. Basically, the Voronoi decomposition from the primal nodes represent the decomposition of the dual mesh, and the primal mesh is constructed topologically from the dual mesh. The procedure of generating a convex mesh from a point set is as follows. Assume a set of primal nodes with positions (and weights for the weighted Voronoi diagram). We compute a Voronoi decomposition, where each cell represents the volume nearest to the current node. Each Voronoi cell, centered by a node, corresponds to a dual body element. Voronoi planes (boundaries between Voronoi cells) correspond to dual face elements, which are associated with primal edge elements between two nodes. Each Voronoi edge (intersection of Voronoi planes) generate both a dual edge and a primal face. The dual nodes and primal bodies are generated from the crossing points of Voronoi edges. In general, the primal

mesh is not a triangulation. Polyhedral elements like cubes, prisms, pyramids or octahedra might exist as well. All these elements fit easily into the DEC framework.

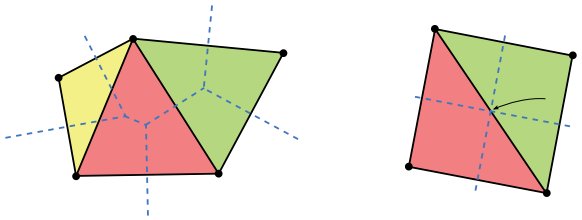


FIGURE 12 A relation between the Voronoi diagram (dotted line) and the Delaunay triangulation (triangles) is illustrated in two-dimensions. On the right-hand side, the Delaunay triangulation has a zero sized dual edge in the middle.

The DEC framework demands positively sized primal and dual cells. The Delaunay triangulation, in which every circumball is empty of nodes, does not offer this automatically. For example, if a Delaunay triangulation is constructed from the four extreme points of a square, there exists a zero sized dual edge in the middle, as shown on the right-hand side of Figure 12. Basically, a Delaunay triangulation can be transformed into acceptable polyhedral mesh by combining the cells with the same circumcenters. We insist that each circumball of a cell does not include any nodes, but all nodes on the boundary must belong to the current cell. To ensure positively sized dual cells at the neighborhood of the domain boundary, we also insist that every circumcenter must lie inside the underlying space of the mesh.

2.7 Mesh optimization

There are certain difficulties in constructing fully unstructured primal and dual meshes. Applying the Voronoi diagrams to a random set of nodes leads almost everywhere to a (weighted) Delaunay triangulation. A random Delaunay triangulation might offer non-optimal dual elements, which are difficult for the DEC method. For optimal meshing, one should guarantee good alignment properties of the primal and dual elements, as discussed in Section 3.3. The element sizes should also be close to each other because the time stepping stability criterion of Section 5.5 is optimal for uniform element sizes. Close to zero element sizes might also cause problems with numerical stability because a division by a small number occurs. In some cases, a sophisticated mesh generation can be an even more challenging problem than the full DEC-based simulation of Maxwell equations.

There are several methods to improve the Delaunay meshing by optimizing certain parameters of the Delaunay triangulation (see Tournois et al., 2009; van Kreveld et al., 2010). As described in Mullen et al. (2011), there is also an optimization method, which is targeted specifically for the DEC framework. The

Hodge-Optimized triangulation (HOT) is designed to improve the quality of the Hodge operator, which is the most important factor for the DEC method accuracy, as we discuss in Chapter 3. The HOT optimization is based on the optimal transport theory, and the idea is to optimize the positions v_i and weights w_i of each primal node. Starting from an initial non-optimized mesh, the positions and weights are optimized alternately by an iterative algorithm. The topological relations of the mesh remain unchanged in the optimization, which means the initial mesh should have been selected wisely. The HOT optimization is applied only for simplicial meshes. Applying the method for polyhedral cells might destroy the circumcentricity of the cells. It still remains unclear, how this optimization method could be applied to polyhedral circumcentric cells.

Our approach is to generate partly structured meshes. In other words, we generate a mesh that consists of structured regions, which are separated by unstructured layers. One advantage of this approach is that we can model boundaries as accurately as with the fully unstructured meshing. We can also modify element sizes inside the domain. The mesh generation is also relatively fast because a very small proportion of elements is unstructured. The HOT optimization method is needed only for the unstructured cells. In the next section, we introduce several grid types, which are based on natural crystal structures. These grids can be applied as structured parts of the partly structured mesh design.

2.8 Natural crystal structures for the structured grid design

Structured grids are successfully applied for the discretization of the Maxwell equations with several frameworks. There are advantages of using structured grids instead of unstructured grids. Usually, the construction of a structured grid is relatively simple, and important element properties can still be controlled. The simulation framework can also be optimized for certain structured systems, and then both memory consumption and CPU time can be reduced. Of course, there are also disadvantages in using structured grids. The structured grids rarely fit curved or oblique boundaries well. The change of element sizes can also be difficult inside the domain. Usually, the structured grids are anisotropic, and thus the system might have different properties in different spatial orientations. The simulation accuracy might depend on the selection of the grid type, as observed in two-dimensions by Keränen et al. (2004). In this section, we introduce several structure types, which can be applied to structured mesh design. The properties of the following grids are tested by numerical experiments in Section 8.5.

The cubic tiling is the simplest and the most common way to fill a three-dimensional space. It is also the basis for the cubic crystal systems, which are used in crystallography to explain natural crystal structures. There are three main structures in cubic crystal systems: primitive cubic, face-centered cubic (FCC) and body-centered cubic (BCC) (Conway and Sloane, 1999). The primitive cubic structure leads to a regular grid, which has a regular dual grid. The regular

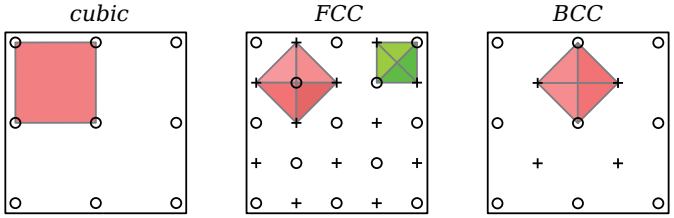


FIGURE 13 The vertices of cubic, FCC and BCC grids are presented by the **o** symbol on even layers and by the **+** symbol on odd layers. One body element of every kind is illustrated on the grids.

grid is applied in several discretization methods, including the conventional Yee scheme. The FCC and BCC structures and their dual structures are explained below.

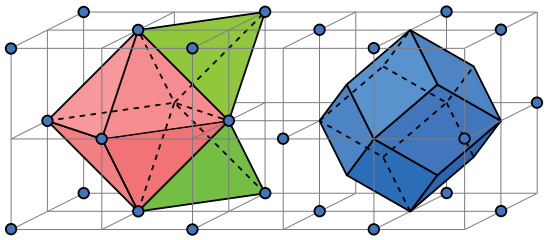


FIGURE 14 Primal and dual elements of the FCC grid are illustrated with respect to the regular base grid. The primal nodes of the construction are represented by small spheres.

The FCC structure can be constructed by the Voronoi diagram from the vertex set, where the cubic grid vertices and the center points of each face element are united, as shown in Figure 13. The space tiling of the FCC structure is composed of alternating regular octahedra and regular tetrahedra at a ratio of 1:2. Each primal edge is of equal length, and the primal faces are equilateral triangles of equal areas. The circumcentric dual faces are rhombuses (a parallelogram with four sides of equal length) with a relation of $\sqrt{2}$ between the longer and shorter diagonal length. The circumcentric dual body elements are Kepler's rhombic dodecahedra, which consist of 12 congruent rhombic faces (Kepler, 1966). The primal and dual elements of the FCC structure are illustrated in Figure 14.

The BCC structure is a tetrahedron structure, which is one of the Sommerville's grids (Sommerville, 1923). It is constructed from the cubic grid by adding a vertex at each body center, as illustrated in Figure 13. The space tiling of the BCC grid is composed of congruent tetrahedra, where each face is a congruent isosceles triangle (see Figure 15). The bottom edges are longer than the side edges by a ratio of $2 : \sqrt{3}$. The dual space tiling consists of truncated octahedra, each having 6 square faces and 8 regular hexagon faces. All the dual edges have the same length. The dual tessellation is known as the Kelvin's structure, which was introduced by Thomson (1887). According to the Kelvin's conjecture, the Kelvin's structure was proposed to be the best space-tiling to minimize the

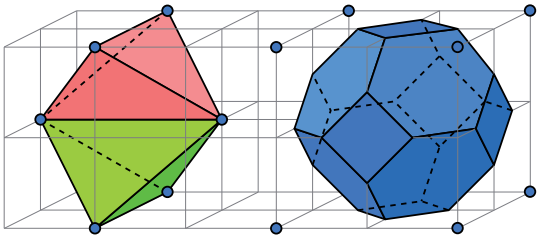


FIGURE 15 Primal and dual elements of the BCC grid are illustrated with respect to the regular base grid. The primal nodes of the construction are represented by small spheres.

total surface area of the interfaces between the elements of equal volume. As the BCC structure is the simplest well-centered tetrahedral grid, it is preferred for the Yee-like simulation schemes in several instances (Bossavit, 2003; VanderZee et al., 2008).

Bossavit (2001, 2003) suggests to use meshes with sharp dihedral angles, when a Yee-like scheme is applied on the tetrahedral grids. By the following theoretical consideration, we can see that the BCC construction is not likely to be the optimal one. The dihedral angle of a regular tetrahedron is $\arccos \frac{1}{3}$. Hence, in a hypothetical space-tiling configuration, we would need $2\pi / \arccos \frac{1}{3} \approx 5.1$, regular tetrahedra to share the same edge. From this, it can be concluded that in a densely packed, nearly regular tetrahedral tiling, there should be five or six tetrahedra sharing the same edge (Sullivan, 2000). Therefore the dual elements should have either pentagonal or hexagonal faces, which is not the case with the Kelvin's structure. Thus, we recall the concept of tetrahedrally close-packed (TCP) structures (described by Frank and Kasper, 1958, 1959), which have this property.

In TCP structures, there are four combinatorial types of the Voronoi cells, which all have only pentagonal and hexagonal faces, with no adjacent hexagons (Sullivan, 1998). In the literature, the most common TCP structures are known as A15, C15 and Z lattices. In fact, the other known TCP structures can be viewed as combinations of these three basic ones (Kusner and Sullivan, 1996; Du and Wang, 2005; Sikiric et al., 2010). The maximum dihedral angles of A15, C15 and Z elements are relatively sharp compared to the BCC structure, as discussed by Eppstein et al. (2004). However, it is not obvious that the TCP structures are optimal for the Yee-like schemes. For example, the edge lengths differ more in the TCP structures than in the cubic crystal structures (Sullivan, 1998).

The A15 structure was first discovered in a molecule structure by Hartmann et al. (1931). To construct A15, one can start with a BCC lattice and insert two vertices on each face of the original cubic grid, as illustrated in Figure 16. The dual body elements of A15 are irregular dodecahedrons (12-hedron with pentagonal faces) centered at each BCC lattice vertex, and a tetrakaidecahedron (14-hedron with two hexagonal and twelve pentagonal faces) around each of the other vertices. Weaire and Phelan (1994) found that a stretched version of the dual struc-

TABLE 1 Relative vertex coordinates for TCP structures are scaled into a unit cube.

A15	C15	Z
$(0,0,0)$	$(0,0,0)$	$(0,0,0)$, $\left(0,0,\frac{1}{2}\right)$
$\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$	$\left(0,\frac{1}{2},\frac{1}{2}\right)$, $\left(\frac{1}{2},0,\frac{1}{2}\right)$, $\left(\frac{1}{2},\frac{1}{2},0\right)$	$\left(\frac{1}{2},\frac{1}{2},0\right)$, $\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$
$\left(0,\frac{1}{2},\frac{1}{4}\right)$	$\left(\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$, $\left(\frac{1}{4},\frac{3}{4},\frac{3}{4}\right)$, $\left(\frac{3}{4},\frac{1}{4},\frac{3}{4}\right)$, $\left(\frac{3}{4},\frac{3}{4},\frac{1}{4}\right)$	$\left(\frac{1}{6},\frac{1}{2},\frac{1}{4}\right)$, $\left(\frac{5}{6},\frac{1}{2},\frac{1}{4}\right)$
$\left(0,\frac{1}{2},\frac{3}{4}\right)$		$\left(\frac{2}{6},0,\frac{1}{4}\right)$, $\left(\frac{4}{6},0,\frac{1}{4}\right)$
$\left(\frac{1}{2},\frac{1}{4},0\right)$	$\left(\frac{1}{8},\frac{1}{8},\frac{5}{8}\right)$, $\left(\frac{1}{8},\frac{3}{8},\frac{7}{8}\right)$, $\left(\frac{3}{8},\frac{1}{8},\frac{7}{8}\right)$, $\left(\frac{3}{8},\frac{3}{8},\frac{5}{8}\right)$	
$\left(\frac{1}{2},\frac{3}{4},0\right)$	$\left(\frac{1}{8},\frac{5}{8},\frac{1}{8}\right)$, $\left(\frac{1}{8},\frac{7}{8},\frac{3}{8}\right)$, $\left(\frac{3}{8},\frac{5}{8},\frac{3}{8}\right)$, $\left(\frac{3}{8},\frac{7}{8},\frac{1}{8}\right)$	$\left(\frac{1}{4},\frac{1}{4},\frac{3}{4}\right)$, $\left(\frac{1}{4},\frac{3}{4},\frac{3}{4}\right)$
$\left(\frac{1}{4},0,\frac{1}{2}\right)$	$\left(\frac{5}{8},\frac{1}{8},\frac{1}{8}\right)$, $\left(\frac{5}{8},\frac{3}{8},\frac{3}{8}\right)$, $\left(\frac{7}{8},\frac{1}{8},\frac{3}{8}\right)$, $\left(\frac{7}{8},\frac{3}{8},\frac{1}{8}\right)$	$\left(\frac{3}{4},\frac{3}{4},\frac{3}{4}\right)$, $\left(\frac{3}{4},\frac{3}{4},\frac{3}{4}\right)$
$\left(\frac{3}{4},0,\frac{1}{2}\right)$	$\left(\frac{5}{8},\frac{5}{8},\frac{5}{8}\right)$, $\left(\frac{5}{8},\frac{7}{8},\frac{7}{8}\right)$, $\left(\frac{7}{8},\frac{5}{8},\frac{7}{8}\right)$, $\left(\frac{7}{8},\frac{7}{8},\frac{5}{8}\right)$	$\left(\frac{1}{2},0,\frac{3}{4}\right)$, $\left(0,\frac{1}{2},\frac{3}{4}\right)$

ture provides a counter-example to Kelvin's conjecture on minimal surfaces. In modern knowledge, this Weaire-Phelan structure partitions a three-dimensional space into cells of equal volume with the smallest area of surface between them. Therefore, it is applied, for example, in modeling bubble foams (Lautensack and Sych, 2008).

The C15 structure was first observed in a molecule structure by Friauf (1927) and Laves (Paufler, 2011). The fundamental structure of C15 is constructed from 24 vertices, as listed in Table 1. The C15 structure can be built on both the cubic or the FCC base grid (see Figure 16). From the three basic TCP structures, C15 has the smallest maximum dihedral angle of 74.20 degrees (Eppstein et al., 2004). The replicable dual structure consists of sixteen 12-hedra and eight 16-hedra.

The FCC, BCC, A15 and C15 structures are constructed from the cubic base, and they are symmetric in positive and negative x-, y- and z-directions. The Z-grid, defined as A_4B_3 by Frank and Kasper (1959), has different properties in all three coordinate directions, but the grid is symmetric on the x-y-plane in 60-degree increments. The z-direction is divergent to any other direction. The fundamental structure of the Z-grid is repeated in h increments in y- and z-directions and in $\sqrt{3}h$ increments in the x-direction, as illustrated in Figure 16. The listed x-coordinates of Table 1 should then be multiplied by $\sqrt{3}$ to obtain a correct tiling. The replicable dual structure consists of six 12-hedra, four 14-hedra, and four 15-hedra (Sullivan, 1998).

2.9 Refining of structured grids

The structured grids offer uniform element properties throughout the region. Thus, tuning of the element size affects the whole domain. In many problems, the non-uniform mesh properties are desired. With DEC meshing, the element

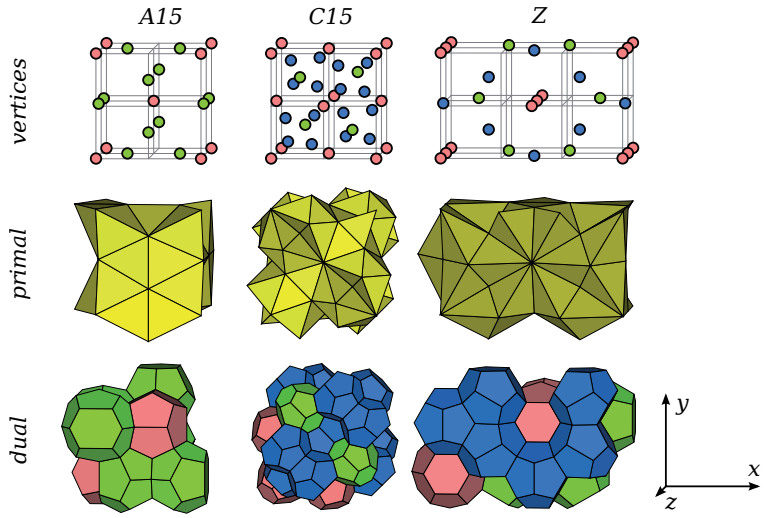


FIGURE 16 The TCP structures A15, C15 and Z are illustrated in three different aspects. The relative primal node positions, which are listed in Table 1, are shown on the top row. The recurring primal and dual structures are illustrated on the middle and bottom row, respectively.

properties can be changed by applying an unstructured layer between different structures. In this approach, discretization of the unstructured elements might be difficult, and the elements might require optimization. Luckily, there is another approach. Since several grid types can be applied to the DEC framework, the structures can be combined. The element properties between different structures can still be controlled. A two-dimensional example of such grid refining is illustrated in Figure 17.

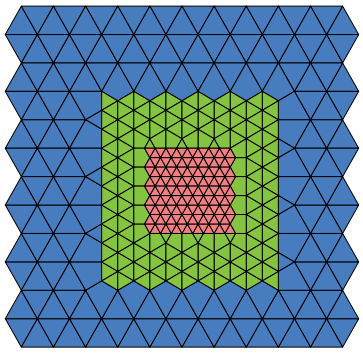


FIGURE 17 A refining of a structured grid in two-dimensions: Starting from a regular triangular grid, a finer grid is achieved by adding nodes to each face center positions. The same procedure can be repeated for the resulting grid. The mesh is updated by the Voronoi construction at each step.

The same procedure can also be applied in three-dimensions. As presented in the previous section, the structured grids FCC, BCC, A15 and C15 are based on

a regular grid. The regular grid can be modified to any of these grids by adding necessary nodes at wanted positions. The new mesh is obtained by updating the cells by the Voronoi construction method. If a regular grid is modified to the FCC or BCC grid, the grid can be converted back to a finer regular grid by adding a few more points. Then, the resulting regular grid has the half edge length compared to the original base grid. Another possibility is to modify the FCC grid to the C15 grid or modify the BCC grid to the A15 grid. The number of possible element shapes between the modified structure and the base structure is limited. Thus, the element properties can be known in advance, and the method can be called a structured grid refining method. There are several variations for refining a three-dimensional structured grid. A nice topic for future research would be to study different mesh refining possibilities. It might be possible to build a fast mesh generator, which produces automatically structured grids that are suitable for DEC discretization.

3 DISCRETIZATION OF MAXWELL EQUATIONS

In this chapter, we discretize the Maxwell equations on the mesh, which is formulated in the previous chapter. The discretization is carried out using the notations of discrete exterior calculus (DEC), which is presented in several previous studies (Hirani, 2003; Desbrun et al., 2005, 2008; Stern et al., 2009). The discretization leads to a time-dependent simulation method for electromagnetic waves. Using the terminology of Bossavit and Kettunen (1999, 2000), the resulting method can be considered a Yee-like scheme on polyhedral meshes. We introduce the discrete differential forms using both classical and differential form presentations of Maxwell equations.

3.1 Discrete forms

A smooth differential p-form is a tailor-made object for the integration over p-manifolds, as described in Abraham et al. (1988). The discrete differential form is a discrete analogue of the smooth differential form. Thus, the discrete p-form is defined as an object, which is tailor made for integration over discrete p-cells. We describe the discrete p-form α as a column vector, which assigns a real number α_i to each p-cell c_i^p of a mesh. The discrete form values can be produced from the smooth differential form using a so called **de Rham map** (see Hirani, 2003, Chapter 3). If $\tilde{\alpha}$ is a smooth differential p-form, then the corresponding discrete form value on c_i^p is defined by

$$\alpha_i = \langle \alpha, c_i^p \rangle := \int_{c_i^p} \tilde{\alpha}.$$

To discretize the Maxwell equations, the vector fields **E**, **B**, **H** and **D** of classical presentation are transferred to discrete forms E, B, H and D. We define E on primal edges (one-form), B on primal faces (two-form), H on dual edges (one-form) and D on dual faces (two-form). The discrete form values are computed

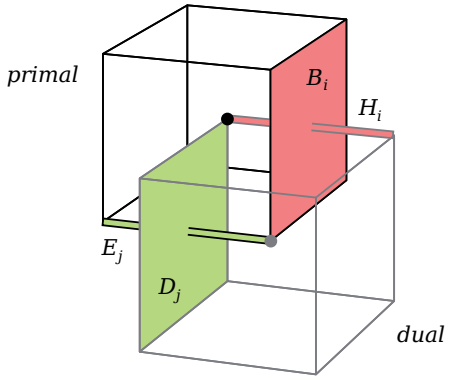


FIGURE 18 Illustration of discrete forms on primal and dual elements of a staggered mesh. E and B are placed on primal elements, and H and D are placed on dual elements. E and H lie on edges (discrete one-form), and B and D lie on faces (discrete two-form).

from vector fields by

$$
\begin{aligned}
E_j &:= \langle E, \mathcal{E}_j \rangle = \int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l}, \\
B_i &:= \langle B, \mathcal{F}_i \rangle = \int_{\mathcal{F}_i} \mathbf{B} \cdot d\mathbf{a}, \\
H_i &:= \langle H, \mathcal{E}_i^* \rangle = \int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l}, \\
D_j &:= \langle D, \mathcal{F}_j^* \rangle = \int_{\mathcal{F}_j^*} \mathbf{D} \cdot d\mathbf{a}.
\end{aligned}
$$

Here, $\mathcal{E}_j$ and $\mathcal{E}_i^*$ are primal and dual edge elements, and $\mathcal{F}_j^*$ and $\mathcal{F}_i$ are corresponding dual and primal face elements, respectively. The disposition of discrete form values on primal and dual elements are illustrated in Figure 18. In this thesis, the discrete forms E and D are presented as column vectors of n rows, and discrete forms H and B are presented as column vectors of m rows.

To fully discretize the Maxwell equations, we need to introduce two discrete two-forms J and J^* corresponding to the electric and magnetic currents $\mathbf{J}$ and $\mathbf{J}^*$, respectively. We also introduce two discrete three-forms ϱ and ϱ^*, which correspond to electric and magnetic charges ρ and ρ^*. We have

$$
\begin{aligned}
J_i^* &:= \langle J^*, \mathcal{F}_i \rangle = \int_{\mathcal{F}_i} \mathbf{J}^* \cdot d\mathbf{a}, \\
J_j &:= \langle J, \mathcal{F}_j^* \rangle = \int_{\mathcal{F}_j^*} \mathbf{J} \cdot d\mathbf{a}, \\
\varrho_l^* &:= \langle \varrho^*, \mathcal{B}_l \rangle = \int_{\mathcal{B}_l} \rho^* \, dv, \\
\varrho_k &:= \langle \varrho, \mathcal{B}_k^* \rangle = \int_{\mathcal{B}_k^*} \rho \, dv.
\end{aligned}
$$

3.2 Discrete Maxwell equations

The curl operator $\nabla \times \mathbf{E}$ in the Faraday law (1) corresponds to exterior derivative d in a differential form presentation (8). Using the Stokes' theorem, we obtain the integral formulations of Maxwell equations, which are presented in (7) or (12). The integral formulation of Faraday law can be written exactly on surface $\mathcal{F}_i$ using the discrete forms introduced in the previous section. As illustrated in the left-hand side of Figure 19, the integral of $\mathbf{E}$ over the boundary $\partial\mathcal{F}_i$ can be obtained by a signed sum of the bounding discrete one-form values E_j. The sign depends on the relative orientation of edge and face elements. In fact, the sign is obtained by the incidence matrix d_1, which is presented in Section 2.3. We write the Faraday law on surface $\mathcal{F}_i$ as

$$J_i^* + \partial_t B_i = -\sum_{j=1}^{n} (d_1)_{i,j} E_j. \tag{15}$$

After applying the Faraday law to every primal face, we obtain the discrete Faraday law $J^* + \partial_t B = -d_1 E$, where d_1 has analogue to the discrete version of the exterior derivative d.

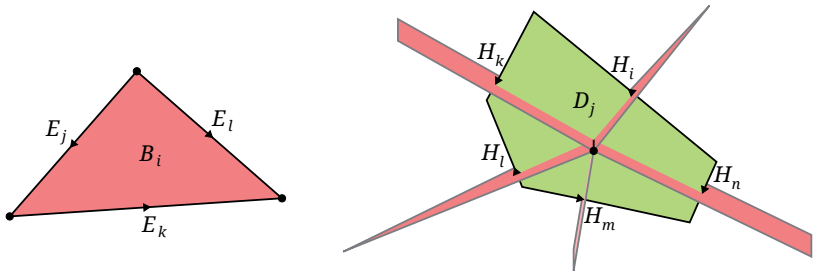


FIGURE 19 The discrete Faraday law is formulated on primal faces (illustrated on the left), where the integral of $\mathbf{E}$ over the face boundary can be expressed as a signed sum of E components. Similarly, the discrete Ampére law is formulated on dual faces (illustrated on the right).

The Ampére law (2) or (9) is discretized in a similar manner on the dual elements. We write the Ampére law on the interior dual face $\mathcal{F}_j^*$. The integration of $\mathbf{H}$ over the boundary $\partial\mathcal{F}_j^*$ is obtained by a signed sum of discrete form values H_i, as illustrated in the right-hand side of Figure 19. The sign is obtained by the incidence matrix of the dual mesh, which is actually a transpose of a primal incidence matrix, as explained in Section 2.4. On the interior dual faces, the Ampére law is written as

$$J_j + \partial_t D_j = \sum_{i=1}^{m} (d_1)_{i,j} H_i, \tag{16}$$

The dual faces, which are cut at the domain boundary, require a specific treatment. This is carried out by describing a boundary condition on such elements.

The boundary conditions are discussed in Section 4.2. At this point, we are satisfied with writing the discrete Ampére law on the full domain as $J + \partial_t D = d_1^T H$.

Next, we consider the last two equations of the Maxwell system, which are called the Gauss laws. We show that these equations are automatically conserved in the discretized Maxwell system. The divergence operators $\nabla \cdot$ in Gauss laws (3) and (4) correspond to the exterior derivative d of a differential two-form in Equations (10) and (11). The discrete divergence operators on primal and dual meshes are represented by the incidence matrix d_2 and transpose incidence matrix d_0^T, respectively. The Gauss laws are then written as

$$d_2 B = \varrho^*,$$
$$d_0^T D = \varrho.$$

The discrete version of the continuity of charge is written on primal and dual meshes as $\partial_t \varrho^* + d_2 J^* = 0$ and $\partial_t \varrho + d_0^T J = 0$. The conservation of the Gauss laws is obtained by multiplying the discrete Faraday law (15) from the left by d_2 and multiplying the discrete Ampére law (16) from the left by d_0^T. Using the continuity of charge on the remaining equation, we get

$$-d_2 d_1 E = d_2 \left(J^* + \partial_t B \right) = \partial_t \left(-\varrho^* + d_2 B \right),$$
$$d_0^T d_1^T H = d_0^T \left(J + \partial_t D \right) = \partial_t \left(-\varrho + d_0^T D \right).$$

The left-hand side is zero, since $d_2 d_1 = 0$ and $d_0^T d_1^T = 0$. The conservation of the Gauss laws is proven by this observation. The Gauss laws can be viewed simply as constraints on initial conditions, while the Faraday and Ampére laws completely describe the time evolution of the system.

3.3 Material parameters as diagonal discrete Hodge operators

The equations (15) and (16) can be imposed exactly as they involve no metric. However, the Maxwell system is not complete because there is no connection between the primal and dual discrete forms. This connection is obtained by the discrete Hodge operator, which is a discrete version of the smooth Hodge star. The Hodge star is an operator that involves the metric and thus is the most important component for the accuracy of the DEC framework. Traditionally, the discrete Hodge operator $\star$ is defined as a map, which transforms a discrete p-form α to a discrete $(n-p)$-form $\star\alpha$ by the relation

$$\frac{1}{|c^*|} \langle \star\alpha, c^* \rangle = \frac{1}{|c|} \langle \alpha, c \rangle.$$

Here we recall that $|c|$ is the p-volume of the cell c. For more details, see Sen et al. (2000); Desbrun et al. (2005); Hirani et al. (2012).

Basically, the discrete Hodge operator is a map from the primal p-form to its dual $(n-p)$-form. In the three-dimensional DEC framework, we consider maps

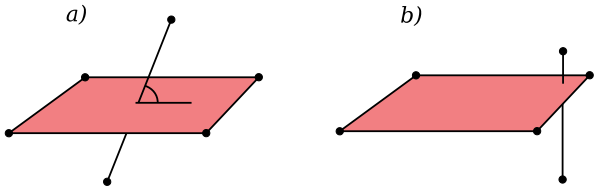


FIGURE 20 The orthogonality and good center alignment of corresponding primal and dual elements are desirable properties for a discrete Hodge star. Figure a) illustrates non-orthogonal pair, and figure b) illustrates badly aligned elements.

between one-forms and two-forms, which lie on the edges and faces, respectively. The discrete Hodge star is always an approximative geometric operator, and the accuracy of the approximation depends on the properties of the discretization. To obtain maximal accuracy, the discrete Hodge operator should transform values from edges to faces as locally as possible. Thus, we prefer the diagonal presentation of the discrete Hodge operator, where each one-form value on an edge has a linear relation to the corresponding two-form value on the face. Using a diagonal Hodge matrix, we can consider the pairwise properties primal-dual element pairs, as illustrated in Figure 20. It is widely reported in the literature that the orthogonality of primal-dual pairs is the main demand for obtaining a diagonal Hodge operator (Stern et al., 2009). The orthogonality can be obtained by the (weighted) circumcentric duality, which is discussed in Section 2.5 of this thesis. A good relative alignment of element pairs is the secondary demand for the mesh construction. Ideally, all of the corresponding primal and dual elements should be weighted at the corresponding positions, unlike in b) of Figure 20. This property is not easily obtained in unstructured orthogonal dual meshing, even though the HOT mesh optimization method is an attempt to overcome this challenge (Mullen et al., 2011). In this thesis, we prefer using structured grids as much as possible to control the alignment of primal-dual pairs.

Within the DEC framework, the constitutive relations (5) are replaced by Hodge operators $\star\epsilon$ and $\star\mu$, which correspond to permittivity ϵ and permeability μ, respectively. The constitutive relations are written as

$$D = \star\epsilon E,$$
$$B = \star\mu H.$$

Since the discrete forms E and D are represented as column vectors of n rows, the Hodge operator $\star\epsilon$ is represented by $n \times n$ matrix. Similarly, $\star\mu$ is represented by $m \times m$ matrix. For simplicity, we prefer considering diagonal Hodge operators, which lead to the relations $D_j = \star\epsilon_{j,j} E_j$ and $B_i = \star\mu_{i,i} H_i$.

To compute the diagonal terms $\star\epsilon_{j,j}$ and $\star\mu_{i,i}$, we assume locally constant vector fields $\mathbf{E}$ and $\mathbf{H}$ and material tensors ϵ and μ. We denote the unit direction vectors of the edge elements $\mathcal{E}_j$ and $\mathcal{E}_i^*$ by $\mathbf{n}_{\mathcal{E}_j}$ and $\mathbf{n}_{\mathcal{E}_i^*}$, respectively. The unit normal vectors of the face elements $\mathcal{F}_j^*$ and $\mathcal{F}_i$ are denoted respectively by $\mathbf{n}_{\mathcal{F}_j^*}$ and

$\mathbf{n}_{\mathcal{F}_i}$. To obtain a linear relation between the corresponding discrete field values we assume that there are real numbers $\epsilon_1, \mu_1 \in \mathbb{R}$ such that

$$\begin{aligned} \epsilon_1 \mathbf{E} \cdot \mathbf{n}_{\mathcal{E}_j} &= \epsilon \mathbf{E} \cdot \mathbf{n}_{\mathcal{F}_j^*}, \\ \mu_1 \mathbf{H} \cdot \mathbf{n}_{\mathcal{E}_i^*} &= \mu \mathbf{H} \cdot \mathbf{n}_{\mathcal{F}_i}, \end{aligned} \tag{17}$$

for any selection of $\mathbf{E}, \mathbf{H} \in \mathbb{R}^3$. If the material parameters are isotropic i.e. $\epsilon = \epsilon_1 \mathcal{I}$ and $\mu = \mu_1 \mathcal{I}$, these relations are obtained simply by the orthogonal duality, where the corresponding direction vectors are equal. The anisotropic material parameters require a specifically discretized mesh. The diagonal Hodge operators, which are presented by Bossavit (2000), are obtained by the following consideration

$$\begin{aligned} \star\epsilon_{j,j} = \frac{D_j}{E_j} &= \frac{\int_{\mathcal{F}_j^*} \epsilon \mathbf{E} \cdot d\mathbf{a}}{\int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l}} = \frac{\epsilon \mathbf{E} \cdot \mathbf{n}_{\mathcal{F}_j^*} |\mathcal{F}_j^*|}{\mathbf{E} \cdot \mathbf{n}_{\mathcal{E}_j} |\mathcal{E}_j|} = \epsilon_1 \frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|}, \\ \star\mu_{i,i} = \frac{B_i}{H_i} &= \frac{\int_{\mathcal{F}_i} \mu \mathbf{H} \cdot d\mathbf{a}}{\int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l}} = \frac{\mu \mathbf{H} \cdot \mathbf{n}_{\mathcal{F}_i} |\mathcal{F}_i|}{\mathbf{H} \cdot \mathbf{n}_{\mathcal{E}_i^*} |\mathcal{E}_i^*|} = \mu_1 \frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|}. \end{aligned}$$

In the third equality, we assume locally constant vector fields $\mathbf{E}$, $\mathbf{H}$ and material parameters ϵ, μ. For non-constant material parameters, we apply averaged material parameters $\bar{\epsilon} := \frac{1}{|\mathcal{F}_j^* \diamond \mathcal{E}_j|} \int_{\mathcal{F}_j^* \diamond \mathcal{E}_j} \epsilon \, dv$ and $\bar{\mu} := \frac{1}{|\mathcal{F}_i \diamond \mathcal{E}_i^*|} \int_{\mathcal{F}_i \diamond \mathcal{E}_i^*} \mu \, dv$. The real valued factors are then obtained by integrating

$$\begin{aligned} \epsilon_1 &:= \bar{\epsilon} \mathbf{n}_{\mathcal{E}_j} \cdot \mathbf{n}_{\mathcal{F}_j^*} = \frac{1}{|\mathcal{F}_j^* \diamond \mathcal{E}_j|} \int_{\mathcal{F}_j^* \diamond \mathcal{E}_j} \epsilon \mathbf{n}_{\mathcal{E}_j} \cdot \mathbf{n}_{\mathcal{F}_j^*} \, dv, \\ \mu_1 &:= \bar{\mu} \mathbf{n}_{\mathcal{E}_i^*} \cdot \mathbf{n}_{\mathcal{F}_i} = \frac{1}{|\mathcal{F}_i \diamond \mathcal{E}_i^*|} \int_{\mathcal{F}_i \diamond \mathcal{E}_i^*} \mu \mathbf{n}_{\mathcal{E}_i^*} \cdot \mathbf{n}_{\mathcal{F}_i} \, dv. \end{aligned}$$

Here we denote a small neighborhood of edge $\mathcal{E}$ and face $\mathcal{F}$ by a diamond product $\mathcal{F} \diamond \mathcal{E}$. We do not offer a formal expression for the neighborhood, but one could use for example the minimal convex hull containing both edge and face elements. The integration of material parameters can be performed by either analytical or numerical integration. If the material parameters are declared constantly on each discretization elements, analytical integration might be considered. For an arbitrary declaration of material parameters, a small Monte-Carlo style integration (Rubinstein, 1981; Gentle, 1998) might be needed to compute each Hodge term.

The discrete Hodge star is applied also to discretize the equations for electric and magnetic current densities (6). We introduce discrete Hodge operators $\star\sigma$ and $\star\sigma^*$ corresponding to electric and magnetic conductivities σ and σ^*, respectively. The equations for the electric and magnetic currents are then

$$\begin{aligned} J &= \star\sigma E, \\ J^* &= \star\sigma^* H. \end{aligned}$$

Similarly to the Hodge operators $\star\epsilon$ and $\star\mu$, we define $\star\sigma$ and $\star\sigma^*$ as diagonal $n \times n$ and $m \times m$ matrices, respectively. The diagonal terms of the Hodge operators are then

$$\star\sigma_{j,j} = \sigma_1 \frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|}\ ,\ \star\sigma^*_{i,i} = \sigma_1^* \frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|},$$

where the real valued material parameters are integrated by

$$\sigma_1 := \frac{1}{|\mathcal{F}_j^* \diamond \mathcal{E}_j|} \int_{\mathcal{F}_j^* \diamond \mathcal{E}_j} \sigma \mathbf{n}_{\mathcal{E}_j} \cdot \mathbf{n}_{\mathcal{F}_j^*}\, dv,$$
$$\sigma_1^* := \frac{1}{|\mathcal{F}_i \diamond \mathcal{E}_i^*|} \int_{\mathcal{F}_i \diamond \mathcal{E}_i^*} \sigma^* \mathbf{n}_{\mathcal{E}_i^*} \cdot \mathbf{n}_{\mathcal{F}_i}\, dv.$$

These diagonal Hodge operators are obtained by assuming locally constant vector fields. This assumption leads to a spatial discretization scheme, which is similar to the classical Yee scheme. Thus, in this thesis, we call these formulations as Yee's Hodge operators. The assumption of locally constant vector fields might not be the most accurate approximation in the spatial discretization. There are several other ways to define the diagonal discrete Hodge operators. For example, in Section 8.1, we derive the Hodge operators using the assumption of harmonic waves.

The use of the discrete Hodge operators with the discretized Maxwell equations (15) and (16) reduces the need for variables B, D, J and J^*. The full Maxwell system is now carried out by two discrete one-forms E and H, where E lies on primal edges and H lies on dual edges. The Maxwell system at the interior domain is now presented by

$$\begin{aligned} \star\sigma E + \star\epsilon\, \partial_t E &= d_1^T H, \\ \star\sigma^* H + \star\mu\, \partial_t H &= -d_1 E. \end{aligned} \tag{18}$$

3.4 Time-harmonic expression

The time-harmonic solutions can be uniquely expressed by complex numbers, where real and imaginary parts represent the solution at different time phases. In this thesis, we use the hat-symbol $\hat{}$ to denote a time-harmonic complex valued variable. Using the complex time-harmonic expression $\hat{E}$ and $\hat{H}$ for discrete electric and magnetic fields, the field values at time instance t are obtained by

$$E(t) = \text{real}(\hat{E} e^{-i\omega t}),$$
$$H(t) = \text{real}(\hat{H} e^{-i\omega t}),$$

where ω is the angular frequency of the time-harmonic wave. The material parameters can also be expressed in complex presentation. The imaginary terms of complex permittivity $\hat{\epsilon}$ and permeability $\hat{\mu}$ include the conductivity terms σ

and σ^*, respectively. Thus, the conductivity terms are not needed in the complex presentation. The relations between the complex material parameters and real valued material parameters can be written as $\hat{\epsilon} = \epsilon + \frac{i}{\omega}\sigma$ and $\hat{\mu} = \mu + \frac{i}{\omega}\sigma^*$. With the same ideology, we introduce the complex discrete Hodge operators $\star\hat{\epsilon}$ and $\star\hat{\mu}$ for time-harmonic problems. The relations to real valued discrete Hodge operators are written as

$$\begin{aligned} \star\hat{\epsilon} &:= \star\epsilon + \tfrac{i}{\omega}\star\sigma, \\ \star\hat{\mu} &:= \star\mu + \tfrac{i}{\omega}\star\sigma^*. \end{aligned} \tag{19}$$

The complex presentation offers a shorter notation for the Maxwell system. The Equations (18) can be now written as

$$\begin{aligned} \star\hat{\epsilon}\,\partial_t\hat{E} &= d_1^T\hat{H}, \\ \star\hat{\mu}\,\partial_t\hat{H} &= -d_1\hat{E}. \end{aligned} \tag{20}$$

In this thesis, we formulate the time-dependent solution method using the real valued terms. Thus, the complex presentation is always transformed to a real valued expression before continuing with the time discretization. The complex formulation is restored to the real valued Maxwell system by applying the following transformation

$$\begin{aligned} \star\epsilon &:= \mathrm{real}(\star\hat{\epsilon}), & \star\sigma &:= \omega\,\mathrm{imag}(\star\hat{\epsilon}), \\ \star\mu &:= \mathrm{real}(\star\hat{\mu}), & \star\sigma^* &:= \omega\,\mathrm{imag}(\star\hat{\mu}). \end{aligned} \tag{21}$$

4 CONDITIONS FOR SCATTERING PROBLEMS

The spatially discretized Maxwell system is presented in Equation (18). To apply the equations to physical applications, the initial and boundary conditions must be considered. In this chapter, we present the most common conditions for scattering problems. For forthcoming arguments, we include the source terms f_E and f_H into the Maxwell system and write it as

$$\begin{aligned} \star\sigma E + \star\epsilon \partial_t E &= d_1^T H + f_E, \\ \star\sigma^* H + \star\mu \partial_t H &= -d_1 E + f_H. \end{aligned} \tag{22}$$

The terms of discrete Hodge operators $\star\epsilon$, $\star\mu$, $\star\sigma$, $\star\sigma^*$, discrete exterior derivatives d_1, d_1^T and source terms f_E, f_H are rewritten by each condition, as described in the following sections.

4.1 Scattered field formulation

In scattering problems, it is often convenient to operate on scattered fields instead of total electric and magnetic fields, as discussed in Holland (1977). The scattered field formulation is based on the linearity of Maxwell equations, decomposing the total fields into components of the incident and scattered fields. In the discrete form presentation this is

$$\begin{aligned} E &= E_{inc} + E_{sca}, \\ H &= H_{inc} + H_{sca}. \end{aligned}$$

The incident values E_{inc} and H_{inc} are known in advance. Usually they are described as simple analytical functions, usually like a polarized plane wave. The scattered field values E_{sca} and H_{sca} are the ones to be solved, while the total field values E and H fulfill the Maxwell equations (18). The Maxwell system is rewritten as

$$\begin{aligned} \star\sigma(E_{inc} + E_{sca}) + \star\epsilon\, \partial_t(E_{inc} + E_{sca}) &= d_1^T(H_{inc} + H_{sca}), \\ \star\sigma^*(H_{inc} + H_{sca}) + \star\mu\, \partial_t(H_{inc} + H_{sca}) &= -d_1(E_{inc} + E_{sca}). \end{aligned} \tag{23}$$

The incidence field is usually set to obey the Maxwell equations with certain material parameters ϵ_0, μ_0, σ_0 and σ_0^*. Thus, we assume that the discrete Maxwell equations with corresponding Hodge operators hold

$$\begin{aligned} \star\sigma_0 E_{inc} + \star\epsilon_0\, \partial_t E_{inc} &= d_1^T H_{inc}, \\ \star\sigma_0^* H_{inc} + \star\mu_0\, \partial_t H_{inc} &= -d_1 E_{inc}. \end{aligned}$$

We decrease this formulation from the Maxwell system (23) and get the non-homogeneous system

$$\begin{aligned} \star\sigma E_{sca} + \star\epsilon\, \partial_t E_{sca} &= d_1^T H_{sca} + f_E, \\ \star\sigma^* H_{sca} + \star\mu\, \partial_t H_{sca} &= -d_1 E_{sca} + f_H, \end{aligned}$$

where the source terms f_E and f_H are expressed by the incident wave data as

$$\begin{aligned} f_E &:= (\star\sigma_0 - \star\sigma) E_{inc} + (\star\epsilon_0 - \star\epsilon)\, \partial_t E_{inc}, \\ f_H &:= (\star\sigma_0^* - \star\sigma^*) H_{inc} + (\star\mu_0 - \star\mu)\, \partial_t H_{inc}. \end{aligned}$$

In scattering problems, the scatterer is usually the dielectric object, where material parameters differ from the incident material parameters. In other words, this is $\star\epsilon \neq \star\epsilon_0$, $\star\mu \neq \star\mu_0$, $\star\sigma \neq \star\sigma_0$ or $\star\sigma^* \neq \star\sigma_0^*$. This leads to the non-zero source terms $f_E \neq 0$ or $f_H \neq 0$ inside the scatterer. Thus, the scattered field formulation is a method to generate a non-zero field from the interior of the simulation domain.

4.2 Boundary conditions

Generally, the physical applications are defined in unbounded domains. To solve such problems numerically, the domain must be truncated to a finite computational domain, and this is achieved by introducing boundary conditions at the outer boundary. One of the greatest challenges in Yee-like schemes has been to efficiently and accurately model the extension of the domain to infinity. The **Silver-Müller absorbing boundary condition** has been introduced to achieve this goal. The Silver-Müller boundary condition is a first-order absorbing boundary condition; for a detailed description see Monk (2003); Hanouzet and Sesques (1993); Barucq et al. (1998). In this section, we introduce the Silver-Müller type boundary condition, which leads to two special cases to Dirichlet and Neumann type boundary conditions.

Let $\mathbf{n}$ be an outward normal vector on the domain boundary. The vector field formulation for the Silver-Müller boundary condition is then given by

$$\mathbf{H} \times \mathbf{n} = -\sqrt{\frac{\epsilon}{\mu}}\, (\mathbf{E} \times \mathbf{n}) \times \mathbf{n}.$$

The Silver-Müller boundary condition is implemented in the DEC framework by introducing and modifying the H^∂-component at the domain boundary, as

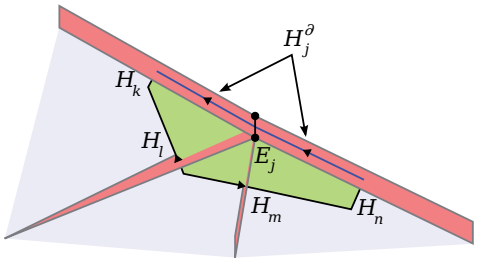


FIGURE 21 The additional magnetic field values H_j^∂ are introduced on the missing dual edges at the domain boundaries. The Ampére law is then written on the dual face elements using these values.

illustrated in Figure 21. This component completes the circulation for dual face, and the Ampére law is then written like in Equation (16). Let $\mathcal{E}_j$ be a primal edge on the boundary of mesh $\mathcal{K}$. As described in Section 2.4, we introduce its boundary dual edge as the dual of $\mathcal{E}_j$ on two dimensional boundary mesh $\partial\mathcal{K}$. The boundary dual edge is denoted by $*^\partial\mathcal{E}_j$. In this consideration, the dual $*^\partial\mathcal{E}_j$ is assumed to have a positive relative orientation compared to $\mathcal{E}_j$. The discrete presentation of the Silver-Müller condition is then

$$\frac{1}{|*^\partial\mathcal{E}_j|}\int_{*^\partial\mathcal{E}_j}\mathbf{H}\cdot d\mathbf{l} = -\sqrt{\frac{\epsilon}{\mu}}\frac{1}{|\mathcal{E}_j|}\int_{\mathcal{E}_j}\mathbf{E}\cdot d\mathbf{l}.$$

The integral on the right-hand side equals the discrete form value E_j, and the integral on the left-hand side corresponds to the missing magnetic field value H_j^∂. Using the Silver-Müller condition, the term H^∂ can be expressed as

$$H_j^\partial := \int_{*^\partial\mathcal{E}_j}\mathbf{H}\cdot d\mathbf{l} = -\sqrt{\frac{\epsilon}{\mu}}\frac{|*^\partial\mathcal{E}_j|}{|\mathcal{E}_j|}E_j. \tag{24}$$

We reformulate the Ampére law at the boundary edge $\mathcal{E}_j$ by completing the circular integration by term H_j^∂. The discrete Ampére law on this single element is then formulated as

$$\star\sigma_{j,j}E_j + \star\epsilon_{j,j}\,\partial_t E_j = (d_1^T H)_j + H_j^\partial. \tag{25}$$

Using the presentation of Equation (24), the term H_j^∂ is expressed as a product of term E_j. Thus, the Silver-Müller condition can be included in the conductivity term $\star\sigma_{j,j}$. The Silver-Müller boundary condition fits the discrete Maxwell formulation (22) by replacing the certain diagonal terms $\star\sigma_{j,j}$ with the Silver-Müller terms $\star\sigma_{j,j}^{\mathrm{SM}}$. The Silver-Müller terms are defined as

$$\star\sigma_{j,j}^{\mathrm{SM}} := \star\sigma_{j,j} + \sqrt{\frac{\epsilon}{\mu}}\frac{|*^\partial\mathcal{E}_j|}{|\mathcal{E}_j|}.$$

In scattering problems, the Silver-Müller boundary condition is usually applied for scattered fields $E_{sca} := E - E_{inc}$ and $H_{sca} := H - H_{inc}$. Using the scat-

tered field formulation, the Silver-Müller condition, which is expressed in Equation (24), can be written as

$$\left(H^{\partial} - H^{\partial}_{inc}\right)_j := -\sqrt{\frac{\epsilon}{\mu}} \frac{|{*}^{\partial}\mathcal{E}_j|}{|\mathcal{E}_j|} \left(E - E_{inc}\right)_j .$$

We transfer the H^{∂}_{inc} term on the right-hand side and apply the term H^{∂}_j in the discrete Ampére law (25). As a result, we obtain

$$\star\sigma^{\mathrm{SM}}_{j,j} E_j + \star\epsilon_{j,j}\, \partial_t E_j = (d_1^T H)_j + (f_E^{\mathrm{SM}})_j, \tag{26}$$

where the source term $(f_E^{\mathrm{SM}})_j$ reads as

$$(f_E^{\mathrm{SM}})_j := \sqrt{\frac{\epsilon}{\mu}} \frac{|(\mathcal{E}^*_{\partial})_j|}{|\mathcal{E}_j|} \left(E_{inc}\right)_j + \left(H^{\partial}_{inc}\right)_j .$$

The Silver-Müller type boundary conditions do not require additional memory storage. As an approximation of the radiation condition, the accuracy of the Silver-Müller boundary condition, however, crucially depends on the difference of the wave propagation direction and the boundary normal vector. Thus, the distance between the boundary and the radiation source has an important influence on the accuracy. As a result, there is a trade-off between accuracy and computational overhead due to the simulation of additional space.

The material parameters ϵ and μ of the Silver-Müller boundary condition can be considered to represent the material parameters at the external domain. Thus, the Silver-Müller condition can be employed to estimate an arbitrary material interface at the domain boundary. The special case $\epsilon \to \infty$ models the perfectly conducting material outside the simulation region. Sometimes, this condition is called the **Dirichlet boundary condition** with respect to the electric field. The discretization of the Dirichlet condition is obtained by dividing both sides of the Equation (26) by

$$\sqrt{\frac{\epsilon}{\mu}} \frac{|(\mathcal{E}^*_{\partial})_j|}{|\mathcal{E}_j|}.$$

Since $\epsilon \to \infty$, the resulting system is written as $E_j := (E_{inc})_j$. Naturally, in terms of Equation (22), this implies that $\star\sigma_{j,j} := 1$, $(f_E)_j := (E_{inc})_j$, $\star\epsilon_{j,j} := 0$ and $(d_1^T)_{j,k} := 0$ for all $k = 1, \ldots, n$.

The boundary condition with the zero permittivity $\epsilon = 0$ at the external domain is called the **Neumann boundary condition**. Applying the parameter $\epsilon = 0$ to Equation (26) gives the discretization of the Neumann condition as

$$\star\sigma_{j,j} E_j + \star\epsilon_{j,j}\, \partial_t E_j = (d_1^T H)_j + \left(H^{\partial}_{inc}\right)_j .$$

This is identical to general Ampére law (25) on boundary elements, where the boundary magnetic field values are defined as $H^{\partial}_j := \left(H^{\partial}_{inc}\right)_j$. Thus, the Neumann boundary condition affects magnetic field values similarly as the Dirichlet condition affects electric field values. Both conditions can be used to formulate perfectly reflecting boundaries.

4.3 Matched layer

An alternate approach for an absorbing boundary condition is to determine an absorbing layer outside the region of interest. The simplest implementation of such a layer is a medium of absorbing material, which is analogous to the physical treatment of the walls of an anechoic chamber. Ideally, the absorbing medium is reflectionless to all impinging waves over their full frequency spectrum. An early implementation of absorbing material layer, reported by Holland and Williams (1983), is called matched layer (ML). This tactic is based on conventional lossy dispersionless absorbing medium, where electric and magnetic conductivities σ and σ^* are set to satisfy the condition

$$\frac{\sigma}{\epsilon} = \frac{\sigma^*}{\mu} = \beta x. \tag{27}$$

Here x is the distance between the inner boundary and the current position. The factor β determines the degree of the absorption. The conductivities are amplified linearly to minimize the dispersion affected by the discretization. The factor β can be adjusted using knowledge of the wavelength, the discretization level and the thickness of the layer. The layer is easily implemented by the DEC by only changing the material parameters inside the ML. The layer is a good alternative for the first implementation of an absorbing layer because it can be run on any mesh discretization. The problem of this strategy is that the incident wave, which is not orthogonal to the layer surface, is not absorbed perfectly.

4.4 Perfectly matched layer

The perfectly matched layer (PML) is designed to absorb perfectly both orthogonal and non-orthogonal incident plane waves. PML was originally formulated for the Yee scheme by Berenger (1994). Berenger's formulation splits the electromagnetic fields into two unphysical fields in the PML region; thus, it is sometimes called split-field PML. The Berenger's idea of splitting fields has become popular, and the approach has several variations, including Modified Berenger PML (Chen et al., 1995) and a simplified PML (Sullivan, 1996). Another approach, in which the PML is described as an artificial anisotropic absorbing material, is called uniaxial PML or UPML, and it was first described by Gedney (1996). Although Berenger's formulation and UPML were initially derived by different approaches, both formulations were later shown to be equivalent with each other and with another approach, called the stretched-coordinate PML (Chew and Weedon, 1994; Teixeira and Chew, 1998). Today, PML is a standard method in Yee-like schemes to model electromagnetic wave interaction problems in unbounded domains. It can be applied to rectangular boundaries (Taflove and Hagness, 2005), on spherical boundaries (Teixeira and Chew, 1997) and on many other variations.

We derive the PML using the UPML formulation, which was originally written for FDTD framework by Gedney (1996). This formulation is derived for time-harmonic problems, where the material parameters can be expressed in complex formulation $\hat{\epsilon} = \epsilon + \frac{i}{\omega}\sigma$ and $\hat{\mu} = \mu + \frac{i}{\omega}\sigma^*$. In the original approach, each coordinate axis direction is assigned a specified conductivity, denoted by σ_x, σ_y and σ_z. Then both permittivity and permeability are represented as complex valued tensors $\hat{\epsilon} = \epsilon\bar{\bar{\epsilon}}$ and $\hat{\mu} = \mu\bar{\bar{\mu}}$, where

$$\bar{\bar{\epsilon}} = \bar{\bar{\mu}} = \begin{pmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_z s_x}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{pmatrix},$$

and

$$s_x = 1 + \frac{i\sigma_x}{\omega\epsilon}, \qquad s_y = 1 + \frac{i\sigma_y}{\omega\epsilon}, \qquad s_z = 1 + \frac{i\sigma_z}{\omega\epsilon}.$$

The UPML is not restricted to the previous Cartesian formulation. The layer can be formulated for an arbitrary orthogonal coordinate system, where the local coordinate axes are expressed by orthogonal unit vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. We assign the directional conductivities σ_x, σ_y and σ_z with each direction $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$, respectively. The rotated material tensor is then locally expressed as

$$\bar{\bar{\epsilon}} = \bar{\bar{\mu}} = \begin{pmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{pmatrix} \begin{pmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_z s_x}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{pmatrix} \begin{pmatrix} \mathbf{x}^T \\ \mathbf{y}^T \\ \mathbf{z}^T \end{pmatrix}.$$

The discretation error is decreased by amplifying the absorption terms linearly inside the PML region, as we did with the matched layer formulation. For simplicity, we assume that the layer is designed to absorb waves only in the direction of $\mathbf{x}$. This means that $\sigma_y = \sigma_z = 0$ and the conductivity σ_x have a linear relation to the depth x inside the layer. Starting from zero at the inner level $x = 0$, the conductivity is determined by the relation

$$\frac{\sigma_x}{\epsilon} = \beta x. \tag{28}$$

On corner regions, the conductivities σ_y and σ_z are treated separately in the same manner and the material tensors $\hat{\epsilon}$ and $\hat{\mu}$ are obtained by the previous consideration.

The UPML formulation is simply discretized by transforming the complex material parameters $\hat{\epsilon}$ and $\hat{\mu}$ to the corresponding real valued material parameters ϵ, μ, σ and σ^*. The real valued anisotropic material parameters are then transformed to diagonal discrete Hodge operators by the method of Section 3.3. Thus, the spatial discretization of PML can be expressed in the discrete Maxwell formulation of Equation (18). The anisotropic material parameters set certain demands for the mesh construction. The required relations between primal and dual elements are expressed specifically in Equation (17). The orthogonal duality is not a sufficient property, as the material parameters are anisotropic. Additionally, the primal and dual elements must be oriented in accordance with the local

absorption directions. In practice, this means that each edge element is either parallel or orthogonal to every local absorption direction **x**, **y** or **z**. Illustrations of such meshes are shown later in this thesis (see Figures 39 and 54).

With a closer look at the UPML formulation, one can easily see that certain diagonal components of the discrete Hodge operators $\star\sigma$ and $\star\sigma^*$ become negative (Gedney, 1996). This implies negative absorption for certain discrete form values. Unfortunately, the classical leapfrog procedure fails in time discretization of such a system. The time discretization of the PML system requires specific consideration, and the solution for this problem is presented in Section 5.3.

5 TIME DISCRETIZATION

In this chapter, we consider the time discretization of the spatially discretized Maxwell system (22). The time discretisation offers a forward-in-time simulation scheme, which is applied to solve time-dependent problems. The Yee's leapfrog time discretization scheme, which is described in Section 5.1, relies on finite difference approximations. By assuming a time-harmonic solution, these approximations can be replaced by exact finite differences, and the resulting harmonic leapfrog scheme is described in Section 5.2. The specific treatment of negative absorption terms, which occur in the PML discretization, is considered in Section 5.3. The equations for synchronizing the time-dependent solution on uniform time instance are expressed in Section 5.4. In time-harmonic problems, the harmonic leapfrog scheme produces exact time discretization. Thus, the accuracy of the harmonic leapfrog method is independent of the time step size Δt. Nevertheless, the stability criterion sets an upper bound for Δt, as discussed in Section 5.5. The conservation of the system energy is considered in Section 5.6, and Yee's leapfrog and the harmonic leapfrog schemes are compared by numerical experiments in Section 5.7.

5.1 Leapfrog discretization procedure

The discrete forms E and H of Maxwell system (22) are expressed as functions of time. Since we are considering time-dependent problems, we must proceed with the discretization of the time evolution. The time discretization is established with a staggered leapfrog-type iteration, which is a well known technique for Yee-like schemes (Yee, 1966; Taflove and Brodwin, 1975; Weiland, 1996; Mattiussi, 2000; Desbrun et al., 2005; Taflove and Hagness, 2005). In the leapfrog-type iteration, the values for the variables E and H are computed at separate time instances, as shown in Figure 22. The electric vector E^1 is first solved at a given time instance using the initial values E^0 and H^0. Then magnetic vector H^1 is solved at the next time instance using the updated initial values H^0 and E^1. This process is

repeated over and over again to obtain the forward-in-time simulation method.

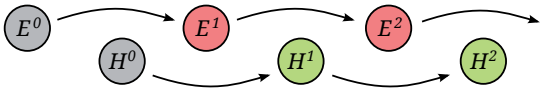


FIGURE 22 The basic idea of the leapfrog time discretization procedure.

Next, we formulate the equations for leapfrog style time discretization using the time step size Δt. Let t^0 be the initial time and recursively $t^k = t^{k-1} + \Delta t$. We define magnetic vectors H^k at time instance t^k and electric vectors E^k at half time step $\frac{\Delta t}{2}$ earlier. Formally, we write

$$\begin{aligned} E^k &:= E(t^k - \tfrac{\Delta t}{2}), \\ H^k &:= H(t^k). \end{aligned}$$

To discretize the time derivatives of the Maxwell system (22), we express the first equation at time instance t^k and the second equation at time instance $t^k + \frac{\Delta t}{2}$. The following finite difference approximations are applied

$$\begin{aligned} E(t^k) &:= \frac{E^k + E^{k+1}}{2}, \\ \partial_t E(t^k) &:= \frac{E^{k+1} - E^k}{\Delta t}, \\ H(t^k + \tfrac{\Delta t}{2}) &:= \frac{H^k + H^{k+1}}{2}. \\ \partial_t H(t^k + \tfrac{\Delta t}{2}) &:= \frac{H^{k+1} - H^k}{\Delta t}, \end{aligned} \tag{29}$$

Simply, by inserting these approximation at the Maxwell system (22), we obtain the relations

$$\begin{aligned} E^{k+1} &:= E^k + \left(\frac{\star\epsilon}{\Delta t} + \frac{\star\sigma}{2}\right)^{-1} \left[d_1^T H^k - \star\sigma E^k + f_E(t^k)\right], \\ H^{k+1} &:= H^k + \left(\frac{\star\mu}{\Delta t} + \frac{\star\sigma^*}{2}\right)^{-1} \left[-d_1 E^{k+1} - \star\sigma^* H^k + f_H(t^k + \tfrac{\Delta t}{2})\right]. \end{aligned} \tag{30}$$

These equations are now called the **value update equations** for the Yee's leapfrog time discretization method. The inverse matrices of the formulation are diagonal, since discrete Hodge operators are diagonal. Thus, the method is explicit; the values for each time step are determined from the values of the previous time step. The value updates can be performed without additional memory storage, simply by updating the vectors E_k and H_k to E^{k+1} and H^{k+1} at each iteration stage in the leapfrog manner.

5.2 Exact leapfrog discretization for time-harmonic problems

The Yee's leapfrog time discretization procedure, which is presented is Section 5.1, is based on finite difference approximation. The accuracy of the discretization de-

pends on the selection of Δt. By the time-harmonic assumption, the discretization error is known in advance, and it can be eliminated. Then one obtains an exact leapfrog time discretization scheme for time-harmonic problems. Even though some of the details here differ from what has been presented in the past, we note that this type of strategy has been used before in the FDTD context, for example by Ma and Chen (2005). We assume a time-harmonic solution, which can be expressed in a complex presentation $E(t) = \text{real}(\hat{E}e^{-i\omega t})$ and $H(t) = \text{real}(\hat{H}e^{-i\omega t})$. Next, we formulate **harmonic leapfrog** discretization scheme, which is in many ways similar to the Yee's leapfrog scheme. The k:th instances of discrete form vectors are expressed by a complex presentation as

$$\begin{aligned} E^k &:= \text{real}(\hat{E}e^{-i\omega(t^k - \frac{\Delta t}{2})}), \\ H^k &:= \text{real}(\hat{H}e^{-i\omega t^k}), \end{aligned}$$

where $t^k := t^0 + k\Delta t$ and Δt is the time step size. The next instance of the electric vector is expressed as $E^{k+1} = \text{real}(\hat{E}e^{-i\omega(t^k + \frac{\Delta t}{2})})$. Using this, we can formulate the following relations

$$\begin{aligned} E^{k+1} + E^k &= \text{real}\left[\hat{E}e^{-i\omega t^k}\left(e^{-i\omega\frac{\Delta t}{2}} + e^{i\omega\frac{\Delta t}{2}}\right)\right] = E(t^k)\left(2\cos\tfrac{\omega\Delta t}{2}\right), \\ E^{k+1} - E^k &= \text{real}\left[-i\omega\hat{E}e^{-i\omega t^k}\left(\frac{e^{-i\omega\frac{\Delta t}{2}} - e^{i\omega\frac{\Delta t}{2}}}{-i\omega}\right)\right] = \partial_t E(t^k)\left(\tfrac{2}{\omega}\sin\tfrac{\omega\Delta t}{2}\right), \end{aligned}$$

where the formula $\partial_t E(t^k) = \text{real}(-i\omega\hat{E}e^{-i\omega t^k})$ is applied in the last equality. With these relations, we obtain exact expressions for $E(t^k)$ and $\partial_t E(t^k)$. The similar consideration is applied for vector H to obtain $H(t^k + \frac{\Delta t}{2})$ and $\partial_t H(t^k + \frac{\Delta t}{2})$. Denoting $\phi := \frac{\omega\Delta t}{2}$, the resulting relations are written as

$$\begin{aligned} E(t^k) &= \frac{E^{k+1} + E^k}{2\cos\phi}, \\ \partial_t E(t^k) &= \frac{E^{k+1} - E^k}{\frac{2}{\omega}\sin\phi}, \\ H(t^k + \tfrac{\Delta t}{2}) &= \frac{H^{k+1} + H^k}{2\cos\phi}, \\ \partial_t H(t^k + \tfrac{\Delta t}{2}) &= \frac{H^{k+1} - H^k}{\frac{2}{\omega}\sin\phi}. \end{aligned} \tag{31}$$

The expressions are similar to the approximative Equations (29), but these equations are exact for a time-harmonic solution. The first equation of the Maxwell system (22) is written at time instance t^k, and the second equation is written at time instance $t^k + \frac{\Delta t}{2}$. Using the relations (31), we have

$$\begin{aligned} E^{k+1} &:= E^k + 2\left(\frac{\omega\star\epsilon}{\sin\phi} + \frac{\star\sigma}{\cos\phi}\right)^{-1}\left[d_1^T H^k - \frac{\star\sigma}{\cos\phi}E^k + f_E(t^k)\right], \\ H^{k+1} &:= H^k + 2\left(\frac{\omega\star\mu}{\sin\phi} + \frac{\star\sigma^*}{\cos\phi}\right)^{-1}\left[-d_1 E^{k+1} - \frac{\star\sigma^*}{\cos\phi}H^k + f_H(t^k + \tfrac{\Delta t}{2})\right]. \end{aligned} \tag{32}$$

The value update equations of time-harmonic leapfrog iteration are very similar to the value update equations (30) of Yee's leapfrog method. In fact, the correction of the harmonic leapfrog method can be applied by scaling the discrete Hodge operators $\star\epsilon$ and $\star\mu$ by $\frac{\phi}{\sin\phi}$ and the discrete Hodge operators $\star\sigma$ and $\star\sigma^*$ by $\frac{1}{\cos\phi}$. This means the Yee's leapfrog and the harmonic leapfrog discretization methods have the same convergence and energy conservation properties.

5.3 Treatment of negative absorption

In this section, we formulate the time discretization procedure for single discrete form values E_j and H_i, for which the corresponding Hodge terms $\star\sigma_{j,j}$ and $\star\sigma^*_{i,i}$ are negative. The negative terms occur, particularly when the UPML formulation is discretized. For details, see Section 4.4. Unfortunately, the classical leapfrog procedure fails in time discretization on term E_j and H_i, since the system becomes dispersive due to negative absorption. The time discretization at those elements can be carried out in number of ways, but we follow the procedure, which is found to be the most efficient by Gedney (1996). Assume that there is negative conductivity at term $\star\sigma_{j,j} < 0$. We write a single row of discretized Maxwell system (18) in complex formulation (20) as

$$\left(\star\epsilon_{j,j} + \tfrac{i}{\omega}\star\sigma_{j,j}\right)\partial_t\hat{E}_j = \left(d_1^T\hat{H}\right)_j.$$

We introduce an auxiliary variable $\hat{D}_j$ and reformulate the equation by the following two-step procedure

$$\begin{aligned}\partial_t\hat{D}_j &= \left(d_1^T\hat{H}\right)_j,\\ \partial_t\hat{E}_j &= \left(\star\epsilon_{j,j} + \tfrac{i}{\omega}\star\sigma_{j,j}\right)^{-1}\partial_t\hat{D}_j.\end{aligned}$$

The downside for this formulation is that it requires additional storage for the auxiliary variable $\hat{D}_j$ and it consist of two equations. We reformulate the latter equation such that there exists only real valued terms. The inverse of the complex expression can be rewritten, and we have

$$\partial_t\hat{E}_j = \frac{\star\epsilon_{j,j} - \frac{i}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2}\partial_t\hat{D}_j.$$

Using the time-harmonic assumption, we reformulate the remaining imaginary term by applying the relation $\frac{i}{\omega}\partial_t\hat{D}_j = \hat{D}_j$. Since the resulting equation includes only real valued factors, the system can be expressed by real valued discrete forms E and D. The two-step procedure is now written as

$$\begin{aligned}\partial_t D_j &= \left(d_1^T H\right)_j,\\ \partial_t E_j &= \frac{1}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2}\left(\star\epsilon_{j,j}\partial_t D_j - \star\sigma_{j,j}D_j\right).\end{aligned} \tag{33}$$

Now, we can apply the time discretization. The first equation can be discretized by either of the leapfrog methods, which are presented in Sections 5.1 and 5.2. Since the time-harmonic assumption is employed in the UPML formulation, we calculate the time discretization using the time-harmonic approach, described in Section 5.2. The second equation is discretized at t^k by replacing the terms $\partial_t E_j(t^k)$, $\partial_t D_j(t^k)$ and $D_j(t^k)$ by the exact formulations, which are similar to Equations (31). The terms are written as

$$\partial_t E_j(t^k) = \frac{E_j^{k+1} - E_j^k}{\frac{2}{\omega}\sin\phi}, \qquad \partial_t D_j(t^k) = \frac{D_j^{k+1} - D_j^k}{\frac{2}{\omega}\sin\phi}, \qquad D_j(t^k) = \frac{D_j^{k+1} + D_j^k}{2\cos\phi}.$$

The time discretized version of the second equation of System (33) is then

$$E_j^{k+1} - E_j^k = \frac{\star\epsilon_{j,j} - \frac{\tan\phi}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^{k+1} + \frac{-\star\epsilon_{j,j} - \frac{\tan\phi}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^k.$$

The value update procedure is obtained by the following three-step method, which can be implemented using one memory storage for both E_j and D_j values. The following equations are performed consecutively

$$\begin{aligned} E_j^{k+1/2} &:= E_j^k + \frac{-\star\epsilon_{j,j} - \frac{\tan\phi}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^k, \\ D_j^{k+1} &:= D_j^k + \frac{2\sin\phi}{\omega}\left(d_1^T H^k\right)_j, \\ E_j^{k+1} &:= E_j^{k+1/2} + \frac{\star\epsilon_{j,j} - \frac{\tan\phi}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^{k+1}. \end{aligned} \tag{34}$$

Notice that the term $E_j^{k+1/2}$ does not denote the discrete form value of E at halfway time instance. The term can be considered as an auxiliary variable in the three step procedure. This notation illustrates, that the term $E_j^{k+1/2}$ can replace the memory storage of the term E_j^k in the first step, and it is overwritten by E_j^{k+1} in the last step.

The similar value update procedure is needed for magnetic field values H_i, for which $\star\sigma_{i,i}^* < 0$. The auxiliary variable B_i is required, and the three-step value update procedure is written as

$$\begin{aligned} H_i^{k+1/2} &:= H_i^k + \frac{-\star\mu_{i,i} - \frac{\tan\phi}{\omega}\star\sigma_{i,i}^*}{\star\mu_{i,i}^2 + \frac{1}{\omega^2}\star{\sigma_{i,i}^*}^2} B_i^k, \\ B_i^{k+1} &:= B_i^k - \frac{2\sin\phi}{\omega}\left(d_1 E^{k+1}\right)_i, \\ H_i^{k+1} &:= H_i^{k+1/2} + \frac{\star\mu_{i,i} - \frac{\tan\phi}{\omega}\star\sigma_{i,i}^*}{\star\mu_{i,i}^2 + \frac{1}{\omega^2}\star{\sigma_{i,i}^*}^2} B_i^{k+1}. \end{aligned} \tag{35}$$

5.4 Obtaining a solution at synchronized time instance

In the leapfrog discretization procedure, the electric and magnetic field values are assigned at different time instances. The vector H^k is determined at time instance t^k, and the vector E^k is determined at time instance $t^k - \frac{\Delta}{2}$, which is half time step earlier. To obtain a full solution at time instance t^k, we apply a half time step transformation for the vector E. As illustrated in Figure 23, the transformation from E^k to $E(t^k)$ should be applied at the end of the leapfrog iteration to obtain the full solution at t^k. Before starting the leapfrog iteration again, the inverse transformation from $E(t^k)$ to E^k is applied in the beginning of the new iteration procedure.

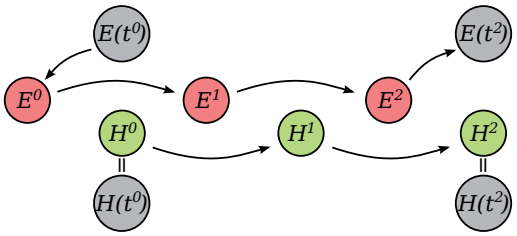


FIGURE 23 An illustration of leapfrog iteration, which is started from time instance t^0 and ended at time instance t^2. The transformation from $E(t^0)$ to E^0 is applied at the beginning of the iteration. The inverse transformation from E^2 to $E(t^2)$ is performed at the end of the leapfrog iteration.

The transformation equations depend on the selected leapfrog procedure. For the Yee's leapfrog method (see Section 5.1), the transformation is obtained by using the first approximation of Equations (29) and the first formulation of Equations (30). Applying the equations consecutively, we obtain

$$E(t^k) := \frac{E^k + E^{k+1}}{2} = E^k + \left(\frac{2\star\epsilon}{\Delta t} + \star\sigma\right)^{-1} \left[d_1^T\, H^k - \star\sigma\, E^k + f_E(t^k)\right].$$

Now vector E^k can be transformed to the solution $E(t^k)$, which is synchronized with the H^k vector. Likewise, the vector $E(t^k)$ can be transformed to vector E^k by the inverse transformation, which is obtained by modifying the previous equation. We have

$$E^k := E(t^k) - \left(\frac{2\star\epsilon}{\Delta t}\right)^{-1} \left[d_1^T\, H^k - \star\sigma\, E(t^k) + f_E(t^k)\right].$$

We formulate the transformation equations for the time-harmonic leapfrog system, which is described in Section 5.2. After applying consecutively the first

equation of Expressions (31) and the first equation of System (32), we obtain

$$\begin{aligned} E(t^k) &= \frac{E^{k+1} + E^k}{2\cos\phi} \\ &= \frac{1}{\cos\phi}\left(E^k + \left(\frac{\omega\star\epsilon}{\sin\phi} + \frac{\star\sigma}{\cos\phi}\right)^{-1}\left[d_1^T H^k - \frac{\star\sigma}{\cos\phi}E^k + f_E(t^k)\right]\right). \end{aligned}$$

By modifying this equation, we get forward and backward relations

$$\begin{aligned} E(t^k) &:= E^k + \left(\frac{\omega\cos\phi\star\epsilon}{\sin\phi} + \star\sigma\right)^{-1}\left[d_1^T\, H^k + \alpha E^k + f_E(t^k)\right], \\ E^k &:= E(t^k) - \left(\frac{\omega\star\epsilon}{\sin\phi}\right)^{-1}\left[d_1^T H^k + \alpha E(t^k) + f_E(t^k)\right], \end{aligned} \tag{36}$$

where

$$\alpha := \frac{\omega(1-\cos\phi)\star\epsilon}{\sin\phi} - \star\sigma.$$

The transformation from E^k to $E(t^k)$ is obtained by the first equation. The inverse transformation from $E(t^k)$ to E^k can be performed by the second equation.

If the value update of E_j is treated by the three-step procedure, presented in Section 5.3, the value update procedure include additional term D_j^k. In this case, both values E_j^k and D_j^k need to be transformed to $E_j(t^k)$ and $D_j(t^k)$ to obtain the result at synchronized time instance t^k. Since the value D_j is updated by the harmonic leapfrog procedure, the transformation of D_j is obtained in a similar manner as that of Equations (36). The transformation of E_j is obtained by writing the second equation of (33) at time instance $t^k - \frac{\Delta t}{4}$. We apply the exact formulations

$$\begin{aligned} \partial_t E(t^k - \tfrac{\Delta t}{4}) &= \frac{E(t^k) - E^k}{\frac{2}{\omega}\sin(\phi/2)}, \qquad \partial_t D(t^k - \tfrac{\Delta t}{4}) = \frac{D(t^k) - D^k}{\frac{2}{\omega}\sin(\phi/2)}, \\ D(t^k - \tfrac{\Delta t}{4}) &= \frac{D(t^k) + D^k}{2\cos(\phi/2)}, \end{aligned}$$

which are derived similarly to Equations (31). The following time discretization is obtained

$$E_j(t^k) - E_j^k = \frac{\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j(t^k) + \frac{-\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^k.$$

This relation can be applied in either direction. The transformation should be separated into the three-step procedure to minimize the amount of required ad-

ditional storage. The transformation from E_j^k, D_j^k to $E_j(t^k), D_j(t^k)$ is obtained by

$$
\begin{aligned}
E_j^{k+1/4} &:= E_j^k + \frac{-\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^k, \\
D_j(t^k) &:= D_j^k + \frac{\tan\phi}{\omega}\left[\left(d_1^T H^k\right)_j + \frac{\omega(1-\cos\phi)}{\sin\phi} D_j^k\right], \\
E_j(t^k) &:= E_j^{k+1/4} + \frac{\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j(t^k).
\end{aligned}
$$

Vice versa, the transformation from $E_j(t^k), D_j(t^k)$ to E_j^k, D_j^k is obtained by

$$
\begin{aligned}
E_j^{k+1/4} &:= E_j(t^k) - \frac{\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j(t^k), \\
D_j^k &:= D_j(t^k) - \frac{\sin\phi}{\omega}\left[\left(d_1^T H^k\right)_j + \frac{\omega(1-\cos\phi)}{\sin\phi} D_j(t^k)\right], \\
E_j^k &:= E_j^{k+1/4} - \frac{-\star\epsilon_{j,j} - \frac{\tan(\phi/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^k.
\end{aligned}
$$

5.5 Stability criterion

Stability of the leapfrog manner iteration depends on the selection of time step size Δt. If the time step is too long, it will cause E and H components to bounce between negative and positive values, and the absolute values will increase without limit as the simulation proceeds. The correct numerical stability criterion for Yee's algorithm was first reported by Taflove and Brodwin (1975). The so called Courant-Friedrichs-Lewy (CFL) condition can also be written for unstructured grids, as discussed by Schuhmann and Weiland (1998) and Cinalli and Schiavoni (2006). We assume that the Yee's leapfrog discretization procedure (30) is applied with the non-negative absorbing terms $\star\sigma_{j,j}$ and $\star\sigma^*{}_{i,i}$. If χ_{max} is the largest eigenvalue of the system matrix $\star\epsilon^{-1} d_1^T \star\mu^{-1} d_1$, then the CFL limit for the time step size is

$$
\Delta t < \frac{2}{\sqrt{\chi_{max}}}. \tag{37}
$$

The same CFL limit can also be applied for the harmonic leapfrog discretization formulas (32). The harmonic leapfrog method can be obtained by modifying the Yee's leapfrog method by multiplying the discrete Hodge operators $\star\epsilon$ and $\star\mu$ by $\frac{\phi}{\sin\phi}$ and multiplying the discrete Hodge operators $\star\sigma$ and $\star\sigma^*$ by $\frac{1}{\cos\phi}$. Thus, the corresponding system matrix of the harmonic leapfrog system is

$$
\frac{\sin^2\phi}{\phi^2}\star\epsilon^{-1} d_1^T \star\mu^{-1} d_1.
$$

The system's largest eigenvalue equals to $\frac{\sin^2\phi}{\phi^2}\chi_{max}$. This is bounded from above by χ_{max}, which means the CFL condition (37) is also sufficient for the harmonic leapfrog discretization.

It is important to use a tight limit for Δt to obtain an efficient time-dependent simulation method. The use of too small time step size leads to unnecessary consumption of time in the iteration process. The harmonic leapfrog method offers an exact time discretization for time-harmonic problems. The numerical experiments in Section 5.7 show that the solution method is then independent of Δt, provided that the stability condition is satisfied. In time-harmonic problems, it is obvious that one should use the maximal time step size, which fulfills the CFL condition. With the Yee's leapfrog method, the result varies according to the selected time step size.

5.6 Conservation of the energy norm

In this section, we introduce the concept of the energy norm. For any Maxwell system, the energy norm should be defined such that the energy is conserved in a system, where absorption and emission are excluded. First, we describe the energy norm for the discretized Maxwell system (18), which is a variation of System (22) with homogeneous Neumann boundary conditions at the boundary. We multiply the first equation of System (18) from left by E^T and the second equation from left by H^T. Then we obtain

$$E^T \star\sigma E + E^T \partial_t \star\epsilon E = E^T d_1^T H,$$
$$H^T \star\sigma^* H + H^T \partial_t \star\mu H = -H^T d_1 E.$$

We sum these equations together and see that the d_1 terms on the right-hand side vanish. The second terms on the left-hand side are considered as the time derivatives of quadratic functions $\frac{1}{2}E^T\star\epsilon E$ and $\frac{1}{2}H^T\star\mu H$, respectively. The resulting system is written as

$$\frac{d}{dt}\left(\tfrac{1}{2}E^T\star\epsilon E + \tfrac{1}{2}H^T\star\mu H\right) = -E^T\star\sigma E - H^T\star\sigma^* H.$$

If the conductivity terms of $\star\sigma$ and $\star\sigma^*$ are zero, the system implies the conservation of the derived value on the left-hand side. Inspired by this derivation, we write the energy norm for the spatially discretized Maxwell system (22) as

$$P := \frac{1}{2}\left(E^T\star\epsilon E + H^T\star\mu H\right). \tag{38}$$

The similar formulation for the energy norm is obtained by Bossavit (2005, Chapter III).

Next, we perform the similar consideration on the time discretized Maxwell system. For simplicity, we consider the Yee's leapfrog Equations (30) without the

absorbing or source terms, meaning $\star\sigma = 0$, $\star\sigma^* = 0$, $f_E = 0$ and $f_H = 0$. The system is then written as

$$\star\epsilon \left(E^{k+1} - E^k \right) = \Delta t d_1^T H^k,$$
$$\star\mu \left(H^{k+1} - H^k \right) = -\Delta t d_1 E^{k+1}.$$

The consecutive instances of the first equation are summed together and multiplied from the left by $(E^{k+1})^T$. The second equation is multiplied from the left by $(H^{k+1} + H^k)^T$. Our results were as follows

$$(E^{k+1})^T \star\epsilon \left(E^{k+2} - E^{k+1} + E^{k+1} - E^k \right) = (E^{k+1})^T \Delta t d_1^T (H^{k+1} + H^k),$$
$$(H^{k+1} + H^k)^T \star\mu \left(H^{k+1} - H^k \right) = -(H^{k+1} + H^k)^T \Delta t d_1 E^{k+1}.$$

The right-hand side terms of the equations are opposite to each other. Thus, these terms vanish when the equations are summed together. After reorganizing the terms, we get the following formulation

$$(E^{k+2})^T \star\epsilon E^{k+1} + (H^{k+1})^T \star\mu H^{k+1} = (E^{k+1})^T \star\epsilon E^k + (H^k)^T \star\mu H^k.$$

The formulation is similar on both sides of the equation. The indices $k+2$ and $k+1$ on the left-hand side correspond to the indices $k+1$ and k on the right-hand side, respectively. Thus, the energy of the leapfrog style formulation can be defined in this manner. Similarly to the time-continuous energy norm of Equation (38), we define the energy norm for the leapfrog discretization at time instance t^k as

$$P^k := \frac{1}{2} \left((E^{k+1})^T \star \epsilon E^k + (H^k)^T \star \mu H^k \right). \tag{39}$$

5.7 Numerical comparison of the leapfrog methods

In this experiment, we try to indicate the properties of Yee's leapfrog time stepping scheme and the harmonic leapfrog method in a simulation of a time-harmonic problem. We are modeling a circularly polarized harmonic plane wave of wavelength $\lambda = 1$ and time period $T = 1$. The wave is propagating to the x-direction through a simulation domain. More specifically, the components of the wave are expressed as

$$\mathbf{E}(x,t) = \begin{pmatrix} 0 \\ \cos(2\pi(x-t)) \\ \sin(2\pi(x-t)) \end{pmatrix}, \qquad \mathbf{H}(x,t) = \begin{pmatrix} 0 \\ -\sin(2\pi(x-t)) \\ \cos(2\pi(x-t)) \end{pmatrix}. \tag{40}$$

The simulation domain is a cube with edges of length 2 oriented in the directions of the coordinate axis. The domain is discretized by a regular cubic grid, where each edge element is 0.05 in length. The material parameters in the entire simulation domain are $\epsilon = \mu = 1$. The boundary of the domain is modeled as an

absorbing Silver-Müller type absorbing boundary condition, which is discussed in Section 4.2. Using the scattered field formulation, the system has source terms at the boundaries.

The simulation starts from zero initial fields $E(t^0) = 0$ and $H(t^0) = 0$, and the fields are then updated as the time-dependent simulation proceeds. On thousand total wave periods are iterated forward in time to obtain a stable result. Both the Yee's leapfrog and the harmonic leapfrog equations (30) and (32) are employed in the discretized system. The time step size Δt is varied, dividing the wave period $T = 1$ into 36, 42, 50, 70, 100, 140 and 200 time steps. The value $\Delta t = T/36$ was found to be the largest time step size to produce convergence. The solution is compared to the exact plane wave solution at the current time instance by integrating the precise vector fields $(\mathbf{E}, \mathbf{H})$ over mesh elements. The relative electric error δE, the relative magnetic error δH and the relative solution error δS are computed using the following formulations

$$\begin{aligned}
\delta E &:= \sqrt{\frac{\sum_j \star\epsilon_{j,j} \left(E_j - \int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l} \right)^2}{\sum_j \star\epsilon_{j,j} \left(\int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l} \right)^2}}, \qquad \delta H := \sqrt{\frac{\sum_i \star\mu_{i,i} \left(H_i - \int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l} \right)^2}{\sum_i \star\mu_{i,i} \left(\int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l} \right)^2}}, \\
\delta S &:= \sqrt{\frac{\sum_j \star\epsilon_{j,j} \left(E_j - \int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l} \right)^2 + \sum_i \star\mu_{i,i} \left(H_i - \int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l} \right)^2}{\sum_j \star\epsilon_{j,j} \left(\int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l} \right)^2 + \sum_i \star\mu_{i,i} \left(\int_{\mathcal{E}_i^*} \mathbf{H} \cdot d\mathbf{l} \right)^2}}.
\end{aligned} \tag{41}$$

The results are also compared to a reference solution, which represents the best solution of the current spatial discretization level. The reference solution is computed using time-harmonic leapfrog equations with very small time step size $\Delta t = T/1000$. The relative error is obtained by comparing the discrete values E_j and H_i to the corresponding reference solution E_j^{ref} and H_i^{ref} by the following formulations

$$\begin{aligned}
\delta E^{ref} &:= \sqrt{\frac{\sum_j \star\epsilon_{j,j} \left(E_j - E_j^{ref} \right)^2}{\sum_j \star\epsilon_{j,j} \left(E_j^{ref} \right)^2}}, \qquad \delta H^{ref} := \sqrt{\frac{\sum_i \star\mu_{i,i} \left(H_i - H_i^{ref} \right)^2}{\sum_i \star\mu_{i,i} \left(H_i^{ref} \right)^2}}, \\
\delta S^{ref} &:= \sqrt{\frac{\sum_j \star\epsilon_{j,j} \left(E_j - E_j^{ref} \right)^2 + \sum_i \star\mu_{i,i} \left(H_i - H_i^{ref} \right)^2}{\sum_j \star\epsilon_{j,j} \left(E_j^{ref} \right)^2 + \sum_i \star\mu_{i,i} \left(H_i^{ref} \right)^2}}.
\end{aligned} \tag{42}$$

The errors are illustrated as functions of the time step size in Figure 24. The comparison to the reference solution (norms δE^{ref} and δH^{ref}) indicates that the harmonic leapfrog method produces a solution, which is independent of the time step size Δt. The largest error compared to the reference is approximately 0.0020%. With Yee's leapfrog method, the solution advances towards the reference solution as the time step size decreases. The error seems to approximately

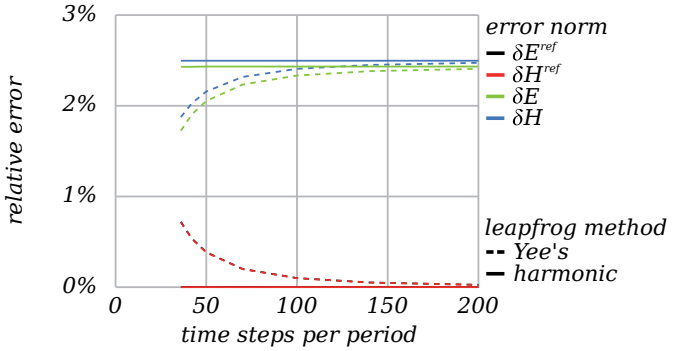


FIGURE 24 The error norms δE^{ref}, δH^{ref}, δE and δH are illustrated as functions of the time step number. Both the Yee's leapfrog and the harmonic leapfrog formulations are considered. The harmonic leapfrog method seems to be independent of the time step size as long as the stability limit is realized.

obey the formula $c\Delta t^2$ having the value 0.025% at $\Delta t = T/200$. The errors δE^{ref} and δH^{ref} behave similarly to each other.

The comparison to the exact solution (norms δE and δH) indicates that the most accurate solution is actually produced by Yee's leapfrog method using the maximal time step size $\Delta t = T/36$. In this case, the error in time discretization improves the accuracy of the solution. This can be explained by considering the methods more closely. Compared to Yee's leapfrog method, the harmonic leapfrog strategy multiplies the material parameters ϵ and μ by the factor $\frac{\phi}{\sin\phi} > 1$. This means the simulated wavelength $\lambda = \frac{T}{\sqrt{\epsilon\mu}}$ is shortened compared to the Yee's procedure. Thus, the inaccurate Yee's leapfrog method produces a wavelength that is too long. In Chapter 8, we will show that the Yee's Hodge operator yields too short wavelength, which is the inverse effect. The error of Yee's leapfrog method is then opposite the error of Yee's Hodge operator, which means the errors are deductive.

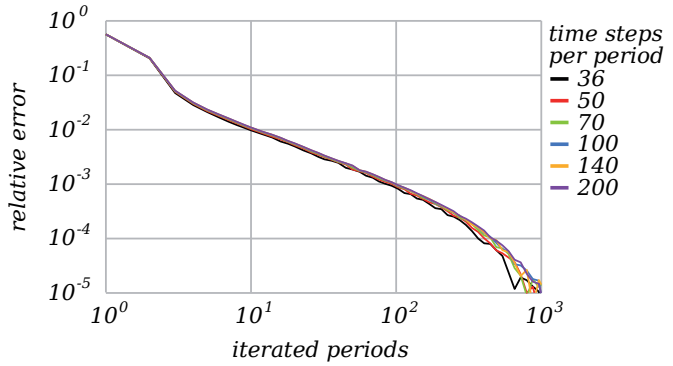


FIGURE 25 The solution error δS^{ref} is plotted as a function of iterated periods to illustrate the convergence speed of the time-dependent simulation. The time step size has no remarkable effect on the convergence speed.

In the harmonic leapfrog method, the time step size has no effect on the accuracy. Next, we consider what effect the time step size has on the convergence speed of the time-dependent iteration. During the harmonic leapfrog iteration, the intermediate solutions at several time instances are compared to the reference result. The progress of the solution error δS^{ref} is illustrated in Figure 25. The results indicate that the convergence speed is almost independent of the time step size. Thus, we cannot name a reason to use a shorter time step size than necessary for the stability criterion. The maximal time step size implies the minimal amount of iterations per time period, which indicates the minimal time consumption for time-dependent iteration. With the harmonic leapfrog method, we recommend the use of the maximal time step size as long as stability condition is realized.

6 FROM DISCRETE FORMS TO NUMERICAL SOLUTIONS

The DEC simulations are carried out by discrete differential forms, which assign an integral value to each mesh element. In physical applications, the discrete form expression is hardly ever the desired formulation for the numerical solution. In this chapter, we consider methods to transform the discrete solutions into several other formulations. In Section 6.1, we introduce a method to transform the time-dependent leapfrog solution into a complex time-harmonic expression. In Section 6.2, we present a method to estimate the vector fields at the desired position. In scattering problems, the result is usually expressed as a far field solution. Thus, we present the near to far field transformation in Section 6.3. The far field Mueller matrix is formulated in Section 6.4.

6.1 Time-harmonic solution

A simulation of a time-harmonic problem can be carried out efficiently by the leapfrog time discretization procedure, where the time-harmonic source terms are applied. The result of a forward-in-time simulation can be transformed into the complex presentation of a time-harmonic solution using the following consideration. We denote a solution of time-dependent leapfrog iteration by $E^k = E(t^k - \frac{\Delta t}{2})$ and $H^k = H(t^k)$. With a time-harmonic assumption, the solution can be expressed as $E^k = \text{real}(\hat{E}^k)$ and $H^k = \text{real}(\hat{H}^k)$, where the complex valued terms are defined as

$$\hat{E}^k := \hat{E} e^{-i\omega(t^k - \frac{\Delta t}{2})},$$
$$\hat{H}^k := \hat{H} e^{-i\omega t^k}.$$

After performing l more leapfrog iterations, the new instance of the solution can be expressed as $E^{k+l} = \text{real}(\hat{E}^k e^{-i\omega l \Delta t})$ and $H^{k+l} = \text{real}(\hat{H}^k e^{-i\omega l \Delta t})$. These relations can be reformulated by expressing the complex exponential function by

trigonometric operators, as

$$E^{k+l} = E^k \cos(\omega l \Delta t) + \text{imag}(\hat{E}^k) \sin(\omega l \Delta t),$$
$$H^{k+l} = H^k \cos(\omega l \Delta t) + \text{imag}(\hat{H}^k) \sin(\omega l \Delta t).$$

By assuming that the term $\sin(\omega l \Delta t)$ is not zero, the imaginary terms can be written as

$$\text{imag}(\hat{E}^k) = \frac{E^{k+l} - E^k \cos(\omega l \Delta t)}{\sin(\omega l \Delta t)},$$
$$\text{imag}(\hat{H}^k) = \frac{H^{k+l} - H^k \cos(\omega l \Delta t)}{\sin(\omega l \Delta t)}.$$

Finally, the expression for the complex terms $\hat{E}$ and $\hat{H}$ is obtained by the following consideration,

$$\hat{E} = \hat{E}^k e^{i\omega(t^k - \frac{\Delta t}{2})} = \left(E^k + i \frac{E^{k+l} - E^k \cos(\omega l \Delta t)}{\sin(\omega l \Delta t)} \right) e^{i\omega(t^k - \frac{\Delta t}{2})},$$
$$\hat{H} = \hat{H}^k e^{i\omega t^k} = \left(H^k + i \frac{H^{k+l} - H^k \cos(\omega l \Delta t)}{\sin(\omega l \Delta t)} \right) e^{i\omega t^k}.$$

Within this approach, the complex discrete form vectors are computed using two separate time instances of the leapfrog solution. If the leapfrog solution obeys a time-harmonic formulation exactly, then the consecutive solutions can be used, and $l = \pm 1$. In practice, the leapfrog solution is rarely perfectly time-harmonic due to unfinished iteration or numerical errors. Thus, the absolute of the divider term $|\sin(\omega l \Delta t)|$ should be maximized to obtain the best approximation. The integer l should then be selected such that the term $\pm \omega l \Delta t$ is as close to the right angle $\frac{\pi}{2}$ as possible. In other words, $l \approx \pm \frac{T}{4\Delta t}$.

6.2 From discrete forms to vector fields

The transformation from discrete differential forms to smooth differential forms is always an approximative operator. The transformation can be applied using Whitney forms (Whitney, 1957), and the idea is presented briefly in Hirani (2003) and Desbrun et al. (2008); with more detail in Bossavit (2005). Next, we present a flexible method to compute approximative vector field values at any domain position using discrete form values.

To obtain the electric field vector at position $\mathbf{p}$, we first find at least three linearly independent edge elements $\mathcal{E}_j$ in the neighborhood of position $\mathbf{p}$. For example, one could find the node element nearest to position $\mathbf{p}$ and select the edges, which are connected to the node. The indices of these edges are listed in set $\mathcal{J}$. We assume a locally constant electric field vector $\mathbf{E} \in R^3$. Then each

discrete form value is presented by

$$E_j = \int_{\mathcal{E}_j} \mathbf{E} \cdot d\mathbf{l} = \mathbf{E} \cdot \int_{\mathcal{E}_j} d\mathbf{l} =: \mathbf{E} \cdot \mathbf{l}_{\mathcal{E}_j},$$

where the vector $\mathbf{l}_{\mathcal{E}_j}$ is the orientation vector of element $\mathcal{E}_j$, meaning the vector between two boundary nodes. We search for a vector $\mathbf{E}$ such that the vector estimates the discrete form values as closely as possible. We formulate a least squares problem, where the cost function to be minimized is

$$J_P = \sum_{j \in \mathcal{J}} \frac{\star\epsilon_{j,j}}{2} \left(\mathbf{E} \cdot \mathbf{l}_{\mathcal{E}_j} - E_j \right)^2.$$

The term J_P is minimized by the weighted linear least square method, as described by Lawson and Hanson (1974). The solution is obtained by solving a linear group of three equations, which are determined by

$$\left(\sum_{j \in \mathcal{J}} \star\epsilon_{j,j} \mathbf{l}_{\mathcal{E}_j}^T \mathbf{l}_{\mathcal{E}_j} \right) \mathbf{E} = \sum_{j \in \mathcal{J}} \star\epsilon_{j,j} \mathbf{l}_{\mathcal{E}_j} E_j.$$

The computation of vector field $\mathbf{H}$ is carried out in similar manner after defining the set $\mathcal{J}$ of dual edge elements $\mathcal{E}_i^*$.

6.3 Near to far field transformation

A far field solution is usually required in scattering problems. The far field solution is obtained from a near field result by applying a powerful and flexible near field to far field transformation, which was first applied to the FDTD method by Umashankar and Taflove (1982; 1983). The scattered far fields are given by transformation of the equivalent currents over the free-space Green's function (for a background of this method, we refer to Bladel, 1964; Harrington, 1961; Jackson, 1975). For the full derivation of the far field transformation, we refer to Taflove and Hagness (2005, Chapter 8). In this section, we introduce a practical method to compute the far field result discretely within the DEC framework.

First, a harmonic near field solution must be computed. The harmonic solution can be obtained by a time-dependent simulation using the time-harmonic source terms. The time-dependent solution can be transformed to the time-harmonic solution by the method described in Section 6.1. The harmonic result is presented using the complex discrete form components $\hat{E}_k$ and $\hat{H}_l$. The harmonic near field data is collected on a boundary Γ, which encloses the scatterer object. The material parameters ϵ and μ are assumed to be constant outside the enclosed area and also in the small neighborhood of the boundary Γ. In scattering problems, a narrow layer must be left between the scatterer and Γ to obtain reliable near field data.

In our implementation, the enclosing boundary Γ is a union of primal face elements $\mathcal{F}_j$, where the indices j are listed in a set $\mathcal{J}$. The tangential vectors $\hat{\mathbf{E}}_j$ and $\hat{\mathbf{H}}_j$ are approximated on each boundary face $\mathcal{F}_j$ by applying the form to vector transformation method, which is described in Section 6.2. The tangential electric field component can be computed using only the edge elements on the boundary of $\mathcal{F}_j$. The tangential magnetic field component is computed using the dual edges, which are linked to the neighboring dual node elements, as illustrated in Figure 26.

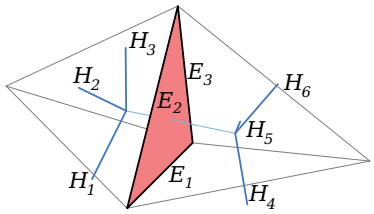


FIGURE 26 Neighboring edge and dual edge elements are utilized for the approximation of tangential vector field components $\hat{\mathbf{E}}_j$ and $\hat{\mathbf{H}}_j$ on a face element $\mathcal{F}_j$. Tangential vector fields are computed from complex discrete form values $\hat{E}_k$ and $\hat{H}_l$ by the least squares transformation method of Section 6.2.

Following the derivation presented in Taflove and Hagness (2005), the scattered far field is obtained by the field equivalence theorem, which is described by Schelkunoff (1951). Let $\mathbf{n}_j$ be the normal vector of $\mathcal{F}_j$ pointing outside of the closed volume. Then, the equivalent electric and magnetic currents $\hat{\mathbf{J}}_j$ and $\hat{\mathbf{M}}_j$ are computed on each surface $j \in \mathcal{J}$ by

$$\hat{\mathbf{J}}_j = \mathbf{n}_j \times \hat{\mathbf{H}}_j,$$
$$\hat{\mathbf{M}}_j = \hat{\mathbf{E}}_j \times \mathbf{n}_j.$$

We consider the scattering direction $\mathbf{x}$ with its associated orthogonal polarisation directions $\mathbf{y}$ and $\mathbf{z}$. The directions are presented by polar coordinates as

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta\cos\phi & \sin\theta\sin\phi \\ -\sin\theta & \cos\theta\cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}.$$

The far field potentials are computed by discrete integration of the electric and magnetic currents over boundary Γ and multiplied by harmonic wave term $e^{i\kappa\mathbf{x}\cdot\mathbf{p}_j}$. More explicitly, we write the far field potentials as

$$\hat{\mathbf{N}} = \sum_{j\in\mathcal{J}} |\mathcal{F}_j| e^{i\kappa\mathbf{x}\cdot\mathbf{p}_j} \hat{\mathbf{J}}_j,$$
$$\hat{\mathbf{L}} = \sum_{j\in\mathcal{J}} |\mathcal{F}_j| e^{i\kappa\mathbf{x}\cdot\mathbf{p}_j} \hat{\mathbf{M}}_j.$$

Here $\mathbf{p}_j$ is the center position of the face $\mathcal{F}_j$, and $|\mathcal{F}_j|$ is the area of surface $\mathcal{F}_j$. The angular wave number $\kappa = \omega\sqrt{\epsilon\mu}$ determines the far field wavelength. The far

field potentials are applied to determine the far field terms

$$\begin{aligned}
\hat{E}_{\mathbf{y}} &= -\hat{\mathbf{L}} \cdot \mathbf{z} - \sqrt{\frac{\mu}{\epsilon}} \hat{\mathbf{N}} \cdot \mathbf{y}, \\
\hat{E}_{\mathbf{z}} &= \hat{\mathbf{L}} \cdot \mathbf{y} - \sqrt{\frac{\mu}{\epsilon}} \hat{\mathbf{N}} \cdot \mathbf{z}, \\
\hat{H}_{\mathbf{y}} &= \hat{\mathbf{N}} \cdot \mathbf{z} - \sqrt{\frac{\epsilon}{\mu}} \hat{\mathbf{L}} \cdot \mathbf{y}, \\
\hat{H}_{\mathbf{z}} &= -\hat{\mathbf{N}} \cdot \mathbf{y} - \sqrt{\frac{\epsilon}{\mu}} \hat{\mathbf{L}} \cdot \mathbf{z},
\end{aligned} \tag{43}$$

which represent the scaled electric and magnetic components on directions **y** and **z** at the far field region. The far field components on direction **x** are zero, since **x** is the considered wave propagation direction. Using the terms of Equations (43), the electric and magnetic far fields $\hat{\mathbf{E}}$ and $\hat{\mathbf{H}}$ in direction **x** are obtained by relations

$$\begin{aligned}
\lim_{r\to\infty} \frac{4\pi r e^{i\kappa r}}{i\kappa} \hat{\mathbf{E}} &= \mathbf{y}\hat{E}_{\mathbf{y}} + \mathbf{z}\hat{E}_{\mathbf{z}}, \\
\lim_{r\to\infty} \frac{4\pi r e^{i\kappa r}}{i\kappa} \hat{\mathbf{H}} &= \mathbf{y}\hat{H}_{\mathbf{y}} + \mathbf{z}\hat{H}_{\mathbf{z}}.
\end{aligned}$$

The values of electric and magnetic far fields can be utilized to obtain the scattering power in the **x** direction. The scattering power can be written in the formula $P_{sca} = \frac{1}{2}|\hat{\mathbf{E}} \times \hat{\mathbf{H}}|$. Using this formula, we get the following equation in the far field region

$$P_{sca} = \left| \frac{i\kappa}{4\pi r e^{i\kappa r}} \right|^2 \left(\mathbf{y}\hat{E}_{\mathbf{y}} + \mathbf{z}\hat{E}_{\mathbf{z}}\right) \times \left(\mathbf{y}\hat{H}_{\mathbf{y}} + \mathbf{z}\hat{H}_{\mathbf{z}}\right) = \frac{\kappa^2}{16\pi^2 r^2} \left(\hat{E}_{\mathbf{y}}\hat{H}_{\mathbf{z}} - \hat{E}_{\mathbf{z}}\hat{H}_{\mathbf{y}}\right).$$

Since $\hat{H}_{\mathbf{z}} = \sqrt{\frac{\epsilon}{\mu}}\hat{E}_{\mathbf{y}}$ and $\hat{H}_{\mathbf{y}} = -\sqrt{\frac{\epsilon}{\mu}}\hat{E}_{\mathbf{z}}$, we can express the scattering power in either of the following two ways

$$P_{sca} = \frac{\sqrt{\epsilon}\kappa^2}{32\sqrt{\mu}\pi^2 r^2} \left(\hat{E}_{\mathbf{y}}^2 + \hat{E}_{\mathbf{z}}^2\right) = \frac{\sqrt{\mu}\kappa^2}{32\sqrt{\epsilon}\pi^2 r^2} \left(\hat{H}_{\mathbf{y}}^2 + \hat{H}_{\mathbf{z}}^2\right).$$

In a similar manner, the electric and magnetic far fields can be utilized to obtain scattering properties like radar cross sections, phase functions and Mueller matrices. In the next section, we describe the calculation of a Mueller matrix using the obtained far field solution.

6.4 Mueller matrix

A Mueller matrix is a 4×4 matrix, which is used together with the Stokes vector to reproduce the effect of a given optical element. The Mueller calculus was developed in 1943 by Hans Mueller, a professor of physics at the Massachusetts

Institute of Technology (McLean, 2008). The method is a representation for coherent and partially polarized electromagnetic waves, and it has several applications with the ray tracing techniques (Bohren and Huffman, 1983).

The Stokes vector is a presentation for harmonic plane waves. The Stokes vector $(I, Q, U, V)^T$ has four real valued components, where each component illustrates certain polarization properties of the electromagnetic wave. We consider a harmonic plane wave propagating in the x-axis direction. Then, the electric field vector can be expressed as $\mathbf{E} = \text{real}(\hat{\mathbf{E}} e^{i\kappa x - i\omega t})$, where $\hat{\mathbf{E}}$ is a complex valued vector. We denote the y- and z-components of $\hat{\mathbf{E}}$ by the complex valued terms $\hat{\mathbf{E}}_y$ and $\hat{\mathbf{E}}_z$, respectively. Then, the wave can be expressed in the Stokes vector presentation, as

$$\begin{aligned} I &= |\hat{\mathbf{E}}_y| + |\hat{\mathbf{E}}_z|, \\ Q &= |\hat{\mathbf{E}}_y| - |\hat{\mathbf{E}}_z|, \\ U &= 2\,\text{real}(\hat{\mathbf{E}}_y \cdot \hat{\mathbf{E}}_z'), \\ V &= -2\,\text{imag}(\hat{\mathbf{E}}_y \cdot \hat{\mathbf{E}}_z'). \end{aligned}$$

Here $\hat{\mathbf{E}}_z'$ represents the complex conjugate of $\hat{\mathbf{E}}_z$. In the Stokes vector formulation, the first component I simply represents the intensity of the plane wave. The other components correspond to the polarization properties of the plane wave. The Stokes vector presentation can be applied for a plane wave of any propagation direction by modifying the coordinate basis.

The Mueller calculus is based on multiplying the Stokes vectors by Mueller matrices. If the incident wave is presented by the Stokes vector formulation $(I, Q, U, V)^T$, a scattered wave $(I', Q', U', V')^T$ is computed by the matrix product

$$\begin{pmatrix} I' \\ Q' \\ U' \\ V' \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix}.$$

For more details, see Bohren and Huffman (1983). For scattering problems, the Mueller matrix is generally presented as a function of the scattering angle θ. The scattering angle is defined such that the direction of the incident wave implies the zero scattering angle $\theta = 0$. The 90 degree scattering implies $\theta = \frac{\pi}{2}$ and the backscattering implies $\theta = \pi$. If the scatterer object is symmetric around the x axis, the scattering properties can be expressed comprehensively by the Mueller matrix. A sphere and an x-oriented disc are examples of such scatterer objects.

Next, we introduce a method, which is applied to compute the Mueller matrix by the DEC framework. First, the scattering geometry must be discretized by the DEC framework. The unbounded domain of the physical problem is truncated to a finite discrete domain by an absorbing boundary condition or by an absorbing layer. In many instances, the usage of an absorbing PML layer is preferred. The convex volume inside the PML layer should include the target object. Most efficiently, the simulation is carried out with the scattered field formulation, which is described in Section 4.1. The incident wave is a fully polarized harmonic plane wave, which is propagating to the x-direction. The polarization is

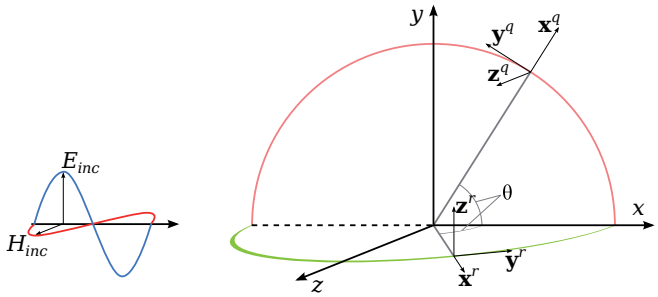


FIGURE 27 The geometry for the Mueller matrix calculation. The incident wave is polarized as illustrated on the left. The scattered far fields for angle θ are computed in the two directions $\mathbf{x}^q$ and $\mathbf{x}^r$, which are illustrated on the right.

selected such that the E field vibrates in the y-direction, and the H field vibrates in the z-direction, as illustrated on the left of Figure 27. After the simulation has proceeded, the near to far field transformation of Section 6.3 is applied to obtain the far field solutions. For each scattering angle θ, we compute far field terms of Equations (43) at two scattering directions $\mathbf{x}^q$ and $\mathbf{x}^r$. The scattering directions, which are illustrated on the right-hand side of Figure 27, are written formally as

$$\begin{pmatrix} \mathbf{x}^q \\ \mathbf{y}^q \\ \mathbf{z}^q \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} \mathbf{x}^r \\ \mathbf{y}^r \\ \mathbf{z}^r \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta & 0 & -\cos\theta \\ 0 & 1 & 0 \end{pmatrix}.$$

Specifically, we obtain the first two terms of Equations (43) at scattering direction $\mathbf{x}^q$ and denote them $\hat{E}^q_{\mathbf{y}}$ and $\hat{E}^q_{\mathbf{z}}$, respectively. The same terms are also obtained at scattering direction $\mathbf{x}^r$, and these are denoted by $\hat{E}^r_{\mathbf{y}}$ and $\hat{E}^r_{\mathbf{z}}$. We employ these terms to obtain four more scattering terms, which represent the terms of 45-degree polarization and the terms of circular polarization

$$\hat{E}^s_{\mathbf{y},\mathbf{z}} := \frac{\hat{E}^q_{\mathbf{y},\mathbf{z}} - \hat{E}^r_{\mathbf{y},\mathbf{z}}}{\sqrt{2}}, \; \hat{E}^t_{\mathbf{y},\mathbf{z}} := \frac{\hat{E}^q_{\mathbf{y},\mathbf{z}} + \hat{E}^r_{\mathbf{y},\mathbf{z}}}{\sqrt{2}},$$
$$\hat{E}^u_{\mathbf{y},\mathbf{z}} := \frac{\hat{E}^q_{\mathbf{y},\mathbf{z}} + i\hat{E}^r_{\mathbf{y},\mathbf{z}}}{\sqrt{2}}, \; \hat{E}^v_{\mathbf{y},\mathbf{z}} := \frac{\hat{E}^q_{\mathbf{y},\mathbf{z}} - i\hat{E}^r_{\mathbf{y},\mathbf{z}}}{\sqrt{2}}.$$

The first component m_{11} of the Mueller matrix represents the average scattering power for the scattering angle θ. Sometimes the term m_{11} is used without the other Mueller matrix components, and the function $m_{11}(\theta)$ is then called the scattering phase function. The scale of the Mueller matrix can be determined in many possible ways. A common way to scale the matrix is to set

$$m_{11} := \frac{P^q_{sca} + P^r_{sca}}{2P_{inc}},$$

where P_{inc} is the incident field power and P^q_{sca} and P^r_{sca} represent the scattered field power at radius $r = 1$. Then the term m_{11} can be expressed as

$$m_{11} := c\left[(\hat{E}^q_{\mathbf{y}})^2 + (\hat{E}^q_{\mathbf{z}})^2 + (\hat{E}^r_{\mathbf{y}})^2 + (\hat{E}^r_{\mathbf{z}})^2\right],$$

where

$$c := \frac{\epsilon\omega^2\sqrt{\epsilon\mu}}{64\pi^2 P_{inc}}.$$

The rest of the Mueller matrix components are expressed as

$$\begin{aligned}
m_{12} &:= c\left[(\hat{E}^q_{\mathbf{y}})^2 - (\hat{E}^q_{\mathbf{z}})^2 + (\hat{E}^r_{\mathbf{y}})^2 - (\hat{E}^r_{\mathbf{z}})^2\right],\\
m_{13} &:= c\left[2\,\mathrm{real}(\hat{E}^q_{\mathbf{y}}(\hat{E}^q_{\mathbf{z}})' + \hat{E}^r_{\mathbf{y}}(\hat{E}^r_{\mathbf{z}})')\right],\\
m_{14} &:= c\left[2\,\mathrm{imag}(\hat{E}^q_{\mathbf{y}}(\hat{E}^q_{\mathbf{z}})' + \hat{E}^r_{\mathbf{y}}(\hat{E}^r_{\mathbf{z}})')\right],\\
m_{21} &:= c\left[(\hat{E}^q_{\mathbf{y}})^2 + (\hat{E}^q_{\mathbf{z}})^2 - (\hat{E}^r_{\mathbf{y}})^2 - (\hat{E}^r_{\mathbf{z}})^2\right],\\
m_{22} &:= c\left[(\hat{E}^q_{\mathbf{y}})^2 - (\hat{E}^q_{\mathbf{z}})^2 - (\hat{E}^r_{\mathbf{y}})^2 + (\hat{E}^r_{\mathbf{z}})^2\right],\\
m_{23} &:= c\left[2\,\mathrm{real}(\hat{E}^q_{\mathbf{y}}(\hat{E}^q_{\mathbf{z}})' - \hat{E}^r_{\mathbf{y}}(\hat{E}^r_{\mathbf{z}})')\right],\\
m_{24} &:= c\left[2\,\mathrm{imag}(\hat{E}^q_{\mathbf{y}}(\hat{E}^q_{\mathbf{z}})' - \hat{E}^r_{\mathbf{y}}(\hat{E}^r_{\mathbf{z}})')\right],\\
m_{31} &:= c\left[(\hat{E}^s_{\mathbf{y}})^2 + (\hat{E}^s_{\mathbf{z}})^2 - (\hat{E}^t_{\mathbf{y}})^2 - (\hat{E}^t_{\mathbf{z}})^2\right],\\
m_{32} &:= c\left[(\hat{E}^s_{\mathbf{y}})^2 - (\hat{E}^s_{\mathbf{z}})^2 - (\hat{E}^t_{\mathbf{y}})^2 + (\hat{E}^t_{\mathbf{z}})^2\right],\\
m_{33} &:= c\left[2\,\mathrm{real}(\hat{E}^s_{\mathbf{y}}(\hat{E}^s_{\mathbf{z}})' - \hat{E}^t_{\mathbf{y}}(\hat{E}^t_{\mathbf{z}})')\right],\\
m_{34} &:= c\left[2\,\mathrm{imag}(\hat{E}^s_{\mathbf{y}}(\hat{E}^s_{\mathbf{z}})' - \hat{E}^t_{\mathbf{y}}(\hat{E}^t_{\mathbf{z}})')\right],\\
m_{41} &:= c\left[(\hat{E}^u_{\mathbf{y}})^2 + (\hat{E}^u_{\mathbf{z}})^2 - (\hat{E}^v_{\mathbf{y}})^2 - (\hat{E}^v_{\mathbf{z}})^2\right],\\
m_{42} &:= c\left[(\hat{E}^u_{\mathbf{y}})^2 - (\hat{E}^u_{\mathbf{z}})^2 - (\hat{E}^v_{\mathbf{y}})^2 + (\hat{E}^v_{\mathbf{z}})^2\right],\\
m_{43} &:= c\left[2\,\mathrm{real}(\hat{E}^u_{\mathbf{y}}(\hat{E}^u_{\mathbf{z}})' - \hat{E}^v_{\mathbf{y}}(\hat{E}^v_{\mathbf{z}})')\right],\\
m_{44} &:= c\left[2\,\mathrm{imag}(\hat{E}^u_{\mathbf{y}}(\hat{E}^u_{\mathbf{z}})' - \hat{E}^v_{\mathbf{y}}(\hat{E}^v_{\mathbf{z}})')\right].
\end{aligned}$$

7 NON-UNIFORM TIME STEPPING

In the leapfrog time discretization scheme, the upper bound for the time step size Δt is determined by the CFL condition, as discussed in Section 5.5. The CFL condition offers a uniform time step size, which can be used everywhere in the simulation region. On unstructured spatial discretization, the uniform time step size is rarely optimal in the full domain. In this chapter, we introduce a non-uniform leapfrog scheme, in which the time step size can be varied on different parts of the domain. We start with formulating an estimate for piecewise stability criterion in Section 7.1. The piecewise stability criterion is applied for the non-uniform leapfrog time stepping scheme, which is presented in Section 7.2. At the end of this chapter, we present numerical experiments, which show that the symmetric non-uniform time stepping scheme can be applied efficiently to problems where the properties of spatial discretization alternate heavily.

7.1 An estimate for piecewise stability criterion

Our aim is to formulate a piecewise limit for Δt, which could be used to automatically determine optimal non-uniform time step sizes for an unstructured system. The CFL condition (37) does not help in this case because it gives a stability criterion only for the uniform time stepping scheme. Ideally, we would like to be able to determine the optimal time step sizes Δt_{E_j} and Δt_{H_i} for each discrete form values E_j and H_i, respectively. Then each value update could be performed using an individual time step size instead of a general time step size. In this section, we estimate a piecewise stability criterion.

The instability of the leapfrog iteration method is generally caused by the dispersion of simulated values, as shown in Figure 28. This distribution leads us to formulate a stability criterion, which prevents this kind of behavior. Assume arbitrary positive real values E^{max} such that $|E_j| \leq E_j^{max}$ for all j. We try to

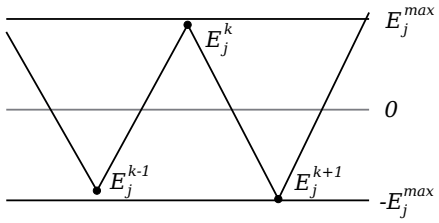


FIGURE 28 The simulation will be unstable if discrete form components bounce too much on a single time step.

eliminate the dispersion by setting

$$\left| E_j^{k+1} - 2E_j^k + E_j^{k-1} \right| \leq CE_j^{max}. \tag{44}$$

The selection of constant C defines how small changes are allowed to influence discrete form values. Constant $C = 4$ would allow E_j to bounce between $-E_j^{max}$ and E_j^{max} on consecutive time steps. Hence, we require $C \leq 4$. Still, the convergence can not be guaranteed for any non-uniform time stepping scheme. We consider the selection of C more specifically in Section 7.4, where we apply numerical experiments to a non-uniform time stepping scheme.

We derive a limit for the piecewise time step size Δt_{E_j} using the assumption of Equation (44). We assume the Yee's leapfrog discretization scheme (30) has zero source terms ($f_E = 0$, $f_H = 0$) and zero absorbing terms ($\star\sigma = 0$, $\star\sigma^* = 0$). The system is then expressed as

$$\begin{aligned} E^{k+1} - E^k &= \Delta t \star \epsilon^{-1} d_1^T H^k, \\ H^{k+1} - H^k &= -\Delta t \star \mu^{-1} d_1 E^{k+1}. \end{aligned}$$

These equations are applied consecutively to reformulate the left-hand side of Equation (44). We have

$$(E^{k+1} - E^k) - (E^k - E^{k-1}) = \Delta t \star \epsilon^{-1} d_1^T (H^k - H^{k-1}) = -\Delta t^2 \star \epsilon^{-1} d_1^T \star \mu^{-1} d_1 E^k.$$

Applying this to Equation (44), we have the relation

$$\Delta t^2 \leq \frac{CE_j^{max}}{\left| \left(\star\epsilon^{-1} d_1^T \star \mu^{-1} d_1 E^k \right)_j \right|},$$

which is realized for all $k \geq 0$. We write an upper bound for the divisor using the assumption $|E_j^k| \leq E_j^{max}$. Using the notation $|\mathcal{M}|_{i,j} := |\mathcal{M}_{i,j}|$, we estimate

$$\left| \left(\star\epsilon^{-1} d_1^T \star \mu^{-1} d_1 E^k \right)_j \right| \leq \left(\left| \star\epsilon^{-1} d_1^T \star \mu^{-1} d_1 \right| E^{max} \right)_j.$$

Now, we can write a sufficient condition for time step size. We denote the local time step for a value update of E_j by Δt_{E_j}. Assuming that E_j and its neighbor

values H_i are discretized using this same time step size, a sufficient condition to fulfill the stability criterion (44) can be written as

$$\Delta t_{E_j} \leq \sqrt{\frac{CE_j^{max}}{\left(\left|\star\epsilon^{-1}d_1^T\star\mu^{-1}d_1\right| E^{max}\right)_j}}. \tag{45}$$

The similar consideration is applied on the magnetic field values H_i to obtain the local time step size Δt_{H_i}. The corresponding formulation for the stability criterion is

$$\Delta t_{H_i} \leq \sqrt{\frac{CH_i^{max}}{\left(\left|\star\mu^{-1}d_1\star\epsilon^{-1}d_1^T\right| H^{max}\right)_i}}. \tag{46}$$

The time step consideration is carried out using an arbitrary selection of E^{max} and H^{max}. The selection of maximal values E_j^{max} and H_i^{max} determines how the condition is weighted between elements. For example, if the E^{max} is selected as the eigenvector corresponding to an eigenvalue χ of $|\star\epsilon^{-1}d_1^T\star\mu^{-1}d_1|$, then the uniform criterion $\Delta t_{E_j} \leq \sqrt{C/\chi}$ is obtained. This presentation is close to the presentation of the CFL condition. We select the maximal value vectors E^{max} and H^{max} inspired by the energy norm formulation (38). Letting each value have the same maximal weight on the energy norm, we obtain

$$E_j^{max} = \frac{1}{\sqrt{\star\epsilon_{j,j}}}, \qquad H_i^{max} = \frac{1}{\sqrt{\star\mu_{i,i}}}.$$

7.2 Non-uniform leapfrog method

Instead of picking the same time step size for every element, it is often more efficient to assign each element its own optimized time step. There are several ways to implement a non-uniform (or asynchronous) time stepping scheme. Lew et al. (2003) developed an asynchronous time stepping scheme for elastodynamics using asynchronous variational integrators (AVI). Later Stern et al. (2009) applied the same idea to electromagnetism. The idea of AVI is based on preserving optimal energy conservation in time-dependent simulations.

The value update of the AVI iteration is illustrated in Figure 29. As previous studies have shown (Lew et al., 2003; Stern et al., 2009), the explicit AVI scheme can be implemented by selecting mesh elements from a priority queue, sorting by time and iterating forward-in-time. However, the scheme is not strictly iterative since the update of a value can depend on several past time instances (dotted arrows in the figure). This is carried out by using extra variables to store potential. Note that if all elements take uniform time steps, the AVI method is reduced to a basic leapfrog scheme.

Unfortunately, we have not identified an adequate analysis for the energy conservation or the stability criterion of the AVI based method. Our experience is that the energy is not conserved during the iteration, even if the small time step

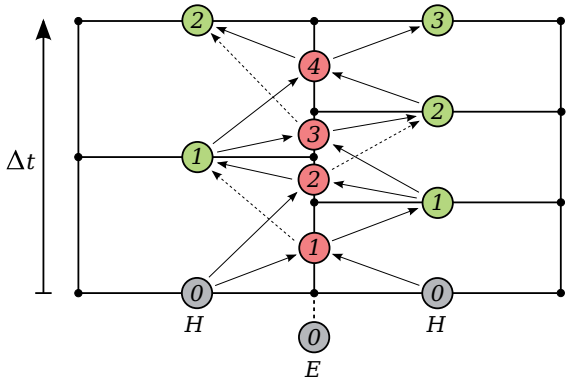


FIGURE 29 One-dimensional illustration of asychronous time stepping using the AVI method. Each H term has its specific time step size, and time instances are represented by circles with numbers. The E terms are computed on the sections between neighboring time instances. The arrows show which neighboring instances are used to compute the current instance. The dotted arrows are used for relations with past values, and those relations require a special consideration.

sizes are applied. This is the reason why we prefer the following non-uniform leapfrog time discretization scheme. The non-uniform leapfrog method begins with computing the maximal local time step sizes $\Delta t_{E_j}^{max}$ and $\Delta t_{H_i}^{max}$ by the stability criterion estimates presented in Equations (45) and (46). Then we select the general time step Δt, which is the smallest uniform iteration block to be repeated over and over again in the forward in time iteration. The local time step sizes are then set as $\Delta t_{E_j} = \Delta t / s_{E_j}$ and $\Delta t_{H_i} = \Delta t / s_{H_i}$, where s_{E_j} and s_{H_i} are integers representing the number of time steps on Δt period. The general time step size Δt is selected between the minimal and maximal values of the local time step sizes $\Delta t_{E_j}^{max}$ and $\Delta t_{H_i}^{max}$.

The derivation of the piecewise stability criterion for $\Delta t_{E_j}^{max}$ assumes that the neighbor values H_i are discretized using the same time step size $\Delta t_{E_j}^{max}$. This can not be guaranteed in non-uniform time stepping. Still, we can guarantee that the neighbor values are discretized at least on the same level. We assume indices i and j such that the primal edge $\mathcal{E}_j$ is a boundary element of the primal face $\mathcal{F}_i$ (this implies $(d_1)_{i,j} \neq 0$). Then we require

$$s_{E_j} \geq \frac{\Delta t}{\min\left\{\Delta t_{E_j}^{max}, \Delta t_{H_i}^{max}\right\}}, \qquad s_{H_i} \geq \frac{\Delta t}{\min\left\{\Delta t_{H_i}^{max}, \Delta t_{E_j}^{max}\right\}}.$$

These inequalities offer a small enough time step size for each element. Still, the time stepping is asynchronous, meaning that the energy conservation is not as clear as we discussed in Section 5.6. We synchronize the time stepping scheme by selecting the numbers s_{E_j} and s_{H_i} such that they can be written as a power of three, i.e. $s = 3^u \in \{1, 3, 9, 27, \dots\}$. Using this selection, each neighbor element, which has finer time stepping than the current element, also has a half-way instance

for the current value update (see Figure 30). This property is the base for the leapfrog methods, and for this reason we call this scheme a non-uniform leapfrog method. The half-way instance is not available for those cases where the current element has a finer time step size than the neighboring element. In those cases, the iteration is still symmetric on Δt period, as illustrated in Figure 30.

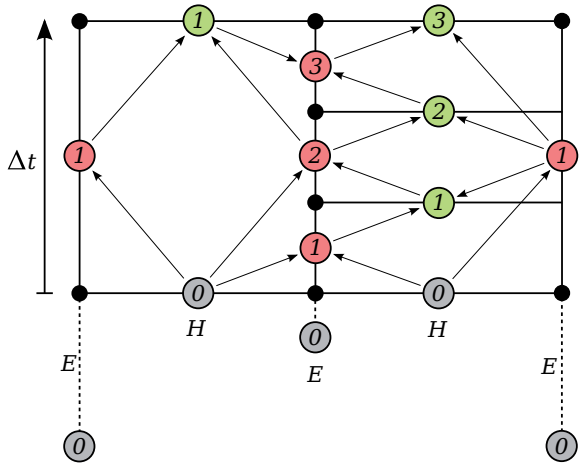


FIGURE 30 In non-uniform time stepping, Δt is divided into smaller steps only where this is needed. In this one-dimensional illustration, the circles with a number inside represent time instances of E_j and H_i. The arrows illustrate the chronological order of computation.

The time stepping of the non-uniform leapfrog method is carried out in chronological order. The value updates are strictly iterative, meaning that a new value E_j^{new} or H_i^{new} can be computed from the latest instance of values. Thus, no extra storage is required during the computation. The following two equations illustrate the value updates in the non-uniform leapfrog method, where the *"new"* values replace the previous values immediately. Based on Yee's leapfrog Equations (30), we write

$$E_j^{new} := E_j + \left(\frac{s_{E_j} \star \epsilon_{j,j}}{\Delta t} + \frac{\star \sigma_{j,j}}{2} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i - \star \sigma_{j,j} E_j + f_E(t)_j \right], \qquad (47)$$

$$H_i^{new} := H_i + \left(\frac{s_{H_i} \star \mu_{i,i}}{\Delta t} + \frac{\star \sigma^*_{i,i}}{2} \right)^{-1} \left[- \sum_{j=1}^{n} (d_1)_{i,j} E_j - \star \sigma^*_{i,i} H_i + f_H(t)_i \right].$$

In the non-uniform leapfrog method, the H values are declared at integer time instances at the beginning and at the end of Δt iteration (see Figure 30). The E values are declared, in principle, at the half-integer time instance, which is at $\Delta t_{E_j}/2$ earlier for E_j. The chronological order of the value updates is obtained by ordering the relative update times

$$t^k_{E_j} := \Delta t \frac{k-1}{s_{E_j}}, \qquad t^k_{H_i} := \Delta t \frac{k-\frac{1}{2}}{s_{H_i}},$$

which correspond to the k:th value update inside the Δt period. To obtain the full solution at synchronized time instance, we apply the transformation for E field values, as described in Section 5.4. We denote the synchronized value by E_j^{sync}. The transformation from E_j to E_j^{sync} and the inverse transformation from E_j^{sync} to E_j are obtained, respectively, by equations

$$E_j^{sync} := E_j + \left(\frac{2s_{E_j} \star \epsilon_{j,j}}{\Delta t} + \star\sigma_{j,j} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i - \star\sigma_{j,j} E_j + f_E(t)_j \right], \tag{48}$$

$$E_j := E_j^{sync} - \left(\frac{2s_{E_j} \star \epsilon_{j,j}}{\Delta t} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i - \star\sigma_{j,j} E_j^{sync} + f_E(t)_j \right].$$

7.3 Harmonic non-uniform leapfrog method

The Equations (47) and (48) of the non-uniform leapfrog method are based on Yee's leapfrog approximations. The equations can be also formulated using the time-harmonic approach of Section 5.2. In the harmonic non-uniform leapfrog method, the value update Equations (47) are replaced by

$$E_j^{new} := E_j + 2 \left(\frac{\omega \star \epsilon_{j,j}}{\sin \phi_{E_j}} + \frac{\star\sigma_{j,j}}{\cos \phi_{E_j}} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i - \frac{\star\sigma_{j,j}}{\cos \phi_{E_j}} E_j + f_E(t)_j \right], \tag{49}$$

$$H_i^{new} := H_i + 2 \left(\frac{\omega \star \mu_{i,i}}{\sin \phi_{H_i}} + \frac{\star\sigma_{i,i}^*}{\cos \phi_{H_i}} \right)^{-1} \left[- \sum_{j=1}^{n} (d_1)_{i,j} E_j - \frac{\star\sigma_{i,i}^*}{\cos \phi_{H_i}} H_i + f_H(t)_i \right],$$

where $\phi_{E_j} := \frac{\omega \Delta t}{2 s_{E_j}}$ and $\phi_{H_i} := \frac{\omega \Delta t}{2 s_{H_i}}$. The new synchronization formulas, which replace Equation (48), are written as

$$E_j^{sync} := E_j + \left(\frac{\omega \cos \phi_{E_j} \star \epsilon_{j,j}}{\sin \phi_{E_j}} + \star\sigma_{j,j} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i + \alpha_j E_j + f_E(t)_j \right],$$

$$E_j := E_j^{sync} - \left(\frac{\omega \star \epsilon_{j,j}}{\sin \phi_{E_j}} \right)^{-1} \left[\sum_{i=1}^{m} (d_1^T)_{j,i} H_i + \alpha_j E_j^{sync} + f_E(t)_j \right],$$

where

$$\alpha_j := \frac{\omega (1 - \cos \phi_{E_j}) \star \epsilon_{j,j}}{\sin \phi_{E_j}} - \star\sigma_{j,j}.$$

The value update Equations (49) are exact for time-harmonic waves, if the neighbor values are determined exactly at the middle of the time step. Unfortunately, this is not always the case with non-uniform time stepping. If the neighboring values are updated with fewer time steps, the formulation employs often non-synchronized neighbor instances, and then the value update is not exact. Nevertheless, the approximations are still improved in comparison to Equation (47),

and we prefer using the harmonic non-uniform time stepping for time-harmonic problems.

The treatment of negative absorption can also be transformed to the non-uniform time stepping scheme. As explained in Section 5.3, the value update of E_j or H_i is treated by a three-step procedure, which requires additional memory storage for the D_j or B_i values, respectively. The value update equations for the non-uniform method are very similar to the Equations (34) and (35). We only replace the term ϕ with ϕ_{E_j} and ϕ_{H_i}, respectively. The equations for the E_j and D_j value updates are

$$
\begin{aligned}
E_j^{temp} &:= E_j + \frac{-\star\epsilon_{j,j} - \frac{\tan\phi_{E_j}}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j, \\
D_j^{new} &:= D_j + \frac{2\sin\phi_{E_j}}{\omega}\left(d_1^T H\right)_j, \\
E_j^{new} &:= E_j^{temp} + \frac{\star\epsilon_{j,j} - \frac{\tan\phi_{E_j}}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^{new},
\end{aligned}
$$

and the equations for H_i and B_i value updates are

$$
\begin{aligned}
H_i^{temp} &:= H_i + \frac{-\star\mu_{i,i} - \frac{\tan\phi_{H_i}}{\omega}\star\sigma_{i,i}^*}{\star\mu_{i,i}^2 + \frac{1}{\omega^2}\star{\sigma_{i,i}^*}^2} B_i, \\
B_i^{new} &:= B_i - \frac{2\sin\phi_{H_i}}{\omega}\left(d_1 E\right)_i, \\
H_i^{new} &:= H_i^{temp} + \frac{\star\mu_{i,i} - \frac{\tan\phi_{H_i}}{\omega}\star\sigma_{i,i}^*}{\star\mu_{i,i}^2 + \frac{1}{\omega^2}\star{\sigma_{i,i}^*}^2} B_i^{new}.
\end{aligned}
$$

The synchronization of the E_j and D_j values is performed using the following equations. The transformation from E_j, D_j to E_j^{sync}, D_j^{sync} is obtained by

$$
\begin{aligned}
E_j^{temp} &:= E_j + \frac{-\star\epsilon_{j,j} - \frac{\tan(\phi_{E_j}/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j, \\
D_j^{sync} &:= D_j + \frac{\tan\phi_{E_j}}{\omega}\left[\left(d_1^T H\right)_j + \frac{\omega(1-\cos\phi)}{\sin\phi} D_j\right], \\
E_j^{sync} &:= E_j^{temp} + \frac{\star\epsilon_{j,j} - \frac{\tan(\phi_{E_j}/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^{sync}.
\end{aligned}
$$

The inverse transformation from E_j^{sync}, D_j^{sync} to E_j, D_j is obtained by

$$E_j^{temp} := E_j^{sync} - \frac{\star\epsilon_{j,j} - \frac{\tan(\phi_{E_j}/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j^{sync},$$

$$D_j := D_j^{sync} - \frac{\sin\phi_{E_j}}{\omega}\left[d_1^T H^k + \frac{\omega(1-\cos\phi)}{\sin\phi} D_j^{sync}\right],$$

$$E_j := E_j^{temp} - \frac{-\star\epsilon_{j,j} - \frac{\tan(\phi_{E_j}/2)}{\omega}\star\sigma_{j,j}}{\star\epsilon_{j,j}^2 + \frac{1}{\omega^2}\star\sigma_{j,j}^2} D_j.$$

7.4 Numerical consideration of the non-uniform leapfrog method

Next, we consider stability, efficiency and energy conservation of the non-uniform time stepping scheme, which is presented in Section 7.2. We perform numerical experiments, where the initialized wave is simulated forward-in-time in a domain with perfectly reflecting boundaries. The simulation domain is a cube of edge length 2. The domain is discretized by a randomly generated mesh, which is illustrated in the left-hand side of Figure 31. The boundary of a cube is constructed by squares with an edge length 0.1. The interior is generated by Voronoi construction from randomly positioned nodes. The nodes are dropped randomly using the uniform density function, where the node density is one node per volume 10^{-3}. The nodes are not allowed to be too near to each other. If the latest node is less than 0.01 away from any previous node, it is regenerated by the same uniform random function, and the same test is repeated. The range of produced element sizes is wide, as shown in Table 2.

TABLE 2 The element sizes of the random Voronoi cube.

	Minimum	Mean	Maximum
Primal edge length	0.0100	0.129	0.309
Primal face area	7.95e-5	0.00628	0.0273
Dual edge length	1.44e-6	0.0442	0.201
Dual face area	1.03e-11	0.00384	0.0288

At the initial stage, the discrete field values $E(t^0)$ and $H(t^0)$ are initialized by an initial wave, which is similar to Equation (40). The initial wave is a circularly polarized plane wave of wavelength $\lambda = 1$ and period $T = 1$, which propagates perpendicularly to the two faces of the domain. Material parameters in the domain region are $\epsilon = \mu = 1$. The domain boundaries are modeled by the perfectly reflecting Dirichlet boundary condition. After the initialization, the system should not include any absorption or emission. We apply both uniform and non-uniform time stepping schemes in these simulations by using Yee's value update Equations (30) and (47). The simulations were iterated forward in time through

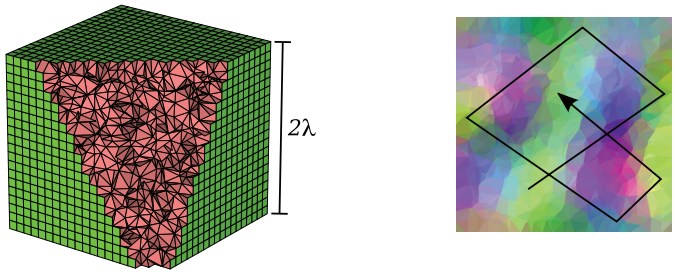


FIGURE 31 The random mesh is illustrated on the left-hand side. The initialized wave is trapped inside the domain, as shown on the right-hand side. The energy conservation is studied by long term simulation.

1000 periods. The discrete energy norm of Equation (38) is computed after each iterated period. The E value instances are synchronized before the computation.

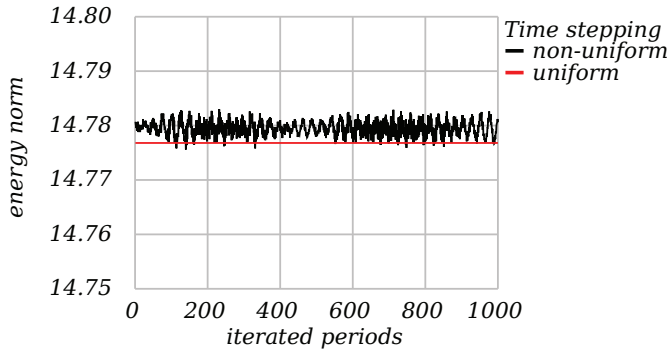


FIGURE 32 The uniform time stepping scheme conserves the energy norm exactly. Within the non-uniform time stepping, the energy norm remains almost constant during long term simulation.

The energy norms are plotted in Figure 32. The energy conservation of the uniform time stepping scheme is exact, as we noticed analytically in Section 5.6. The energy conservation of the non-uniform time stepping scheme also seems to be satisfactory. The standard deviation of the energy norm is 0.0014, and the difference between the maximal and minimal values is 0.0074. The relative standard deviation is 0.010%, and the relative maximal difference is 0.050%. The results also show that the system is stable with both uniform and non-uniform time stepping schemes. The stability constant $C = 4$ was applied with both methods using the stability criterion of Section 7.1. This constant is now seen to be sufficient for the convergence of this problem. Using another simulation, we found that neither uniform nor non-uniform time stepping schemes converged with the constant $C = 4.5$. Thus, the stability criterion nominates a quite tight bound for the time step size in the current random mesh discretization. Still, we have found that with certain spatial discretizations, the constant $C = 4$ is not sufficient for the convergence. Then we apply $C = 2$, which has been a reliable criterion in all of our simulations.

We also compared the iteration times of both schemes in our test problem. We performed a single core simulation on a Intel Xeon E5-2670 processor at 2.60 GHz. The time consumption of non-uniform time stepping scheme was 0.773 seconds per one period simulation. The maximum of time step divisors s_{E_j} and s_{H_i} was 81, which means there was a remarkable difference between the smallest and largest time step sizes. The uniform time step scheme is scaled to obey the stability criterion at every part of the domain, which means the minimum time step is used for every element. An iteration time for period T of the uniform time stepping scheme was 13.9 seconds, which is about 18 times the corresponding iteration time of non-uniform time stepping scheme. Thus, non-uniform time stepping can cause a significant improvement in the method efficiency.

8 HARMONIC HODGE OPERATOR

In the DEC framework, the error in spatial discretization is packed into one source, which is the Hodge operator. Therefore, it is important to examine this operator more closely. The selection of the Hodge operator can significantly affect the accuracy of the method; therefore, the development of DEC is partly shifting to the consideration of the Hodge operator (Hirani et al., 2012; Mullen et al., 2011). In this chapter, we introduce a diagonal discrete Hodge operator, which is optimized for problems including harmonic waves. We compare the harmonic Hodge presentation to the Yee's Hodge operator by the numerical experiments and observe significant improvement in accuracy. In this context, we also compare the structured grids of Section 2.8 in association with DEC simulations. We observe that the isotropic properties of grids have a significant influence on the method accuracy.

8.1 The Yee's Hodge operators for the Maxwell system

The four discrete Hodge operators $\star\epsilon$, $\star\mu$, $\star\sigma$ and $\star\sigma^*$ are introduced in Section 3.3 to formulate the discrete Maxwell system. The relations for the Hodge operators are written as

$$\begin{aligned} D &= \star\epsilon E, \qquad B = \star\mu H, \\ J &= \star\sigma E, \qquad J^* = \star\sigma^* H. \end{aligned}$$

These relations are supposed to estimate the corresponding physical relations as accurately as possible. The Hodge operators are presented as diagonal matrices, when the orthogonal duality is required. The Yee's Hodge operator is based on an assumption of constant material parameters ϵ, μ, σ and σ^*, and locally constant vector fields $\mathbf{D}$, $\mathbf{E}$, $\mathbf{B}$, $\mathbf{H}$, $\mathbf{J}$ and $\mathbf{J}^*$. The discrete form values D_j, E_j, B_i, H_i, J_j and J_i^* can then be expressed by the dot product between the vector field and the element volume vector: for example, relations $E_j = |\mathcal{E}_j|\mathbf{n}_{\mathcal{E}_j} \cdot \mathbf{E}$ and $D_j = |\mathcal{F}_j^*|\mathbf{n}_{\mathcal{F}_j^*} \cdot \mathbf{D}$. Assuming real valued material parameters ϵ, μ, σ and σ^*, the diagonal discrete

Hodge operators have formulations

$$\star\epsilon_{j,j} = \epsilon\frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|}, \qquad \star\mu_{i,i} = \mu\frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|},$$
$$\star\sigma_{j,j} = \sigma\frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|}, \qquad \star\sigma^*_{i,i} = \sigma^*\frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|}$$

We recall that $|\mathcal{F}|$ is the area of face $\mathcal{F}$ and $|\mathcal{E}|$ is the length of edge $\mathcal{E}$. The dual element is denoted by a star * after the element symbol.

The assumption of locally constant vector fields is not accurate. Many problems have harmonic or curved solutions, where the curvature can be approximated in advance. For example, time-harmonic problems usually have a solution, which is locally space-harmonic in some direction. If one knows the local material parameters and the period of the solution, the wavelength λ can be estimated. This information can be used to improve the estimate for the Hodge operator and to obtain more accurate results.

8.2 Principle of harmonic Hodge

From now on, we use a complex time-harmonic expression for Hodge operators and discrete form values. As discussed in Section 6.1, the four discrete Hodge operators $\star\epsilon$, $\star\mu$, $\star\sigma$ and $\star\sigma^*$ can be expressed by two complex valued discrete Hodge operators $\hat{\star\epsilon}$ and $\hat{\star\mu}$, which are defined by Equation (19). This notation allows us to express the discretized Maxwell system in a shorter form as presented in Equation (20).

From now on, we consider only the Hodge operator $\hat{\star\epsilon}$ and the fields corresponding to it. The other Hodge operator, $\hat{\star\mu}$, can be computed in a similar manner. We make an assumption that the time-harmonic property will generate locally space-harmonic solutions. In other words, we assume that in a small, environment the solution behaves like a harmonic plane wave. Let $\mathbf{d}$ be the propagation direction of a plane wave. Then the harmonic discrete form values $\hat{E}_j^{\mathbf{d}}$ and $\hat{D}_j^{\mathbf{d}}$ corresponding to plane wave solution are

$$\hat{E}_j^{\mathbf{d}} := \int_{\mathcal{E}_j} \hat{\mathbf{E}}_0^{\mathbf{d}} e^{i\omega\sqrt{\hat{\epsilon}\hat{\mu}}(\mathbf{p}\cdot\mathbf{d})} \cdot d\mathbf{l}, \qquad \hat{D}_j^{\mathbf{d}} := \int_{\mathcal{F}_j^*} \hat{\epsilon}\hat{\mathbf{E}}_0^{\mathbf{d}} e^{i\omega\sqrt{\hat{\epsilon}\hat{\mu}}(\mathbf{p}\cdot\mathbf{d})} \cdot d\mathbf{a}.$$

Here $\hat{\mathbf{E}}_0^{\mathbf{d}}$ is a complex-valued field vector in $\mathbb{C}^3$ and $\mathbf{p}$ represents a position vector. Physically, the diagonal Hodge operator $\hat{\star\epsilon}$ is designed to estimate the relation $\hat{D}_j^{\mathbf{d}} = \hat{\star\epsilon}_{j,j}\hat{E}_j^{\mathbf{d}}$. The error of the Hodge transformation for a plane wave propagating in the direction $\mathbf{d}$ is

$$\hat{r}_j^{\mathbf{d}} = \hat{E}_j^{\mathbf{d}} - \hat{\star\epsilon}_{j,j}^{-1}\hat{D}_j^{\mathbf{d}}.$$

Since the propagation direction is not usually known in advance, we want to minimize the error in all directions. We minimize the squared error norm $\sum_{\mathbf{d}} |\hat{r}_j^{\mathbf{d}}|^2$,

where $\sum_{\mathbf{d}}$ goes through a wide range of propagation directions. The least square minimization gives the best estimate for the Hodge operator, which is obtained by the equation

$$\hat{\star\epsilon}_{j,j} = \frac{\sum_{\mathbf{d}} \hat{D}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}'}}{\sum_{\mathbf{d}} \hat{E}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}'}}. \tag{50}$$

The minimization can be carried out numerically, simply by selecting a set of propagation directions for the computation. Another way is to integrate propagation directions analytically over a unit sphere surface. The analytical approach is discussed in Section 8.3.

There is an interesting question of how much does harmonic Hodge operator differ from the Yee's Hodge operator? For computing we assume a regular grid, where the length of edge elements is 1. Further, we assume that the solution of the example is a plane wave that propagates to the x-direction with wavelength $\lambda = \frac{2\pi}{\omega\sqrt{\epsilon\mu}}$. Here, ω is the angular frequency, and the material parameters ϵ and μ are constant real numbers. The x-components of the resulting vector fields are zero. Every non-zero discrete value of the solution lies on the edges in a y- or z-direction or on their corresponding orthogonal faces. Since the problem is symmetric for the y- and z-directions, we need to compute only one direction of the wave. We select primal edge $\mathcal{E}_j$ as the unit line on the z-axis with its center at the origin. Let dual face $\mathcal{F}_j^*$ be a unit square on the x-y-plane with its center at the origin. We know the propagation direction; therefore, we can compute the exact harmonic Hodge operator for this one direction. The diagonal term of the harmonic Hodge operator is

$$\hat{\star\epsilon}_{j,j} = \frac{\hat{D}_j^{\mathbf{d}}}{\hat{E}_j^{\mathbf{d}}} = \frac{\int_{-0.5}^{0.5}\int_{-0.5}^{0.5} \epsilon \hat{\mathbf{E}}_0 e^{i\omega\sqrt{\epsilon\mu}x}\, dydx}{\int_{-0.5}^{0.5} \hat{\mathbf{E}}_0 e^{i\omega\sqrt{\epsilon\mu}x}\, dz} = \epsilon \frac{e^{\frac{i\omega\sqrt{\epsilon\mu}}{2}} - e^{-\frac{i\omega\sqrt{\epsilon\mu}}{2}}}{i\omega\sqrt{\epsilon\mu}} = \epsilon \frac{\sin(\frac{\omega\sqrt{\epsilon\mu}}{2})}{\frac{\omega\sqrt{\epsilon\mu}}{2}}.$$

In this case, the component of the harmonic Hodge operator has a real value. The expression is now compared to the Yee's Hodge operator. Since $|\mathcal{F}_j^*| = |\mathcal{E}_j| = 1$, the diagonal term for the Yee's Hodge operator is $\star\epsilon_{j,j} = \epsilon$. This means the relative difference of the Yee's Hodge and the harmonic Hodge operators is

$$\frac{\hat{\star\epsilon}_{j,j} - \star\epsilon_{j,j}}{\star\epsilon_{j,j}} = \frac{\sin\left(\frac{\omega\sqrt{\epsilon\mu}}{2}\right)}{\frac{\omega\sqrt{\epsilon\mu}}{2}} - 1.$$

If the wavelength is ten times the element edge length (i.e. $\lambda = 10$), then the value of term $\frac{\omega\sqrt{\epsilon\mu}}{2}$ equals $\frac{\pi}{10}$. Using this in the previous equation, we get the difference of Hodge operators. The harmonic Hodge operator has 1.64% smaller values than Yee's Hodge operator. Since the harmonic Hodge operator is exact in this case, the system with the Yee's Hodge operator produces a 1.64% too short wavelength. We verify this result by numerical experiments in Section 8.5.1.

8.3 Integration of the harmonic Hodge operator

In this section, we perform an analytical derivation for the harmonic Hodge operator. We express the wave propagation direction $\mathbf{d}$ by the polar coordinate presentation

$$\mathbf{d} = \begin{pmatrix} \cos\theta \\ \sin\theta\cos\phi \\ \sin\theta\sin\phi \end{pmatrix}.$$

The discrete formulation (50) of the harmonic Hodge operator can be expressed in continuous form by replacing the discrete sum of propagation directions by the integration of all propagation directions over a unit sphere. The harmonic Hodge operator is now written as

$$\star\hat{\epsilon}_{j,j} = \frac{\int_0^{\pi}\int_0^{2\pi} \hat{D}_j^{\mathbf{d}}\hat{D}_j^{\mathbf{d}} \sin\theta \, d\phi d\theta}{\int_0^{\pi}\int_0^{2\pi} \hat{E}_j^{\mathbf{d}}\hat{D}_j^{\mathbf{d}} \sin\theta \, d\phi d\theta}. \tag{51}$$

To evaluate the integrations, we estimate the terms $\hat{D}_j^{\mathbf{d}}\hat{D}_j^{\mathbf{d}}$ and $\hat{E}_j^{\mathbf{d}}\hat{D}_j^{\mathbf{d}}$. We assume that the primal edge $\mathcal{E}_j$ is a line segment of length l and the corresponding dual face $\mathcal{F}_j^*$ is a circle of radius r. As we illustrate in Figure 33, the edge element is pointing at the x-direction, and the dual face element is orthogonal to it, lying on the y-z-plane. Both elements are centered at the origin.

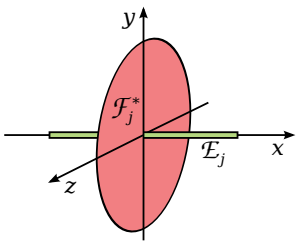


FIGURE 33 An illustration of the elements, which are applied in the derivation of the harmonic Hodge operator.

The layout of the elements is symmetric for all azimuth angles ϕ, which let us simplify the consideration assuming $\phi = 0$. The x-component of the field vector is denoted by $(\hat{\mathbf{E}}_0^{\mathbf{d}})_x$. Then the discrete form value $\hat{D}_j^{\mathbf{d}}$ is computed by

$$\hat{D}_j^{\mathbf{d}} = \int_{-r}^{r}\int_{-\sqrt{r^2-z^2}}^{\sqrt{r^2-z^2}} \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x e^{i\omega\sqrt{\hat{\epsilon}\hat{\mu}}y\sin\theta} dydz.$$

The face element $\mathcal{F}_j^*$ is supposed to be small compared to the wavelength. Thus, the Taylor polynomial expression offers a good formulation for the exponential function. To simplify the forthcoming formulation, we introduce an auxiliary term $\hat{\alpha} := i\omega\sqrt{\hat{\epsilon}\hat{\mu}}\sin\theta$. The Taylor polynomial for the exponential function is

then $e^{\hat{\alpha}y} = \sum_{n=0}^{\infty} \frac{\hat{\alpha}^n y^n}{n!}$. We integrate the expression for the discrete form value as

$$\begin{aligned}
\hat{D}_j^{\mathbf{d}} &= \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \int_{-r}^{r} \int_{-\sqrt{r^2-z^2}}^{\sqrt{r^2-z^2}} \sum_{n=0}^{\infty} \frac{\alpha^n y^n}{n!} dy dz \\
&= \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \int_{-r}^{r} \sum_{n=0}^{\infty} \frac{\hat{\alpha}^n \left((\sqrt{r^2-z^2})^{n+1} - (-\sqrt{r^2-z^2})^{n+1} \right)}{(n+1)!} dz \\
&= \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \int_{-r}^{r} \sum_{n=0}^{\infty} \frac{2\hat{\alpha}^{2n} \left(\sqrt{r^2-z^2} \right)^{2n+1}}{(2n+1)!} dz.
\end{aligned}$$

To continue the integration, we change the variable z to t such that $z = r \sin t$. By this selection, we have $dz = r \cos t dt$ and $\sqrt{r^2 - z^2} = r \cos t$. The precomputed integrals of Table 3 are applied to get the few first terms of the resulting polynomial

$$\begin{aligned}
\hat{D}_j^{\mathbf{d}} &= \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \sum_{n=0}^{\infty} \frac{2\hat{\alpha}^{2n} r^{2n+2}}{(2n+1)!} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (\cos t)^{2n+2} \, dt \\
&= \pi r^2 \hat{\epsilon}(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \left(1 + \frac{(\hat{\alpha} r)^2}{8} + \frac{(\hat{\alpha} r)^4}{192} + \dots \right). \qquad (52)
\end{aligned}$$

TABLE 3 Precomputed integrals of trigonometric functions.

$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^2 t dt = \frac{\pi}{2}$	$\int_0^{\pi} \sin^3 \theta d\theta = \frac{4}{3}$	$\int_0^{\pi} \sin^5 \theta \cos^2 \theta d\theta = \frac{16}{105}$
$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^4 t dt = \frac{3\pi}{8}$	$\int_0^{\pi} \sin^5 \theta d\theta = \frac{16}{15}$	$\int_0^{\pi} \sin^3 \theta \cos^2 \theta d\theta = \frac{4}{15}$
$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^6 t dt = \frac{5\pi}{16}$	$\int_0^{\pi} \sin^7 \theta d\theta = \frac{32}{35}$	$\int_0^{\pi} \sin^3 \theta \cos^4 \theta d\theta = \frac{4}{35}$

A similar computation is performed for the term $\hat{E}_j^{\mathbf{d}}$. This integration is applied through an edge element, which makes the consideration a little easier. Again, the edge element is relatively small, which allows us to use the Taylor formulation. We initialize an auxiliary variable $\hat{\beta} := i\omega\sqrt{\hat{\epsilon}\hat{\mu}} \cos\theta$. The use of the

Taylor formulation $e^{\hat{\beta}y} = \sum_{n=0}^{\infty} \frac{\hat{\beta}^n y^n}{n!}$ leads to the following formulation

$$\begin{aligned}
\hat{E}_j^{\mathbf{d}} &= \int_{-\frac{l}{2}}^{\frac{l}{2}} (\hat{\mathbf{E}}_0^{\mathbf{d}})_x e^{i\omega\sqrt{\hat{\epsilon}\hat{\mu}}x\cos\theta} dx \\
&= (\hat{\mathbf{E}}_0^{\mathbf{d}})_x \int_{-\frac{l}{2}}^{\frac{l}{2}} \sum_{n=0}^{\infty} \frac{\hat{\beta}^n x^n}{n!} dx \\
&= (\hat{\mathbf{E}}_0^{\mathbf{d}})_x \sum_{n=0}^{\infty} \frac{\hat{\beta}^n \left((\frac{l}{2})^{n+1} - (-\frac{l}{2})^{n+1} \right)}{(n+1)!} \\
&= l(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \sum_{n=0}^{\infty} \frac{(\hat{\beta}l)^{2n}}{2^{2n}(2n+1)!} \\
&= l(\hat{\mathbf{E}}_0^{\mathbf{d}})_x \left(1 + \frac{(\hat{\beta}l)^2}{24} + \frac{(\hat{\beta}l)^4}{1920} + \dots \right). \qquad (53)
\end{aligned}$$

Equations (52) and (53) let us reformulate the product terms $\hat{D}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}}$ and $\hat{E}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}}$, which are needed for Equation (51). The first terms of the Taylor polynomials are

$$\begin{aligned}
D_j^{\mathbf{d}} D_j^{\mathbf{d}} &\approx \pi^2 r^4 \epsilon^2 (\mathbf{E}_0^{\mathbf{d}})_x^2 \left(1 + \frac{(\alpha r)^2}{4} + \frac{5(\alpha r)^4}{192} \right), \\
E_j^{\mathbf{d}} D_j^{\mathbf{d}} &\approx l\pi r^2 \epsilon (\mathbf{E}_0^{\mathbf{d}})_x^2 \left(1 + \frac{(\alpha r)^2}{8} + \frac{(\beta l)^2}{24} + \frac{(\alpha r \beta l)^2}{192} + \frac{(\alpha r)^4}{192} + \frac{(\beta l)^4}{1920} \right).
\end{aligned}$$

Next, we introduce the curvature terms $\hat{\kappa}_{\mathcal{F}_j^*}$ and $\hat{\kappa}_{\mathcal{E}_j}$ corresponding to the elements $\mathcal{F}_j^*$ and $\mathcal{E}_j$, respectively, and set

$$\hat{\kappa}_{\mathcal{F}_j^*} := \omega^2 \hat{\epsilon}\hat{\mu} r^2, \qquad \hat{\kappa}_{\mathcal{E}_j} := \omega^2 \hat{\epsilon}\hat{\mu} l^2. \qquad (54)$$

The terms $\hat{\alpha}$ and $\hat{\beta}$ are then eliminated by the relations $(\hat{\alpha}r)^2 = -\hat{\kappa}_{\mathcal{F}_j^*} \sin^2\theta$ and $(\hat{\beta}l)^2 = -\hat{\kappa}_{\mathcal{E}_j} \cos^2\theta$. The x-component of the field vector is expressed by the relation $(\hat{\mathbf{E}}_0^{\mathbf{d}})_x = \hat{E}_0 \sin\theta$. Using the new notations and Table 3 of precomputed

integrals, we obtain

$$\int_0^\pi 2\pi \sin\theta \hat{D}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}} d\theta \approx 2\pi^3 r^4 \hat{\epsilon}^2 \hat{E}_0^2 \int_0^\pi \left(\sin^3\theta - \frac{\hat{\kappa}_{\mathcal{F}_j^*} \sin^5\theta}{4} + \frac{5\hat{\kappa}_{\mathcal{F}_j^*}^2 \sin^7\theta}{192} \right) d\theta$$
$$= \frac{8}{3}\pi^3 r^4 \hat{\epsilon}^2 \hat{E}_0^2 \left(1 - \frac{\hat{\kappa}_{\mathcal{F}_j^*}}{5} + \frac{\hat{\kappa}_{\mathcal{F}_j^*}^2}{56} \right).$$

$$\int_0^\pi 2\pi \sin\theta \hat{E}_j^{\mathbf{d}} \hat{D}_j^{\mathbf{d}} d\theta \approx 2\pi^2 l r^2 \hat{\epsilon} \hat{E}_0^2 \int_0^\pi \left(\sin^3\theta - \frac{\hat{\kappa}_{\mathcal{F}_j^*} \sin^5\theta}{8} - \frac{\hat{\kappa}_{\mathcal{E}_j} \sin^3\theta \cos^2\theta}{24} + \right.$$
$$\left. \frac{\hat{\kappa}_{\mathcal{F}_j^*} \hat{\kappa}_{\mathcal{E}_j} \sin^5\theta \cos^2\theta}{192} + \frac{\hat{\kappa}_{\mathcal{F}_j^*}^2 \sin^7\theta}{192} + \frac{\hat{\kappa}_{\mathcal{E}_j}^2 \sin^3\theta \cos^4\theta}{1920} \right) d\theta$$
$$= \frac{8}{3}\pi^2 l r^2 \hat{\epsilon} \hat{E}_0^2 \left(1 - \frac{\hat{\kappa}_{\mathcal{F}_j^*}}{10} - \frac{\hat{\kappa}_{\mathcal{E}_j}}{120} + \frac{\hat{\kappa}_{\mathcal{F}_j^*}^2}{280} + \frac{\hat{\kappa}_{\mathcal{F}_j^*} \hat{\kappa}_{\mathcal{E}_j}}{1680} + \frac{\hat{\kappa}_{\mathcal{E}_j}^2}{22400} \right).$$

The harmonic Hodge operator $\star\hat{\epsilon}_{j,j}$ is obtained by a division of the above formulation, as expressed in Equation (51). We replace the term πr^2 by the area of the dual face $|\mathcal{F}_j^*|$ and the term l by the length of primal edge $|\mathcal{E}_j|$. Then we finally have

$$\star\hat{\epsilon}_{j,j} \approx \hat{\epsilon} \frac{|\mathcal{F}_j^*|}{|\mathcal{E}_j|} \left(\frac{1 - \frac{\hat{\kappa}_{\mathcal{F}_j^*}}{5} + \frac{\hat{\kappa}_{\mathcal{F}_j^*}^2}{56}}{1 - \frac{\hat{\kappa}_{\mathcal{F}_j^*}}{10} - \frac{\hat{\kappa}_{\mathcal{E}_j}}{120} + \frac{\hat{\kappa}_{\mathcal{F}_j^*}^2}{280} + \frac{\hat{\kappa}_{\mathcal{F}_j^*} \hat{\kappa}_{\mathcal{E}_j}}{1680} + \frac{\hat{\kappa}_{\mathcal{E}_j}^2}{22400}} \right). \tag{55}$$

The corresponding formulation for the harmonic Hodge operator $\star\hat{\mu}_{i,i}$ is obtained by a similar consideration between primal face element $\mathcal{F}_i$ and dual edge element $\mathcal{E}_i^*$. Applying the curvature terms $\hat{\kappa}_{\mathcal{F}_i}$ and $\hat{\kappa}_{\mathcal{E}_i^*}$ that correspond to the elements, respectively, we write

$$\star\hat{\mu}_{i,i} \approx \hat{\mu} \frac{|\mathcal{F}_i|}{|\mathcal{E}_i^*|} \left(\frac{1 - \frac{\hat{\kappa}_{\mathcal{F}_i}}{5} + \frac{\hat{\kappa}_{\mathcal{F}_i}^2}{56}}{1 - \frac{\hat{\kappa}_{\mathcal{F}_i}}{10} - \frac{\hat{\kappa}_{\mathcal{E}_i^*}}{120} + \frac{\hat{\kappa}_{\mathcal{F}_i}^2}{280} + \frac{\hat{\kappa}_{\mathcal{F}_i} \hat{\kappa}_{\mathcal{E}_i^*}}{1680} + \frac{\hat{\kappa}_{\mathcal{E}_i^*}^2}{22400}} \right). \tag{56}$$

8.4 A generalization for convex polygon elements

The formulations (55) and (56) are derived by assuming circular face elements. The harmonic Hodge formulation for $\star\hat{\epsilon}$ differs from Yee's Hodge operators $\star\epsilon$ by the Taylor polynomial factors, which include the curvature terms (54). In this section, we estimate the curvature term $\hat{\kappa}_{\mathcal{F}_j^*}$ by assuming an arbitrary convex polygon element. More specifically, we determine a proper estimate for radius

$r_{\mathcal{F}_j^*}$ such that

$$\hat{\kappa}_{\mathcal{F}_j^*} := \omega^2 \hat{\epsilon} \hat{\mu} r^2_{\mathcal{F}_j^*}.$$

The maximum internal sphere (radius r) and the minimal external sphere (radius R) determine lower and upper bounds for $r_{\mathcal{F}_j^*}$ respectively, as shown on the left-hand side of Figure 34. To select a proper radius between these limits, we turn to regular polygons.

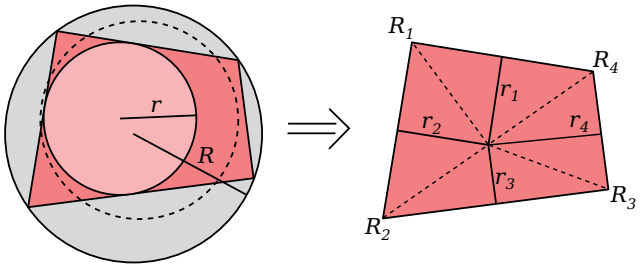


FIGURE 34 On the left, the internal and external spheres determine lower and upper limits for model radius (dotted sphere). On the right, we illustrate the computation of the average internal distance r_k and the average edge length l_k.

Using a polynomial integration on regular triangles and squares, we can observe that the formulation $r^2_{\mathcal{F}_j^*} = \frac{1}{3}(2r^2 + R^2)$ leads to the Hodge operators, where the first three terms of Taylor polynomials are equal to Equation (55). The integration is similar to the integration presented in the previous Section 8.3, and we omit it at this stage. Nevertheless, we keep this approximation as our starting point. For irregular polygons, the exact computation of the internal and external radii is inappropriate. Instead, we approximate these parameters by distances between elements. Intuitively, the internal radius corresponds to the distance between the face center and a boundary edge center. Similarly, the external radius corresponds to the distance between the face center and an extreme node position. We denote these distances by r_k and R_k, respectively, as illustrated on the right-hand side of Figure 34. Assume that n is the number of boundary elements. Then the intuition gives us the following approximation, where internal and external radii are estimated by averages of square distances

$$r^2_{\mathcal{F}_j^*} \approx \frac{1}{3n} \sum_{k=1}^{n} (2r_k^2 + R_k^2).$$

The curvature term $\kappa_{\mathcal{F}_j^*}$ can now be written by this estimate. Applying the same procedure to estimate $\kappa_{\mathcal{F}_i}$, we obtain formulations

$$\kappa_{\mathcal{F}_j^*} = \frac{\omega^2 \hat{\epsilon} \hat{\mu}}{3n} \sum_{k=1}^{n} (2r_k^2 + R_k^2), \qquad \kappa_{\mathcal{F}_i} = \frac{\omega^2 \hat{\epsilon} \hat{\mu}}{3n} \sum_{k=1}^{n} (2r_k^2 + R_k^2). \tag{57}$$

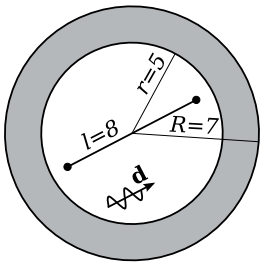


FIGURE 35 The geometry for the numerical experiment to study the wavelength of the harmonic solution.

8.5 Numerical experiments

In this section, we report the numerical experiments considering approximations of the Hodge operators. Simulations are run with the crystal grids presented in Section 2.8 and the properties of each grid are analyzed. The Yee's Hodge and the harmonic Hodge approximations are compared. We begin the experiments by considering the simulated wavelength in different wave propagating directions. Then, we study how the wavelength error correlates to the overall efficiency of the method in physical scattering problems.

8.5.1 Simulated wavelength

In the first experiment, we focus on testing how close the simulated wavelength is to the theoretical wavelength. We solve a time-harmonic problem with constant material parameters $\epsilon = 1$ and $\mu = 1$. The simulation is run with a time-harmonic source function, which has time period $T = 1$. The theoretical value for the wavelength is $\lambda = \frac{T}{\sqrt{\epsilon\mu}} = 1$, which is the reference for the simulated wavelength.

The computational domain is a sphere of radius $R = 7$, as illustrated in Figure 35. An internal sphere of radius $r = 5$ is surrounded by an absorbing ML layer, which is introduced in Section 4.3. The thickness of the layer is 2, and the absorbing parameter for Equation (27) is $\beta = 0.8\pi$. The Dirichlet boundary condition is applied at the external boundary. The scattered field formulation of Section 4.1 is applied in the absorbing layer to generate a scattered plane wave solution at the internal domain. The incident wave is a circularly polarized plane wave such that the x directional wave can be expressed as in Equation (40). Since the total wave of the system is zero, the scattered wave solution should then be opposite to the incident wave. The time-harmonic solution is observed at the line of length $l = 8$, which is parallel to the wave propagation direction $\mathbf{d}$ and centered at the domain center. The phase error of the simulated wave is computed at several positions on the line. The position and the phase error are plotted on an x-y-grid, and simple linear regression curve is fitted on the result. The slope defines the error of the simulated wavelength.

To validate the spatial isotropy, the simulations are run in 10 different wave

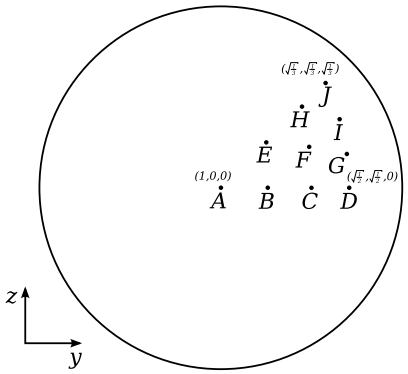


FIGURE 36 The wave propagation directions of the simulations are illustrated on the cross section of the unit sphere.

propagation directions **d** labelled from A to J. The directions are illustrated on the cross section of the unit sphere in Figure 36. All the grids, except the Z grid, are symmetric in positive and negative x-, y- and z-directions. With those grids, the selection of directions is comprehensive because the symmetrical directions of the 10 cases cover 218 directions of the full unit sphere. The Z grid is tested in 6 different coordinate axis permutation to cover all these directions. The angles between adjacent directions are about 15 degrees.

The simulation domain is discretized by six different grid types, which are presented in Section 2.8. The grids are cubic grid, FCC, BCC, A15, C15 and Z grid. The lengths of edge elements in the cubic grid are $h = 1/10$. The other grids are scaled by the base scale factors presented in Table 4 to keep the computing time fixed. The computing times are obtained by simulations of this problem. The iteration times on the bottom row represent the median iteration times needed for a simulation of time period T forward-in-time. The simulations are carried out by 200 periods of time-dependent simulation on 32 Intel Xeon E7-8837 processors at 2.67 GHz. The uniform harmonic leapfrog iteration scheme is applied in the simulations to eliminate the error of time discretization. Both the Yee's Hodge and the harmonic Hodge operators are considered, and the errors of directional wavelengths are illustrated in Figures 37 and 38.

The statistics of the wavelength error are reported in Table 5. In the case of the Yee's Hodge operator (see Figure 37), the largest relative error, -1.63%, occurs with the cubic grid in orthogonal case A. This is close to the value we computed analytically at the end of Section 8.2. Naturally, the simulated wavelength varies with the propagation direction. On average, the most accurate results are achieved with the FCC grid. In that case, the average error is 0.82%, while it is between 0.96% and 1.00% for the other grids. However, the tetrahedral grids have the smallest directional dependencies for the wavelength. The C15 grid is the most isotropic, having a standard deviation of 0.04% for the relative error. The cubic grid has the largest directional dependency with the standard deviation 0.36%. On the whole, the same characteristics are observed with the harmonic

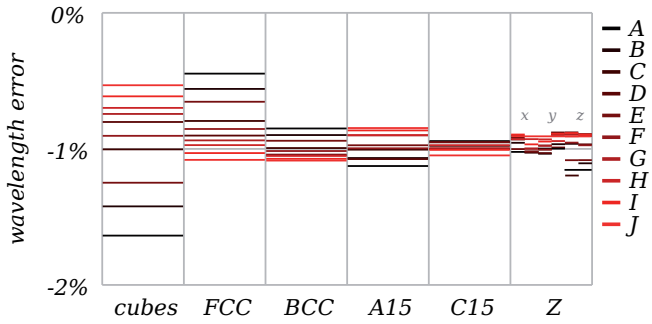


FIGURE 37 The relative errors of the simulated wavelength are plotted for the Yee's Hodge operator. Each grid is tested with 10 propagation directions, labelled as *A* - *J*. The non-symmetric Z grid is additionally rotated in 6 different orientations to cover all the directions.

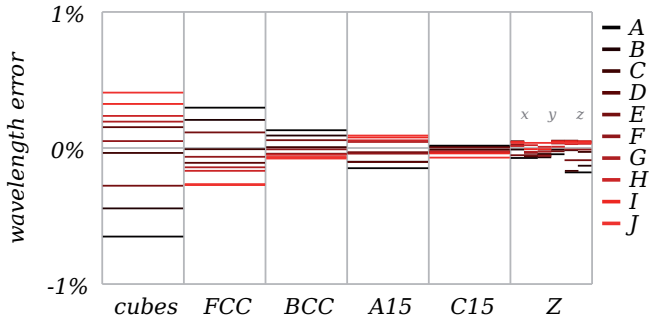


FIGURE 38 The relative errors of the simulated wavelength are plotted for the harmonic Hodge operator. In comparison to the Yee's Hodge operator (see Figure 37), the average errors are now located much closer to the zero value.

TABLE 4 Primal and dual (*) element sizes are scaled to correspond a cubic grid of edge length 1. The base scale indicates the edge length of the base rectangle in current tiling. The sizes of edges, faces and bodies are presented as lengths, areas and volumes respectively. The iteration times of the test runs are illustrated at the bottom line.

	cubic	FCC	BCC	A15	C15	Z
base scale	1	2.12	1.84	2.92	4.20	2.92
|edge|	1	1.50	1.59–1.84	1.46–1.79	1.48–1.82	1.46–1.84
|*edge|	1	0.92	0.65	0.42–0.91	0.54–0.87	0.54–0.97
|face|	1	0.97	1.20	1.07–1.38	0.95–1.35	0.92–1.38
|*face|	1	0.79	0.42–1.10	0.63–1.30	0.75–1.05	0.65–1.62
|body|	1	0.40–1.59	0.52	0.52–0.58	0.39–0.58	0.45–0.60
|*body|	1	2.38	3.11	3.04–3.14	2.87–3.53	2.79–3.46
iter. time	1.1360s	1.1364s	1.1654s	1.1552s	1.0912s	1.1663s

TABLE 5 Statistics of the wavelength error: average, standard deviation, and minimum and maximum of the relative errors.

	Yee's Hodge				Harmonic Hodge			
	average	stdev	min	max	average	stdev	min	max
cubic	-0.96%	0.36%	-1.63%	-0.53%	0.00%	0.35%	-0.65%	0.41%
FCC	-0.82%	0.21%	-1.08%	-0.44%	-0.04%	0.19%	-0.27%	0.29%
BCC	-1.00%	0.08%	-1.08%	-0.85%	0.00%	0.07%	-0.08%	0.13%
A15	-0.98%	0.10%	-1.13%	-0.85%	-0.02%	0.08%	-0.15%	0.09%
C15	-0.98%	0.04%	-1.05%	-0.92%	-0.02%	0.03%	-0.07%	0.01%
Z	-0.97%	0.07%	-1.20%	-0.88%	-0.01%	0.06%	-0.18%	0.05%

Hodge operators (see Figure 38). However, the average errors of the simulated wavelengths are moved very close to zero (-0.04%–0.00%). The C15 grid, which has the smallest directional dependency, provides the most accurate results when the harmonic Hodge operator is applied.

The current consideration shows how the wavelength is varied by the propagation direction in staggered grids. A similar result was observed earlier by Keränen et al. (2004), who compared two-dimensional regular grid to the equilateral triangular grid. Different grids have unique properties, and the wavelength error in each direction depends on both the element size and the selection of the grid. If the harmonic Hodge operator is used, the systematic error of the wavelength is very small.

8.5.2 Scattering by a sphere

In this section, we study how the error in wavelength correlates to the accuracy of a scattering simulation. We consider a scattering problem, where a time-harmonic electromagnetic wave is scattered by a spherical obstacle. The incident

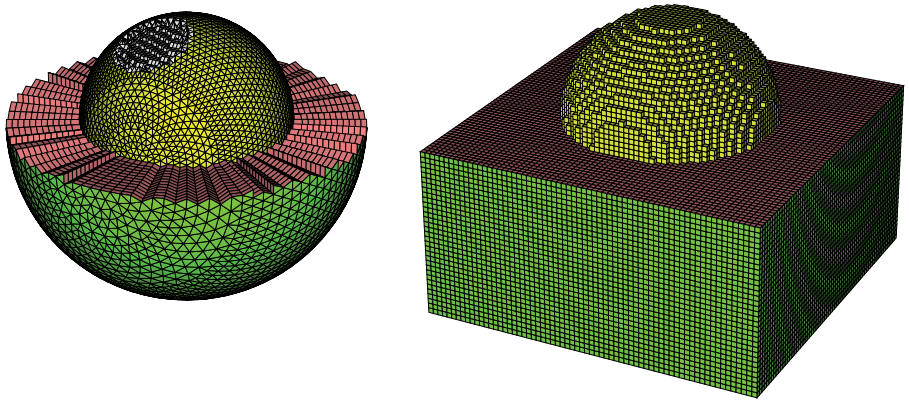


FIGURE 39 An illustration of the mesh construction in the sphere scattering simulations. The spherical mesh with cubic grid inside the scatterer is illustrated on the left. The corresponding cubic mesh for the Yee scheme simulation is illustrated on the right. The figures represent the lowest discretization levels used in these simulations.

wave is chosen to be a fully polarized plane wave of wavelength $\lambda = 1$ propagating in the direction of the positive x-axis. Formally, the incident wave is written in vector field presentation as

$$\mathbf{E}(x,t) = \begin{pmatrix} 0 \\ \cos(2\pi(x-t)) \\ 0 \end{pmatrix}, \qquad \mathbf{H}(x,t) = \begin{pmatrix} 0 \\ 0 \\ \cos(2\pi(x-t)) \end{pmatrix}. \tag{58}$$

The scattering target object is a sphere of radius 2.5. The refractive index of the sphere is $1.6 + i0.01$, which means $\hat{\epsilon}_{sphere} = 2.5599 + i0.032$ and $\hat{\mu}_{sphere} = 1$. The material parameters outside the scatterer are $\hat{\epsilon}_0 = 1$ and $\hat{\mu}_0 = 1$.

The interior of the scatterer is discretized using all the six grid types presented in Section 2.8. Five different discretization levels are applied for each grid type. The resolutions for the cubic grid are defined such that the edge element lengths are $\frac{1}{11.2}$, $\frac{1}{16}$, $\frac{1}{22.4}$, $\frac{1}{32}$ and $\frac{1}{44.8}$. The wavelength inside the scatterer is approximately 0.625, which means there are 7, 10, 14, 20 and 28 elements per scatterer wavelength, respectively. The grids of other types are scaled by the factors presented in Table 4 to keep the computing time fixed. The boundary of the sphere is discretized by triangles, and the region between the boundary and the interior grid is constructed by the Voronoi diagram construction method. To improve element quality, the boundary elements are optimized by the HOT optimization method, which is discussed in Section 2.7. The boundary surface is stretched in a radial direction such that totally a 1.7 thick layer is generated outside the scatterer. The full domain radius is then 4.2, as illustrated in the left-hand side of Figure 39. The elements outside the scatterer are scaled to match approximately each discretization level, meaning 7, 10, 14, 20 or 28 elements per wavelength. The simulations are carried out for both Yee's Hodge and harmonic Hodge operators. The harmonic non-uniform time stepping scheme is applied for the time

discretization.

For comparison, we perform the similar simulations with the classical Yee scheme, where the simulation domain is discretized by uniform cubic elements. The spherical scatterer is centered in a cubic domain, of edge length 8.4, corresponding to the smallest cube that can contain the spherical domain of radius 4.2 (see Figure 39). To maintain the same level of time consumption as in the other scattering simulations, the edge element lengths $\frac{1}{8}$, $\frac{1}{11.2}$, $\frac{1}{16}$, $\frac{1}{22.4}$ and $\frac{1}{32}$ are applied in the whole simulation domain. In the Yee scheme simulations, the Yee's Hodge operator and the Yee's uniform leapfrog time discretization scheme are applied.

The simulations are carried out by the scattered field formulation, which is discussed in Section 4.1. In this formulation, the scattered field is produced inside the target object, and the energy is then absorbed by absorbing boundary outside the target object. A 1.5 thick PML layer is initialized outside the scatterer using the absorbing factor $\beta = 1.0\pi$, presented in Equation (28). The simulations are performed by forward-in-time iteration through 200 time periods T. The iteration method is started from zero initial values $E(t^0) = 0$ and $H(t^0) = 0$, and the simulations are performed on 16 Intel Xeon E5-2670 processors at 2.60 GHz.

The resulting fields of the DEC simulations are measured at the inner boundary of the PML at a radius of 2.7. The near field solution is transferred to the far field scattering results by applying a powerful and flexible near field to far field transformation, which is discussed in Section 6.3. The far field scattering data is applied to produce the Mueller matrix, as discussed in Section 6.4.

We validate the quality of the scattering results by comparing the obtained Mueller matrices to the analytical Mie scattering solution, which is described by Bowman et al. (1969). The analytical solution of the Mueller matrix $M^{mie}(\theta)$ is computed by the Mie scattering code, presented by Mätzler (2002). The relative error of simulated Mueller matrix $M(\theta)$ is determined by integrating the Mueller matrix components over a unit sphere. The relative error can be expressed as

$$\delta M := \frac{\int_0^\pi \sin\theta \left\| M(\theta) - M^{mie}(\theta) \right\| d\theta}{\int_0^\pi \sin\theta \left\| M^{mie}(\theta) \right\| d\theta}, \tag{59}$$

where the norm $\| \cdot \|$ denotes the L^2-norm. Figure 40 illustrates the relative error of Mueller matrices in each simulation case. The simulation time increases as the element size becomes smaller. Naturally, the smaller element size implies a smaller error. For each grid type, the relative error is smaller in the case of the harmonic Hodge operator compared to the Yee's Hodge operator. The simple cubic grid with the harmonic Hodge operator seems to produce about equally efficient simulations as the FCC grid with the Yee's Hodge operator. The grid types affect the accuracy in similar way as the results of Section 8.5.1 predicted.

With the Yee's Hodge operator, the FCC grid produces the most accurate results. By using the FCC grid, a particular accuracy, expressed as relative error, is achieved with considerable lower CPU time than by using any other grid with the Yee's Hodge operator (see Figure 41). For instance, if we set the target relative error to be 5%, the CPU time required to solve the problem with the FCC grid is around one fifth of that required with the simple cubic mesh. This accuracy

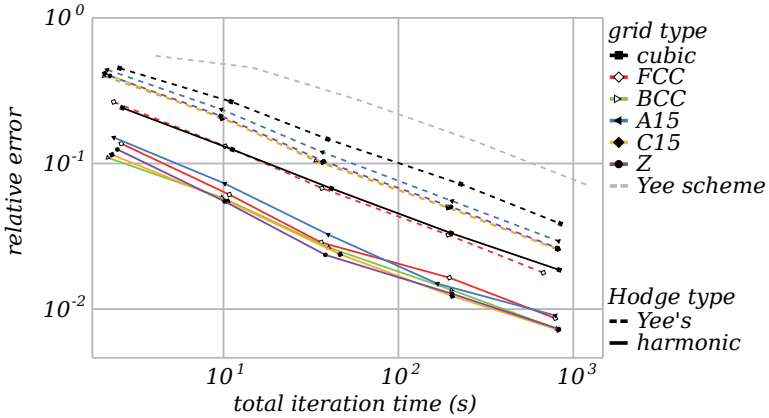


FIGURE 40 The relative error of the Mueller matrix is plotted as a function of total iteration time.

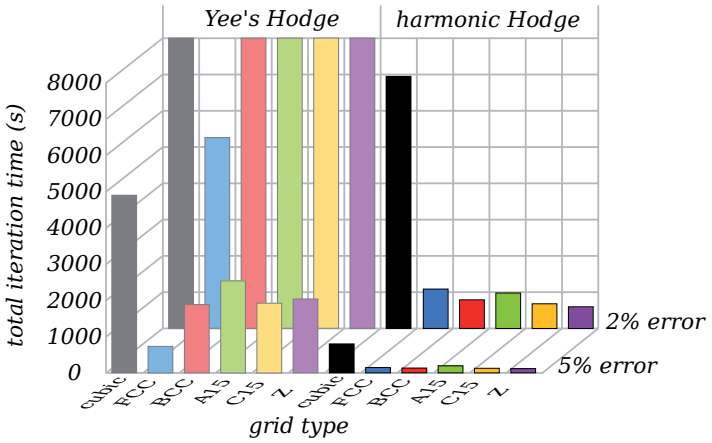


FIGURE 41 The required simulation times to obtain given error levels 2% and 5% are illustrated by bars. The bars are cut at the 8000 second level.

level is not achieved by the Yee scheme simulations, but at the 10% level, the time consumption of FCC simulation is less than one tenth compared to the Yee approach (see Figure 40).

Even higher CPU time saving is gained by utilizing the harmonic Hodge operator. The simulation with the simple cubic mesh is the most CPU time demanding, while the most accurate results are obtained with C15, Z and BCC grids. The simulations with the those grids reach a prescribed accuracy about ten times faster than the simulations with the simple cubic mesh. Hence, by selecting an appropriate grid and a sophisticated Hodge operator we managed to decrease the time consumption down to a few percent, or even less, compared to the conventional methods based on cubical grids and the Yee's Hodge operator.

8.5.3 Scattering by a more complicated particle

We introduce another scattering problem to test the grid types and the Hodge operators. In this problem, we apply a more complicated scatterer object, which obeys the shape of the Stanford bunny (see Figure 42). The Stanford bunny is a standard three-dimensional object, which was originally built by the surface reconstruction technique by Turk and Levoy (1994). Several instances of object surface meshes are freely available on the internet. The surface of the Stanford bunny object was transformed to a Delaunay two-mesh. The object is rotated such that the ground is in the negative z-direction and the bunny tail points in the negative x-direction. The object is scaled such that the bunny height (maximal difference between z-values from ear tip to the ground) is 6.173.

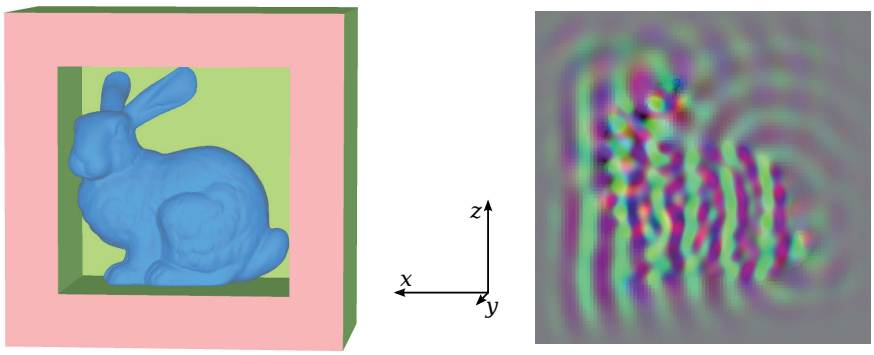


FIGURE 42 The Stanford bunny object is illustrated on the left-hand side. The x-z-plane cross section of the scattered E field is illustrated on the right. The red, green and blue components of the figure present the x- y- and z-components of the electric field, respectively.

The incident wave is chosen to be a fully polarized plane wave of wavelength $\lambda = 1$ propagating in the direction of the positive x-axis. The incident wave obeys the formulation of Equation (58). The refractive index of the scatterer is set to $2.0 + 0.01i$, meaning that the material parameters inside the scatterer are $\hat{\epsilon}_{bunny} = 3.9999 + 0.04i$ and $\hat{\mu}_{bunny} = 1$. The outside material parameters are $\hat{\epsilon}_0 = 1$ and $\hat{\mu}_0 = 1$. The scatterer object is placed in the middle of a rectangle of edge lengths 9.5, 8.1 and 9.4 in x-, y- and z-directions, respectively. A rectangular perfectly matched layer (PML) of thickness 1.5 is applied around the object to absorb the outgoing wave. The absorption term $\beta = 1.0\pi$ is employed in the PML. The external part of the domain is discretized by a regular cubic grid with element edge length $\frac{1}{10}$. The harmonic Hodge operator is applied outside the scatterer in each simulation to keep the external discretization constant. The discretization at the scatterer interior is varied using different grid types, element sizes and Hodge operators.

The interior of the scatterer is discretized by the six grid types of Section 2.8. Five discretization levels are applied for each grid type. In the cubic discretization, the edge element lengths $\frac{1}{10}$, $\frac{1}{14}$, $\frac{1}{20}$, $\frac{1}{28}$ and $\frac{1}{40}$ are utilized. The other grid

types are scaled using the factors of Table 4. Since the real part of the scatterer refractive index is 2, the interior wavelength is approximately $\lambda/2$. Thus, the number of elements per wavelength is approximately 5, 7, 10, 14 and 20, respectively, for each discretization level. From now on, these values are applied to indicate which discretization level is applied. The interior grid is attached to the scatterer surface using unstructured elements, which are obtained by the Voronoi diagram construction method. Both the Yee's Hodge and the harmonic Hodge operators are applied inside the scatterer.

The analytical solution is not available for this problem, hence the accuracy of each simulation is compared to a reference solution. The reference solution is computed on a mesh having the BCC grid with 40 elements per wavelength inside the scatterer. The scatterer boundary and the external domain are discretized similarly to the other cases. The harmonic Hodge operator is applied to the whole domain. All the simulations are carried out by a forward-in-time simulation method, where 200 total time periods are iterated. The harmonic non-uniform leapfrog scheme is applied for the time discretization.

The Mueller matrices are computed by the near field solution, which is obtained at the inner boundary of the absorbing layer. The equations of Section 6.4 are utilized, even though the physical meaning of the Mueller matrix is ambiguous due to the lack of symmetry. The Mueller matrix is compared to the reference solution in each case. The relative Mueller matrix error δM is computed by the formulation of Equation (59). Mueller matrix errors are illustrated in Figure 43. Naturally, the relative error decreases by decreasing the element size. The harmonic Hodge operator seems to induce more accurate solutions than the Yee's Hodge operator. The error, which is produced by the Yee's Hodge operator, is 1.99–6.63 times the error, produced by the harmonic Hodge operator. The grid type does not have as remarkable an effect on the accuracy as we observed in the previous section. With the Yee's Hodge operator, the largest error is less than 1.53 times the smallest error at each discretization level. In each case, the smallest error is induced by the FCC grid. The cubic grid suffers in comparison, especially when the harmonic Hodge operator is applied. With a discretization level 14 elements per wavelength, the error of cubic grid simulation is 1.98 times the error of C15 grid simulation. The C15 and FCC grids induce the smallest error levels, when the simulations are carried out by the harmonic Hodge operator.

The similar comparison is carried out by comparing the internal fields at the target object. The reference interior fields $\mathbf{E}^{ref}(\mathbf{p}_i)$ and $\mathbf{H}^{ref}(\mathbf{p}_i)$ are obtained from the reference solutions at 2012 uniformly distributed positions $\mathbf{p}_i \in \mathbb{R}^3$ inside the target object. The form-to-vector transformation of Section 6.2 is applied to obtain the fields at certain time instance. The corresponding vector fields $\mathbf{E}(\mathbf{p}_i)$ and $\mathbf{H}(\mathbf{p}_i)$ are computed for different discretizations to perform the comparison. The relative interior field error δS^p is obtained by formulation

$$\delta S^p := \sqrt{\frac{\sum_i \epsilon(\mathbf{p}_i) \left\|\mathbf{E}(\mathbf{p}_i) - \mathbf{E}^{ref}(\mathbf{p}_i)\right\|^2 + \sum_i \mu(\mathbf{p}_i) \left\|\mathbf{H}(\mathbf{p}_i) - \mathbf{H}^{ref}(\mathbf{p}_i)\right\|^2}{\sum_i \epsilon(\mathbf{p}_i) \left\|\mathbf{E}^{ref}(\mathbf{p}_i)\right\|^2 + \sum_i \mu(\mathbf{p}_i) \left\|\mathbf{H}^{ref}(\mathbf{p}_i)\right\|^2}}. \quad (60)$$

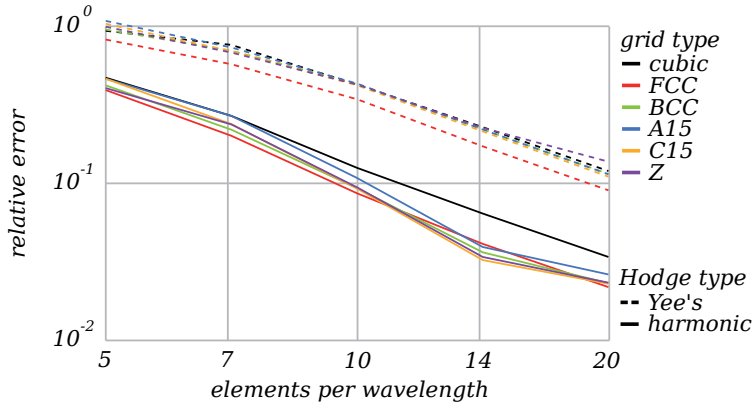


FIGURE 43 The Mueller matrix error δM is plotted as a function of the discretization level. The results of six grid types and two Hodge operators are illustrated in separate lines.

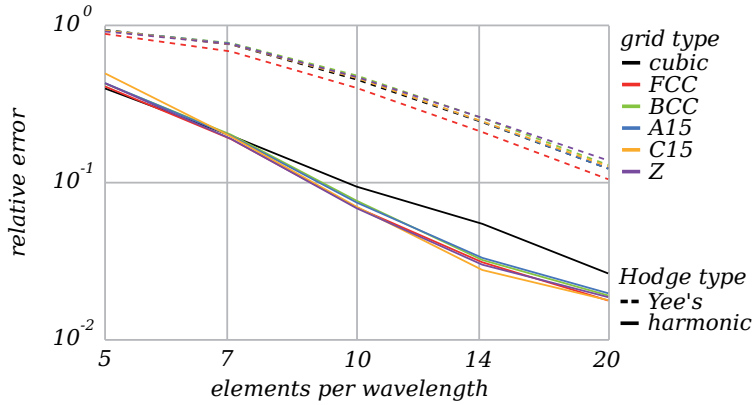


FIGURE 44 The interior field error δM is plotted as a function of the discretization level. The grid types and the Hodge operators are illustrated in separate lines.

The relative errors of the interior fields are illustrated in Figure 44. The behavior of the error norm δS^p is similar to the behavior of the Mueller matrix error δM. The Hodge operator seems to have very significant effect on the accuracy. The grid type has a smaller, but still remarkable, effect on the accuracy.

8.5.4 Conclusions

We formulated a harmonic Hodge operator to optimize the DEC framework in cases of time-harmonic problems. The harmonic Hodge operator has a similar diagonal matrix presentation as the Yee's Hodge operator and offers a small correction on the diagonal terms. Thus, the convergence properties of the modified Hodge operator remain unchanged. The properties of different Hodge operators were analyzed by numerical experiments. The DEC simulations with a harmonic Hodge operator produce a better adjusted wavelength. Then the accuracy of the DEC simulation depends on the isotropy of the grid. Several grid constructions were compared, and the tetrahedrally close-packed structures offered the most isotropic grid constructions.

9 ITERATION METHODS FOR TIME-PERIODIC PROBLEMS

In this thesis, the Maxwell equations are reformulated in a discrete domain by the method based on discrete exterior calculus (DEC). We used leapfrog style time discretization to obtain a time-dependent solver, where electromagnetic wave propagation is modeled forward-in-time. A time-dependent solver can be applied for many types of problems, including time-periodic and time-harmonic ones. However, the efficiency of the forward-in-time iteration is not always as good as the efficiency of the other solution methods. In this chapter, we recall the concept of controllability methods for solving time-harmonic problems. The controllability method leads to a minimization problem, which is then solved by the iterative conjugate gradient (CG) algorithm.

Let us now express the forward-in-time iteration scheme in a shorter notation, which enables us to follow the forthcoming arguments and basic ideas more easily. The leapfrog iteration methods can be performed by updating a single column vector by matrix operators. We write the discrete forms E and H and the additional PML terms D and B in a single column vector and denote it by

$$u := \begin{pmatrix} E \\ H \\ D \\ B \end{pmatrix}.$$

The additional terms D and B are needed only for the value update equations, which have negative absorption due to the PML formulation. Nevertheless, we consider D and B as full-sized column vectors to simplify the notation. Then most of the terms are zero throughout the simulation. Both E and D are considered as column vectors of m rows. The column vectors H and B include n rows.

The forward-in-time simulation scheme is now considered as a series of matrix multiplications. An update of a single value of u is carried out by a matrix multiplication $u^{new} := Q_j u^{curr} + b_j$, where u^{curr} is a vector of the current values and u^{new} is a vector of the next values. The matrix Q_j differs from the identity matrix in a single row, and the source vector b_j has at maximum one non-zero

component such that only one component of u is affected. Starting from the initial values $u^0 = (E^0, H^0, D^0, B^0)^T$, the iteration over one time step Δt is operated by

$$\begin{aligned} u^{k+1} &= Q_s \left(\cdots \left(Q_2 \left(Q_1 u^k + b_1 \right) + b_2 \right) \cdots \right) + b_s \\ &=: Qu^k + b^k. \end{aligned}$$

In the uniform leapfrog time stepping scheme, each value update equation is performed once during the uniform Δt iteration. If the PML terms are excluded, the number of matrix products is then $s = m + n$. In the non-uniform time stepping scheme, some values are updated several times during the Δt iteration. The number of matrix products is then

$$s = \sum_{i=j}^{m} s_{E_j} + \sum_{j=i}^{n} s_{H_i},$$

where s_{E_j} and s_{H_i} are the number of time steps for E_j and H_i elements, respectively, as defined in Section 7.2. Including the PML increases the number of matrix products. The matrix $Q := Q_s \cdots Q_2 Q_1$ is not necessarily a sparse matrix. Thus, this matrix is not expressed explicitly in an efficient solver. Instead, each pair $\{Q_j, b_j\}$ is expressed as a sparse matrix. Since the pair $\{Q_j, b_j\}$ has only one significant row, the expression can be reduced into a sparse row formulation. Each value update is then performed by the operators that are determined by this single row. The matrix product Q can be expressed as an ordered list of sparse rows. In this approach, the number of computing operations and the amount of memory consumption are reduced.

The forward-in-time simulation through time period T is carried out by several iterations over Δt. Assume that Δt is selected such that $T = l\Delta t$ holds for a selected integer l. The simulation through time period T is then applied by consecutive matrix multiplications

$$\begin{aligned} u^{k+l} &= Q \left(\cdots \left(Q \left(Qu^k + b^k \right) + b^{k+1} \right) \cdots \right) + b^{k+l-1} \\ &=: Q^l u^k + b^{k,l}. \end{aligned}$$

Here, the multiplication matrix Q remains unchanged during the iterations and the source vector b^k varies depending on the current time instance. In the shorter notation, the Q^l represents the l:th power of Q and the $b^{k,l}$ represents the combined source terms for the time period.

A time-periodic system can now be expressed with the new notation. In the time-periodic solution, the same sequence of time instances is repeated on time intervals T, as illustrated in Figure 45. Thus, the discrete solution is time-periodic if $u^{k+l} = u^k$ for every k. In a discrete system, this is obtained only if the source terms $b^{k,l}$ are also repeated on time intervals T. We call the discrete system time-periodic if $b^{k,l} = b^{k+l,l}$ for every k.

Due to the nature of the leapfrog time discretization, the electric field values of u^k are determined at different time instances than the magnetic field values.

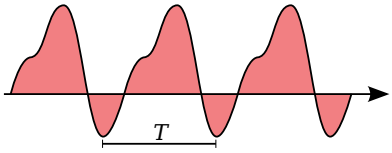


FIGURE 45 In a time-periodic solution, the same sequence of instances is repeated on time intervals T.

The values can be transformed to a synchronized time instance by applying a transformation on electric field values, as discussed in Section 5.4. The transformation can be considered as a matrix vector product, similarly to the forward-in-time iteration. We denote the synchronized time instance solution using the underlined symbols and define $\underline{u}^k := \left(E(t^k), H(t^k), D(t^k), B(t^k)\right)^T$. The synchronization transformation is applied by operation $\underline{u}^k = \underline{Q}\, u^k + \underline{b}^k$, where $\underline{Q}$ is a transformation matrix and $\underline{b}^k$ is a source function. The inverse transformation is obtained by $u^k = \underline{Q}^{-1}(\underline{u}^k - \underline{b}^k)$. The iteration between synchronized time instances over one period is obtained by the system

$$\underline{u}^{k+l} = \underline{Q}\left(Q^l \underline{Q}^{-1}\left(\underline{u}^k - \underline{b}^k\right) + b^{k,l}\right) + \underline{b}^k =: \underline{Q}Q^l\underline{Q}^{-1}\underline{u}^k + \underline{b}^{k,l}. \tag{61}$$

9.1 Asymptotic iteration

If the forward-in-time iteration is applied for a time-periodic problem, then the solution method relies on asymptotic convergence. To express the convergence as a sequence, we change to the lower corner indices u_n to illustrate time instances between time period T. By the terms of Equation (61), we define $u_n := \underline{u}^{nl}$, where $l\Delta t = T$. Then we obtain series $u_0, u_1, u_2 \ldots$, which represents solutions at synchronized time instances on T periods. By assuming a time-periodic system, we can use constant source terms on each iteration and we write $b := \underline{b}^{0,l}$. Then the iteration step over one time period can be expressed as

$$u_{n+1} = \underline{Q}Q^l\underline{Q}^{-1}u_n + b. \tag{62}$$

Since the terms $\underline{Q}$, Q^l and b are constants, the solution is naturally time-periodic if the consecutive instances are equal i.e. $u_{n+1} = u_n$. Thus, the convergence of the iterative method can be controlled by the difference $u_{n+1} - u_n$. Inspired by the energy norm of Equation (38), we write the control norm as

$$P_\Delta := \frac{1}{2}(u_{n+1} - u_n)^T \Lambda (u_{n+1} - u_n), \tag{63}$$

where

$$\Lambda := \begin{pmatrix} \star\epsilon & 0 & 0 & 0 \\ 0 & \star\mu & 0 & 0 \\ 0 & 0 & \star\epsilon^{-1} & 0 \\ 0 & 0 & 0 & \star\mu^{-1} \end{pmatrix}.$$

The iterative system (62), starting from an initial value u_0, is called an **asymptotic iteration method**. The asymptotic iteration is performed simply by time-dependent simulation, and one iteration implies a full time period T of time-dependent iteration. The asymptotic iteration method is a natural way of solving time-periodic problems. In this procedure, the electromagnetic energy is basically produced by a source and absorbed due to either a boundary condition or absorbing materials. The asymptotic iteration method converges only if the solution tends to oscillate like a time-periodic solution. Thus, the system must have a balance between the emission and the absorption. The asymptotic iteration method suffers in the geometries, where the energy is trapped without reaching absorbing elements. In this thesis, the asymptotic iteration method is applied by using the zero initial values $u_0 = 0$, if not otherwise mentioned.

9.2 Controllability method for time-periodic problems

In this section, we consider an iteration method, which is aimed to accelerate the solution convergence in time-periodic problems. The iteration method is based on the exact controllability concept presented by Bristeau et al. (1998, 1999). Essentially, the approach is a controlled variation of the asymptotic approach with periodic constraints, and the iteration is applied by the **conjugate gradient** (CG) minimization algorithm. The controllability method is derived for the generalized Maxwell equations by Pauly and Rossi (2011), where the method is based on an approach, which was proposed in the short paper of Glowinski and Rossi (2006). The control method has been recently applied to acoustics in Kähkönen et al. (2011) and elastodynamics in Mönkölä et al. (2008).

We recall the classical CG algorithm, originated by Hestenes and Stiefel (1952). The CG method is developed for solving a linear system $Ax + y = 0$ where the matrix A is symmetric ($A^T = A$) and positive definite ($x^T Ax > 0$ for all non-zero vector x). The CG method can be defined as an iterative algorithm closely related to gradient descent or the deepest descent method. Starting from an initial value x_0, the solution is updated on each iteration by

$$x_n = x_{n-1} + \alpha d_n,$$

where d_n is the current minimizing direction and $\alpha \in \mathbb{R}$ defines the optimal distance for minimization. In the CG algorithm, the minimization directions d_n are conjugate with respect to A, meaning that $d_i^T A d_j = 0$ for all integers $i \neq j$. This property can be guaranteed without storing previous minimization directions by the CG algorithm 1.

Theoretically, the CG method produces the exact solution after a finite number of iterations, where the number of iterations is not larger than the dimension of the matrix A. Fortunately, the CG method, viewed as an iterative algorithm, may reach the required tolerance even for a much smaller number of iterations. The iterative method provides monotonically improving approximations x_n to

Algorithm 1 Conjugate gradient algorithm for solving x from $Ax + y = 0$ where A is a symmetric and positive definite matrix.

set iteration count $n \leftarrow 0$
get initial control vector x_0
compute residual $r_0 \leftarrow Ax_0 + y$
compute norm $s_0 \leftarrow r_0^T r_0$
set first minimizing direction $d_1 = r_0$
while $s_n > \epsilon^2$ **and** $n < N$ **do**
  update iteration count $n \leftarrow n + 1$
  compute gradient $d \leftarrow Ad_n$
  compute parameter $\alpha \leftarrow -\frac{s_{n-1}}{d^T d_n}$
  update control vector $x_n \leftarrow x_{n-1} + \alpha d_n$
  update residual vector $r_n \leftarrow r_{n-1} + \alpha d$
  compute norm $s_n \leftarrow r_n^T r_n$
  compute parameter $\beta \leftarrow \frac{s_n}{s_{n-1}}$
  update minimizing direction $d_{n+1} \leftarrow r_n + \beta d_n$
end while
return x_n as solution

the exact solution, and the improvement is typically linear (Saad, 2003). Still, the linearity of the convergence is not guaranteed. The speed of the convergence depends on the condition number $\kappa(A)$ of the system matrix A. A larger condition number indicates slower convergence, and vice versa.

A standard procedure for controlling the condition number of the system matrix is called preconditioning (Saad, 2003; Eisenstat, 1981; Axelsson, 1974). The preconditioning means that instead of directly solving the linear system $Ax + y = 0$, we apply the CG algorithm 1 to a linear system

$$P^{-1}(A(P^T)^{-1}x' + y) = 0,$$

where P is an invertible matrix and $x' = P^T x$. The matrix $M = PP^T$ is called the preconditioner, and it must be symmetric and positive definite. The aim of the preconditioning is to minimize the condition number of the system matrix $P^{-1}A(P^T)^{-1}$. The preconditioned CG algorithm is written in Algorithm 2.

The CG algorithm can also be applied for system of linear equations where the system matrix A is not symmetric. In case of an arbitrary system matrix A, the standard procedure is to solve normal equations

$$A^T(Ax + y) = 0$$

instead of original equations $Ax + y = 0$ (Saad, 2003, Chapter 9.5). The matrix $A^T A$ is symmetric and positive semidefinite, which means the CG algorithm is valid. However, the downside of forming the normal equations is that the condition number $\kappa(A^T A)$ is equal to $\kappa^2(A)$, and the rate of convergence may be slow.

Algorithm 2 Preconditioned Conjugate Gradient Algorithm for solving x from $Ax + y = 0$, where A is a symmetric and positive definite matrix. Using the symmetric and positive definite preconditioner matrix $M = PP^T$, the algorithm is identical to Algorithm 1 when applied for problem $P^{-1}(A(P^T)^{-1}x' + y) = 0$ where $x' = P^T x$.

set iteration count $n \leftarrow 0$
get initial control vector x_0
compute residual $r_0 \leftarrow M^{-1}(Ax_0 + y)$
compute norm $s_0 \leftarrow r_0^T M r_0$
set first minimizing direction $d_1 = r_0$
while $s_n > \epsilon^2$ **and** $n < N$ **do**
  update iteration count $n \leftarrow n + 1$
  compute gradient $d \leftarrow M^{-1}Ad_n$
  compute parameter $\alpha \leftarrow -\frac{s_{n-1}}{d^T M d_n}$
  update control vector $x_n \leftarrow x_{n-1} + \alpha d_n$
  update residual vector $r_n \leftarrow r_{n-1} + \alpha d$
  compute norm $s_n \leftarrow r_n^T M r_n$
  compute parameter $\beta \leftarrow \frac{s_n}{s_{n-1}}$
  update minimizing direction $d_{n+1} \leftarrow r_n + \beta d_n$
end while
return x_n as solution

The controllability method is described to offer a solution for time-periodic Maxwell problems. We formulate the controllability as a minimization problem, where the control norm P_Δ of Equation (63) is the term to be minimized. The control norm can be rewritten as a function of u_n by applying the Formulation (62). Then we have

$$P_\Delta(u_n) := \frac{1}{2}\left[\left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)u_n + b\right]^T \Lambda \left[\left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)u_n + b\right].$$

The problem is to find x such that the norm $P_\Delta(x)$ is minimized. Then x represents the solution of the time-periodic problem at the initial time instance. The norm can be minimized by finding the zero of its gradient, which can be expressed as

$$\nabla P_\Delta(x) = \left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)^T \Lambda \left[\left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)x + b\right].$$

Setting $\nabla P_\Delta(x) = 0$, we formulate a linear system, where the system matrix is symmetric and positive definite. Thus, we can solve the system by the CG algorithm 1 using the parameters

$$A := \left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)^T \Lambda \left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right),$$
$$y := \left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)^T \Lambda b.$$

The control vector for the CG algorithm can be initialized by any initial solution $x_0 := u_0$. In this thesis, we use zero initial control vector $x_0 = 0$, if not mentioned otherwise. The result x_n of the CG iteration is taken as the solution x.

The transpose matrix $(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I})^T$ is called the **adjoint state matrix** and sometimes is also considered for backward-in-time equations (Glowinski and Lions, 1995; Mönkölä, 2011). The multiplication by the adjoint state matrix is quite straightforward. We write

$$\left(\underline{Q}Q^l\underline{Q}^{-1} - \mathcal{I}\right)^T = \left(\underline{Q}^{-1}\right)^T \left(Q^T\right)^l \underline{Q}^T - \mathcal{I}.$$

Since $Q := Q_s \cdots Q_2 Q_1$, we have $Q^T := Q_1^T Q_2^T \cdots Q_s^T$, where each Q_j^T differs from the identity matrix only in a single column. Thus, the operation $u^{new} := Q_j^T u^{curr}$ updates a small number of values in u only by adding a single scaled value on each affected row. Each value update consists of one multiplication and one plus operation, which means the transpose operation is as fast as the original operation.

The efficiency of the controllability method relies on the convergence of the CG algorithm. Thus, we call the method a CG iteration method. To improve the convergence of the CG method, we recommend applying a preconditioner and using the CG algorithm 2. Practically, we have found that the matrix $M = \Lambda$ is a good choice for the preconditioner.

An iteration of the CG algorithm is twice as demanding as the asymptotic iteration because one iteration includes a period T of forward-in-time iteration and a period T of adjoint state iteration. The parallel implementation of the CG method is as simple as the parallel implementation of the traditional asymptotic method. In practice, the most time demanding part of the algorithm is the iterations over time period T. The forward-in-time iteration can be parallelized in a very efficient manner by mesh partitioning, where a small amount of cross-processor communication is needed. The same mesh partitioning can be applied for adjoint state iteration. In the CG algorithm 2, all the vectors x_n, r_n, d_n and d and diagonal matrix Λ can be separated for different threads, and thus the amount of cross-processor communication remains small. An advantage of the conjugate gradient algorithm is that the source terms b are needed only once at the initial stage of the method. The rest of the iteration is carried out by performing only homogeneous matrix multiplications.

9.3 Controllability method with half-periodic constraints

The controllability method of the previous section was formulated for the time-periodic problems. A similar iteration method can also be formulated for problems with half-periodic constraints $u(t) + u(t + \frac{T}{2}) = 0$ where t is any time instance and T is the time period. A wave with the half-periodic constraints is symmetric on $T/2$ interval, as illustrated in Figure 46. For example, time-harmonic waves coincide with this category. In this thesis, we call the following iteration method the half-periodic conjugate gradient (HPCG) method. The iteration time of the HPCG method is, generally, half of the iteration time of the CG method and similar to the iteration time of the asymptotic iteration method.

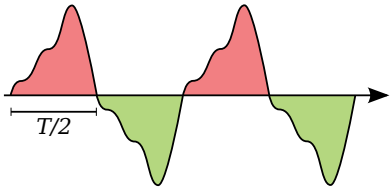


FIGURE 46 A time-periodic wave with half-periodic constraints $u(t) + u(t + \frac{T}{2}) = 0$ is symmetric on time intervals $T/2$.

Assume that the iteration over time period T is divided into an even number of iterations over Δt. In other words, the integer l, for which $l\Delta t = T$, is now even, meaning that the number $l/2$ is also an integer. By the terms of Equation (61), the iteration over a half time period is expressed as

$$\underline{u}^{k+l/2} = \underline{Q}Q^{l/2}\underline{Q}^{-1}\underline{u}^{k} + \underline{b}^{k,l/2}.$$

For the time discretized expression, the half-periodic constraints are written as $u^{k+l/2} + u^{k} = 0$. This relation can hold only if the system has the half-periodic constraints $\underline{b}^{k,l/2} = \underline{b}^{k+l/2,l/2}$ for all k.

We define sequence $v_0, v_1, v_2 \ldots$ by setting $v_n := (-1)^n \underline{u}^{nl/2}$. Starting from the initial value $v_0 := u_0$, the same sequence can be expressed recursively as

$$v_{n+1} = -(\underline{Q}Q^{l/2}\underline{Q}^{-1}v_n + c).$$

where $c := \underline{b}^{0,l/2}$. Naturally, the half-periodic constraints are realized only if $v_{n+1} = v_n$. Similarly to the previous section, we formulate a controllability method by minimizing the control norm

$$P_{\Delta}^{1/2} := \frac{1}{2}(v_{n+1} - v_n)^T \Lambda (v_{n+1} - v_n).$$

The control norm can be expressed as a function of v_n by writing

$$P_{\Delta}^{1/2}(v_n) := \frac{1}{2}\left[\left(\underline{Q}Q^{l/2}\underline{Q}^{-1} + \mathcal{I}\right)v_n + c\right]^T \Lambda \left[\left(\underline{Q}Q^{l/2}\underline{Q}^{-1} + \mathcal{I}\right)v_n + c\right].$$

The control norm is minimized at solution x by finding the zero of gradient i.e. $\nabla P_{\Delta}^{1/2}(x) = 0$. This equation formulates a linear system, which can be solved by the CG algorithm 1, where the following terms are used

$$\begin{aligned} A &:= \left(\underline{Q}Q^{l/2}\underline{Q}^{-1} + \mathcal{I}\right)^T \Lambda \left(\underline{Q}Q^{l/2}\underline{Q}^{-1} + \mathcal{I}\right), \\ y &:= \left(\underline{Q}Q^{l/2}\underline{Q}^{-1} + \mathcal{I}\right)^T \Lambda c. \end{aligned}$$

To improve the convergence of the iterative method, we recommend using the preconditioned CG algorithm 2, similarly to the previous section, with the preconditioner $M = \lambda$. In this thesis, the iteration is started with a zero initial value $x_0 = 0$, if not mentioned otherwise.

9.4 Combined methods

The iteration methods, described in Sections 9.1, 9.2 and 9.3, can be easily combined together using the solution of first method as the initial value for the second method. As we show by the numerical experiments in the next section, it is usually beneficial to start the iteration with the asymptotic iteration method. If the problem includes a significant amount of absorption, the asymptotic iteration method converges quite well. Then the asymptotic method gives a moderate approximation for the solution before starting one of the controllability methods. With good initial values, the convergence of the conjugate gradient method might be significantly faster than with zero initial values. The final result is controlled by the last iteration method.

Each of the original methods are derived by using different constraints for the solution. A time-harmonic solution fulfills all of the constraints and, thus, all of the methods can be applied for time-harmonic problems. Still, the methods might converge to different solutions because none of the methods guarantee its solution to be time-harmonic. Methods with looser restrictions might find a wider range of acceptable solutions, as illustrated by numerical experience in Section 9.5.2. This happens specifically with wave trapping obstacles, as discussed in Bardos and Rauch (1994) and Zuazua (2007).

9.5 Numerical experiments

In this section, we present numerical experiments to compare the previously introduced iteration methods. All the iteration methods are suitable for solving time-harmonic problems and, thus, we formulate the time-harmonic test problems. We are interested in the convergence speed of each iteration method. The natural choice is to consider the convergence of the error norm, which represents the solution error with respect to the exact solution. The iteration of the discretized Maxwell system never reaches the exact solution due to the inaccuracy of spatial and time discretization. Thus, the exact error norm is not the best norm to illustrate the convergence properties of the iteration methods. We also consider the error norm, which represents the solution error with respect to the so-called converged solution. The converged solution is obtained by using the particular discretization and performing the iteration much further than the convergence consideration proceeds. This error norm eliminates discretization error and allows the convergence properties to be illustrated much better.

9.5.1 Plane wave in a cube

We start with comparing the iteration methods in a very simplified geometry. In these experiments, the domain is a cube, where the edges are parallel to coordi-

nate axis x, y and z, and the edge length is 2. The domain boundary is modeled by the absorbing Silver-Müller boundary condition such that the source terms generate a circularly polarized plane wave inside the domain region. The wavelength is $\lambda = 1$, and the propagation direction is diagonal with respect to the domain cube. The plane wave can be expressed as

$$\mathbf{E}(\phi, t) = \begin{pmatrix} -\frac{\cos\phi}{\sqrt{2}} - \frac{\sin\phi}{\sqrt{6}} \\ \frac{\cos\phi}{\sqrt{2}} - \frac{\sin\phi}{\sqrt{6}} \\ \frac{\sin\phi}{\sqrt{6}} \end{pmatrix}, \qquad \mathbf{H}(\phi, t) = \begin{pmatrix} \frac{\sin\phi}{\sqrt{2}} - \frac{\cos\phi}{\sqrt{6}} \\ -\frac{\sin\phi}{\sqrt{2}} - \frac{\cos\phi}{\sqrt{6}} \\ \frac{\cos\phi}{\sqrt{6}} \end{pmatrix},$$

where the angular term is $\phi := 2\pi\left(\frac{x+y+z}{\sqrt{3}} - t\right)$. The material parameters $\epsilon = 1$ and $\mu = 1$ are applied in the whole domain. The domain is discretized using regular grids with cubic cells. Four discretization levels are applied where the length h of edge elements are $\frac{1}{10}$, $\frac{1}{20}$, $\frac{1}{50}$ and $\frac{1}{100}$. The discretization is operated using the harmonic Hodge operator and the harmonic uniform leapfrog method.

We compare the four iteration methods, which are the asymptotic iteration method (Section 9.1), the CG iteration method (Section 9.2), the HPCG method (Section 9.3) and the combined method (Section 9.4). Each iteration method began with zero initial values $u_0 = 0$. The combined method starts with 10 asymptotic iterations, and the rest of the iteration is carried out by the HPCG iterations. We consider the iteration stage by the term "iterated periods," which indicate the number of time periods simulated in the iteration process. In the asymptotic iteration method, the HPCG method and the combined method, this implies one iteration step of the particular method. The periodic CG iteration takes two iterated periods per iteration step.

Two error norms are evaluated during the iterations. The first error norm δS is obtained by comparing the numerical solution to the exact field $\mathbf{E}(\phi, t)$, $\mathbf{H}(\phi, t)$ and using error formulation, as described in Equation (41). The second error norm δS^{ref} represents error with respect to a converged solution. The converged solution E^{ref}, H^{ref} is computed separately for each iteration method, and it is obtained by executing the particular iteration procedure over 1000 iterated periods. The methods, which are based on the conjugate gradient algorithm, are alternatively stopped if residual norm becomes smaller than 10^{-13}. The formulation for δS^{ref} is expressed in Equation (42).

The error norms δS and δS^{ref} are computed at several iteration stages, and the convergence of the norms is illustrated in Figure 47. The results indicate that the controllability methods (CG and HPCG) converge slower than the asymptotic iteration at the beginning of the process. The asymptotic iteration slows down quickly, and the controllability methods outperform the asymptotic method at some stages. Both CG and HPCG methods have similar behavior at the beginning. The HPCG seems to have more steady convergence compared to the CG at the latter part of the iteration. The fastest convergence of δS^{ref} is obtained by the combined iteration method.

The progress of the exact error norm δS stops at a certain level, which depends on the element size h. Naturally, a smaller element size h implies a smaller

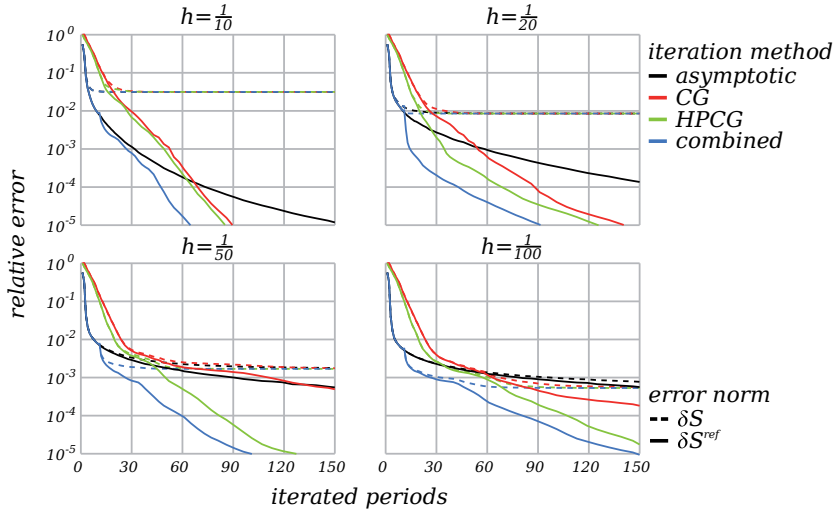


FIGURE 47 Convergence of the error norms δS and δS^{ref}, when the mesh element size h is varied.

error and the error seems to decrease in relation to h^2. With respect to the error norm δS, the controllability methods do not improve the convergence compared to asymptotic iteration on the smaller tasks ($h = \frac{1}{10}$ and $h = \frac{1}{20}$). Nevertheless, the controllability methods become valuable, especially with larger tasks, where the element size is small and the solution is more accurate ($h = \frac{1}{50}$ or $h = \frac{1}{100}$). On case $h = \frac{1}{100}$, the accuracy level $\delta S = 0.00078$, which takes 150 periods of asymptotic iteration, is obtained by 46, 69 and 81 iterated periods by the combined method, the HPCG method and the CG method, respectively.

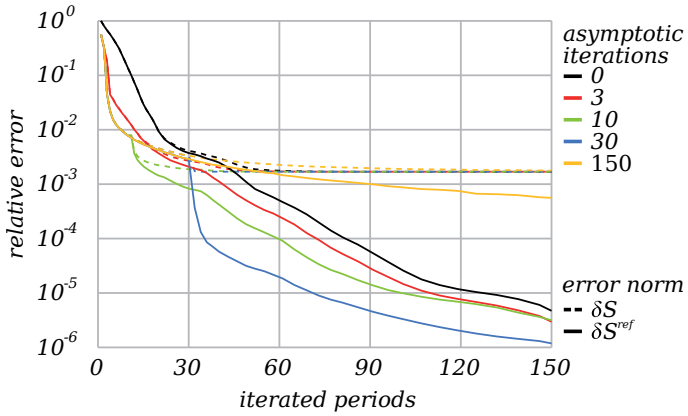


FIGURE 48 Convergence of the error norms by different combinations of asymptotic and HPCG iteration methods.

The combined method takes advantage of both asymptotic and HPCG iteration methods. The asymptotic iteration method converges quickly in the beginning, and the convergence of the HPCG is accelerated by taking the asymptotic

solution as an initial value. There is no evidence of what is the optimal number of asymptotic iterations before starting the HPCG iteration. We tried to figure that out by simulating the case $h = \frac{1}{50}$ with a few options. The number of asymptotic iterations varied between 0, 3, 10, 30 and 150. The value 0 indicates a completely HPCG procedure, and the value 150 indicates a completely asymptotic procedure. The results are illustrated in Figure 48. The results show that the HPCG method seems to converge faster the more asymptotic iterations are performed in advance. Naturally, the asymptotic iteration method also consumes time, which means that the number of asymptotic iterations should be selected as a compromise between these facts.

9.5.2 Trapped wave simulations

The problem of the previous section includes a lot of absorption. The asymptotic iteration process, which requires the absorbing terms, converges quite well in such cases. In this section, we modify the problem by changing the boundary conditions on domain faces. Instead of having the absorbing Silver-Müller boundary condition on all faces, we employ the perfectly reflecting Dirichlet boundary condition on certain areas. By using the Dirichlet boundary condition, we decrease the amount of absorption in the system.

The geometry of this section is similar to the geometry of the previous section. Thus, the simulation domain is a cube of edge length 2, and the circularly polarized plane wave of wavelength $\lambda = 1$ crosses the domain diagonally. The domain is discretized by a regular cubic grid, where the lengths of edge elements are $\frac{1}{20}$. We vary the number of Dirichlet faces D such that the opposite faces always have the same boundary condition. Let us consider the domain as a dice. In the first case ($D = 0$), all the faces 1–6 are modeled using the absorbing boundary condition. This is exactly the same problem as the $\lambda/20$ case in the previous section. The second case ($D = 2$) has two Dirichlet faces; such as 1 and 6 on a dice. The third case ($D = 4$) has four Dirichlet faces; for example numbers 1, 2, 5 and 6 on a dice. In the last case ($D = 6$), all six faces are modeled by Dirichlet boundary. Like in the previous section, the error norms δS and δS^{ref} are computed at several simulation stages, and the results are illustrated in Figure 49.

Figure 49 indicates that the convergence of δS^{ref} becomes slower with all iteration methods when the amount of absorption is decreased. The asymptotic iteration method suffers the most, as we expected. The asymptotic method does not converge at all when all six boundary faces are perfectly reflecting ($D = 6$). The controllability methods converge well in every case. The progress of the exact error norm δS indicates that the iterative methods converge to different solutions when the Dirichlet boundaries are employed. The HPCG method and the combined iteration method converge close to the expected time-harmonic solution. The asymptotic method and the CG method converge to a time-periodic solution, which is relatively far from the time-harmonic solution. In fact, all four iteration methods find the solutions that they are looking for. For example, the converged solution of the CG method fulfills the periodic constraints very well,

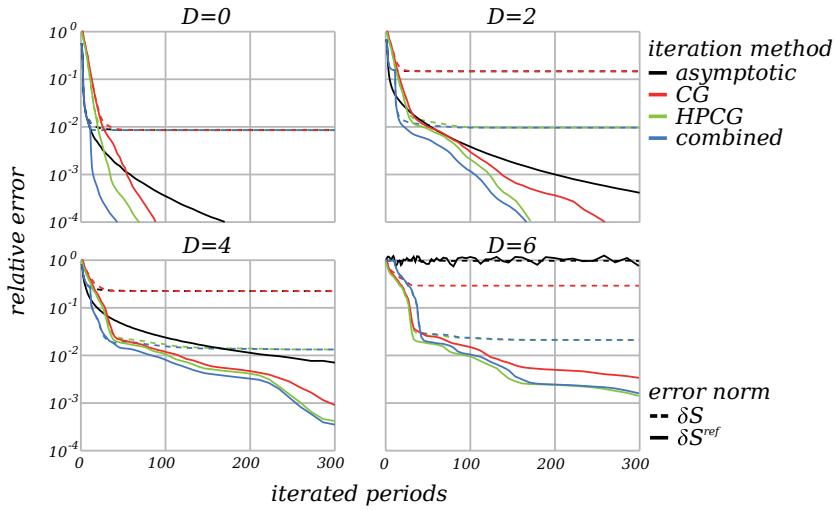


FIGURE 49 Convergence of the error norms δS and δS^{ref}, when the amount of absorption is varied. The term D denotes the number of domain faces, which are modeled by perfectly reflecting Dirichlet boundary conditions. The other faces are modeled by the Silver-Müller boundary condition.

since the relative residual norm is less than 10^{-6} in all cases after 1000 iterated periods. In this experiment, the half-periodic constraints are important to narrow down the solution space. Thus, if a time-harmonic solution is required, we recommend using the HPCG method instead of the CG method.

9.5.3 Electromagnetic wave in a wave guide

In this section, we consider a wave trapping problem in a more complicated geometry. The domain of this problem is a narrow cylindrical wave guide of length 12 and diameter 2. The cylinder axis is oriented in the x-direction, and it is centered at the origin. In the middle of the domain, there is a spherical bump of diameter 6. The material parameters $\epsilon = 1$ and $\mu = 1$ are applied in the whole domain. The curved boundaries of the domain are modeled as the perfectly reflecting Neumann type boundary condition. The cylinder ends are modeled by the absorbing Silver-Müller boundary condition. An incident wave is initialized at the cylinder end in the negative x-direction. The incident wave is a circularly polarized plane wave of wavelength $\lambda = 1$ and period $T = 1$, and it obeys the formulation of Equation (40).

Several types of cells are used in the domain discretization. The boundary of the bump is discretized by structured triangular elements, as illustrated in the left-hand side of Figure 50. The interior of the bump is discretized by the body centered cubic grid (BCC), and the Voronoi diagram is applied near the boundary to connect the grid with the boundary. The cylinder outside the bump is discretized using prism elements. Figure 50 is a scaled illustration of the mesh construction and much finer mesh is used in the simulations. The simulations are

run on a mesh, where the element lengths are scaled to be approximately $\frac{1}{20}$. Since the exact solution is not available for the current problem, we compute the reference solution using a much finer discretization. In the reference discretization, the average element lengths are approximately $\frac{1}{40}$. The number of primal body elements are 1 761 560 and 13 747 476 for $\frac{1}{20}$ and $\frac{1}{40}$ discretizations, respectively.

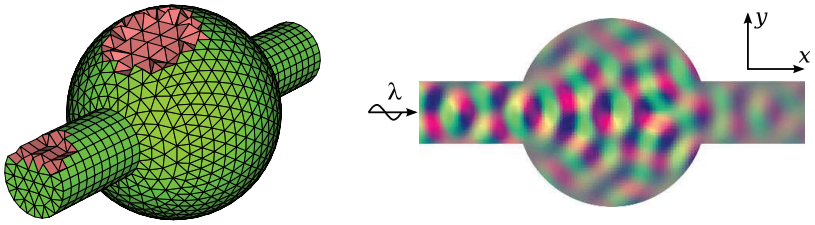


FIGURE 50 The mesh structure is illustrated on the left with approximately 6 times larger elements than in our simulation. The mesh is cut to show how volume consists of tetrahedra and prisms. The simulation result is presented on the right-hand side image. Electric field components x, y, and z are illustrated by the colors red, green and blue, respectively, on $z = 0$ plane.

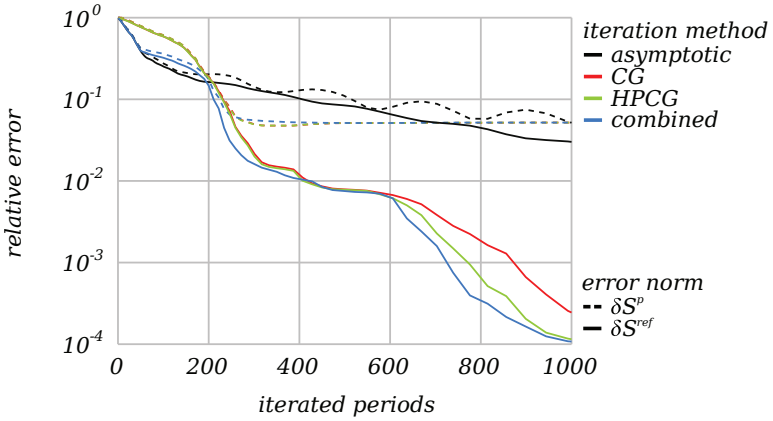


FIGURE 51 The convergence of the error norms δS^p and δS^{ref} during the iteration process.

The convergence is experienced by considering relative errors at each iteration stage. Mainly, we consider error norm δS^{ref}, which represents the solution error with respect to a converged solution. The converged solution is obtained separately for each iteration method by iterating the particular method over 2000 periods. Since the exact result is not available, we consider the error δS^p instead of the exact error. The error δS^p is computed with respect to a reference solution, which is obtained by the reference simulation with the finer spatial discretization. The iteration for the reference solution is preformed using 500 consecutive iterations of the asymptotic method, HPCG method and CG method. The reference solution is transformed to the electric and magnetic fields $\mathbf{E}^{ref}(\mathbf{p}_i)$ and $\mathbf{H}^{ref}(\mathbf{p}_i)$

at 4173 different positions $\mathbf{p}_i$, which are uniformly distributed inside the domain. The transformation is performed by the method described in Section 6.2. During the iteration, the similar fields $\mathbf{E}(\mathbf{p}_i)$ and $\mathbf{H}(\mathbf{p}_i)$ are computed in several iteration stages. The formulation for the error norm δS^p is expressed in Equation (60).

The convergence of the error norms is illustrated in Figure 51. The asymptotic iteration scheme converge slowly compared to the controllability methods. The controllability methods achieve the desired accuracy in about 250–300 periods of iteration. After that, the error δS^p does not improve significantly because the accuracy level of current discretization is found. The asymptotic iteration method needs 800 or more iterations to achieve the same accuracy level. The results show that the controllability methods are important, especially in trapped wave problems. In these simulations, the CG and the HPCG methods converge at the same rate. The combined method, which includes 50 periods of asymptotic iteration followed by the HPCG process, converges slightly better at some stages, but the improvement is not remarkable.

9.5.4 A sphere scattering problem

In scattering problems, the amount of absorption is usually significant, since the unbounded physical domain is truncated to bounded region by absorbing boundary condition or absorbing layer. Thus, the asymptotic iteration method might produce a solution relatively efficiently compared to the controllability techniques. In this section, we consider the sphere scattering problem, which is already described in Section 8.5.2. For the spatial discretization, we use the same meshes, which were applied in that section for the BCC simulations. Thus, the scatterer object is constructed by the BCC grid, and the rest of the mesh construction is illustrated in the left-hand side of Figure 39. We consider the four finest discretization levels, which correspond to 10, 14, 20 and 28 elements per wavelength, as explained in Section 8.5.2. The harmonic Hodge operator and the harmonic non-uniform leapfrog method are applied in the simulations.

We consider the convergence of the Mueller matrix error δM during the iteration. At several iteration stages, the simulated Mueller matrix $M(\theta)$ is compared to the Mueller matrix $M^{mie}(\theta)$ of the analytical Mie scattering solution. The error norm δM is then obtained by the formulation of Equation (59). Similarly to the previous sections, we also compare the iteration stage to the converged solution. The Mueller matrix $M^{ref}(\theta)$ of the converged solution is computed separately for each discretization level. The converged solution is obtained as a result of the combined iteration method, which consists of 300 asymptotic iterations, 300 HPCG iterations and 150 CG iterations, respectively. The convergence error is denoted by δM^{ref}, and it is computed by

$$\delta M^{ref} := \frac{\int_0^\pi \sin\theta \left\| M(\theta) - M^{ref}(\theta) \right\| d\theta}{\int_0^\pi \sin\theta \left\| M^{ref}(\theta) \right\| d\theta}.$$

The convergence results are illustrated in Figure 52. The convergence of δM^{ref} is almost identical with all four discretization levels. The asymptotic iter-

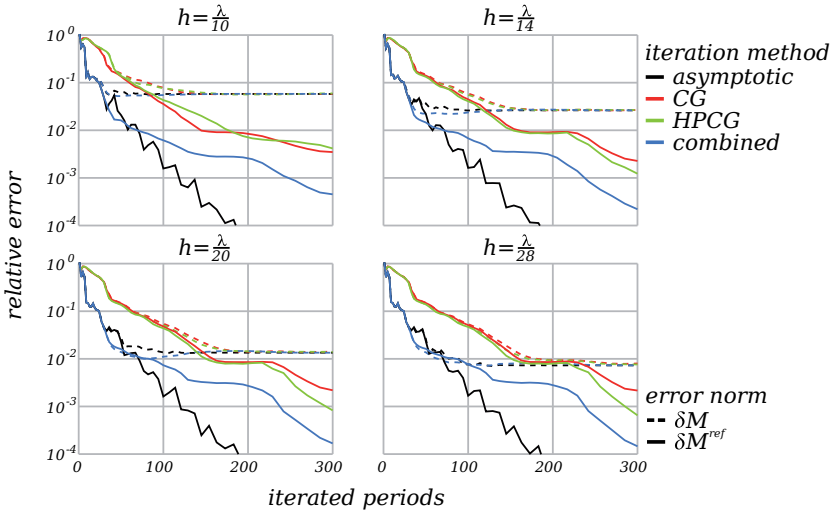


FIGURE 52 Convergence of the Mueller matrix error in a sphere scattering simulations, where the PML absorbing layer is applied.

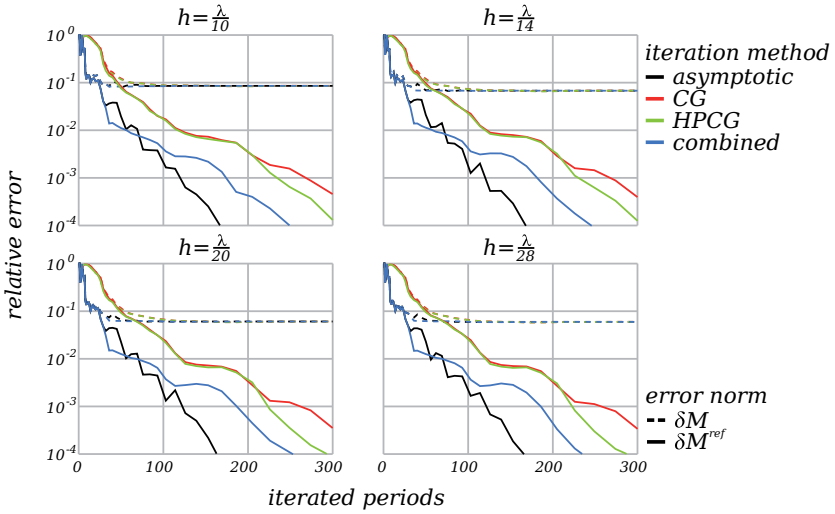


FIGURE 53 Convergence of the Mueller matrix error in a sphere scattering simulation using the Silver-Müller type absorbing boundary condition.

ation method seems to offer the fastest convergence, though the convergence is far from monotonic. The controllability methods do not converge that well. The combined method still offers a temporary improvement for the convergence. The combined method begins with 30 periods of the asymptotic iteration method, and the iteration is continued with the HPCG method.

To show that the slower convergence of the controllability methods is not due to the PML layer, we performed the same simulations without the absorbing layer. Figure 53 illustrated the convergence for the problem, where the Silver-Müller absorbing boundary condition is applied at the external boundary. The exact error δM does not improve significantly by refining the mesh. This is due to the inferior approximation of unbounded domain offered by the Silver-Müller boundary condition. The reference error δM^{ref} converges slightly faster than we observed with the experiments, including the PML.

9.5.5 Conclusions

The iteration methods for time-harmonic problems are compared by numerical experiments. The asymptotic iteration method is a natural consequence from the forward-in-time iteration scheme. The numerical experiments show that the controllability methods can speed up the convergence compared to the asymptotic method. The controllability methods improve the convergence speed especially in trapped wave problems. In scattering problems, where the wave is absorbed almost immediately after its generation, the asymptotic iteration can outperform the controllability methods. The combined method, where the iteration begins with the asymptotic approach and continues with a control method, seems to be the most versatile alternative to obtain fast convergence.

10 NUMERICAL COMPARISON TO DDA

In this chapter, we consider the produced DEC framework as a numerical method for solving electromagnetic scattering problems. More specifically, we compare the DEC method to another approach, the discrete-dipole approximation (DDA), which is a well-known technique in the context of electromagnetic scattering (Yurkin and Hoekstra, 2007). The first comparison between the DEC and the DDA is published in Räbinä et al. (2014), where the efficiency of the DEC simulation method was compared to the DDA method in regards to electromagnetic scattering problems. The efficiency of a numerical simulation method corresponds to the level of accuracy in a given amount of simulation time. In the first comparison, the accuracy was qualified by the error of the scattering phase function (the first component of the Mueller matrix). Here, we repeat the numerical experiments of the article using a full Mueller matrix in the comparison.

10.1 Discrete-dipole approximation

The discrete-dipole approximation (DDA) is a well-established technique in the context of solving scattering problems numerically. Its details and formulation are already covered in many publications. We recommend that the interested reader should see Yurkin and Hoekstra (2007) for a general description of the method. The method has several excellent open-source implementations, which have been reviewed and tested for example by Penttilä et al. (2007).

In a DDA approach, the domain is divided into sub-volumes, which are called the dipoles. The linear system of DDA equations is usually quite large, since it has $3n$ unknowns, where n is the number of dipoles. Direct solution becomes impossible, and iterative methods are used. Although it is not required that the dipoles should lie in a rectangular grid, in practice, this is often desired. With its rectangular grid, the discrete fast Fourier transform (FFT) can be used to speed up the matrix-vector multiplications in the iterative solver. With FFT, the number of computational operations needed for solving the scattering problem

with the DDA is of order $\mathcal{O}(n \log n)$, where n is the number of dipoles used for the space discretization. The possible errors in the DDA method can be eliminated by decreasing the dipole size. It seems that the theoretical results in this field are difficult, as discussed by Yurkin et al. (2006b). Nevertheless, there are many numerical studies that show good convergence in practice with decreasing dipole size (Draine and Flatau, 1994; Yurkin et al., 2006a; Yurkin and Kahnert, 2013).

The DDA results of this chapter were produced by Antti Penttilä from the Department of Physics in the University of Helsinki. The results are obtained by the same DDA simulations, which were produced for the first comparison and published by Räbinä et al. (2014). The DDA simulations were performed by the ADDA code, version 1.2, which is described in detail by Yurkin and Hoekstra (2011). Several iterative methods for solving the DDA equations are implemented in the ADDA code, and the bi-conjugate gradient stabilized method and the quasi minimal residual method are employed in these simulations.

10.2 Comparison by numerical experiments

The efficiencies of DEC and DDA are compared by performing electromagnetic scattering simulations. By the efficiency of a method we refer to the ratio between accuracy and the computational work required to achieve that level of accuracy. The accuracy of the discrete solutions depends on the spatial discretization, the approximation of the geometry and the stopping criterion of the iterative solving procedure. In the DEC simulations, the radiation condition, which is constructed by PML, in another source of error. The time discretization is performed using the harmonic non-uniform leapfrog method, presented in Section 7.3. The harmonic Hodge operator of Chapter 8 is used for the spatial discretization.

To compare the methods, we compute scattering Mueller matrices for three different target objects. The incident wave for computation is a fully polarized harmonic plane wave with a wavelength $\lambda = 2\pi$ and a time period $T = 2\pi$. The plane wave propagates in the direction of the positive x-axis, and the $\mathbf{E}$ field of the incident wave is orthogonal to x- and z-axis. The target objects have material parameters $\hat{\epsilon}_{target} = 2.5575 + 0.16i$ and $\hat{\mu}_{target} = 1.0$ implying to the refractive index $\hat{r} = 1.6 + 0.05i$. The material parameters outside the scatterer are $\hat{\epsilon}_0 = 1$ and $\hat{\mu}_0 = 1$. Analytical solutions for the scattering problems of this type exist only for some simple geometries, like a sphere. That is why we have chosen a sphere of diameter 5λ as the first obstacle in the numerical experiments. In this case, the accuracy of the methods is validated by comparing the solution with the known analytical solution, which is available by the Mie-theory (Bowman et al., 1969). The exact Mueller matrices are produced by the Mie-scattering code of (Mätzler, 2002). The other two obstacles, which we use, are a cube and a torus. Both the edge length of the cube and the outer diameter of the torus are set to be 5λ. The inner diameter of the torus is set to be 3λ, which means that the thickness of torus ring is λ. The incident wave propagates orthogonal to one of the cubic faces and

through the torus hole. All the obstacles and the DEC meshes are illustrated in Figure 54.

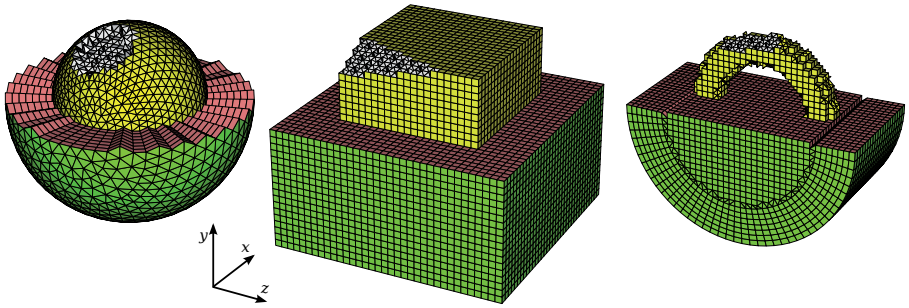


FIGURE 54 The DEC discretizations by polygons of different shapes (tetrahedron, prism, cube, octahedron). For clarity, these figures are made with 5 elements per wavelength, but 10 to 28 elements per wavelength are used in the simulations. An absorbing layer of thickness 1.3λ is surrounding the scattering objects.

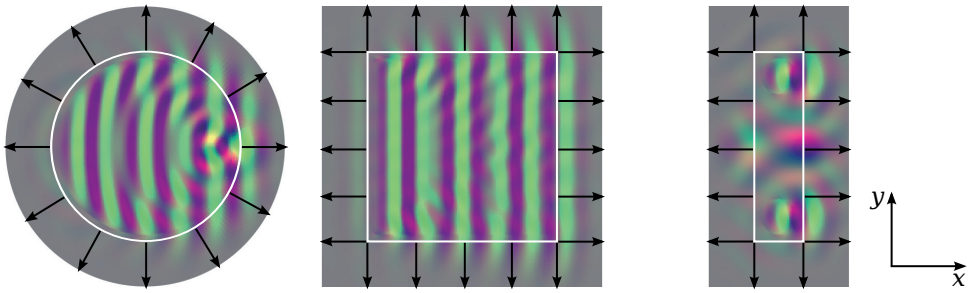


FIGURE 55 The xy-plane cross-section of the DEC scattered fields are presented by background colors, where red, green, and blue components represent the x-, y- and z-components of the electric field, respectively. The white line represents the physical domain boundary, where the PML begins. The black arrows illustrate the absorption directions of the PML.

The DEC meshes are constructed by combining structured grids. With spherical and toroidal target objects, we obtain unstructured elements between the structured regions. The unstructured elements are then optimized by the Hodge optimization method, as described in Section 2.7. The structured grid refining protocol is applied with the cube and torus meshes to refine the mesh at the target object region. At the target object region, the BCC grid is obtained by inserting nodes at each body center location of the regular base grid, as described in Section 2.9. The torus mesh is not in accordance with the target object surface and the material parameters are then converted to discrete Hodge matrices by numerical integration over volume elements, as discussed in Section 3.3. The DEC meshes are generated for four different discretization levels, which correspond approximately to 10, 14, 20 and 28 elements per wavelength. The element shapes

vary within the meshes, which means these numbers are given for illustrative purpose only. By the same ideology, the resolution in Figure 54 is 5 elements per wavelength. The element sizes are adjusted to fit the local wavelength, which is $\frac{\lambda}{|\hat{r}|}$ inside the target object and λ outside the target object. We recall, that λ is the wavelength of the incident wave.

The 1.5λ thick structured grid is applied in each mesh outside the target object. The absorbing PML layer of thickness 1.3λ is surrounding the obstacles, starting from the 0.2λ distance outside the scatterer. The PML, described in Section 4.4, requires a specific mesh design, where primal and dual edges are either parallel or orthogonal to the absorption direction. The absorption directions of each geometry are illustrated by the black arrows in Figure 55. The cube and torus geometries include corner regions, where several absorption directions are combined. The absorbing term $\beta = 0.7$ is applied for the PML.

The elements of the DDA system are dipoles placed on a regular grid. The computational domain is the minimal rectangular parallelepiped that encompasses the scatterer. No absorbing layers are needed for the DDA. The resolutions of the spatial discretization used with the DDA are 10, 20, and 30 dipoles per wavelength. The cube and the sphere are discretized by the systems of $80 \times 80 \times 80$, $160 \times 160 \times 160$ and $240 \times 240 \times 240$ dipoles, whereas the torus is discretized by the systems of $16 \times 80 \times 80$, $32 \times 160 \times 160$, and $48 \times 240 \times 240$ dipoles, respectively. For the non-spherical target objects, the reference solution is computed by the DDA method with 80 dipoles per wavelength. Thus, the numbers of dipoles are $640 \times 640 \times 640$ for the cube and $128 \times 640 \times 640$ for the torus. The fine discretization gives a highly accurate solution. The Muller matrix error for cube is known to be less than one percent at every measurement angle.

The method accuracy is considered as the relative error of the Mueller matrix as already formulated in Equation (59). The far field scattering data are obtained from the DEC result by applying a powerful and flexible near field to far field transformation, which is discussed in Sections 6.3 and 6.4. The near field data was collected at distance of 0.2λ outside the scatterer, as illustrated by white lines in Figure 55. The transformation from discrete field to Mueller matrix is another source of error in the DEC approach.

The DEC simulations were iterated using the combined iteration method of Section 9.4. The simulations were started with 20 asymptotic iterations and followed by HPCG iterations until the residual norm dropped down to 0.005. From 7 to 21 HPCG iretations were needed in the procedure. The stopping criterion for the iterative solver in ADDA, concerning the relative residual norm, was set to 10^{-8}. In DDA simulations, the Bi-CG stabilized method was used with the cubic target object, but with the sphere and torus target objects the iteration was successful using the default method of quasi minimal residual. All the computations have been carried out on an Intel Xeon E5-2670 processor at 2.6 GHz.

To illustrate the efficiency of numerical simulation, the relative error of the Mueller matrix is presented as a function of simulation time in Figure 56. With both simulation methods, it is possible to improve the accuracy at the expense of computational time by using finer spatial discretization. It can be seen that the

computational costs are of the same order of magnitude for the both methods. The DDA method seems to be slightly more efficient with cubical and spherical target objects. With the toroidal target object, the DEC outperforms the DDA. The results are quite similar to the article of Räbinä et al. (2014). Both methods have their own strengths and weaknesses, depending on the geometry of the scatterer and the details of the discretization.

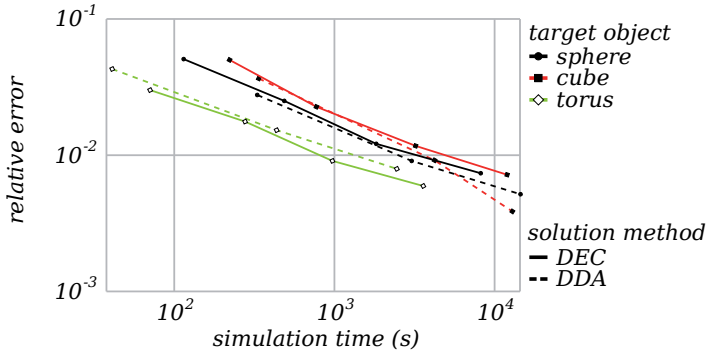


FIGURE 56 The Mueller matrix errors are plotted in function of simulation time for both DEC and DDA methods.

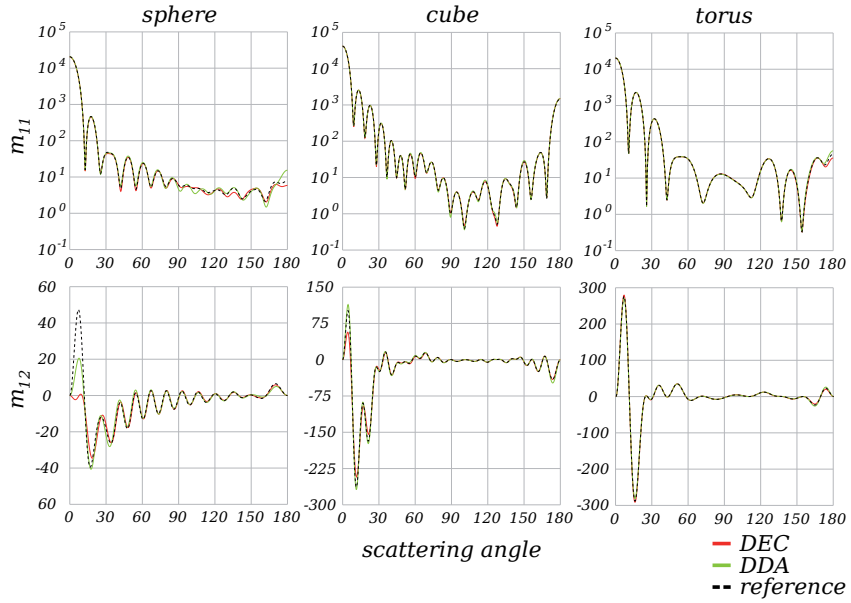


FIGURE 57 The first two components of the Mueller matrix are presented for each target object.

The first two components of the Mueller matrices are illustrated in Figure 57 for each target object. In this illustration, the best corresponding discretization levels are selected for DEC and DDA curves. The densest, the second densest and

the third densest discretization levels are used in torus, cube and sphere graphs, respectively.

10.3 Conclusions

We compared the DEC to the discrete-dipole approximation (DDA) for solving electromagnetic scattering problems. Three different target obstacles were considered, and the computational efficiencies were of the same order of magnitude with both methods. The results were similar to the article of Räbinä et al. (2014). Both methods have their own strengths and weaknesses, depending on the geometry of the scatterer and the details of the discretization. Certainly, the DEC is a more flexible method than the DDA, but the efficiency of the DEC method depends on the quality of the spatial discretization. Thus, an efficient DEC implementation requires a good mesh generator. Generating a good mesh for DEC is often much more difficult problem than solving the Maxwell equations. Since there exist no optimized and easy-to-use DEC software, which includes the mesh generation, the DEC method is not yet competitive with DDA (or with FDTD) in simple scattering problems. An interesting direction to continue these studies would be to produce such an interface, which could be operated by physicists who have deeper interest and understanding on scattering phenomena. Thus, an automated grid generator might be a good approach for this task, as it is with DDA and with FDTD. The crystal structures and the structured mesh refining might be an answer for this demand.

11 SUMMARY

This study considers a numerical solution method based on discrete exterior calculus (DEC). The thesis concentrates on electromagnetic waves, meaning that the mathematical model is given by the Maxwell equations. The DEC offers a natural-driven time-dependent simulation scheme, where the wave propagation can be tracked forward-in-time. The implementation is formulated for three-dimensional problems. The time axis brings in the fourth dimension.

The space and time discretizations are separated in the derivation. The spatial discretization is described by the mesh, which divides the simulation domain into small elements. The elements are either nodes, edges, faces or bodies depending on the dimension of the element. We also introduce a dual mesh, which has dual elements corresponding to each element of the primal mesh. In three dimensions, the dual of a node is a body and the dual of an edge is a face, and vice versa. The vector fields of the Maxwell equations are described by discrete differential forms, which are defined as integrals over mesh elements. The discrete Faraday and Ampére laws of the Maxwell equations are discretized exactly on primal and dual meshes, respectively. The constitutive relations between flux densities and fields are expressed by the discrete Hodge operators. Using the orthogonal relation of primal and dual mesh elements, the discrete Hodge operators are expressed as diagonal matrices, and the outcoming discretization scheme has exact energy conservation properties. The discrete Hodge operator is the only source of error in spatial discretization.

The time discretization is described in the leapfrog manner, where electric and magnetic field values are updated by turns. As a result, a forward-in-time iteration scheme is obtained. A new non-uniform leapfrog scheme with variable time step size is introduced, and an estimate for stability criterion is formulated. Numerical experiments on a random mesh show good energy conservation properties for the non-uniform leapfrog method. The improvement of the method efficiency is significant.

There are many practical problems that can be described as time-harmonic problems. Based on the harmonic approach, the harmonic formulations for space and time discretization are presented. The time-harmonic leapfrog equations pro-

duce an exact time discretization for time-harmonic problems. The harmonic Hodge operator is derived from space harmonic assumption by minimizing the error based on the least squares method. The harmonic corrections have no effect on energy conservation. The numerical experiments show that the correction can significantly improve the simulation accuracy.

Alternative iteration methods are introduced to solve time-periodic problems. The alternative methods are based on a controllability approach, where the formulated cost function is minimized by conveniently preconditioned conjugate gradient method. The controllability method is compared to asymptotic forward-in-time iteration method by comparing the development of solution error during the iterations. The controllability method increases the speed of convergence, especially in tasks, where the wave can be trapped inside the domain. The asymptotic iteration can be faster in problems with a lot of absorption. A combined method, where the iteration begins with the asymptotic approach and continues with a control method seems to give the fastest convergence in almost every test cases.

The mesh quality is one of the most significant factors to affect the efficiency of the DEC solver. The relations between primal and dual elements have a tremendous effect on the mesh quality. Generating a high-quality mesh can sometimes be an even more laborious task than solving wave equations. The meshes of this thesis are constructed using crystal structures, and even the most complicated meshes are mostly made up of structured elements. By comparing the crystal structures in DEC simulations, the isotropy is found to be one of the most important properties for simulation efficiency. The building of the meshes by Voronoi diagrams enables the combination of crystal structures, which could be an efficient way for controlled mesh refining inside the simulation domain. The comparison to discrete dipole approximation shows that a simplified and optimized DEC implementation could be a very competitive method for solving scattering problems.

YHTEENVETO (FINNISH SUMMARY)

Väitöskirjan suomenkielinen nimi on *Maxwellin yhtälöiden numeerinen ratkaiseminen diskreettiä differentiaalilaskentaa hyödyntäen*. Väitöskirja käsittelee numeerista ratkaisumenetelmää, joka perustuu diskreettiin differentiaalilaskentaan, ja josta käytetään lyhennettä DEC (englanniksi discrete exterior calculus). Numeerinen ratkaisumenetelmä johdetaan ja muotoillaan sähkömagneettista aaltoliikettä mallintaville Maxwellin yhtälöille. DEC tarjoaa hyvin luonnollisen lähestymistavan ajasta riippuvien tehtävien ratkaisuun, jossa sähkömagneettisen aallon etenemistä voidaan seurata ajassa edeten. Ratkaisija on toteutettu kolmiulotteiseen paikka-avaruuteen. Neljännen ulottuvuuden tuo mukanaan aika-avaruus.

Paikka- ja aikadiskretointi määritellään erillisinä vaiheina DEC-toteutuksessa. Paikkadiskretoinnin määrittämiseen tarvitaan kolmiulotteinen verkko, joka jakaa simulointialueen pieniin osasiin. Osaset ovat joko pisteitä, janoja, monikulmioita tai näiden rajaamia tilavuuselementtejä. Verkolle määritellään myös duaaliverkko, jonka osaset muodostuvat alkuperäisen verkon duaaliosasista. Kolmiulotteisessa tapauksessa pisteen duaaliosanen on tilavuuselementti ja janan duaaliosanen on monikulmio, ja toisinpäin. Sähkö- ja magneettikentät esitetään diskreettien differentiaalimuotojen avulla siten, että diskreetin muodon alkiot määritellään integraaleina osasten yli. Näin Maxwellin yhtälöt pystytään diskretoimaan tarkasti alkuperäisen verkon ja duaaliverkon avulla. Verkkojen välinen riippuvuus määritellään matriisilla, jota kutsutaan diskreetiksi Hodge-operaattoriksi. Kun duaaliverkon elementit noudattavat tiettyä kohtisuoruussääntöä, voidaan diskreetti Hodge-operaattori esittää diagonaalisena matriisina. Tällöin systeemin energia säilyy täydellisesti ajan edetessä, mikä on tärkeä ominaisuus menetelmän toimivuuden kannalta. Diskreetti Hodge-operaattori on paikkadiskretoinnin ainoa virhelähde.

Aikadiskretointi toteutetaan yleisesti käytössä olevalla pukkihyppelytekniikalla, missä sähkö- ja magneettiarvot päivitetään vuorotellen toisiaan hyödyntäen. Tuloksena syntyy ajassa etenevä iterointimenetelmä. Väitöskirjassa esitetään uusi epävakioitu pukkihyppelymenetelmä, missä aika-askeleen kokoa voidaan vaihdella simulointialueen eri osissa. Samassa yhteydessä esitetään arvioitu yläraja aika-askeleen koolle. Numeeriset tulokset osoittavat, että menetelmän energian säilyvyys on erinomainen. Satunnaisverkossa simuloituna epävakioitu pukkihyppelymenetelmä parantaa menetelmän tehokkuutta merkittävästi.

Sähkömagneettiseen aaltoliikkeeseen liittyvät käytännön ongelmat voidaan usein esittää aikaharmonisina tehtävinä. Väitöskirjassa esitellään harmoniseen oletukseen perustuen muotoilut sekä paikka- että aikadiskretoinnille. Harmoninen pukkihyppelymenetelmä johtaa tarkkaan aikadiskretointiin, mikäli tehtävä on aikaharmoninen. Harmoninen Hodge-operaattori puolestaan johdetaan paikkaharmoniselle aaltoliikkeelle minimoimalla operaattorin aiheuttama virhe pienimmän neliösumman menetelmään perustuen. Harmoniset korjaukset eivät vaikuta energian säilymiseen, mutta numeeriset laskelmat osoittavat korjauksilla olevan merkittäviä vaikutuksia tulosten tarkkuuteen.

Väitöskirjassa on esitelty myös vaihtoehtoisia iterointimenetelmiä aikajaksollisten tehtävien ratkaisuun. Vaihtoehtoiset menetelmät ovat kontrollimenetelmiä, joissa aikajaksolliselle tehtävälle muotoiltu kustannusfunktio minimoidaan sopivasti pohjustetulla konjugaattigradienttimenetelmällä. Kontrollimenetelmiä verrataan suoraan ajassa etenevään simulointiin vertaamalla todellisen virheen kehitystä iteroinnin aikana. Kontrollimenetelmä nopeuttaa ratkaisun löytymistä erityisesti tehtävissä, joissa aalto jää kimpoilemaan simulointialueen sisään. Tehtävissä, joissa absorptiota on paljon, voi suora menetelmä olla kontrollimenetelmää nopeampi. Yhdistetty menetelmä, jossa iterointi aloitetaan suoralla menetelmällä ja viimeistellään kontrollimenetelmällä, on tulosten mukaan toimivin ratkaisu lähes kaikissa testitapauksissa.

Verkon laatu on yksi merkittävimmistä DEC-ratkaisijan tehokkuuteen vaikuttavista tekijöistä. Verkon laatuun vaikuttavat ensisijaisesti alkuperäisen verkon osasten ja duaaliosasten keskinäiset suhteet. Laadukkaan verkon luominen voi olla työläämpi tehtävä kuin itse aaltoyhtälöiden ratkaiseminen. Väitöskirjassa hyödynnetään verkkojen luomisessa kiderakenteita ja monimutkaisetkin verkot muodostetaan pääasiassa näitä rakenteita hyödyntäen. Vertailemalla kiderakenteita DEC-simulaatioissa huomataan rakenteiden isotrooppisuudella olevan suuri vaikutus simulaation tarkkuuteen. Verkkojen luominen Voronoin kaaviota käyttämällä mahdollistaa kristallirakenteiden hallitun yhdistelemisen, mikä voi olla erinomainen keino rakenteen hallittuun tihentämiseen simulointialueen sisällä. Vertailu yleisemmin tunnettuun sironnan simulointimenetelmään (DDA) osoittaa, että yksinkertainen ja optimoitu DEC-toteutus voisi olla erittäin kilpailukykyinen menetelmä sirontatehtävien ratkaisuun.

REFERENCES

Abraham, R., Marsden, J. E. & Ratiu, T. 1988. Manifolds, tensor analysis, and applications. Springer-Verlag, New York: Applied Mathematical Sciences.

Aurenhammer, F. & Klein, R. 2000. Voronoi diagrams. In J.-R. Sack & J. Urrutia (Eds.) Handbook of computational geometry. Amsterdam, Netherlands: North-Holland, 201–290.

Axelsson, O. 1974. On preconditioning and convergence acceleration in sparse matrix problems. Geneva: CERN European organization for nuclear research. CERN 74–10.

Bardos, C. & Rauch, J. 1994. Variational algorithms for the Helmholtz equation using time evolution and artificial boundaries. Asymptotic Analysis 9 (2), 101–117.

Barucq, H., Delaurens, F. & Hanouzet, B. 1998. Method of absorbing boundary conditions: Phenomena of error stabilization. SIAM Journal on Numerical Analysis 35 (3), 1113–1129.

Bell, N. & Hirani, A. N. 2012. PyDEC: Software and algorithms for discretization of exterior calculus. (Preprint, arXiv:1103.3076v2 [cs.NA]).

Berenger, J.-P. 1994. A perfectly matched layer for the absorption of electromagnetic waves. Journal of Computational Physics 114 (2), 185–200.

de Berg, M., van Kreveld, M., Overmars, M. & Schwarzkopf, O. 2000. Computational geometry (2nd edition). Springer-Verlag.

Bladel, J. V. 1964. Electromagnetic fields. New York: McGraw-Hill.

Bohren, C. F. & Huffman, D. R. 1983. Absorption and scattering of light by small particles. New York: Wiley & Sons.

Bossavit, A. & Kettunen, L. 1999. Yee-like schemes on a tetrahedral mesh, with diagonal lumping. International Journal of Numerical Modelling 12 (1–2), 129–142.

Bossavit, A. & Kettunen, L. 2000. Yee-like schemes on staggered cellular grids: A synthesis between FIT and FEM approaches. IEEE Transactions on Magnetics 36 (4), 861–867.

Bossavit, A. 1998. Computational electromagnetism: Variational formulations, complementarity, edge elements. San Diego: Academic Press.

Bossavit, A. 2000. The discrete Hodge operator in electromagnetic wave propagation problems. In Fifth international conference on mathematical and numerical aspects of wave propagation, 753–759.

Bossavit, A. 2001. 'Generalized finite differences' in computational electromagnetics. Progress in Electromagnetics Research 32, 45–64.

Bossavit, A. 2003. The Sommerville mesh in Yee-like schemes. In Scientific computing in electrical engineering: SCEE-2002, Vol. 4. Berlin: Springer. Mathematics in Industry, 128–136.

Bossavit, A. 2005. Discretization of electromagnetic problems: The "generalized finite differences" approach. Handbook of Numerical Analysis 13, 105–197.

Bowman, J. J., Senior, T. B. A. & Uslenghi, P. L. E. 1969. Electromagnetic and acoustic scattering by simple shapes. Amsterdam: North-Holland publishing company.

Bristeau, M. O., Glowinski, R., Périaux, J. & Rossi, T. 1999. 3D harmonic Maxwell solutions on vector and parallel computers using controllability and finite element methods. INRIA-0073072. (Research Report RR-3607).

Bristeau, M. O., Glowinski, R. & Périaux, J. 1998. Controllability methods for the computation of time-periodic solutions; Application to scattering. Journal of Computational Physics 147 (2), 265–292.

Cartan, H. 2006. Differential forms. Dover Publications. Dover Books on Mathematics Series.

Cessenat, M. 1996. Mathematical methods in electromagnetism: Linear theory and applications. Singapore: World Scientific Publishing Co. Pte. Ltd.

Chen, B., Fang, D. G. & Zhou, B. H. 1995. Modified Berenger PML absorbing boundary condition for FD-TD meshes. IEEE microwave and guided wave letters 5 (11), 399–401.

Chew, W. C. & Weedon, W. H. 1994. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. Microwave and Optical Technology Letters 7 (13), 599–604.

Chung, E. T. & Engquist, B. 2005. Convergence analysis of fully discrete finite volume methods for Maxwell's equations in nonhomogeneous media. SIAM Journal on Numerical Analysis 43 (1), 303–317.

Cinalli, M. & Schiavoni, A. 2006. A stable and consistent generalization of the FDTD technique to nonorthogonal unstructured grids. IEEE Transactions on antennas and propagation 54 (5), 1503–1512.

Clemens, M. & Weiland, T. 2001. Discrete electromagnetism with the finite integration technique. Progress In Electromagnetics Research 32, 65–87.

Conway, J. & Sloane, N. J. A. 1999. Sphere packings, lattices and groups (3rd edition). Springer. A series of comprehensive studies in mathematics.

Delaunay, B. N. 1934. Sur la sphére vide. Bulletin of Academy of Sciences of the USSR 7, 793–800.

Desbrun, M., Kanso, E. & Tong, Y. 2008. Discrete differential forms for computational modeling. Discrete Differential Geometry Oberwolfach Seminars 38, 287–324.

Desbrun, M., Hirani, A. N., Leok, M. & Marsden, J. E. 2005. Discrete exterior calculus. (Preprint, arXiv:math/0508341v2 [math.DG]).

Draine, B. T. & Flatau, P. J. 1994. The discrete dipole approximation for scattering calculations. The Journal of the Optical Society of America A 11, 1491–1499.

Du, Q. & Wang, D. 2005. The optimal centroidal Voronoi tessellations and the Gersho's conjecture in the three-dimensional space. Computers & Mathematics with Applications 49 (9–10), 1355–1373.

Eisenstat, S. C. 1981. Efficient implementation of a class of preconditioned conjugate gradient methods. SIAM Journal on Scientific and Statistical Computing 2 (1), 1–4.

Eppstein, D., Sullivan, J. M. & Üngör, A. 2004. Tiling space and slabs with acute tetrahedra. Computational Geometry: Theory and Applications 27 (3), 237–255.

Flanders, H. 1989. Differential forms with applications to the physical sciences. New York: Dover Publications.

Frank, F. C. & Kasper, J. S. 1958. Complex alloy structures regarded as sphere packings. I. Definitions and basic principles. Acta Crystallographica 11 (3), 184–190.

Frank, F. C. & Kasper, J. S. 1959. Complex alloy structures regarded as sphere packings. II. Analysis and classification of representative structures. Acta Crystallographica 12 (7), 483–499.

Friauf, J. B. 1927. The crystal structures of two intermetallic compounds. Journal of the American Chemical Society 49 (12), 3107–3114.

Gedney, S. D. 1996. An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices. IEEE Transactions on Antennas and Propagation 44 (12), 1630–1639.

Gentle, J. E. 1998. Random number generation and Monte Carlo methods, statistics and computing. New York: Springer-Verlag.

Glowinski, R. & Lions, J. L. 1995. Exact and approximate controllability for distributed parameter systems (II). Acta Numerica 4 (4), 159–328.

Glowinski, R. & Rossi, T. 2006. A mixed formulation and exact controllability approach for the computation of the periodic solutions of the scalar wave equation. (I) controllability problem formulation and related iterative solution. Comptes Rendus de l'Académie des Sciences Paris 343 (7), 493–498.

Gross, P. W. & Kotiuga, P. R. 2004. Electromagnetic theory and computation: A topological approach. Mathematical Sciences Research Institute Publications 48.

Hanouzet, B. & Sesques, M. 1993. Absorbing boundary conditions for Maxwell's equations. Notes on Numerical Fluid Mechanics 43, 315–322.

Harrington, R. F. 1961. Time harmonic electromagnetic fields. New York: McGraw-Hill.

Hartmann, H., Ebert, F. & Bretschneider, O. 1931. Elektrolysen in phosphatschmelzen. I. die elektrolytische gewinnung von α- und β-Wolfram. Zeitschrift für anorganische und allgemeine Chemie 198 (1), 116–140.

Hatcher, A. 2002. Algebraic topology. Cambridge University Press.

Hestenes, M. R. & Stiefel, E. 1952. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49 (6), 409–436.

Hiptmair, R. 2001. Discrete Hodge-operators: An algebraic perspective. Progress In Electromagnetics Research 32, 247–269.

Hiptmair, R. 2002. Finite elements in computational electromagnetism. Acta Numerica 11, 237–339.

Hirani, A. N., Kalyanaraman, K. & VanderZee, E. B. 2012. Delaunay Hodge star. (Preprint, arXiv:0707.4470v3 [math.NA]).

Hirani, A. N. 2003. Discrete Exterior Calculus. California Institute of Technology. Dissertation (Ph.D.). (http://resolver.caltech.edu/CaltechETD:etd-05202003-095403).

Holland, R. & Williams, J. 1983. Total-field versus scattered-field finite-difference codes: A comparative assessment. IEEE Transactions on Nuclear Science 30 (6), 4583–4588.

Holland, R. 1977. Threde: A free-field EMP coupling and scattering code. IEEE Transactions on Nuclear Science 24 (6), 2416–2421.

Jackson, J. D. 1975. Classical electrodynamics. New York: Wiley.

Jin, J.-M. 2002. The finite element method in electromagnetics (2nd edition). New York: Wiley-IEEE Press.

Kepler, J. 1966. Strena, seu nive sexangula, 1611. Oxford, UK: Clarendon Press. (The Six-cornered Snowflake, English translation by Colin Hardie).

Keränen, J., Koljonen, E., Tarhasaari, T. & Kettunen, L. 2004. Effect of cell type on convergence of wave propagation schemes. IEEE transactions on magnetics 40 (2), 1452–1455.

van Kreveld, M., Löffler, M. & Silveira, R. I. 2010. Optimization for first order Delaunay triangulations. Computational Geometry 43 (4), 377–394.

Kusner, R. & Sullivan, J. M. 1996. Comparing the Weaire-Phelan equal-volume foam to Kelvin's foam. Forma 11 (3), 233–242.

Kähkönen, S., Glowinski, R., Rossi, T. & Mäkinen, R. 2011. Solution of time-periodic wave equation using mixed finite-elements and controllability techniques. Journal of Computational Acoustics 19 (4), 335–352.

Lautensack, C. & Sych, T. 2008. A random Weaire-Phelan foam. In Proceedings of the VIII. international conference on stereology and image analysis in materials science STERMAT.

Lawson, C. L. & Hanson, R. J. 1974. Solving least squares problems. Englewood Cliffs: Prentice-Hall.

Lee, J. M. 2009. Manifolds and differential geometry. American Mathematical Society. Graduate studies in mathematics.

Lew, A., Marsden, J. E., Ortiz, M. & West, M. 2003. Asynchronous variational integrators. Archive for Rational Mechanics and Analysis 167 (2), 85–146.

Ma, C. & Chen, Z. 2005. Stability and numerical dispersion analysis of CE-FDTD method. IEEE Transactions on antennas and propagation 53 (1), 332–338.

Madsen, N. K. 1995. Divergence preserving discrete surface integral methods for Maxwell's curl equations using non-orthogonal unstructured grids. Journal of Computational Physics 119 (1), 34–45.

Marklein, R. 2002. The finite integration technique as a general tool to compute acoustic, electromagnetic, elastodynamic, and coupled wave fields. In W. R. Stone (Ed.) Review of radio science: 1999-2002 URSI. New York: John Wiley & Sons, 201–244.

Mattiussi, C. 2000. The finite volume, finite element, and finite difference methods as numerical methods for physical field problems. Journal of Computational Physics 133, 289–309.

Maur, P. 2002. Delaunay triangulation in 3D. University of West Bohemia in Pilsen. (Technical Report No. DCSE/TR-2002-02).

McLean, I. S. 2008. Electronic imaging in astronomy: Detectors and instrumentation. Praxis Publishing Limited, Chichester, UK. Springer Praxis Books.

Monk, P. 2003. Finite element methods for Maxwell's equations. Oxford University Press.

Mullen, P., Memari, P., de Goes, F. & Desbrun, M. 2011. HOT: Hodge-optimized triangulations. ACM Transactions on Graphics 30 (4), 103.

Munkres, J. R. 1984. Elements of algebraic topology. Perseus Books. Advanced book classics.

Munkres, J. R. 1991. Analysis on manifolds. Addison-Wesley.

Mätzler, C. 2002. MATLAB functions for Mie scattering and absorption. Version 2. Institute of Applied Physics, University of Bern. (IAP Research Report, No. 2002-11).

Mönkölä, S., Heikkola, E., Pennanen, A. & Rossi, T. 2008. Time-harmonic elasticity with controllability and higher order discretization methods. Journal of Computational Physics 227 (11), 5513–5534.

Mönkölä, S. 2011. Numerical simulation of fluid-structure interaction between acoustic and elastic waves. University of Jyväskylä. Dissertation (Ph.D.). (Jyväskylä studies in computing, 133).

Nédélec, J.-C. 1980. Mixed finite elements in $\mathbb{R}^3$. Numerische Mathematik 35 (3), 315–341.

Nédélec, J.-C. 1986. A new family of mixed finite elements in $\mathbb{R}^3$. Numerische Mathematik 50 (1), 57–81.

Nédélec, J.-C. 2001. Acoustic and electromagnetic equations: Integral representations for harmonic problems. Springer.

O'Rourke, J. 1998. Computational geometry in C (2nd edition). Cambridge: Cambridge University Press.

Paufler, P. 2011. Early work on Laves phases in East Germany. Intermetallics 19 (4), 599–612.

Pauly, D. & Rossi, T. 2011. Theoretical considerations on the computation of generalized time-periodic waves. Advances in Mathematical Sciences and Applications 21 (1), 105–131.

Penttilä, A., Zubko, E., Lumme, K., Muinonen, K., Yurkin, M., Draine, B., Rahola, J., Hoekstra, A. & Shkuratov, Y. 2007. Comparison between discrete dipole implementations and exact techniques. Journal of Quantitative Spectroscopy and Radiative Transfer 106, 417–436.

Picard, R. & Freymond, H. 2012. On electromagnetic waves in complex linear media in nonsmooth domains. Mathematical Methods in the Applied Sciences 36 (8), 880–895.

Picard, R. 2009. A structural observation for linear material laws in classical mathematical physics. Mathematical Methods in the Applied Sciences 32 (14), 1768–1803.

Reem, D. 2009. An algorithm for computing Voronoi diagrams of general generators in general normed spaces. In Proceedings of the sixth international symposium on Voronoi diagrams in science and engineering, 144–152.

de Rham, G. 1984. Differentiable manifolds: Forms, currents, harmonic forms. Springer-Verlag. Grundlehren der mathematischen Wissenschaften.

Rubinstein, R. Y. 1981. Simulation and the Monte Carlo method. USA: John Wiley & Sons. Wiley series in probability and mathematical statistics.

Räbinä, J., Mönkölä, S., Rossi, T., Penttilä, A. & Muinonen, K. 2014. Comparison of discrete exterior calculus and discrete-dipole approximation for electromagnetic scattering. Journal of Quantitative Spectroscopy & Radiative Transfer 146, 417–423.

Saad, Y. 2003. Iterative methods for sparse linear systems (2nd edition). Philadelphia: Society for Industrial and Applied Mathematics.

Schelkunoff, S. A. 1951. Kirchhoof's formula, its vector analogue, and other field equivalence theorems. Communications on Pure and Applied Mathematics 4 (1), 43–59.

Schuhmann, R. & Weiland, T. 1998. Stability of the FDTD algorithm on nonorthogonal grids related to the spatial interpolation scheme. IEEE Transactions on magnetics 34 (5), 2751–2754.

Sen, S., Sen, S., Sexton, J. C. & Adams, D. H. 2000. A geometric discretisation scheme applied to the Abelian Chern-Simons theory. Physical Review E 61 (3), 3174–3185.

Sikiric, M. D., Delgado-Friedrichs, O. & Deza, M. 2010. Space fullerenes: A computer search for new Frank-Kasper structures. Acta Crystallographica Section A 66 (5), 602–615.

Sommerville, D. M. Y. 1923. Space-filling tetrahedra in Euclidean space. Proceedings of the Edinburgh Mathematical Society 41, 49–57.

Stern, A., Tong, Y., Desbrun, M. & Marsden, J. E. 2009. Geometric computational electrodynamics with variational integrators and discrete differential forms. (Preprint, arXiv:0707.4470v3 [math.NA]).

Sullivan, D. M. 1996. A simplified PML for use with the FDTD method. IEEE microwave and guided wave letters 6 (2), 97–99.

Sullivan, J. M. 2000. New tetrahedrally close-packed structures. In P. Zitha, J. Banhart & G. Verbist (Eds.) Foams, emulsions and applications; Eurofoam 2000. Bremen: Verlag MIT, 111–119.

Sullivan, J. M. 1998. The geometry of bubbles and foams. Foams and Emulsions (NATO ASI volume E 354, 379–402.

Taflove, A. & Brodwin, M. E. 1975. Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations. IEEE Transactions on Microwave Theory and Techniques 23 (8), 623–630.

Taflove, A. & Hagness, S. C. 2005. Computational electrodynamics: The finite-difference time-domain method (3rd edition). Artech House Inc.

Taflove, A. & Umashankar, K. 1983. Radar cross section of general three-dimensional scatterers. IEEE Transactions on Electromagnetic Compatibility 25 (4), 433–440.

Taflove, A. 1980. Application of the finite-difference time-domain method to sinusoidal steady state electromagnetic penetration problems. IEEE Transactions on Electromagnetic Compatibility 22 (3), 191–202.

Tarhasaari, T., Kettunen, L. & Bossavit, A. 1999. Some realizations of a discrete Hodge operator: A reinterpretation of finite element techniques. IEEE Transactions on Magnetics 35 (3), 1494–1497.

Teixeira, F. L. & Chew, W. C. 1997. PML-FDTD in cylindrical and spherical grids. IEEE Microwave and guided wave letters 7 (9), 285–287.

Teixeira, F. L. & Chew, W. C. 1998. General closed-form PML constitutive tensors to match arbitrary bianisotropic and dispersive linear media. IEEE Microwave and Guided Wave Letters 8 (6), 223–225.

Teixeira, F. L. & Chew, W. C. 1999. Lattice electromagnetic theory from a topological viewpoint. Journal of Mathematical Physics 40, 169–187.

Thomson, W. 1887. On the division of space with minimum partitional area. Acta Mathematica 11 (1–4), 121–134.

Tonti, E. 1999. On the geometrical structure of electromagnetism. In G. Ferrarese (Ed.) Gravitation, electromagnetism and geometrical structures. Bologna: Pitagora Editrice, 281–308.

Tonti, E. 2001. A direct discrete formulation of field laws: The cell method. Computer Modeling in Engineering & Sciences 2 (2), 237–258.

Tonti, E. 2002. Finite formulation of electromagnetic field. IEEE Transactions on Magnetics 38 (2), 333–336.

Tournois, J., Wormser, C., Alliez, P. & Desbrun, M. 2009. Interleaving Delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. ACM Transactions on Graphics 28 (3), 75:1–75:9.

Turk, G. & Levoy, M. 1994. Zippered polygon meshes from range images. In Proceedings of the 21st annual conference on computer graphics and interactive techniques. New York, NY, USA: ACM. SIGGRAPH '94, 311–318.

Umashankar, K. & Taflove, A. 1982. A novel method to analyse electromagnetic scattering of complex object. IEEE Transactions on Electromagnetic Compatibility 24 (4), 397–405.

VanderZee, E., Hirani, A. N. & Guoy, D. 2008. Triangulation of simple 3D shapes with well-centered tetrahedra. In Proceedings of the 17th international meshing roundtable, 19–35.

Volakis, J. L., Chatterjee, A. & Kempel, L. C. 1998. Finite element method for electromagnetics: Antennas, microwave circuits, and scattering applications. New York: Wiley-IEEE Press.

Voronoi, G. F. 1908. Nouvelles applications des paramétres continus á la théorie de formes quadratiques. Journal für die reine und angewandte Mathematik 133, 97–178.

Weaire, D. & Phelan, R. 1994. A counter-example to Kelvin's conjecture on minimal surfaces. Philosophical Magazine Letters 69 (2), 107–110.

Weiland, T. 1977. A discretization method for the solution of Maxwell's equations for six-component fields. Electronics and Communications AEUE 31 (3), 116–120.

Weiland, T. 1996. Time domain electromagnetic field computation with finite difference methods. International Journal of Numerical Modelling 9 (4), 295–319.

Whitney, H. 1957. Geometric integration theory. Princeton: Princeton Univercity Press.

Yee, K. S. & Chen, J. S. 1997. The finite-difference time-domain (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations. IEEE Transactions on Antennas and Propagation 45 (3), 354–363.

Yee, K. S. 1966. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Transactions on antennas and propagation 14 (3), 302–307.

Yurkin, M. A. & Hoekstra, A. G. 2007. The discrete dipole approximation: An overview and recent developments. Journal of Quantitative Spectroscopy and Radiative Transfer 106 (1–3), 558–589.

Yurkin, M. A. & Hoekstra, A. G. 2011. The discrete-dipole-approximation code ADDA: Capabilities and known limitations. Journal of Quantitative Spectroscopy and Radiative Transfer 112 (13), 2234–2247.

Yurkin, M. A. & Kahnert, M. 2013. Light scattering by a cube: Accuracy limits of the discrete dipole approximation and the T-matrix method. Journal of Quantitative Spectroscopy and Radiative Transfer 123 (0), 176–183.

Yurkin, M. A., Maltsev, V. P. & Hoekstra, A. G. 2006a. Convergence of the discrete dipole approximation. II. an extrapolation technique to increase the accuracy. The Journal of the Optical Society of America 23 (10), 2592–2601.

Yurkin, M. A., Maltsev, V. P. & Hoekstra, A. G. 2006b. Convergence of the discrete dipole approximation. I. theoretical analysis. The Journal of the Optical Society of America 23 (10), 2578–2591.

Zuazua, E. 2007. On the numerical approximation of the Helmholtz equations. Matemática Contemporânea 32, 253–286.