Discontinuous Galerkin Methods in Nanophotonics

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

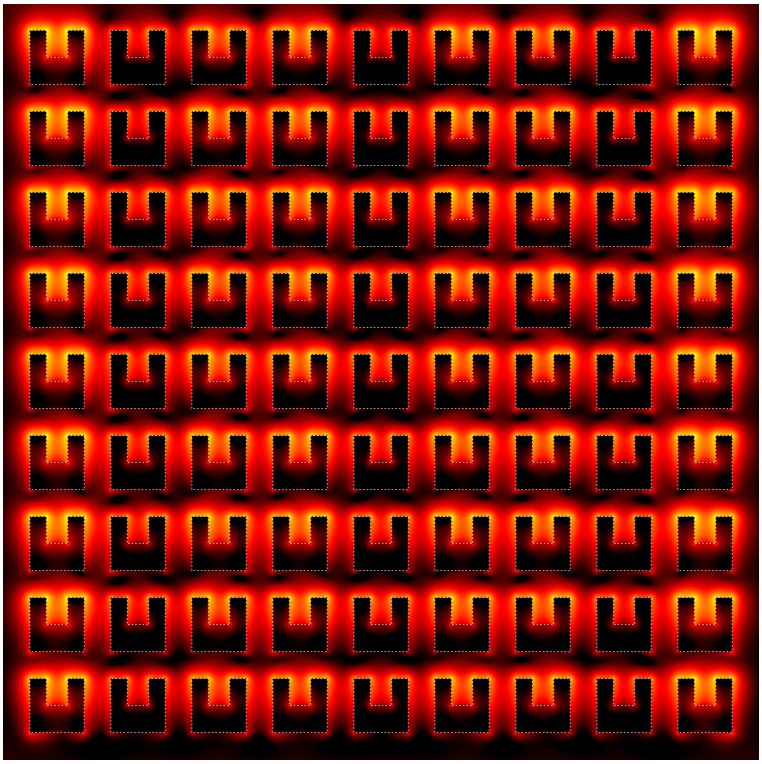
Dipl.-Phys. Michael Christian König
aus Landau in der Pfalz

Tag der mündlichen Prüfung: 1. Juli 2011
Referent: Prof. Dr. Kurt Busch
Korreferent: Prof. Dr. Martin Wegener

Karlsruhe Institute of Technology

Fakultät für Physik
Institut für Theoretische Festkörperphysik

Discontinuous Galerkin Methods in Nanophotonics



PhD Thesis
by
Dipl.-Phys. Michael Christian König
1st July 2011

Instructor:	Prof. Dr. Kurt Busch
2nd Instructor:	Prof. Dr. Martin Wegener

Contents

1 Introduction **1**
- 1.1 The Need for Numerical Simulations . . . 3
- 1.2 A Survey of Common Simulation Methods . . . 3
- 1.3 Overview of the Discontinuous Galerkin Method . . . 4
- 1.4 Outline of the Thesis . . . 5

2 Classical Electrodynamics **7**
- 2.1 Maxwell's Equations . . . 7
  - 2.1.1 Constitutive Relations . . . 8
  - 2.1.2 Material Interfaces . . . 9
  - 2.1.3 Reduction to the Curl Equations . . . 10
  - 2.1.4 Reduction to Dimensionless Units . . . 11
  - 2.1.5 Reduction to Two Spatial Dimensions . . . 12
- 2.2 Solutions to Selected Problems . . . 13
  - 2.2.1 Plane Wave Propagation . . . 13
  - 2.2.2 Normal Incidence on a Dielectric Half Space . . . 15
  - 2.2.3 Normal Incidence on a Thin Dielectric Film . . . 16
- 2.3 Material Dispersion in the Time-Domain . . . 17
  - 2.3.1 Auxiliary Differential Equations . . . 18
  - 2.3.2 The Drude Model . . . 19
  - 2.3.3 The Drude-Lorentz Model . . . 21
  - 2.3.4 Material Parameters . . . 22
- 2.4 Elements of Electromagnetic Scattering . . . 23

3 Spatial Discretisation via the Discontinuous Galerkin Approach **27**
- 3.1 An Illustrated Tour of the Discontinuous Galerkin Method . . . 27
  - 3.1.1 Maxwell's Equations in Conservation Form . . . 28
  - 3.1.2 Tesselation of the Computational Domain . . . 28
  - 3.1.3 Working on Single Elements . . . 29
  - 3.1.4 Connecting Elements via the Numerical Flux . . . 31
  - 3.1.5 Boundary Conditions . . . 33
  - 3.1.6 Expanding the Electromagnetic Fields . . . 34
  - 3.1.7 The Semi-Discrete Form . . . 37
- 3.2 Technicalities . . . 41
  - 3.2.1 Derivation of the Numerical Flux . . . 42
  - 3.2.2 Creation of Basis Functions . . . 45
  - 3.2.3 Operations on the Reference Element . . . 47
  - 3.2.4 Efficient Implementation of Matrix-Vector Products . . . 50

3.3 Discussion . . . 54
3.3.1 Computational Efficiency . . . 55
3.3.2 Error of the Spatial Discretisation . . . 55
3.3.3 The Discontinuous Galerkin Method and Nanophotonics . . . 56

4 The Discontinuous Galerkin Time-Domain Method 57
4.1 General Characteristics of Time-Domain Simulations . . . 57
4.2 Time-Stepping and the Runge-Kutta Method . . . 58
4.3 Eigenvalues, Conditional Stability and Maximum Time Steps . . . 60
4.4 Optimised Runge-Kutta Schemes . . . 62
4.5 Advantages and Drawbacks . . . 63

5 Discontinuous Galerkin Frequency-Domain Methods 65
5.1 General Characteristics of Frequency-Domain Simulations . . . 66
5.2 Eigenvalue Problems . . . 66
5.3 Driven Problems . . . 67
5.3.1 The Structure of the System Matrix . . . 68
5.3.2 Direct and Iterative Methods . . . 71
5.3.3 Preconditioning . . . 73
5.4 Performance Comparison . . . 76
5.4.1 Scattering of Light by an Infinite Cylinder (2D) . . . 76
5.4.2 Light Propagation in Waveguides Coupled to a Slotted Ring Resonator (2D) . 87
5.4.3 Scattering of Light by a Sphere (3D) . . . 92
5.5 Advantages and Drawbacks . . . 93

6 Essential Extensions for Practical Use 95
6.1 Sources . . . 95
6.1.1 The Total-Field/Scattered-Field Technique . . . 96
6.1.2 Spatial Profiles . . . 98
6.1.3 Time Dependencies . . . 98
6.2 Dispersive Media . . . 100
6.3 Curvilinear Elements . . . 101
6.4 Data Recording and Analysis . . . 102

7 Absorbing Boundaries: Perfectly Matched Layers 105
7.1 Uniaxial Perfectly Matched Layers . . . 106
7.2 A Novel Stretched-Coordinate Implementation . . . 109
7.2.1 Stretched Coordinates in Maxwell's Equations . . . 109
7.2.2 Spatial Discretisation via the Discontinuous Galerkin Method . . . 111
7.2.3 The Split-Flux Formulation . . . 112
7.2.4 Stability . . . 113
7.2.5 Some Differences to UPMLs . . . 113
7.3 Parameter Choice and Performance Comparison . . . 113
7.3.1 Test Configuration . . . 114
7.3.2 Numerical Results . . . 115
7.3.3 Computational Efficiency . . . 120

7.4 Summary . . . 120

8 Anisotropic Materials in the Discontinuous Galerkin Framework 123
8.1 The Semi-Discrete Form of the Anisotropic Maxwell Equations . . . 123
8.2 An Upwind Flux for Anisotropic Materials . . . 124
8.3 Validation . . . 126
8.4 Optical Cloaking . . . 129
8.5 Summary . . . 131

9 Numerical Study of Split-Ring Resonators 133
9.1 Split-Ring Resonator Basics . . . 133
9.1.1 Geometry and Material Parameters . . . 133
9.1.2 Physical Principles . . . 135
9.2 Individual Split-Ring Resonators . . . 137
9.2.1 Simulation Setup . . . 137
9.2.2 Scattering, Absorption, and Extinction . . . 139
9.2.3 Field Distributions . . . 140
9.2.4 Parameter Studies . . . 141
9.2.5 Summary . . . 148
9.3 Periodic Arrays of Split-Ring Resonators . . . 148
9.3.1 Simulation Setup . . . 149
9.3.2 Transmittance, Reflectance, and Absorption . . . 150
9.3.3 Influence of the Lattice Constant . . . 152
9.4 Split-Ring Resonator Dimers . . . 153
9.4.1 Simulation Setup and Configurations . . . 153
9.4.2 Side-By-Side Configuration . . . 156
9.4.3 On-Top Configuration . . . 157
9.4.4 Back-To-Back Configuration . . . 158
9.4.5 Face-To-Face Configuration . . . 161
9.4.6 Cheek-To-Cheek Configuration . . . 161
9.4.7 Influence of Substrate Index . . . 163
9.4.8 Summary . . . 164
9.5 From Individual Particles to Finite-Sized Arrays . . . 165
9.5.1 Horizontally Coupled Linear Chains . . . 165
9.5.2 Vertically Coupled Linear Chains . . . 167
9.5.3 Finite-Sized Arrays . . . 170
9.5.4 Conclusions . . . 172
9.6 Summary . . . 173

10 Summary and Outlook 175
10.1 Summary . . . 175
10.2 Outlook . . . 176
10.3 Closing Remarks . . . 179

A The Finite-Difference Time-Domain Method 181
A.1 Finite Differences . . . 181

A.2 Spatial Discretisation and the Yee Grid 182
A.3 A Leapfrog Scheme for Time-Evolution 183
A.4 Discussion 183

Acknowledgements **185**

Publications **187**

Bibliography **189**

Chapter 1

1 Introduction

The last decades have seen amazing improvements in advanced nanofabrication techniques, where tools such as electron beam lithography are successfully applied to structure materials in the nanometre regime. Electronics has benefitted very much from these and other advances in technology and has introduced them to everyday life. At the same time, nanostructured materials have facilitated completely new ways of how to control light.

The most interesting effects occur when the feature sizes of the system are comparable to or much smaller than the wavelength of incident electromagnetic waves [1–4]. In particular, photonic crystals—which consist of periodically structured dielectric materials—prohibit the propagation of light along certain directions for certain frequencies via the formation of band gaps. Deliberately introduced deviations from the periodicity allow to design functional elements such as waveguides, splitters, interferometers, and more.

In a similar fashion, periodically arranged *metallic* building blocks such as nanorods, split-ring resonators (see Fig. 1.1), and fishnets can be combined to form so-called metamaterials [1]: If these building blocks are much smaller than the wavelength of incident light, individual structures are no longer resolved. In this case, the composite material may be considered as an effective medium whose optical properties are largely determined by the building blocks. Intriguingly, this allows researchers to tailor material properties to their needs.

For instance, metamaterials which show a distinct magnetic resonance at optical frequencies have been reported. For certain systems, even a negative effective refractive index has been found, a property unheard of among natural materials [5]. Current research and possible applications include devices such as the perfect lens [6] and the optical cloak, which guides light around an obstacle in a way that it is invisible to an observer [7].

Nanostructures are also appealing for biological and chemical applications. For example, metallic nanostructures tend to enhance electric fields near tips and corners, in turn boosting nonlinear effects such as the Raman effect. This enables researchers to measure and identify the Raman signal of single molecules [8].

Based on the same principle of local field enhancement, it is also possible to construct plasmonic tweezers [9] which trap and manipulate small particles. As the field enhancement comes along with a strong localisation, metallic nanostructures provide a way to overcome the diffraction limit at optical frequencies. Scanning near-field optical microscopes utilise metallic tips to achieve resolutions in

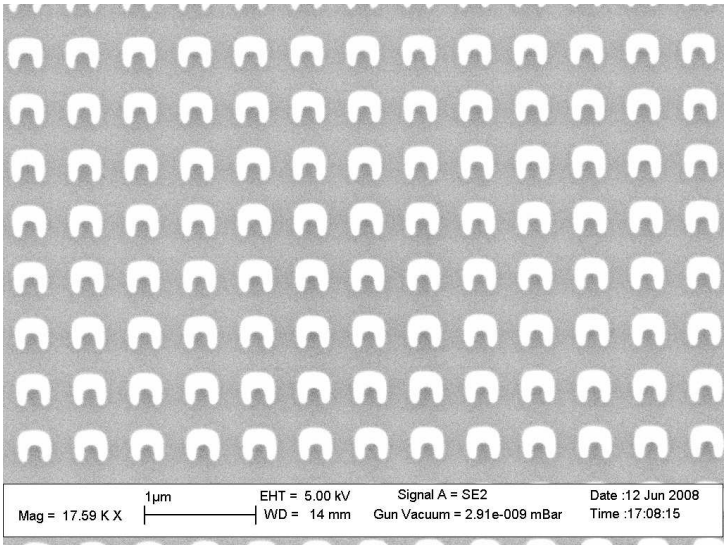


Figure 1.1: *Experimental realisation of nanostructures. The figure shows a scanning electron microscope image of an array of gold split-ring resonators on a silicon substrate, which I have fabricated myself while visiting the group of Thomas Krauss at the University of St Andrews, Scotland.*

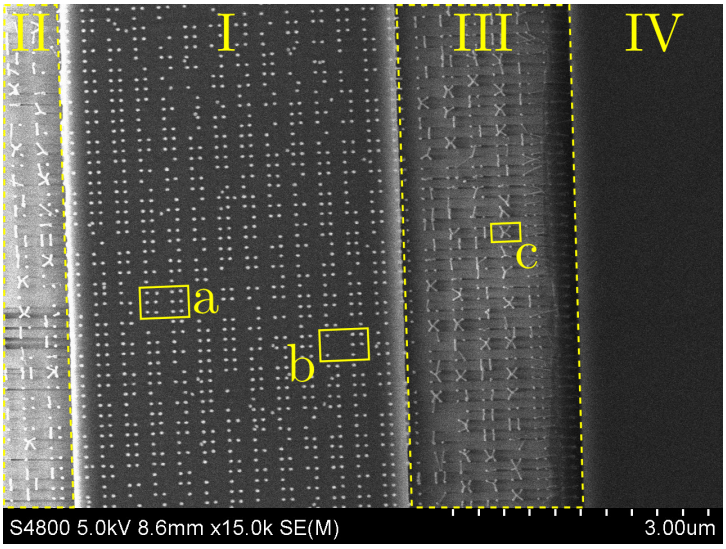


Figure 1.2: *Fabrication issues in modern nanolithography. The sample shows an array of nanoscopic dipole antennas (a) on a silicon strip waveguide (I) next to two trenches (II and III) in a silicon layer (IV). Due to the ill-fated choice of fabricating the dipoles first, many dipoles washed off during the waveguide fabrication steps, leaving a Braille-style pattern (b) behind. In addition, the involved etching process has led to the formation of partly collapsed nanopillars (c), which gives this image a certain aesthetic component.*

the sub-$100\,\mathrm{nm}$ regime [10].

Furthermore, microscopic dielectric resonators show very pronounced resonances with enormous quality factors. At the same time, they are extremely sensitive to their environment—so sensitive, in fact, that even a single molecule can change their resonance frequency [11–13]. Consequently, such resonators are promising candidates for biological and chemical sensing applications.

The list of fascinating achievements and working devices is virtually endless. Despite the impressive bandwidth of these experiments, however, there is one commonality: Experimental research in the field of nanophotonics is challenging, expensive, and suffers from limited resources and infrastructure. Luckily, numerical simulation tools can support and ease experiments in a number of stages.

1.1 The Need for Numerical Simulations

First of all, when planning a research project numerical experiments help to find promising systems and geometries. Proofs of concepts can be done without ever dealing with the fabrication process and the related difficulties (see Fig. 1.2 for an example). From such idealised experiments with optimal control, promising parameter ranges can be determined and the influence of different materials can be studied a priori.

Once the systems of interest are fabricated, numerical simulations can be used to optimise the system parameters in order to tweak the performance, e.g., to increase the quality factor of a resonator structure. On the other hand, numerical simulations also allow to assess the quality of the fabrication process, as imperfections are usually not included in the numerical model.

Finally, simulations can simplify the interpretation of experiments. Using computer programs, it is possible to access quantities such as charge and electromagnetic field distributions or heat dissipation, which may not be accessible in the actual experiment. Physical effects can be switched on and off independently to investigate the dominant contribution to a phenomenon. Thus, we can learn for future designs. Simulation results can be visualised to obtain an intuitive understanding of the underlying physics. In some cases this might even help to understand unexpected effects which are not included in the physical model or have not been considered.

1.2 A Survey of Common Simulation Methods

For nanophotonics, it is usually sufficient to employ a classical description of the electromagnetic fields as given by Maxwell's equations. It is important to understand that the choice of the simulation tool is crucial for the quality of the results. Among the wide range of available methods we find a large number of techniques for specific applications. For example, the multiple multipole technique (MMP) is ideally suited to simulate ellipsoidal particles [14]. The Wannier function expansion can accurately describe large-scale photonic crystals with a comparatively small number of degrees of freedom [15, 16]. Layered periodic structures are easily investigated using the Fourier modal method [17–19].

Not surprisingly, such methods cannot fully display their strengths when being applied to systems other than the ones they were originally intended for. In such cases, converged numerical results re-

quire a disproportionately large amount of computational resources. In contrast, there exist a couple of general purpose solvers which can—some extensions provided—tackle an extremely diverse set of systems with only moderate requirements.

The Finite-Difference Time-Domain (FDTD) algorithm is the most popular simulation tool for nanophotonics [20, 21]. Starting from an initial state, it evolves the electromagnetic fields in time. The field components are discretised at cleverly chosen positions within a rectangular lattice, the so-called Yee (or staggered) grid. Using a Taylor expansion for the derivatives and a leap-frog scheme in time, one ends up with a very simple procedure to evolve the fields step by step. Together with a multitude of extensions, FDTD has been successfully applied to countless systems, even nonlinear ones.

However, FDTD has a couple of weaknesses. It owes its simplicity to the comparatively inflexible Yee grid. Objects which are not parallel to the coordinate axes are subject to aliasing or the staircase effect, which is of special concern for metallic nanostructures. The accuracy of the spatial discretisation is further limited by the underlying Taylor expansion. As soon as material interfaces are present, the accuracy is reduced because the electromagnetic fields are no longer smooth. Despite various efforts [22–24], it remains extremely challenging to improve the spatial accuracy beyond the limits of the basic algorithm.

As a second commonly used class of general purpose solvers we would like to mention the Finite Element Method (FEM) [25, 26]. FEM works on a non-uniform mesh which is adapted to the geometry of interest. At crucial points, the mesh size can be reduced to increase the local resolution (h-refinement). Each element of the mesh holds an electromagnetic field representation of adjustable accuracy (p-refinement) and is coupled to its neighbours. Eventually, these building blocks lead to a sparse system of linear equations (SLE).

Solving such a system appears costly when compared to the computation of a single time step in FDTD. However, if one solves Maxwell's equations in the frequency-domain for just a small to medium (≈ 100) number of frequencies, this can be considerably faster than a corresponding time-domain simulation. Consequently, FEM is most often implemented in the frequency-domain. In contrast, finite element time stepping schemes are rather expensive, as one has to solve an SLE for each time step.

1.3 Overview of the Discontinuous Galerkin Method

It is desirable to find a way to combine the advantages of both FDTD and FEM. An adaptive mesh with adjustable order of accuracy should be woven into a scheme with reasonable efficiency concerning processor (CPU) time and main memory (RAM). Discontinuous Galerkin (DG) methods are one possibility to achieve this goal [27, 28].

The historical development of the DG method is rather intricate with important contributions by many researchers in different fields. Thus, we will restrict our historical account to a few selected works relevant for Maxwell's equations. More detailed chronologies and information on other types of partial differential equations can be found in Refs. [27, 29, 30], for example.

The DG method was first proposed and used in the context of steady-state neutron transport [31] in 1973. Later works applied the method to other research areas like acoustics, plasma physics, and hydrodynamics. Combined with a Runge-Kutta scheme [32], the method showed very promising

results for hyperbolic partial differential equations. After some previous efforts related to Maxwell's equations, Hesthaven and Warburton proposed a nodal[1] DG time-domain scheme for the numerical simulation of electrodynamic problems in 2002 [28]. Besides providing the algorithm itself, their paper also features rigorous mathematical proofs concerning numerical stability and convergence. Since then, a steadily increasing number of groups outside the mathematical community have implemented the DG method for Maxwell's equations (for example, see Refs. [33–36]).

In essence, the DG method is a variant of conventional FEM. As with FEM, we divide the computational domain, i.e., the domain on which we consider the problem, into a set of small subvolumes called elements. All together they form the mesh, which often consists of hundreds of thousands of elements. On each element, we expand the electromagnetic fields in terms of a set of basis functions. The main difference to FEM is that these basis functions are restricted to their respective elements; they are identically zero on all other—and in particular on the neighbouring—elements. Hence, at the interface between neighbours, we have two *different* field values, one for each element. This is the origin of the word "discontinuous" in the method's name and an important point, since it allows us to model and represent discontinuous electromagnetic fields at material interfaces without conceptual difficulties.

At the same time, this choice of basis leads to a vanishing overlap of basis functions defined on neighbouring elements. As a result, the electromagnetic problem on the mesh reduces to a set of electromagnetic problems on much smaller domains; the elements. In particular, any linear algebra involved (matrix-vector products, for example) acts only on the degrees of freedom associated with single elements. This results in small, manageable matrices and renders the algorithm very efficient. On the other hand, it is clear that the elements must be somehow coupled with each other to allow wave propagation from one element to another. Hence, once the potentially expensive linear algebra has been performed, we reintroduce the coupling between neighbouring elements using the concept of the numerical flux, which is borrowed from finite volume methods [37]. With this we obtain a semi-discrete form of Maxwell's curl equations in which the position-dependence is already discretised while the time-dependence is not.

We are free to discretise the time variable in terms of time steps similar to FDTD or we can solve Maxwell's curl equations in the frequency-domain. The time-domain version indeed combines the advantages of both FDTD and FEM because it is as accurate as FEM and—for a comparable level of accuracy—often faster than FDTD [38]. The frequency-domain version is supposedly not as efficient as conventional FEM, but it can be derived from the time-domain algorithm with just a few modifications. This is an advantage in practice, since computer code can be reused to minimise bugs and glitches.

1.4 Outline of the Thesis

The remainder of this thesis[2] is structured as follows: Chapter 2 gives a brief introduction to Maxwell's equations and related topics. It discusses solutions to a few selected problems, how to model dispersion in the time-domain and reviews the basics of electromagnetic scattering theory.

Chapter 3 gives a detailed account on the spatial discretisation via the discontinuous Galerkin

[1] A nodal scheme identifies degrees of freedom with field values at certain positions.

[2] For completeness, it should be mentioned that the text of this thesis is based on some of my publications [39–41].

method. In the course of this chapter, all spatial dependencies in Maxwell's equations are absorbed into expansion coefficients and small matrices. Eventually, this leads to a system of ordinary differential equations in time. In addition to the basic concepts, a few pages are dedicated to efficient implementation techniques.

Once the spatial discretisation has been established, we are left with the decision how to treat the remaining time-dependence. In principle, there are two possibilities. The first route follows the trails of FDTD, and thus describes the discontinuous Galerkin time-domain (DGTD) algorithm in chapter 4. Starting from an initial state, the electromagnetic fields are evolved using a low-storage Runge-Kutta time integrator. We discuss fundamental properties and provide a criterion for time steps which yield numerically stable solutions.

An alternative approach is topic of chapter 5. In contrast to discretising time itself, we investigate steady states using the discontinuous Galerkin frequency-domain (DGFD) method. In particular, we are interested in systems excited by time-harmonic sources. Solving a large system of linear equations replaces the simple time-stepping process of DGTD. The challenging task of balancing execution time and memory requirements is one of the main topics of this chapter.

The three subsequent chapters extend the discontinuous Galerkin technique in a multitude of ways. In chapter 6 we introduce sources, dispersive media, and curvilinear elements. In addition, we describe how to obtain spectral information from time-domain simulations to facilitate a comparison with real-world experiments. Chapters 7 and 8 focus on two extensions developed in the course of this thesis. First, a novel stretched-coordinate based implementation of perfectly matched layers offers an alternative to commonly employed absorbing boundary conditions (chapter 7). Secondly, we provide details on how to include anisotropic materials in the discontinuous Galerkin framework (chapter 8).

Having gathered all necessary tools and techniques, we employ the discontinuous Galerkin method to simulate electromagnetic scattering of split-ring resonators (chapter 9). Starting from individual resonators, we systematically investigate the influence of mutual coupling between split-rings on the properties of a periodic array of such resonators.

Finally, the thesis concludes with a summary and an outlook on potential future research.

Chapter 2
2 Classical Electrodynamics

Before we dive right into the discontinuous Galerkin method, its extensions, and applications, we first want to establish the theoretical background for the remainder of this thesis. For the physical systems we have in mind, quantum effects do not play a perceptible role. Consequently, it is sufficient and accurate to describe light-matter interaction in terms of classical electrodynamics.

Thus, we briefly review Maxwell's equations in section 2.1. We highlight the role of material interfaces and introduce some notations, conventions, and simplifications which will be used in the remainder of this thesis.

Section 2.2 presents solutions to selected wave propagation phenomena like normal incidence on a dielectric half space. The problem of handling dispersive material parameters in the time-dependent Maxwell equations is discussed in section 2.3. Finally, the chapter closes with a short introduction to electromagnetic scattering theory.

2.1 Maxwell's Equations

The classical theory of electrodynamics was established in 1865, when James Clerk Maxwell published his seminal paper "A Dynamical Theory of the Electromagnetic Field" [42] following a presentation held a year before. In modern vector notation as introduced by Oliver Heaviside, Maxwell's equations in the SI unit system read [43]

$$\vec{\nabla} \cdot \vec{\mathrm{D}}(\vec{\mathrm{r}},\, t) = \rho(\vec{\mathrm{r}},\, t), \tag{2.1a}$$

$$\vec{\nabla} \cdot \vec{\mathrm{B}}(\vec{\mathrm{r}},\, t) = 0, \tag{2.1b}$$

$$\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) = \quad \partial_t \vec{\mathrm{D}}(\vec{\mathrm{r}},\, t) + \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t), \tag{2.1c}$$

$$\vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = -\partial_t \vec{\mathrm{B}}(\vec{\mathrm{r}},\, t). \tag{2.1d}$$

These equations feature the electric field $\vec{\mathrm{E}}$, the magnetic field $\vec{\mathrm{H}}$, the electric displacement $\vec{\mathrm{D}}$, and the magnetic induction $\vec{\mathrm{B}}$. Free electric charges ρ and free electric currents $\vec{\mathrm{j}}$ act as sources to these electromagnetic fields, which all depend on both position $\vec{\mathrm{r}} = (x,\, y,\, z)^T$ and time t. All vectors in Eqs. (2.1) are three-dimensional, i.e., they consist of x-, y-, and z-components.

2.1.1 Constitutive Relations

In the above form, Maxwell's equations are not complete. As a closure, we have to provide the constitutive relations

$$\begin{aligned}\vec{\mathrm{D}}(\vec{\mathrm{r}},\, t) &= \epsilon_0 \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) + \vec{\mathrm{P}}[\vec{\mathrm{E}},\, \vec{\mathrm{H}}],\\ \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) &= \frac{1}{\mu_0}\vec{\mathrm{B}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{M}}[\vec{\mathrm{E}},\, \vec{\mathrm{H}}].\end{aligned} \tag{2.2}$$

Here, we have introduced the vacuum permittivity ϵ_0 and the vacuum permeability μ_0. Both quantities are fundamental constants. Furthermore, we have introduced two fields which account for the presence of matter; the polarisation $\vec{\mathrm{P}}$ and the magnetisation $\vec{\mathrm{M}}$. In general, both fields are functionals of the electric and magnetic fields [44] and rely on a macroscopic description of matter: If the wavelength of incident electromagnetic radiation is sufficiently large, the effects of individual bound charges and currents present in atoms and molecules are averaged.[1] In particular, it is not necessary to keep track of each charge separately. Considering typical numbers of atoms this is indeed quite convenient.

As a result, each material can be characterised by specifying the dependence of polarisation and magnetisation on externally applied electric and magnetic fields. Within the context of this thesis, it is reasonable to introduce a few simplifications.

First of all, we are interested in low field strengths as compared to atomic ones. As a result, the response of the materials is approximately linear, i.e.,

$$\begin{aligned}\vec{\mathrm{P}}[\lambda \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t)] &= \lambda \cdot \vec{\mathrm{P}}[\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t)]\\ \vec{\mathrm{M}}[\lambda \cdot \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t)] &= \lambda \cdot \vec{\mathrm{M}}[\vec{\mathrm{H}}(\vec{\mathrm{r}},\, t)]\end{aligned} \tag{2.3}$$

for any $\lambda \in \mathbb{R}$. Furthermore, we have assumed that $\vec{\mathrm{P}}$ ($\vec{\mathrm{M}}$) only depends on $\vec{\mathrm{E}}$ ($\vec{\mathrm{H}}$). For such linear materials, Maxwell's equations form a system of linear partial differential equations. Consequently, any superposition of solutions of Maxwell's equations satisfies the same equations itself. Using the Fourier transform, it is hence possible to split all fields into their respective spectral components. In the case of the electric field this reads

$$\begin{aligned}\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega)\cdot\exp(-\mathrm{i}\omega t)\,\mathrm{d}\omega \qquad \text{with}\\ \vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) &= \int_{-\infty}^{\infty} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t)\cdot\exp(+\mathrm{i}\omega t)\,\mathrm{d}t.\end{aligned} \tag{2.4}$$

The new complex-valued frequency-domain field $\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega)$ represents the spectral component of the time-domain field $\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t)$ which oscillates with angular[2] frequency ω.

In the frequency-domain, the constitutive relations for linear materials are most simply expressed as

$$\begin{aligned}\vec{\mathrm{D}}(\vec{\mathrm{r}},\, \omega) &= \epsilon_0 \underline{\epsilon}_r(\vec{\mathrm{r}},\, \omega)\cdot\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega),\\ \vec{\mathrm{B}}(\vec{\mathrm{r}},\, \omega) &= \mu_0 \underline{\mu}_r(\vec{\mathrm{r}},\, \omega)\cdot\vec{\mathrm{H}}(\vec{\mathrm{r}},\, \omega).\end{aligned} \tag{2.5}$$

[1] This thesis covers wavelengths larger than $100\,\mathrm{nm}$, which is significantly larger than any atomic scale.

[2] In the remainder of this thesis, we will omit the explicit mention of "angular".

All material properties—previously hidden in polarisation and magnetisation—are given by the *relative* permittivity and permeability tensors $\underline{\epsilon}_r$ and $\underline{\mu}_r$, respectively. The frequency-dependence of these material parameters describes dispersion and is especially pronounced for metals such as gold in the visible regime [45, 46]. At the same time, their tensorial nature allows for anisotropic wave propagation, i.e., the velocity of travelling waves in a medium depends on the direction of propagation and the polarisation. Even though both properties are important and relevant at later stages of this thesis, we will restrict ourselves to isotropic, nondispersive materials—such as many dielectrics in the optical regime—for the time being. In this case, the constitutive relations further simplify to

$$\begin{aligned} \vec{\mathrm{D}}(\vec{\mathrm{r}},\,\omega) &= \epsilon_0\epsilon_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{E}}(\vec{\mathrm{r}},\,\omega), \\ \vec{\mathrm{B}}(\vec{\mathrm{r}},\,\omega) &= \mu_0\mu_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{H}}(\vec{\mathrm{r}},\,\omega). \end{aligned} \tag{2.6}$$

Since ϵ_r and μ_r no longer depend on the frequency, these relations are straightforwardly transformed back into time-domain:

$$\begin{aligned} \vec{\mathrm{D}}(\vec{\mathrm{r}},\,t) &= \epsilon_0\epsilon_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{E}}(\vec{\mathrm{r}},\,t), \\ \vec{\mathrm{B}}(\vec{\mathrm{r}},\,t) &= \mu_0\mu_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{H}}(\vec{\mathrm{r}},\,t). \end{aligned} \tag{2.7}$$

Equations (2.7) enable us to eliminate both $\vec{\mathrm{D}}$ and $\vec{\mathrm{B}}$ from Maxwell's equations. We obtain Maxwell's equations for linear, dispersionless, isotropic materials:

$$\vec{\nabla}\cdot\Big(\epsilon_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{E}}(\vec{\mathrm{r}},\,t)\Big) = \frac{\rho(\vec{\mathrm{r}},\,t)}{\epsilon_0}, \tag{2.8a}$$

$$\vec{\nabla}\cdot\Big(\mu_r(\vec{\mathrm{r}})\cdot\vec{\mathrm{B}}(\vec{\mathrm{r}},\,t)\Big) = 0, \tag{2.8b}$$

$$\vec{\nabla}\times\vec{\mathrm{H}}(\vec{\mathrm{r}},\,t) = \quad \epsilon_0\epsilon_r(\vec{\mathrm{r}})\cdot\partial_t\vec{\mathrm{E}}(\vec{\mathrm{r}},\,t) + \vec{\mathrm{j}}(\vec{\mathrm{r}},\,t), \tag{2.8c}$$

$$\vec{\nabla}\times\vec{\mathrm{E}}(\vec{\mathrm{r}},\,t) = -\mu_0\mu_r(\vec{\mathrm{r}})\cdot\partial_t\vec{\mathrm{H}}(\vec{\mathrm{r}},\,t). \tag{2.8d}$$

2.1.2 Material Interfaces

In regions with smoothly varying material parameters, the solutions to Maxwell's equations are smooth themselves. For discontinuous parameters at material interfaces this statement is no longer applicable. When free charges and currents are absent, we can integrate the divergence equations (2.8a) and (2.8b) over a Gaussian pillbox (see Fig. 2.1) and apply Gauss' theorem [43]. For an infinitesimally thin pillbox, contributions from the box's sides vanish. If we assume a small section of a material interface inside the box, and denote its normal vector by $\hat{\mathrm{n}}$, we obtain

$$\begin{aligned} \epsilon_1\hat{\mathrm{n}}\cdot\vec{\mathrm{E}}_1 &= \epsilon_2\hat{\mathrm{n}}\cdot\vec{\mathrm{E}}_2, \\ \mu_1\hat{\mathrm{n}}\cdot\vec{\mathrm{H}}_1 &= \mu_2\hat{\mathrm{n}}\cdot\vec{\mathrm{H}}_2. \end{aligned} \tag{2.9}$$

Please note that the indices "1" and "2" discriminate fields and material parameters on either side of the interface (compare Fig. 2.1). Equations (2.9) state that the normal components of the electric and magnetic fields exhibit well-defined, material-dependent discontinuities. Hence, an accurate numerical description of electromagnetic fields must allow for discontinuities as well.

Likewise, we can investigate the curl equations (2.8c) and (2.8d). This time, however, we integrate over an area bounded by a closed Stokesian loop of infinitesimal height (Fig. 2.1). Exploiting Stoke's

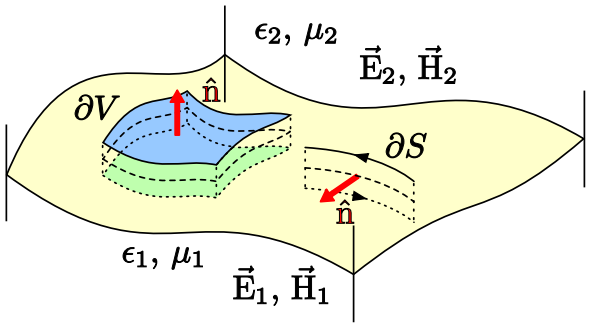


Figure 2.1: *Special integration domains to determine the behaviour of electromagnetic fields at material interfaces (yellow surface). The left-hand volume is called a Gaussian pillbox, while the right-hand contour represents a Stokesian loop. Dashed lines indicate intersections with the interface. Adapted from [47].*

theorem yields [43]

$$\begin{aligned} \hat{\mathrm{n}} \times \vec{\mathrm{E}}_1 &= \hat{\mathrm{n}} \times \vec{\mathrm{E}}_2, \\ \hat{\mathrm{n}} \times \vec{\mathrm{H}}_1 &= \hat{\mathrm{n}} \times \vec{\mathrm{H}}_2. \end{aligned} \tag{2.10}$$

Thus, those components of the electromagnetic fields tangential to an interface are always continuous.

2.1.3 Reduction to the Curl Equations

Maxwell's equations (2.8) represent a set of eight coupled partial differential equations (PDEs) of first order. Since Eqs. (2.8c) and (2.8d) are the only equations which contain temporal derivatives, it is reasonable that these equations suffice to uniquely determine the time-evolution of the electromagnetic fields.

To prove this, we first derive the continuity equation by taking the divergence of (2.8c) and inserting (2.8a). Since the divergence of the curl of any vector field is zero, we obtain the continuity condition

$$\partial_t \rho(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{j}}(\vec{\mathrm{r}}, t) = 0. \tag{2.11}$$

Hence, each change in the local charge density is associated with an electric current. Using this relation, it is easy to show that

$$\partial_t \left(\vec{\nabla} \cdot \left(\epsilon_r(\vec{\mathrm{r}}) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \right) - \frac{\rho(\vec{\mathrm{r}}, t)}{\epsilon_0} \right) = 0, \tag{2.12}$$

$$\partial_t \left(\vec{\nabla} \cdot \left(\mu_r(\vec{\mathrm{r}}) \cdot \vec{\mathrm{B}}(\vec{\mathrm{r}}, t) \right) \right) = 0. \tag{2.13}$$

This implies that if the divergence conditions are satisfied for a given initial state (characterised by $\vec{\mathrm{E}}(\vec{\mathrm{r}}, t_0)$ and $\vec{\mathrm{H}}(\vec{\mathrm{r}}, t_0)$), they will be so for all future times as well.

For the remainder of this thesis, we will omit the divergence conditions and concentrate on Maxwell's curl equations

$$\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) = \quad \epsilon_0 \epsilon_r(\vec{\mathrm{r}}) \cdot \partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) + \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t), \tag{2.14a}$$

$$\vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = -\mu_0 \mu_r(\vec{\mathrm{r}}) \cdot \partial_t \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t). \tag{2.14b}$$

These equations fully determine the time-evolution of the electromagnetic fields. Since the charge density $\rho(\vec{\mathrm{r}},\, t)$ does not enter the curl equations, we can always assume the continuity condition to be implicitly fulfilled. Specifically, we consider (2.11) as the definition of ρ for any current density distribution $\vec{\mathrm{j}}(\vec{\mathrm{r}},\, t)$.

2.1.4 Reduction to Dimensionless Units

To render Maxwell's curl equations (2.14) more accessible to numerical treatment, it is helpful to eliminate ϵ_0 and μ_0. Formally, this can be achieved by absorbing these constants into the electromagnetic fields, e.g., by introducing new variables

$$\vec{\mathrm{E}}'(\vec{\mathrm{r}},\, t') = \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t), \quad \vec{\mathrm{H}}'(\vec{\mathrm{r}},\, t') = \sqrt{\frac{\mu_0}{\epsilon_0}} \cdot \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t), \quad \text{and} \quad t' = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \cdot t. \tag{2.15}$$

In this new system, both the electric and the magnetic field share the same unit, i.e., $[\mathrm{V/m}]$. Similarly, the time variable has been rescaled in order to have the unit of a length, i.e., $[\mathrm{m}]$. At the same time, the vacuum speed of light changes from

$$c = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \quad \text{to} \quad c' = 1. \tag{2.16}$$

Furthermore, it is possible to rescale all fields by a reference field strength E_0 and all lengths by a length scale λ_0:

$$\begin{aligned}
\vec{\mathrm{E}}''(\vec{\mathrm{r}}',\, t'') &= \frac{1}{E_0} \cdot \vec{\mathrm{E}}'(\vec{\mathrm{r}},\, t'), & t'' &= \frac{1}{\lambda_0} \cdot t', \\
\vec{\mathrm{H}}''(\vec{\mathrm{r}}',\, t'') &= \frac{1}{E_0} \cdot \vec{\mathrm{H}}'(\vec{\mathrm{r}},\, t'), & \vec{\mathrm{r}}'' &= \frac{1}{\lambda_0} \cdot \vec{\mathrm{r}}, \\
\vec{\mathrm{j}}''(\vec{\mathrm{r}}',\, t'') &= \frac{1}{E_0} \cdot \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t'). &&
\end{aligned} \tag{2.17}$$

Omitting all primes for notational convenience and solving for the time-derivatives, Maxwell's curl equations read

$$\partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \quad \epsilon^{-1}(\vec{\mathrm{r}}) \cdot \Big(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t) \Big), \tag{2.18a}$$

$$\partial_t \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) = -\mu^{-1}(\vec{\mathrm{r}}) \cdot \vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t). \tag{2.18b}$$

Note that we have also dropped the index r, since in our new unit system it is obvious that ϵ and μ denote relative permittivities and permeabilities, respectively. The conversion of units for the relevant physical quantities is summarised in Table 2.1.

Quantity	In SI units	In Dimensionless Units
Electric field	$\vec{\mathrm{E}}_{\mathrm{SI}}$	$\vec{\mathrm{E}} = \frac{1}{E_0} \cdot \vec{\mathrm{E}}_{\mathrm{SI}}$
Magnetic field	$\vec{\mathrm{H}}_{\mathrm{SI}}$	$\vec{\mathrm{H}} = \frac{1}{E_0}\sqrt{\frac{\mu_0}{\epsilon_0}} \cdot \vec{\mathrm{H}}_{\mathrm{SI}}$
Current	$\vec{\mathrm{j}}_{\mathrm{SI}}$	$\vec{\mathrm{j}} = \frac{1}{E_0} \cdot \vec{\mathrm{j}}_{\mathrm{SI}}$
(Wave-)Length	λ_{SI}	$\lambda = \frac{1}{\lambda_0} \cdot \lambda_{\mathrm{SI}}$
Time	t_{SI}	$t = \frac{c_{\mathrm{SI}}}{\lambda_0} \cdot t_{\mathrm{SI}}$
Frequency	ω_{SI}	$\omega = \frac{\lambda_0}{c_{\mathrm{SI}}} \cdot \omega_{\mathrm{SI}}$

Table 2.1: *Conversion from SI system to dimensionless units. The conversion factors involve the vacuum permittivity ϵ_0, the vacuum permeability μ_0 and the vacuum speed of light $c_{\mathrm{SI}} = (\epsilon_0\mu_0)^{-1/2}$. In addition, we rescale all fields by an arbitrary field E_0. Times and lengths are rescaled by an arbitrary reference length λ_0. For nanophotonic systems, typical scales are given by $E_0 = 1\,\frac{\mathrm{V}}{\mathrm{m}}$ and $\lambda_0 = 1\,\mathrm{nm}$.*

2.1.5 Reduction to Two Spatial Dimensions

For some selected problems, it is not necessary to solve the three-dimensional curl equations (2.18). Consider a physical system in which the material distribution is uniform along a certain direction, e.g., the z-direction. If both the initial field distribution $\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t_0)$, $\vec{\mathrm{H}}(\vec{\mathrm{r}},\, t_0)$ and the current density $\vec{\mathrm{j}}(\vec{\mathrm{r}},\, t)$ are independent of z, the electromagnetic fields at later times $t > t_0$ must not depend on z as well for symmetry reasons. Thus, all derivatives with respect to z vanish identically and the problem is effectively a two-dimensional one.

As a result, the six components of the curl equations can be decomposed into two distinct sets of equations. The transverse-electric (TE) polarisation is given by

$$\begin{aligned}
\partial_t E_x(t) &= \epsilon^{-1} \cdot \Big(\partial_y H_z(t) - j_x(t)\Big),\\
\partial_t E_y(t) &= \epsilon^{-1} \cdot \Big(-\partial_x H_z(t) - j_y(t)\Big),\\
\partial_t H_z(t) &= \mu^{-1} \cdot \Big(\partial_y E_x(t) - \partial_x E_y(t)\Big).
\end{aligned} \tag{2.19}$$

Similarly, the transverse-magnetic (TM) polarisation reads

$$\begin{aligned}
\partial_t H_x(t) &= -\mu^{-1}\partial_y E_z(t),\\
\partial_t H_y(t) &= \;\;\mu^{-1}\partial_x E_z(t),\\
\partial_t E_z(t) &= \epsilon^{-1} \cdot \Big(\partial_x H_y(t) - \partial_y H_x(t) - j_z(t)\Big).
\end{aligned} \tag{2.20}$$

For notational convenience, we have suppressed the obvious dependence of the fields on the spatial variables $(x,\, y)$ in both sets of equations.

The general solution of a system with prerequisites as stated above is a superposition of both polarisation states. Oftentimes, the source current $\vec{\mathrm{j}}$ excites only one of the two polarisations. If the

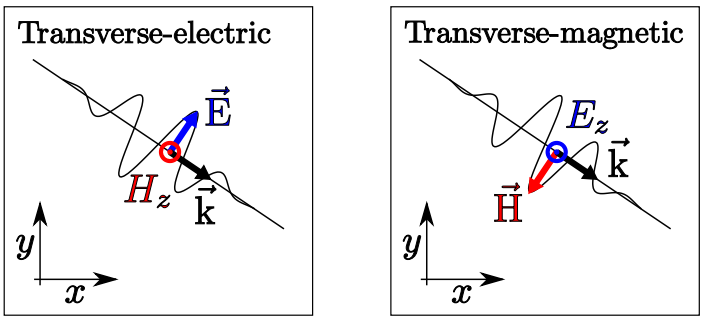


Figure 2.2: *Two-dimensional polarisations. The left-hand side shows the transverse-electric (TE) polarisation, while the right-hand side illustrates the transverse-magnetic (TM) polarisation.*

initial fields vanish and we fix $j_z(t) \equiv 0$, for example, then the TM-polarisation will not be excited. Hence, it would be sufficient to solve Eqs. (2.19). Figure 2.2 illustrates both polarisation states.

It is possible to generalise this result for anisotropic materials, provided the material tensors are of the form

$$\underline{\epsilon} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} & 0 \\ \epsilon_{yx} & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{pmatrix} \quad \text{and} \quad \underline{\mu} = \begin{pmatrix} \mu_{xx} & \mu_{xy} & 0 \\ \mu_{yx} & \mu_{yy} & 0 \\ 0 & 0 & \mu_{zz} \end{pmatrix}. \tag{2.21}$$

Introducing the inverse permittivity tensor $\underline{\eta} \equiv \underline{\epsilon}^{-1}$ to replace ϵ^{-1} in Eqs. (2.19), we obtain

$$\begin{aligned} \partial_t E_x(t) &= \eta_{xx}\partial_y H_z(t) - \eta_{xy}\partial_x H_z(t) - \eta_{xx} j_x(t) - \eta_{xy} j_y(t) \\ \partial_t E_y(t) &= \eta_{yx}\partial_y H_z(t) - \eta_{yy}\partial_x H_z(t) - \eta_{yx} j_x(t) - \eta_{yy} j_y(t) \\ \partial_t H_z(t) &= \mu_{zz}^{-1} \cdot \Big(\partial_y E_x(t) - \partial_x E_y(t)\Big). \end{aligned} \tag{2.22}$$

Similar expressions can be derived for the transverse-magnetic polarisation.

2.2 Solutions to Selected Problems

Maxwell's equations are generally difficult to solve and thus demand numerical simulation tools. For a few selected systems, however, analytical solutions are available. This section discusses the propagation of plane waves in homogeneous regions and deals with interfaces introduced by dielectric half spaces and films. Once these solutions are available, they can be directly employed to investigate scattering phenomena via the total-field/scattered-field method presented in section 6.1.

2.2.1 Plane Wave Propagation

Consider Maxwell's equations in a homogeneous region, i.e.,

$$\epsilon(\vec{\mathrm{r}}) \equiv \epsilon = \text{const} \quad \text{and} \quad \mu(\vec{\mathrm{r}}) \equiv \mu = \text{const}. \tag{2.23}$$

Furthermore, we assume the absence of sources, i.e., $\vec{\mathrm{j}}(\vec{\mathrm{r}},\, t) \equiv 0,\ \rho(\vec{\mathrm{r}},\, t) \equiv 0$. Inserting Eq. (2.18b) into the time-derivative of Eq. (2.18a) yields

$$\partial_t^2 \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) + \frac{1}{\epsilon\mu}\vec{\nabla} \times \Big(\vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t)\Big) = 0. \tag{2.24}$$

Since $\vec{\nabla} \times (\vec{\nabla} \times \vec{\mathrm{E}}) = \vec{\nabla}(\vec{\nabla} \cdot \vec{\mathrm{E}}) - \Delta\vec{\mathrm{E}}$ and $\vec{\nabla} \cdot \vec{\mathrm{E}} \equiv 0$ according to the divergence condition (2.8a), we obtain the wave equation

$$\Delta\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) - \frac{1}{c^2}\partial_t^2 \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = 0 \tag{2.25}$$

with the speed of light given by $c = (\epsilon\mu)^{-\frac{1}{2}}$. The Fourier transform (2.4) leads us to the Helmholtz equation

$$\Delta\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) + \frac{\omega^2}{c^2}\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) = 0. \tag{2.26}$$

It is straightforward to prove that a solution is given by

$$\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) = \vec{\mathrm{E}}_0(\omega) \cdot \exp(\mathrm{i}\vec{\mathrm{k}} \cdot \vec{\mathrm{r}}). \tag{2.27}$$

Equation (2.27) describes the propagation of a plane wave with frequency-dependent amplitude $\vec{\mathrm{E}}_0(\omega)$ and wave vector $\vec{\mathrm{k}}$, where the wave vector is normal to $\vec{\mathrm{E}}_0(\omega)$, i.e., $\vec{\mathrm{E}}_0(\omega) \cdot \vec{\mathrm{k}} \equiv 0$, and obeys the dispersion relation $\omega^2 = k^2c^2,\ k \equiv |\vec{\mathrm{k}}|$. The corresponding magnetic field follows from the frequency-domain version of the curl equation (2.18b):

$$\vec{\mathrm{H}}(\vec{\mathrm{r}},\, \omega) = \frac{1}{\omega\mu}\vec{\mathrm{k}} \times \vec{\mathrm{E}}_0(\omega) \cdot \exp(\mathrm{i}\vec{\mathrm{k}} \cdot \vec{\mathrm{r}}). \tag{2.28}$$

The dispersion relation allow us to split the wave vector into a magnitude and a direction:

$$\vec{\mathrm{k}} = \frac{\omega}{c} \cdot \hat{\mathrm{e}}_k. \tag{2.29}$$

Here, $\hat{\mathrm{e}}_k \equiv \vec{\mathrm{k}}/|\vec{\mathrm{k}}|$ is the unit vector in the propagation direction. With this, we can simplify Eq. (2.28) to

$$\vec{\mathrm{H}}(\vec{\mathrm{r}},\, \omega) = \sqrt{\frac{\epsilon}{\mu}} \cdot \hat{\mathrm{e}}_k \times \vec{\mathrm{E}}_0(\omega) \cdot \exp(\mathrm{i}\vec{\mathrm{k}} \cdot \vec{\mathrm{r}}). \tag{2.30}$$

With the frequency-domain solution in place, we reapply the Fourier transform to obtain the time-domain solution

$$\begin{aligned} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega)\exp(-\mathrm{i}\omega t)\,\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \vec{\mathrm{E}}_0(\omega)\exp\Big(-\mathrm{i}\omega\Big(t - \frac{\hat{\mathrm{e}}_k \cdot \vec{\mathrm{r}}}{c}\Big)\Big)\,\mathrm{d}\omega. \end{aligned} \tag{2.31}$$

The last integral, however, is nothing else but a Fourier transform of the amplitude $\vec{\mathrm{E}}_0(\omega)$. Analogous steps for the magnetic field eventually lead us to time-domain expressions for the propagation of a plane pulse:

$$\begin{aligned} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) &= \vec{\mathrm{E}}_0\Big(t - \frac{\hat{\mathrm{e}}_k \cdot \vec{\mathrm{r}}}{c}\Big), \\ \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) &= \sqrt{\frac{\epsilon}{\mu}} \cdot \hat{\mathrm{e}}_k \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t). \end{aligned} \tag{2.32}$$

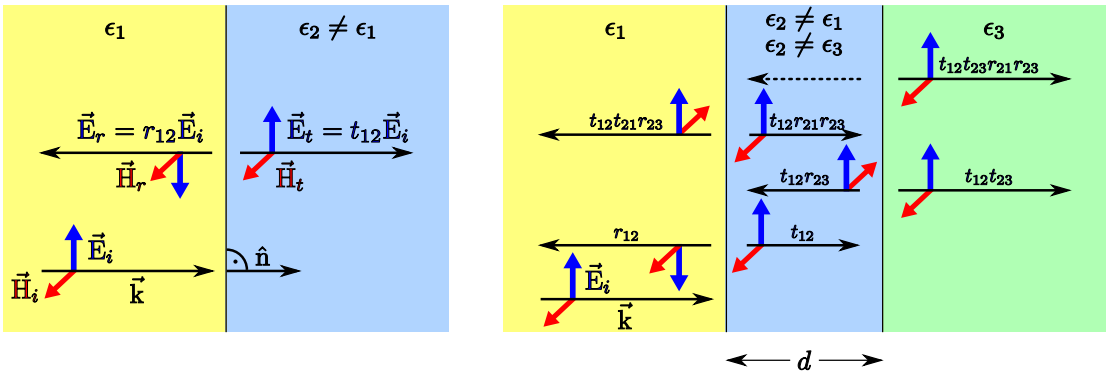


Figure 2.3: *Reflection of a plane wave by a dielectric half space (left panel) and a thin dielectric film (right panel). The black arrows illustrate the direction of propagation. Blue (red) arrows represent the direction of the electric (magnetic) field. The fields are not drawn to scale. However, the labels specify the ratio of the respective electric field amplitudes as compared to the initial field strength* $\vec{E}_i$*. Phase shifts as they appear in Eq.* (2.41) *are omitted for the sake of readability.*

2.2.2 Normal Incidence on a Dielectric Half Space

Let us generalise this result to the case of two half spaces made of different materials. For these materials we assume $\mu_1 = \mu_2 \equiv 1$, since the magnetic response of natural materials is negligible at optical frequencies. Thus, the materials are uniquely defined by specifying their permittivities ϵ_1 and ϵ_2, respectively (see Fig. 2.3). Accordingly, the speed of light in regions 1 and 2 is given by c_1 and c_2.

In half space 1, a plane wave is travelling towards the interface between both materials, which includes the coordinate system's origin. More specifically, its wave vector $\vec{k}$ is parallel to the interface's normal vector $\hat{n}$. For normal incidence, the electric field on both sides of the interface can be written as

$$\vec{E}(\vec{r},\,\omega) = \begin{cases} \vec{E}_i(\omega)\exp\left(i\frac{\omega}{c_1}\hat{n}\cdot\vec{r}\right) + \vec{E}_r(\omega)\exp\left(-i\frac{\omega}{c_1}\hat{n}\cdot\vec{r}\right) & \text{for} \quad \hat{n}\cdot\vec{r} < 0 \\ \vec{E}_t(\omega)\exp\left(i\frac{\omega}{c_2}\hat{n}\cdot\vec{r}\right) & \text{for} \quad \hat{n}\cdot\vec{r} > 0 \end{cases} . \tag{2.33}$$

The indices "i", "r", and "t" represent the incident, reflected, and transmitted electric field, respectively. The associated magnetic fields are easily obtained using Eq. (2.30) with $\hat{e}_k = \pm\hat{n}$. Fresnel's equations [48] establish the connexion between the different amplitudes. We find

$$\vec{E}_r(\omega) = r_{12}\vec{E}_i(\omega), \qquad \vec{E}_t(\omega) = t_{12}\vec{E}_i(\omega), \tag{2.34}$$

with the reflection coefficient from half space a to half space b given by

$$r_{ab} = \frac{n_a - n_b}{n_a + n_b}. \tag{2.35}$$

Similarly, the transmission coefficient is given by

$$t_{ab} = \frac{2n_a}{n_a + n_b}. \tag{2.36}$$

The material properties of both half spaces enter these coefficients via the refractive indices

$$n_a \equiv \sqrt{\epsilon_a} = \frac{1}{c_a} \qquad (\mu_a \equiv 1). \tag{2.37}$$

As in the case of standard plane waves, we are also interested in a time-domain solution. After a Fourier transform we obtain

$$\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \begin{cases} \vec{\mathrm{E}}_i\Big(t - \dfrac{\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}}{c_1}\Big) + r_{12}\vec{\mathrm{E}}_i\Big(t + \dfrac{\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}}{c_1}\Big) & \text{for} \quad \hat{\mathrm{n}}\cdot\vec{\mathrm{r}} < 0 \\ t_{12}\vec{\mathrm{E}}_i\Big(t - \dfrac{\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}}{c_2}\Big) & \text{for} \quad \hat{\mathrm{n}}\cdot\vec{\mathrm{r}} > 0 \end{cases}. \tag{2.38}$$

2.2.3 Normal Incidence on a Thin Dielectric Film

Following the reasoning in the previous section, we can add another material interface to the system. Thus, we investigate normal incidence on a thin film of thickness d and permittivity ϵ_2, which separates two half spaces with permittivities ϵ_1 and ϵ_3 (Fig. 2.3). As before, the permeabilities of all materials are set to unity. At the two interfaces, plane waves are partially reflected and transmitted. Hence, multiple reflections (accounted for by summation indices l) occur and slightly complicate the solution in Fourier space:

$$\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) = \begin{cases} \vec{\mathrm{E}}_i(\omega)\exp\Big(\mathrm{i}\dfrac{\omega}{c_1}\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}\Big) + \displaystyle\sum_{l=0}^{\infty}\vec{\mathrm{E}}_{r,\,l}(\omega)\exp\Big(-\mathrm{i}\dfrac{\omega}{c_1}\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}\Big) & \text{for} \quad \hat{\mathrm{n}}\cdot\vec{\mathrm{r}} < 0 \\ \displaystyle\sum_{l=1}^{\infty}\vec{\mathrm{E}}_{+,\,l}(\omega)\exp\Big(\mathrm{i}\dfrac{\omega}{c_2}\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}\Big) + \sum_{l=1}^{\infty}\vec{\mathrm{E}}_{-,\,l}(\omega)\exp\Big(-\mathrm{i}\dfrac{\omega}{c_2}\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}\Big) & \text{for} \quad 0 < \hat{\mathrm{n}}\cdot\vec{\mathrm{r}} < d \\ \displaystyle\sum_{l=1}^{\infty}\vec{\mathrm{E}}_{t,\,l}(\omega)\exp\Big(\mathrm{i}\dfrac{\omega}{c_3}\hat{\mathrm{n}}\cdot\vec{\mathrm{r}}\Big) & \text{for} \quad \hat{\mathrm{n}}\cdot\vec{\mathrm{r}} > d \end{cases}. \tag{2.39}$$

In addition to the incident, reflected, and transmitted amplitudes, we have also introduced amplitudes for waves in the film, which propagate in the same ($\vec{\mathrm{E}}_+(\omega)$) or the opposite ($\vec{\mathrm{E}}_-(\omega)$) direction as the incident wave.

All amplitudes follow from that of the incident wave. In contrast to the simple half space problem, we have to account for a phase shift

$$\Delta\phi = \exp\Big(\mathrm{i}\frac{\omega\cdot d}{c_2}\Big) \tag{2.40}$$

due to the film's finite thickness. Thus, the amplitudes are given by

$$\begin{aligned}
\vec{\mathrm{E}}_{r,l} &= e_{r,l} \cdot (\Delta\phi)^{2l} \cdot \vec{\mathrm{E}}_i, & e_{r,l} &= \begin{cases} r_{12}, & l = 0 \\ t_{12}t_{21}r_{23} \cdot (r_{21}r_{23})^{l-1}, & l > 0 \end{cases}, \\
\vec{\mathrm{E}}_{+,l} &= e_{+,l} \cdot (\Delta\phi)^{2l-2} \cdot \vec{\mathrm{E}}_i, & e_{+,l} &= t_{12} \cdot (r_{21}r_{23})^{l-1}, \\
\vec{\mathrm{E}}_{-,l} &= e_{-,l} \cdot (\Delta\phi)^{2l} \cdot \vec{\mathrm{E}}_i, & e_{-,l} &= t_{12}r_{23} \cdot (r_{21}r_{23})^{l-1}, \\
\vec{\mathrm{E}}_{t,l} &= e_{t,l} \cdot (\Delta\phi)^{2l-1} \cdot \vec{\mathrm{E}}_i, & e_{t,l} &= t_{12}t_{23} \cdot (r_{21}r_{23})^{l-1}.
\end{aligned} \tag{2.41}$$

Here, we have reused the definition of the reflection and transmission coefficients from Eqs. (2.35) and (2.36). Once back in the time-domain, the phase shifts lead to additional delay times

$$T = \frac{d}{c_2}. \tag{2.42}$$

The full time-dependent solution for the electric field is finally given by

$$\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \begin{cases} \vec{\mathrm{E}}_i\Big(t - \dfrac{\hat{\mathrm{n}} \cdot \vec{\mathrm{r}}}{c_1}\Big) + \displaystyle\sum_{l=0}^{\infty} e_{r,l}\vec{\mathrm{E}}_i\Big(t + \dfrac{\hat{\mathrm{n}} \cdot \vec{\mathrm{r}}}{c_1} - 2lT\Big) & \text{for} \quad \hat{\mathrm{n}} \cdot \vec{\mathrm{r}} < 0 \\ \displaystyle\sum_{l=1}^{\infty} e_{+,l}\vec{\mathrm{E}}_i\Big(t - \dfrac{\hat{\mathrm{n}} \cdot \vec{\mathrm{r}}}{c_2} - 2(l-1)T\Big) & \\ \quad + \displaystyle\sum_{l=1}^{\infty} e_{-,l}\vec{\mathrm{E}}_i\Big(t + \dfrac{\hat{\mathrm{n}} \cdot \vec{\mathrm{r}}}{c_2} - 2lT\Big) & \text{for} \quad 0 < \hat{\mathrm{n}} \cdot \vec{\mathrm{r}} < d \\ \displaystyle\sum_{l=1}^{\infty} e_{t,l}\vec{\mathrm{E}}_i\Big(t - \dfrac{\hat{\mathrm{n}} \cdot \vec{\mathrm{r}} - d}{c_3} - (2l-1)T\Big) & \text{for} \quad \hat{\mathrm{n}} \cdot \vec{\mathrm{r}} > d \end{cases}. \tag{2.43}$$

Please note that the sums over l run up to infinity. For practical considerations, it is necessary to truncate the sums when the desired accuracy is reached. The magnetic field is readily derived from the individual propagating waves using Eqs. (2.30) and (2.32).

2.3 Material Dispersion in the Time-Domain

As already pointed out in section 2.1.1, dispersion is a common phenomenon in nature. In a prism, for example, an incident ray of white light is decomposed into its spectral components, since each colour, i.e., frequency, experiences a slightly different refractive index. In the context of this thesis, the dispersion of metals in the visible and near-infrared spectrum is particularly relevant.

Unfortunately, including dispersion in the time-domain is not as straightforward as one would hope. Let us recall Eqs. (2.5) in dimensionless units. For isotropic, dispersive materials, we get the constitutive relations

$$\begin{aligned}
\vec{\mathrm{D}}(\vec{\mathrm{r}},\, \omega) &= \epsilon(\vec{\mathrm{r}},\, \omega) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega), \\
\vec{\mathrm{B}}(\vec{\mathrm{r}},\, \omega) &= \mu(\vec{\mathrm{r}},\, \omega) \cdot \vec{\mathrm{H}}(\vec{\mathrm{r}},\, \omega).
\end{aligned} \tag{2.44}$$

In order to obtain a time-dependent displacement field $\vec{\mathrm{D}}(\vec{\mathrm{r}},\, t)$, we apply the Fourier transform (2.4) to $\vec{\mathrm{D}}(\vec{\mathrm{r}},\, \omega)$, which results in

$$\vec{\mathrm{D}}(\vec{\mathrm{r}},\, t) = \int_{-\infty}^{t} \epsilon(\vec{\mathrm{r}},\, t-\tau) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}},\, \tau)\, \mathrm{d}\tau. \tag{2.45}$$

Inserting this convolution integral (and a similar result for $\vec{\mathrm{B}}(\vec{\mathrm{r}},\, t)$) into Maxwell's equations (2.1) yields a complicated system of coupled integro-differential equations. Such systems are notoriously hard to solve. In the upcoming sections we will establish techniques to circumvent the convolution integral (2.45).

2.3.1 Auxiliary Differential Equations

As we have seen in section 2.1.3, it is sufficient to consider Maxwell's curl equations only. This statement also holds for dispersive materials. In their original form, i.e., Eqs. (2.1c) and (2.1d), the electric displacement only occurs in combination with a temporal derivative. Hence, our goal will be to find an expression for

$$\partial_t \vec{\mathrm{D}}(t). \tag{2.46}$$

Here and in the following, we omit the spatial dependence of $\vec{\mathrm{D}}$ to emphasise the time-dependence. Fourier-transforming the derivative with respect to time leads to

$$\begin{aligned} \partial_t \vec{\mathrm{D}}(t) &= \frac{1}{2\pi} \partial_t \int_{-\infty}^{\infty} \vec{\mathrm{D}}(\omega) \cdot \exp(-\mathrm{i}\omega t)\, \mathrm{d}\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} -\mathrm{i}\omega \vec{\mathrm{D}}(\omega) \cdot \exp(-\mathrm{i}\omega t)\, \mathrm{d}\omega. \end{aligned} \tag{2.47}$$

To summarise the last equation and Eq. (2.4), we have two rules for performing a Fourier transform:

$$\begin{aligned} \vec{\mathrm{D}}(t) \quad &\overset{\mathrm{FT}}{\longleftrightarrow} \quad \vec{\mathrm{D}}(\omega), \\ \partial_t \vec{\mathrm{D}}(t) \quad &\overset{\mathrm{FT}}{\longleftrightarrow} \quad -\mathrm{i}\omega \vec{\mathrm{D}}(\omega). \end{aligned} \tag{2.48}$$

These rules motivate us to find a frequency-domain expression for $-\mathrm{i}\omega\vec{\mathrm{D}}(\omega)$, which is easily transferred into the time-domain.

As a first step, we decompose the frequency-dependent permittivity into

$$\epsilon(\omega) = \epsilon_\infty + \chi(\omega),$$

where ϵ_∞ is a constant, i.e., frequency-independent background permittivity and $\chi(\omega)$ is the frequency-dependent susceptibility. Inserting this relation in (2.44) and multiplying by $-\mathrm{i}\omega$ leads to

$$-\mathrm{i}\omega\vec{\mathrm{D}}(\omega) = -\mathrm{i}\omega\epsilon_\infty \cdot \vec{\mathrm{E}}(\omega) - \mathrm{i}\omega\chi(\omega) \cdot \vec{\mathrm{E}}(\omega). \tag{2.49}$$

This can be formulated as a system of two equations

$$-\mathrm{i}\omega\vec{\mathrm{D}}(\omega) = -\mathrm{i}\omega\epsilon_\infty \cdot \vec{\mathrm{E}}(\omega) + \vec{\mathrm{j}}_p(\omega), \tag{2.50a}$$

$$\vec{\mathrm{j}}_p(\omega) = -\mathrm{i}\omega\chi(\omega) \cdot \vec{\mathrm{E}}(\omega). \tag{2.50b}$$

Here, we have defined an auxiliary field $\vec{\mathrm{j}}_p(\omega)$ which represents a polarisation current. Transforming (2.50a) into the time-domain yields

$$\partial_t\vec{\mathrm{D}}(t) = \epsilon_\infty\partial_t\vec{\mathrm{E}}(t) + \vec{\mathrm{j}}_p(t). \tag{2.51}$$

This expression is easily inserted into Maxwell's curl equations:

$$\partial_t\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \epsilon_\infty^{-1}(\vec{\mathrm{r}}) \cdot \Big(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{j}}_p(\vec{\mathrm{r}},\, t)\Big). \tag{2.52}$$

Please recall that $\vec{\mathrm{j}}$ represents an external current source which should not be confused with the polarisation current $\vec{\mathrm{j}}_p$. As opposed to Eq. (2.45), Eq. (2.52) does not include a convolution, but an auxiliary field instead. Obviously, the time-evolution of the polarisation current is governed by the Fourier transform of Eq. (2.50b). Consequently, Maxwell's curl equations are accompanied by an auxiliary differential equation (ADE) for $\vec{\mathrm{j}}_p(t)$.

Though we have successfully avoided a convolution integral in the curl equations themselves, the problem appears to have been shifted to the polarisation current instead. As it turns out, however, conveniently chosen susceptibility models, i.e., rational functions with respect to $\mathrm{i}\omega$, lead to simple ADEs for $\vec{\mathrm{j}}_p(t)$ (and probably additional auxiliary fields). The following sections will introduce two susceptibility models of considerable relevance for the modelling of metals in the visible and near-infrared spectrum.

2.3.2 The Drude Model

In a first approximation originally proposed by Drude, metals can be considered as a uniform distribution of identical atoms [49]. These atoms consist of heavy, quasi-immobile nuclei and a number of electrons. Of the latter, most are bound to the nuclei and, consequently, can be considered quasi-immobile as well. The remaining valence electrons, however, are only weakly bound. Furthermore, we assume that the valence electrons interact with neither the other electrons nor the ions formed by the nuclei and the bound electrons. Thus, metals are described as a free electron gas on a jellium background.

In the frequency domain, the equation of motion for a valence electron subject to an external electric field is given by

$$-\mathrm{i}\omega\vec{\mathrm{p}}(\omega) + \gamma_D\vec{\mathrm{p}}(\omega) = -e \cdot \vec{\mathrm{E}}(\omega). \tag{2.53}$$

Here, $\vec{\mathrm{p}}$ is the electron's momentum, $-e$ its negative electric charge, and γ_D a phenomenological collision frequency. Obviously, this leads to the solution

$$\vec{\mathrm{p}}(\omega) = \frac{-e}{\gamma_D - \mathrm{i}\omega} \cdot \vec{\mathrm{E}}(\omega). \tag{2.54}$$

Since all valence electrons are subject to the same electric field, their collective movement leads to an electric charge current

$$\vec{\mathrm{j}}_p(\omega) = -en \cdot \frac{\vec{\mathrm{p}}(\omega)}{m_e}, \tag{2.55}$$

where n is the density of valence electrons and m_e is their effective mass. After inserting the previous result for $\vec{\mathrm{p}}(\omega)$, we obtain

$$\vec{\mathrm{j}}_p(\omega) = \sigma(\omega) \cdot \vec{\mathrm{E}}(\omega) \quad \text{with} \quad \sigma(\omega) = \frac{e^2 n}{m_e \cdot (\gamma_D - \mathrm{i}\omega)}. \tag{2.56}$$

The conductivity $\sigma(\omega)$ is related to the electric susceptibility via

$$\chi_D(\omega) = -\frac{\sigma(\omega)}{\mathrm{i}\omega} = -\frac{e^2 n}{m_e \cdot \omega(\omega + \mathrm{i}\gamma_D)} = -\frac{\omega_D^2}{\omega(\omega + \mathrm{i}\gamma_D)}. \tag{2.57}$$

In the last step, we have introduced the plasma frequency

$$\omega_D = \sqrt{\frac{e^2 n}{m_e}}. \tag{2.58}$$

Please note that this quantity is also given in the unit system presented in section 2.1.4. As such, conversion to the SI system obeys the conversion rules compiled in Table 2.1.

Equation (2.57) describes the response of a Drude metal to an externally applied harmonic electric field. Though the Drude model ignores the solid foundation of quantum mechanics, it provides a surprisingly good description for a wide range of metals in the near-infrared region. In this frequency regime, photon energies are too low to excite interband transitions. Hence, free electrons are a rather good approximation. In practice, it is common to consider ω_D and γ_D as free parameters, which can be chosen to provide the best fit of the permittivity[3] to a set of experimental data in a given spectral range.

Let us now return to the problem of finding the time-dependence of the polarisation current

$$\vec{\mathrm{j}}_D(\omega) = -\mathrm{i}\omega\chi_D(\omega) \cdot \vec{\mathrm{E}}(\omega). \tag{2.59}$$

Please note that we have renamed $\vec{\mathrm{j}}_p$ to $\vec{\mathrm{j}}_D$ to stress that this is the polarisation current associated with the Drude model. Inserting (2.57) into (2.59) yields

$$-\mathrm{i}\omega\vec{\mathrm{j}}_D(\omega) = \omega_D^2 \vec{\mathrm{E}}(\omega) - \gamma_D \vec{\mathrm{j}}_D(\omega)$$

after a few algebraic manipulations. Finally, the time-domain equivalent of this equations follows immediately from the Fourier transform rules (2.48):

$$\partial_t \vec{\mathrm{j}}_D(t) = \omega_D^2 \vec{\mathrm{E}}(t) - \gamma_D \vec{\mathrm{j}}_D(t). \tag{2.60}$$

The presence of a single Drude pole in the susceptibility leads to an auxiliary differential equation for the associated polarisation current $\vec{\mathrm{j}}_D(t)$. It is straightforward to extend the susceptibility by adding additional Drude poles. For each pole, however, we incur the cost of an additional ADE.

[3] Alternatively, of the susceptibility or the refractive index

2.3.3 The Drude-Lorentz Model

The Drude model is a fairly good description of metals in the near-infrared region. For higher frequencies in the visible regime, however, an externally applied electric field triggers interband transitions [49].

To improve on the description of the permittivity for shorter wavelengths, a common ansatz is to accompany the Drude model with Lorentz oscillators [21, 46, 50]. Again, we start with the equation of motion for an electron in an external electric field:

$$-\mathrm{i}\omega\vec{\mathrm{p}}(\omega) + \gamma_L\vec{\mathrm{p}}(\omega) - \frac{\omega_L^2}{\mathrm{i}\omega}\cdot\vec{\mathrm{p}}(\omega) = -e\vec{\mathrm{E}}(\omega). \tag{2.61}$$

In addition to a damping coefficient γ_L as in the Drude model, we have also introduced the resonance frequency ω_L of the undriven Lorentz oscillator. Following the already familiar path we obtain

$$\chi_L(\omega) = \frac{\Delta\epsilon_L\cdot\omega_L^2}{\omega_L^2 - \mathrm{i}\gamma_L\omega - \omega^2} \tag{2.62}$$

as the susceptibility associated with the Lorentz model. Herein, we have defined the oscillator strength

$$\Delta\epsilon_L = \frac{\omega_D^2}{\omega_L^2} \qquad \left(\text{with } \omega_D^2 \equiv \frac{e^2 n}{m_e}\right). \tag{2.63}$$

Although ω_D shares its formal definition with the Drude model's plasma frequency (2.58), it should be acknowledged that ω_D is usually considered a fit parameter. Thus, replacing ω_D by $\Delta\epsilon_L$ highlights that it presents an independent fit parameter of the *Lorentz* model as well. Thus, the Lorentz model features a total of three parameters ω_L, γ_L, and $\Delta\epsilon_L$, which can be used to model dispersion as accurately as possible.

Since Eq. (2.62) is a rational function in $\mathrm{i}\omega$, it easily translates into the time-domain. Inserting (2.62) into (2.50b), one obtains

$$-\mathrm{i}\omega\vec{\mathrm{j}}_L(\omega) = \Delta\epsilon_L\omega_L^2\vec{\mathrm{E}}(\omega) + \vec{\mathrm{q}}_L(\omega), \tag{2.64}$$

where we have introduced a second auxiliary field

$$\vec{\mathrm{q}}_L(\omega) = -\frac{\omega_L^2\vec{\mathrm{j}}_L(\omega) + \gamma_L\Delta\epsilon_L\omega_L^2\vec{\mathrm{E}}(\omega)}{\gamma_L - \mathrm{i}\omega}. \tag{2.65}$$

A few straightforward algebraic manipulations yield

$$-\mathrm{i}\omega\vec{\mathrm{q}}_L(\omega) = -\omega_L^2\vec{\mathrm{j}}_L(\omega) - \gamma_L\Delta\epsilon_L\omega_L^2\vec{\mathrm{E}}(\omega) - \gamma_L\vec{\mathrm{q}}_L(\omega). \tag{2.66}$$

Again, applying the Fourier transform rules (2.48) yields the time-domain formulation

$$\begin{aligned}\partial_t\vec{\mathrm{j}}_L(t) &= \Delta\epsilon_L\omega_L^2\vec{\mathrm{E}}(t) + \vec{\mathrm{q}}_L(t)\\ \partial_t\vec{\mathrm{q}}_L(t) &= -\omega_L^2\vec{\mathrm{j}}_L(t) - \gamma_L\Delta\epsilon_L\omega_L^2\vec{\mathrm{E}}(t) - \gamma_L\vec{\mathrm{q}}_L(t).\end{aligned} \tag{2.67}$$

In contrast to the Drude model, the Lorentz model requires *two* auxiliary fields $\vec{\mathrm{j}}_L$ and $\vec{\mathrm{q}}_L$. Both equations have to be solved in parallel to Maxwell's curl equations.

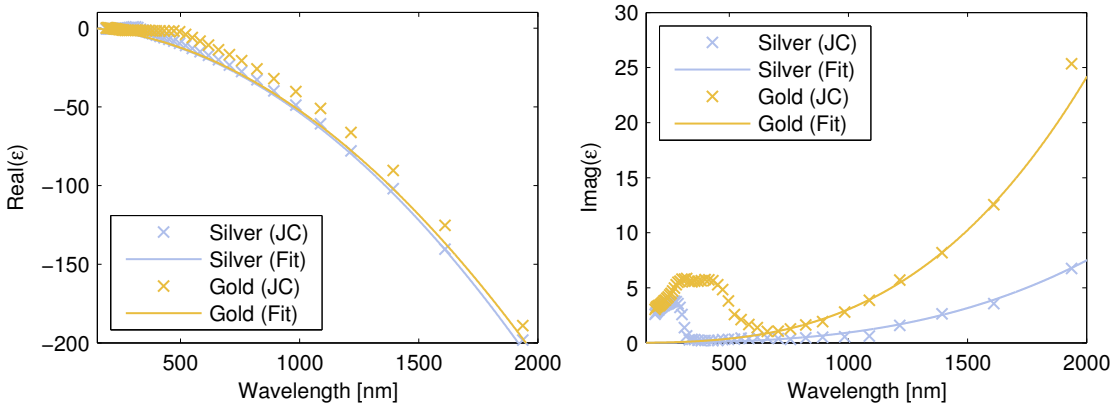


Figure 2.4: *Permittivity of silver and gold. The plots show the real (left panel) and imaginary parts (right panel) of ϵ. The crosses are measured data points taken from Ref. [45]. The solid curves are the results of Drude fits presented in the same publication. For wavelengths in the near-infrared region, i.e., above* $1000\,\mathrm{nm}$*, the Drude model approximates the dispersion of both materials rather well. A close-up of both figures for lower wavelengths is given in Fig.* 2.5.

Finally, we can combine an arbitrary number of both Drude and Lorentz poles to the Drude-Lorentz model, for which the permittivity is given by

$$\epsilon(\omega) = \epsilon_\infty - \sum_{i=1}^{N_D} \frac{\omega_{D,i}^2}{\omega(\omega + \mathrm{i}\gamma_{D,i})} + \sum_{j=1}^{N_L} \frac{\Delta\epsilon_{L,j} \cdot \omega_{L,j}^2}{\omega_{L,j}^2 - \mathrm{i}\gamma_{L,j}\omega - \omega^2}. \tag{2.68}$$

For each of the N_D Drude poles, we have to specify two fit parameters and introduce one auxiliary field $\vec{\mathrm{j}}_{D,i}$. Similarly, each of the N_L Lorentz poles requires two ADEs for $\vec{\mathrm{j}}_{L,j}$ and $\vec{\mathrm{q}}_{L,j}$, which incorporate three fit parameters.

2.3.4 Material Parameters

Quite a few publications have dealt with the experimental acquisition of material parameters for a wide range of metals. Johnson and Christy tabularise the optical properties of gold, silver, and copper for various photon energies in the optical and near-infrared regime [45]. They also provide fit parameters for Drude models commonly used in the literature. Figures 2.4 and 2.5 show measured values of ϵ and corresponding Drude fits for silver and gold. Ordal *et al.* compile tables for a number of metals, including non-noble metals such as aluminium [51]. Fit parameters can also be found in their paper. Vial *et al.* improve on the simple Drude model for gold using an additional Lorentz term [46]. They report a much better fit in the visible spectrum, especially for the imaginary part of the permittivity.

Results in later chapters mostly involve resonances of gold nanostructures in the near-infrared regime. Consequently, we utilise a simple Drude model with parameters according to Johnson and Christy to model the dispersion of gold. This parameter set is widely used in the literature. From

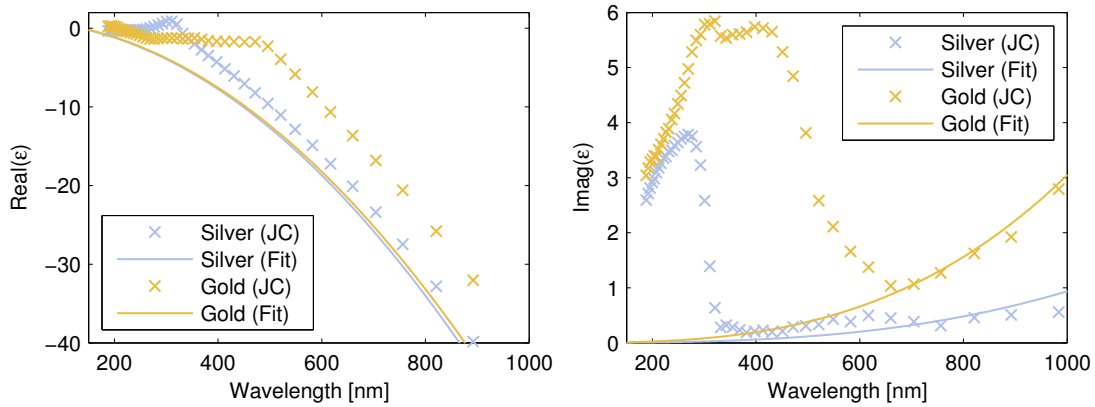


Figure 2.5: *Permittivity of silver and gold. For an explanation of the plots please refer to Fig.* 2.4. *In this zoomed view, it is apparent that a simple Drude model is not sufficient to capture the intricate dispersion of metals in the visible frequency regime. In particular, the imaginary part shows significant deviations due to inter-band transitions.*

their paper, we extract the values

$$\epsilon_\infty = 1, \quad \omega_D = 1.3780 \cdot 10^{16}\,\mathrm{s}^{-1}, \quad \text{and} \quad \gamma_D = 1.0753 \cdot 10^{14}\,\mathrm{s}^{-1}. \tag{2.69}$$

2.4 Elements of Electromagnetic Scattering

One of the most important notions in physics is the concept of energy. Thus, it is of little surprise that energy—more precisely, the transport of energy—is of utmost importance for nanophotonics as well. The transport of electromagnetic energy for all times t and positions $\vec{\mathrm{r}}$ is given by the Poynting vector [43]

$$\vec{\mathrm{S}}(\vec{\mathrm{r}},\, t) = \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t). \tag{2.70}$$

For experiments, however, this general definition is of little relevance. Spectrometers, for example, decompose an incoming signal into its respective frequency components, and thus measure the energy flux of time-harmonic fields $\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega)$ and $\vec{\mathrm{H}}(\vec{\mathrm{r}},\, \omega)$. Furthermore, conventional detectors cannot resolve the time-dependence of the fields since visible and near-infrared electromagnetic waves oscillate at frequencies of $100\,\mathrm{THz}$ and more [50]. Instead, for time-harmonic fields the *time-averaged* Poynting vector

$$\vec{\mathrm{S}}(\vec{\mathrm{r}},\, \omega) = \frac{1}{2}\Re\Big\{\vec{\mathrm{E}}(\vec{\mathrm{r}},\, \omega) \times \vec{\mathrm{H}}^*(\vec{\mathrm{r}},\, \omega)\Big\} \tag{2.71}$$

is the relevant quantity. Please note that Eq. (2.71) is *not* the Fourier transform of Eq. (2.70).

Once it is clear how the transport of energy can be described, we can apply our knowledge to a small section of the broad topic of electromagnetic scattering [50, 52]. Our primary goal is to analyse how an object of arbitrary shape and material composition interacts with an incident plane

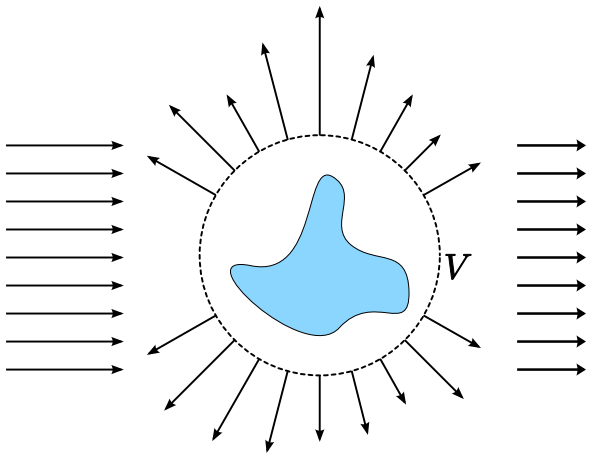


Figure 2.6: *Scattering by an arbitrary object. A fraction of incoming light is scattered into different directions. Integrating the total Poynting vector over the surface of the volume V yields the amount of absorbed energy.*

wave (Fig. 2.6). Outside the scatterer, we decompose the total electromagnetic fields, labelled "tot", into respective incident ("inc") and scattered fields ("scat"):

$$\begin{aligned} \vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}},\,\omega) &= \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}},\,\omega) + \vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega), \\ \vec{\mathrm{H}}_{\mathrm{tot}}(\vec{\mathrm{r}},\,\omega) &= \vec{\mathrm{H}}_{\mathrm{inc}}(\vec{\mathrm{r}},\,\omega) + \vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega). \end{aligned} \tag{2.72}$$

In the absence of scatterers, we would have $\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) = \vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) \equiv 0$. It is highly desirable to summarise the scattering properties of a particle in a small set of numbers. In particular, these numbers should provide an intuitive understanding on how much light is scattered and absorbed by a certain particle.

To this end, we construct a virtual volume V with closed surface ∂V which encloses the scatterer. If we assume the absence of sources in V, electromagnetic energy can only enter through the volume's surface. Thus, the energy

$$W_{\mathrm{abs}}(\omega) = -\int_{\partial V} \vec{\mathrm{S}}_{\mathrm{tot}}(\vec{\mathrm{r}},\,\omega)\cdot\hat{\mathrm{n}}(\vec{\mathrm{r}})\,\mathrm{d}^2 r \tag{2.73}$$

represents energy "lost" within the volume. Herein, we have used the total Poynting vector $\vec{\mathrm{S}}_{\mathrm{tot}}$, which is constructed from the total electromagnetic fields. Moreover, $\hat{\mathrm{n}}$ represents an outwardly directed unit normal with respect to ∂V. The minus sign is a matter of convention; if energy enters the volume, we have $\vec{\mathrm{S}}_{\mathrm{tot}}\cdot\hat{\mathrm{n}} < 0$. Hence, a positive value of W_{abs} indicates energy has entered, but not left the volume. For a lossless medium surrounding the scatterer, W_{abs} can only be attributed to absorption in the scatterer.

Similarly, we analyse the scattered portion of the electromagnetic fields. Thus, the energy scattered by an object is given by

$$W_{\mathrm{scat}}(\omega) = +\int_{\partial V} \vec{\mathrm{S}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega)\cdot\hat{\mathrm{n}}(\vec{\mathrm{r}})\,\mathrm{d}^2 r. \tag{2.74}$$

As compared to Eq. (2.73), the Poynting vector in Eq. (2.74) merely comprises *scattered* fields, i.e.,

$$\vec{\mathrm{S}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) = \frac{1}{2}\Re\Big\{\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) \times \vec{\mathrm{H}}^{*}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega)\Big\}. \tag{2.75}$$

Furthermore, the sign has changed with respect to Eq. (2.73). If scattered fields emerge from the volume, we have $\vec{\mathrm{S}}_{\mathrm{scat}} \cdot \hat{\mathrm{n}} > 0$. Consequently, positive values of W_{scat} indicate outgoing scattered waves. Since $\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) = \vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) \equiv 0$ in the absence of scatterers, those waves must be attributed to the object enclosed by V.

So far, both W_{abs} and W_{scat} depend on the amplitude of the incident plane wave. To eliminate this, we can normalise both quantities by the incident irradiance, which is the magnitude of the Poynting vector of the incident wave. Thus, we obtain the absorption cross section

$$C_{\mathrm{abs}} = -\frac{1}{|\vec{\mathrm{S}}_{\mathrm{inc}}(\omega)|} \cdot \int_{\partial V} \vec{\mathrm{S}}_{\mathrm{tot}}(\vec{\mathrm{r}},\,\omega) \cdot \hat{\mathrm{n}}(\vec{\mathrm{r}})\,\mathrm{d}^2 r \tag{2.76}$$

and the scattering cross section

$$C_{\mathrm{scat}} = \frac{1}{|\vec{\mathrm{S}}_{\mathrm{inc}}(\omega)|} \cdot \int_{\partial V} \vec{\mathrm{S}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,\omega) \cdot \hat{\mathrm{n}}(\vec{\mathrm{r}})\,\mathrm{d}^2 r. \tag{2.77}$$

Finally, the extinction cross section

$$C_{\mathrm{ext}} = C_{\mathrm{abs}} + C_{\mathrm{scat}} \tag{2.78}$$

unites the effects of both absorption and scattering. Intuitively, the extinction cross section can be understood in terms of geometrical optics: There, it corresponds to the size of the shadow cast by a perfectly intransparent obstacle. In the regime of wave optics, however, the "shadow" can be significantly different—both larger and smaller—from the geometrical cross section [50].

Chapter 3

3 Spatial Discretisation via the Discontinuous Galerkin Approach

Maxwell's curl equations (2.18), introduced in the previous chapter, form a system of partial differential equations. Such equations are inherently difficult to solve with conventional analytical tools. Although solutions to a number of (usually highly symmetric) electromagnetic problems are available, they often require considerable algebraic effort and involve special functions. For just a little more complicated problems we have to rely on a numerical treatment of the curl equations.

To this end, we have to develop a method to represent these equations in a computer. In this chapter, we present the Discontinuous Galerkin (DG) method as an attractive technique to discretise the spatial dependence of the electromagnetic fields. The time-dependence of the electromagnetic fields is covered later in chapters 4 and 5.

Section 3.1 familiarises the reader with the basic concepts behind the DG method. Technicalities which have been omitted to avoid unnecessary distraction are subsequently presented in section 3.2. Besides some derivations, these also include important details on the actual implementation. This chapter concludes with a discussion of DG's key advantages and shortcomings in section 3.3.

3.1 An Illustrated Tour of the Discontinuous Galerkin Method

We start with a quick overview of the DG method [27, 28]. Maxwell's curl equations, rewritten in a special form (section 3.1.1), are solved on a finite part of the physical volume, the computational domain. Moreover, the latter is decomposed into a set of mutually distinct elements (section 3.1.2).

On each of these elements, we require the residuum of the numerical solution to be orthogonal to a set of test functions (section 3.1.3). Most importantly, orthogonality is defined in a local sense, i.e., without involving numerical solutions on neighbouring elements. At some point, however, the respective neighbours must interact with an element to allow for wave propagation across element boundaries. This is achieved by introducing the numerical flux (section 3.1.4), which also provides a convenient means to enforce boundary conditions (section 3.1.5).

Once we represent the electromagnetic fields in terms of an expansion basis (section 3.1.6), it is just a few simple steps to obtain the semi-discrete form of Maxwell's curl equations (section 3.1.7).

In this form, all spatial dependencies have been reduced to a set of (time-dependent) expansion coefficients and corresponding matrix operators.

3.1.1 Maxwell's Equations in Conservation Form

As has already been pointed out in section 2.1.3, the time-evolution of the electromagnetic fields is governed by Maxwell's curl equations. For convenience, they shall be repeated here for isotropic, dispersionless materials in the unit system introduced in section 2.1.4:

$$\partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \quad \epsilon^{-1}(\vec{\mathrm{r}}) \cdot \Big(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{j}}(\vec{\mathrm{r}},\, t) \Big), \tag{2.18a}$$

$$\partial_t \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) = -\mu^{-1}(\vec{\mathrm{r}}) \cdot \vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t). \tag{2.18b}$$

For the time being, we want to restrict ourselves to source-free regions, i.e., $\vec{\mathrm{j}}(\vec{\mathrm{r}},\, t) \equiv 0$. A thorough discussion of sources is topic of section 6.1.

A series of definitions allows us to rewrite the curl equations in a form more accessible to the following derivation of the discontinuous Galerkin method. To this end, we introduce the material matrix $\mathcal{Q}$ and the state vector $\mathbf{q}$ as [28]

$$\mathcal{Q}(\vec{\mathrm{r}}) = \begin{pmatrix} \epsilon(\vec{\mathrm{r}}) & 0 \\ 0 & \mu(\vec{\mathrm{r}}) \end{pmatrix}, \quad \mathbf{q}(\vec{\mathrm{r}},\, t) = \begin{pmatrix} \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) \\ \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) \end{pmatrix}. \tag{3.1}$$

Furthermore, we define the flux

$$\vec{\mathrm{F}} = \Big(\mathbf{F}_x,\, \mathbf{F}_y,\, \mathbf{F}_z \Big)^T, \quad \mathbf{F}_i(\mathbf{q}) = \begin{pmatrix} -\hat{\mathrm{e}}_i \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) \\ \hat{\mathrm{e}}_i \times \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) \end{pmatrix}, \tag{3.2}$$

where $\hat{\mathrm{e}}_x$, $\hat{\mathrm{e}}_y$, and $\hat{\mathrm{e}}_z$ represent Cartesian unit vectors. Please note that the state vector $\mathbf{q}$ represents the three components of both electromagnetic fields and, thus, has six components. The flux vector $\vec{\mathrm{F}}$ has three components (x, y, and z), where each component is a state vector of six components itself. This notation seems confusing at first glance, but turns out to be convenient in the following. Table 3.1 shows the vector notations used in this thesis. With these definitions in place, it is possible to reformulate Eq. (2.18) as the conservation law

$$\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}) = 0. \tag{3.3}$$

All efforts in the proceeding sections will be directed to discretise the spatial dependence of this equation.

3.1.2 Tesselation of the Computational Domain

The DG method is a volume method, which means that we sample field values all over the simulation volume, also called the computational domain. Due to the limited amount of available memory, this volume must be of finite extent. Obviously, the computational domain should capture the essential features of the physical setup under investigation. For example, a scattering setup in the computer

Notation	Meaning
$\hat{\mathrm{e}}_i$	Cartesian unit vector in i-direction ($i = x,\ y,\ z$)
$\hat{\mathrm{n}}$	Outwardly directed normal vector of unit length
$\vec{\mathrm{a}}$	Physical vector with x-, y-, and z-components
$\mathbf{a}$	Physical state vector with more than three components
$\tilde{\mathrm{a}}$	Vector of expansion coefficients for a single field component
$\tilde{\vec{\mathrm{a}}}$	A physical vector with x-, y-, and z-components, where each component is a vector of expansion coefficients $\tilde{\mathrm{a}}_x$, $\tilde{\mathrm{a}}_y$, and $\tilde{\mathrm{a}}_z$
$\tilde{\mathbf{a}}$	A physical state vector with more than three components, where each component is a vector of expansion coefficients

Table 3.1: *Various types of vectors used throughout this thesis.*

includes the scatterer itself plus its immediate environment. The remainder of the physical system, i.e., the part which is not contained within the computational domain, must be accounted for by suitable boundary conditions. A thorough discussion of boundary conditions is presented in section 3.1.5.

Depending on the dimensionality of the system, the computational domain is then decomposed into mutually distinct elements which make up the mesh. For one-dimensional simulations, these elements would be line segments. In two and three dimensions, the standard choices of triangles and tetrahedrons, respectively, lead to efficient methods. Other element types such as quadrilaterals, hexahedrons [53], prisms, pyramids, or meshes with multiple types are possible [54] and sometimes advantageous. However, in this thesis we will only consider conforming triangular and tetrahedral meshes.

Either way, those elements need not be equally sized or shaped and, thus, form an unstructured mesh. This considerable freedom can be exploited to resolve spatial features of the computational domain with as many elements as needed to achieve a desired level of accuracy, while large homogeneous regions can be modelled with fewer elements. In Figs. 3.1 to 3.3 we show one-, two-, and three-dimensional examples of meshes for a couple of systems together with sketches of typical elements.

The task of creating a mesh for a given geometric setup is a vast research field of its own. Luckily, there are quite a few meshing tools available. Among the more popular open source tools we find `NETGEN` [55], `TetGen` [56], and `Gmsh` [57]. Virtually all meshes for the computations in this thesis have been created using `NETGEN`; The remaining ones have been created by custom `MATLAB` scripts to enforce certain mesh symmetries.

3.1.3 Working on Single Elements

Let us now consider a single element $\triangle$ of the computational domain. Our goal is to find a numerical approximation $\mathbf{q}_N$ to the correct (and generally unknown) solution $\mathbf{q}$ of Maxwell's equations in

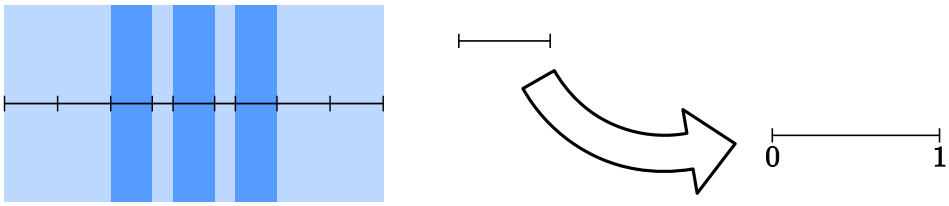


Figure 3.1: *Tesselation of a one-dimensional physical system, in this case a series of Bragg layers. The colours represent different refractive indices. Please note that the elements which make up the various layers are of different sizes. The right-hand side shows a zoomed view of one of these elements, which is mapped to a reference element via a linear transformation (see section* 3.2.3*).*

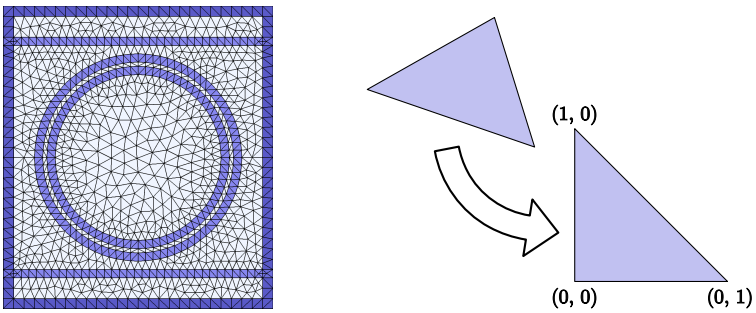


Figure 3.2: *Tesselation of two-dimensional waveguides coupled to a slotted microresonator. The colours represent different material parameters. The right-hand side illustrates the affine transform which maps a single element of the mesh to a reference triangle (see section* 3.2.3*).*

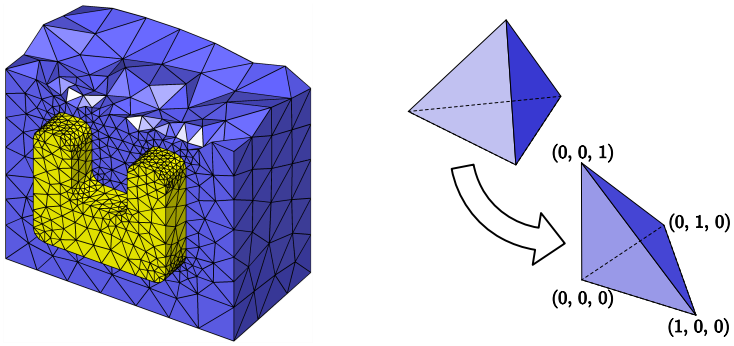


Figure 3.3: *Tesselation of a three-dimensional system which comprises a split-ring resonator (see Fig.* 1.1 *in the introduction) on a substrate. The flexibility of the mesh generation process allows for an accurate modelling of corners smoothed by the fabrication process. The right-hand panel shows one of the elements of the mesh and its relation to a reference tetrahedron (see section* 3.2.3*).*

conservation form. In general, Eq. (3.3) will not hold exactly for $\mathbf{q}_N$, but will have to be modified to

$$\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_N) = \mathrm{res}, \tag{3.4}$$

where res is the residuum of the equation. While ideally the residuum would be zero, this is usually not the case and we have to find—and define—the best approximation.

Within a given, finite, linear function space, there exists an optimal function $\mathbf{q}_N$ minimising the residuum, which then will be orthogonal to the function space. Let $L_i(\vec{\mathrm{r}})$ be a scalar basis function of said function space. Thus, orthogonality between the residuum and the function space is locally expressed via the vanishing scalar product on the element $\triangle$ with volume $V_\triangle$:

$$\int_{V_\triangle} \Big(\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_N) \Big) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r = \int_{V_\triangle} \mathrm{res} \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r \stackrel{!}{=} 0. \tag{3.5}$$

The numerical solution on the element $\triangle$ must satisfy Eq. (3.5) for *all* test functions L_i. The *global* solution must satisfy Eq. (3.5) on *all* elements.

At this point it should be noted that Eq. (3.5) is a completely local statement. It only involves field values and derivatives on the element $\triangle$. From physical reasoning it is clear that we cannot assume to get a correct solution of our problem from Eq. (3.5) alone, because Maxwell's equations are formulated in a continuous space, not in a segmented one. Electromagnetic waves propagate and, thus, must leave an element and enter the neighbouring one. We will account for this in the next section.

3.1.4 Connecting Elements via the Numerical Flux

So far, we have completely ignored the presence of neighbouring elements. However, solutions to Maxwell's equations must obey the continuity conditions. In particular, the tangential components of $\vec{\mathrm{E}}$ and $\vec{\mathrm{H}}$ must be continuous across material interfaces, while the respective normal components exhibit well-defined jumps. This leaves us with two options.

1. We can define boundary conditions on each element. The boundary conditions depend on the field values in neighbouring elements.
2. We can absorb the boundary conditions into modifications of the physical equations. These modifications depend on fields in neighbouring elements as well.

For our purposes we choose the second option. Acknowledging that the coupling to neighbouring elements must be achieved across the element boundary, we integrate (3.5) by parts and obtain

$$\int_{V_\triangle} \Big(\mathcal{Q}(\vec{\mathrm{r}}) \partial_t \mathbf{q}_N(\vec{\mathrm{r}},\, t) \cdot L_i(\vec{\mathrm{r}}) - \vec{\mathrm{F}}(\mathbf{q}_N) \cdot \vec{\nabla} L_i(\vec{\mathrm{r}}) \Big)\, \mathrm{d}^3 r = - \int_{\partial V_\triangle} \Big(\hat{\mathrm{n}} \cdot \vec{\mathrm{F}}(\mathbf{q}_N) \Big) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^2 r. \tag{3.6}$$

In the right-hand side's integral over the element's surface $\partial V_\triangle$, $\hat{\mathrm{n}}$ represents the corresponding outwardly directed normal vector of unit length.

In the spirit of option 2 stated above, we will now replace the flux $\vec{\mathrm{F}}$ on the right-hand side of this equation by a coupling term $\vec{\mathrm{F}}^*$, commonly referred to as the numerical flux, which we will have to define later. This procedure, though seemingly random, allows us to reintroduce the coupling

to neighbouring elements previously eliminated by defining the orthogonality in an element-wise fashion. Undoing the integration by parts then yields

$$\begin{aligned}\int_{V_\triangle} \left(\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_N)\right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r \\ = \int_{\partial V_\triangle} \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}(\mathbf{q}_N) - \vec{\mathrm{F}}^*(\mathbf{q}_N)\right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^2 r.\end{aligned} \tag{3.7}$$

This is the strong variational formulation of Maxwell's curl equations [27].

At this point, it is worthwhile to mention that the left-hand side of Eq. (3.7) is still a local statement, and thus entirely confined to the element $\triangle$. The evaluation of the right-hand side requires both the flux and the numerical flux on the element's boundary. As hinted at above, $\vec{\mathrm{F}}^*$ is used to establish coupling to—and hence must incorporate field values from—neighbouring elements. As a consequence, the right-hand side is a non-local statement.

Hitherto, we have postponed to specify how the numerical flux is actually defined. Given that it mediates coupling between neighbouring elements, it is fairly obvious that the numerical flux $\vec{\mathrm{F}}^*$ is essential for correctness and convergence of the scheme. Most interestingly, however, its choice is not unique. Hesthaven and Warburton have shown [28] that—for nodal schemes as presented later in section 3.1.6—a particular choice called upwind flux leads to a numerically stable and convergent scheme. It is given by

$$\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}(\mathbf{q}_N) - \vec{\mathrm{F}}^*(\mathbf{q}_N)\right) = \begin{pmatrix} \dfrac{1}{\bar{Z}} \left(\alpha \left[\Delta\vec{\mathrm{E}} - \hat{\mathrm{n}}(\hat{\mathrm{n}} \cdot \Delta\vec{\mathrm{E}})\right] + Z^+ \hat{\mathrm{n}} \times \Delta\vec{\mathrm{H}}\right) \\ \dfrac{1}{\bar{Y}} \left(\alpha \left[\Delta\vec{\mathrm{H}} - \hat{\mathrm{n}}(\hat{\mathrm{n}} \cdot \Delta\vec{\mathrm{H}})\right] - Y^+ \hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}\right) \end{pmatrix}. \tag{3.8}$$

Equation (3.8) contains impedances

$$Z^\pm = \sqrt{\frac{\mu^\pm}{\epsilon^\pm}}, \tag{3.9}$$

admittances

$$Y^\pm = \frac{1}{Z^\pm} = \sqrt{\frac{\epsilon^\pm}{\mu^\pm}}, \tag{3.10}$$

and their respective sums

$$\bar{Z} = Z^+ + Z^- \qquad \text{and} \qquad \bar{Y} = Y^+ + Y^-. \tag{3.11}$$

Hence, the numerical flux (3.8) includes material parameters from both the local element (index "$-$") and its neighbour (index "$+$") in the normal direction $\hat{\mathrm{n}}$. Furthermore, it includes the field differences

$$\Delta\vec{\mathrm{E}} = \vec{\mathrm{E}}^+ - \vec{\mathrm{E}}^- \qquad \text{and} \qquad \Delta\vec{\mathrm{H}} = \vec{\mathrm{H}}^+ - \vec{\mathrm{H}}^- \tag{3.12}$$

across the mutual interface of both elements. Stated more clearly, $\vec{\mathrm{E}}^+$ is the limit of the electric field on the interface when approaching from the neighbouring element, while $\vec{\mathrm{E}}^-$ is the limit when

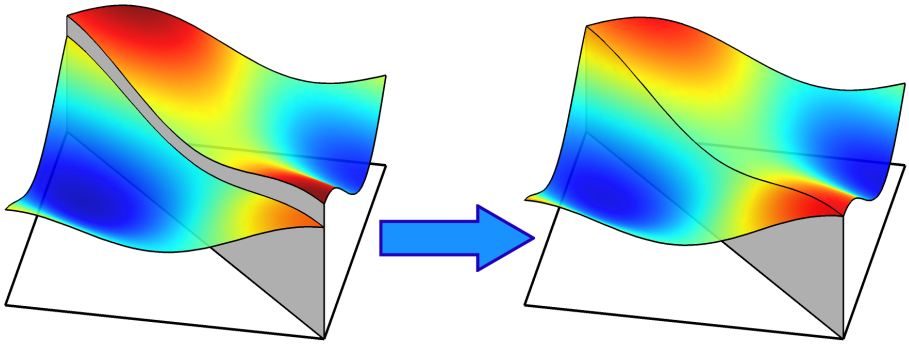


Figure 3.4: *Illustration of the influence of the numerical flux. The panels show two neighbouring elements (triangles) and the component of the electric field tangential to the interface between both elements. The left panel depicts the situation where no numerical flux is present, i.e., where no coupling between the elements exists. As a result, the field distribution on the interface is not unique. The situation changes if one employs the upwind flux given by Eq.* (3.8)*, which penalises jump discontinuities of the tangential fields. As a result, it weakly enforces the continuity of tangential fields across element boundaries, as shown in the right panel.*

coming from the interior of the local element. As both limits are not necessarily identical, $\Delta\vec{\mathrm{E}}$ describes a discontinuity or jump. Such discontinuities naturally occur at material interfaces, where the normal component of the electric field changes according to $\epsilon^{-}\hat{\mathrm{n}}\cdot\vec{\mathrm{E}}^{-} = \epsilon^{+}\hat{\mathrm{n}}\cdot\vec{\mathrm{E}}^{+}$ (section 2.1.2). As a consequence, expression (3.8) weakly enforces that

1. the fields are continuous where they ought to be and
2. satisfy the matching conditions at material interfaces.

This impact of the numerical flux on the electromagnetic fields is schematically illustrated in Fig. 3.4.

We have not yet discussed the upwind parameter α in Eq. (3.8). Any number in $[0,\,1]$ yields a numerically stable and convergent scheme [27]. The value 1 represents the pure upwind flux, whereas we recover the (energy-conserving) central flux for $\alpha = 0$. As it turns out, for a nodal expansion basis as introduced later in section 3.1.6, the upwind flux is preferable because it strongly damps unphysical modes at the expense of strict energy conservation. Thus, the choice of α strongly influences the accuracy of the scheme. Optimal convergence rates are achieved for $\alpha = 1$, while the rates are less clear for other values of α [27, 28]. A derivation of the pure upwind flux is presented in section 3.2.1.

3.1.5 Boundary Conditions

It has already been mentioned that volume methods cannot discretise an infinite space with a finite amount of memory. Hence, the computational domain must be truncated. We need to specify boundary conditions, i.e., how the electromagnetic fields should behave at the interface to the non-simulated part of the universe. The numerical flux allows for an easy inclusion of the most important

Boundary condition	$\Delta\vec{\mathrm{E}}$	$\Delta\vec{\mathrm{H}}$
Perfect electric conductor (PEC)	$-2\vec{\mathrm{E}}^{-}$	0
Perfect magnetic conductor (PMC)	0	$-2\vec{\mathrm{H}}^{-}$
First order absorbing (Silver-Müller)	$-2\vec{\mathrm{E}}^{-}$	$-2\vec{\mathrm{H}}^{-}$

Table 3.2: *Modified field differences for commonly used boundary conditions. For derivations, see Ref. [54]*

boundary conditions, i.e., periodic boundaries, reflecting boundaries, and absorbing boundaries.

Periodic boundary conditions are a special case among the boundary conditions. In contrast to the other conditions presented below, each element on the boundary of the computational domain has a well-defined neighbour. Consequently, we can apply the very same numerical flux as we do for elements in the interior of the computational domain. Implementing this boundary condition is thus reduced to finding correct pairs of neighbours.

The situation is a bit more complicated for actual boundary elements, i.e., elements missing a neighbour. For such elements we have to provide information on the materials and fields in *virtual* neighbouring elements. To do so, we simply set

$$Z^{+} \equiv Z^{-} \qquad \text{and} \qquad Y^{+} \equiv Y^{-} \tag{3.13}$$

and modify the field differences as indicated by Table 3.2. Perfect electric (PEC) and perfect magnetic conductors (PMC) reflect all incident radiation and are used to create cavities or to introduce symmetry planes into the system. Silver-Müller boundary conditions [25,26] mimic an infinite computational domain by partially absorbing outgoing radiation. However, only spherical waves which normally impinge on the (potentially spherical) boundary are (at least theoretically) perfectly absorbed. All other angles of incidence lead to spurious reflections. Thus, to minimise reflections one should place a spherical Silver-Müller boundary sufficiently far away from the radiation source. Still better results can be obtained by adding perfectly matched layers (see chapter 7).

3.1.6 Expanding the Electromagnetic Fields

Equations (3.7) and (3.8) are the strong formulation of Maxwell's curl equations for numerical approximations of the electromagnetic fields. However, the explicit form of the approximation has yet to be specified, which is the task of this section.

To this end, we represent the electromagnetic fields in terms of the previously defined test functions L_i. Using the same function space for both the test functions and the field expansion is called the Galerkin choice. For each field component in each element $\triangle$, we obtain expressions analogous to

$$\mathrm{E}^{\triangle}_{x,\,N}(\vec{\mathrm{r}},\,t) = \sum_{j=1}^{n} \tilde{\mathrm{E}}^{\triangle}_{x,\,j}(t) \cdot L_j(\vec{\mathrm{r}}) \equiv \tilde{\mathrm{E}}^{\triangle}_{x,\,j}(t) \cdot L_j(\vec{\mathrm{r}}). \tag{3.14}$$

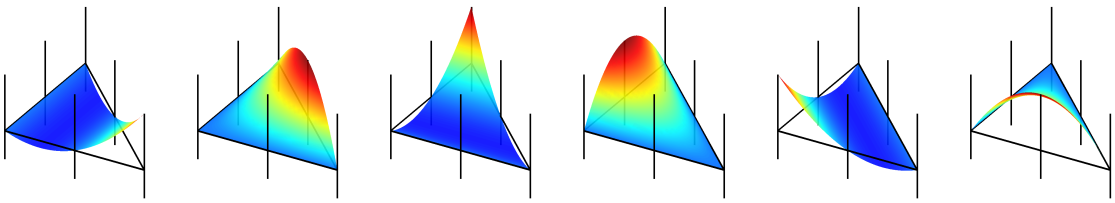


Figure 3.5: *Two-dimensional Lagrange polynomials on a triangle. For $p = 2$ we obtain a total of six Lagrange polynomials which are depicted above. The triangle outline marks the x-y-plane, the values of the Lagrange polynomials are encoded in the z-direction. The node positions are given by the vertical lines which range from $z = -0.5$ to $z = 1$. Apparently, each Lagrange polynomial is assigned to a node where it has unit value; at all other nodes it is zero. In between, Lagrange polynomials show a smooth interpolation behaviour.*

Here, $\tilde{\mathrm{E}}_x^{\triangle}$ was introduced as a vector of n expansion coefficients for the numerical solution of the x-component of the electric field (see Table 3.1 for an overview on notation). Please note the Einstein notation which implies summation for repeating indices and is frequently used in the remainder of this thesis. Furthermore, it is important to realise that Eq. (3.14) separates time and space dependencies from each other. Specifically, the time dependence of the electric field is entirely determined by the expansion coefficients. Likewise, the spatial modulation is uniquely defined by the basis functions.

In general, one has to reconstruct the numerical solution at any point within the element using the expansion coefficients and the basis functions. However, there is a special basis which allows us to immediately connect the expansion coefficients with field values. The basis functions we use are called Lagrange polynomials or interpolating polynomials (Fig. 3.5). Their defining property is given by

$$L_i(\vec{\mathrm{r}}_j) = \delta_{ij} \equiv \begin{cases} 0 & \text{for} \quad i \neq j \\ 1 & \text{for} \quad i = j \end{cases} \qquad i,\, j \in 1 \ldots n, \tag{3.15}$$

where δ_{ij} is the Kronecker symbol. Given a number n of predetermined nodes $\vec{\mathrm{r}}_j$, each basis function will be zero for each but one of them. With this property it is easy to show that

$$\mathrm{E}_{x,\,N}^{\triangle}(\vec{\mathrm{r}}_k,\, t) = \tilde{\mathrm{E}}_{x,\,j}^{\triangle}(t) \cdot L_j(\vec{\mathrm{r}}_k) = \tilde{\mathrm{E}}_{x,\,j}^{\triangle}(t) \cdot \delta_{jk} = \tilde{\mathrm{E}}_{x,\,k}^{\triangle}(t). \tag{3.16}$$

Thus, the expansion coefficients correspond to the field values at the nodes. Hence, we call our DG method a *nodal* one. Usually, the more basis functions—and thus more coefficients—we use, the more accurate our expansion will be (compare Figs. 3.6 and 3.7). With the general shape of Lagrange polynomials of order p being given by

$$L_i(\vec{\mathrm{r}}) = \sum_{k,\,l,\,m=0}^{k+l+m \leq p} a_{k,\,l,\,m}^{(i)} \cdot x^k y^l z^m, \tag{3.17}$$

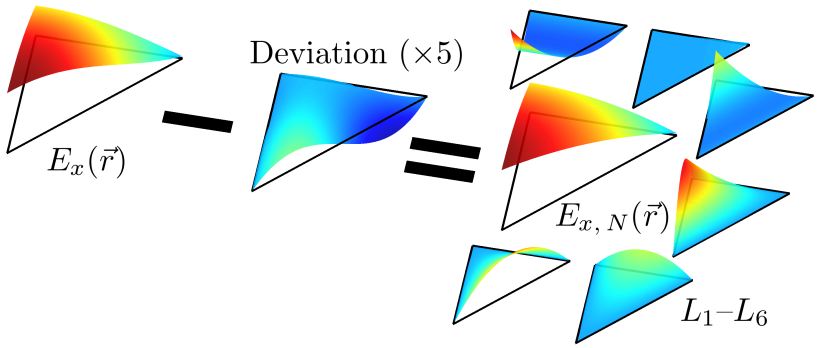


Figure 3.6: *Representation of the electric field in terms of Lagrange polynomials for $p = 2$. The left element shows the correct solution $E_x(\vec{r})$ to Maxwell's equations. The right-hand side shows the contributions of the six Lagrange polynomials to the numerical approximation $E_{x,\,N}(\vec{r})$. The element in between shows the deviation $E_x(\vec{r}) - E_{x,\,N}(\vec{r})$, magnified by a factor of 5 for better visibility. This deviation cannot be captured in the chosen function space; It is orthogonal to it. The overall approximation, however, is quite good since $E_x(\vec{r})$ only varies slowly with position.*

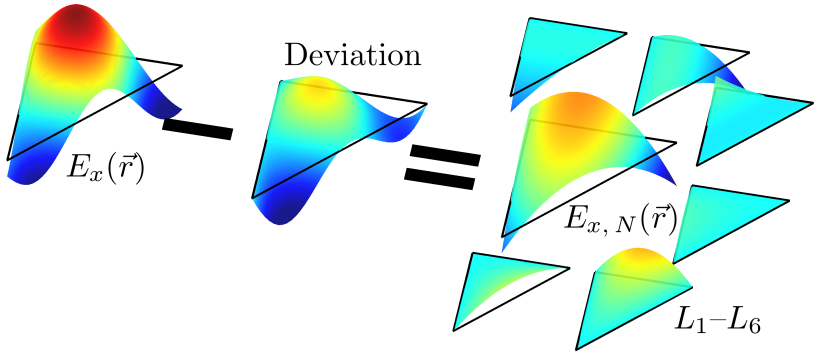


Figure 3.7: *Representation of the electric field in terms of Lagrange polynomials for $p = 2$. For a basic explanation of how to read this figure, see Fig. 3.6. However, here we try to approximate a strongly oscillating field. Obviously, the six Lagrange polynomials are not sufficient to properly resolve these oscillations, since the deviation (not magnified) is of the order of the original field values. More elements or higher-order basis functions can be used to improve on this.*

	1D				2D				3D			
p	n	n'	b	i	n	n'	b	i	n	n'	b	i
1	2	1	2	0	3	2	3	0	4	3	4	0
2	3	1	2	1	6	3	6	0	10	6	10	0
3	4	1	2	2	10	4	9	1	20	10	20	0
4	5	1	2	3	15	5	12	3	35	15	34	1
5	6	1	2	4	21	6	15	6	56	21	52	4
6	7	1	2	5	28	7	18	10	84	28	74	10
7	8	1	2	6	36	8	21	15	120	35	100	20

Table 3.3: *Number of nodes in one-, two-, and three-dimensional simplex elements. p denotes the polynomial order, n the total number of nodes, n' the number of nodes on a* single *face, b the number of nodes on the element's boundary, and i the number of nodes within the element. For higher dimensions, most nodes reside on the element's boundary.*

we get a total of

$$\begin{aligned} n &= p + 1 && \text{(1D)} \\ n &= \frac{1}{2} \cdot (p+1) \cdot (p+2) && \text{(2D)} \\ n &= \frac{1}{6} \cdot (p+1) \cdot (p+2) \cdot (p+3) && \text{(3D)} \end{aligned} \tag{3.18}$$

basis functions and nodal points per element. The coefficients $a^{(i)}_{k,l,m}$ are entirely determined by the node positions and Eq. (3.15). The actual computation of these coefficients is only trivial for one spatial dimension [27]. Section 3.2.2 sheds more light on the computation of Lagrange polynomials for two- and three-dimensional problems. A full set of Lagrange polynomials for $p = 2$ and two-dimensional systems is shown in Fig. 3.5. Table 3.3 shows numbers of nodal points and their distribution on the element boundaries for typical orders.

3.1.7 The Semi-Discrete Form

Finally, we have to insert the field expansion as presented in the previous section into Eq. (3.7), the strong formulation of Maxwell's curl equations. To this end, we consider the following integrals:

$$I^{\triangle}_{1,i} = \int_{V_{\triangle}} \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N(\vec{\mathrm{r}}, t) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r, \tag{3.19a}$$

$$I^{\triangle}_{2,i} = \int_{V_{\triangle}} \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_N) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r, \tag{3.19b}$$

$$I^{\triangle}_{3,i} = \int_{\partial V_{\triangle}} \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}(\mathbf{q}_N) - \vec{\mathrm{F}}^*(\mathbf{q}_N) \right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^2 r. \tag{3.19c}$$

Mass Matrix

Let us first deal with integral $I_{1,\,i}$. For clarity's sake, we restrict ourselves to the first component of $\mathbf{q}$, i.e., the E_x-component. On inserting the field expansion (3.14), we obtain

$$\begin{aligned} I_{1,\,i}^{\triangle,\,E_x} &= \int_{V_\triangle} \epsilon(\vec{\mathrm{r}}) \cdot \partial_t \tilde{\mathrm{E}}_{x,\,j}^{\triangle}(t) \cdot L_j(\vec{\mathrm{r}}) L_i(\vec{\mathrm{r}})\,\mathrm{d}^3 r \\ &= \partial_t \tilde{\mathrm{E}}_{x,\,j}^{\triangle}(t) \int_{V_\triangle} \epsilon(\vec{\mathrm{r}}) \cdot L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\,\mathrm{d}^3 r. \end{aligned} \tag{3.20}$$

We can simplify this further if we assume $\mathcal{Q}(\vec{\mathrm{r}})$ (and thus $\epsilon(\vec{\mathrm{r}})$ and $\mu(\vec{\mathrm{r}})$) to be constant within each element. For two reasons, this is generally not a restriction:

- Smooth material distributions are rarely encountered in nanophotonics. In those circumstances where they are, they are usually *effective* material properties derived from some substructure [7]. In these situations, it is controversial if effective material parameters include all physically relevant effects. It should be noted, however, that an accurate description of liquid crystals, for example, is not possible with elementwise constant material parameters.

- Normally, material parameters change abruptly, i.e., on atomic scales which cannot be resolved by wavelengths in the visible spectrum. Suppose we would allow for a discontinuity of $\epsilon(\vec{\mathrm{r}})$ within an element. In turn, the corresponding normal component of the electric field acquires a discontinuity as well. The latter, though, cannot be resolved with smooth Lagrange polynomials, which we use to represent the electric field *within* aforementioned element. The DG method, however, does allow for discontinuous fields across element interfaces.

Thus, we adapt our mesh in a way that material interfaces always coincide with element interfaces. In this case, integral $I_{1,\,i}^{\triangle}$ reduces to

$$I_{1,\,i}^{\triangle,\,E_x} = \epsilon^{\triangle} \cdot \partial_t \tilde{\mathrm{E}}_{x,\,j}^{\triangle}(t) \int_{V_\triangle} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\,\mathrm{d}^3 r, \tag{3.21}$$

where $\epsilon^{\triangle}$ is the constant permittivity of element $\triangle$. In matrix-vector notation, this is conveniently written as

$$I_{1,\,i}^{\triangle,\,E_x} = \epsilon^{\triangle} \cdot \partial_t \Big(\mathcal{M}^{\triangle} \tilde{\mathrm{E}}_x^{\triangle}(t) \Big)_i \tag{3.22}$$

with the components of the mass matrix $\mathcal{M}^{\triangle}$ given by

$$\Big(\mathcal{M}^{\triangle} \Big)_{ij} = \int_{V_\triangle} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\,\mathrm{d}^3 r. \tag{3.23}$$

Stiffness Matrices

Similarly, we can discuss the second integral. Again, we restrict ourselves to the first component of $\mathbf{q}$ and insert the field expansion:

$$\begin{aligned} I_{2,\,i}^{\triangle,\,E_x} &= \int_{V_\triangle} \vec{\nabla}\cdot\vec{\mathrm{F}}(\mathbf{q}_N)\cdot L_i(\vec{\mathrm{r}})\,\mathrm{d}^3r \\ &= \int_{V_\triangle} \Big(\partial_z \vec{\mathrm{H}}_{y,\,N}^{\triangle}(\vec{\mathrm{r}},\,t) - \partial_y \vec{\mathrm{H}}_{z,\,N}^{\triangle}(\vec{\mathrm{r}},\,t)\Big)\cdot L_i(\vec{\mathrm{r}})\,\mathrm{d}^3r \\ &= \int_{V_\triangle} \tilde{\mathrm{H}}_{y,\,j}^{\triangle}(t)\cdot L_i(\vec{\mathrm{r}})\Big(\partial_z L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3r - \int_{V_\triangle} \tilde{\mathrm{H}}_{z,\,j}^{\triangle}(t)\cdot L_i(\vec{\mathrm{r}})\Big(\partial_y L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3r \\ &= \tilde{\mathrm{H}}_{y,\,j}^{\triangle}(t)\cdot\int_{V_\triangle} L_i(\vec{\mathrm{r}})\Big(\partial_z L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3r - \tilde{\mathrm{H}}_{z,\,j}^{\triangle}(t)\cdot\int_{V_\triangle} L_i(\vec{\mathrm{r}})\Big(\partial_y L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3r. \end{aligned} \tag{3.24}$$

In complete analogy to the mass matrix, we define the stiffness matrices[1]

$$\Big(\mathcal{S}_k^{\triangle}\Big)_{ij} = \int_{V_\triangle} L_i(\vec{\mathrm{r}})\Big(\partial_k L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3r, \qquad k = x,\,y,\,z \tag{3.25}$$

and the three-dimensional vector of stiffness matrices

$$\vec{\mathcal{S}}^{\triangle} = \Big(\mathcal{S}_x^{\triangle},\,\mathcal{S}_y^{\triangle},\,\mathcal{S}_z^{\triangle}\Big)^T. \tag{3.26}$$

With these abbreviations, we can rewrite Eq. (3.24) in matrix-vector notation:

$$\begin{aligned} I_{2,\,i}^{\triangle,\,E_x} &= \Big(\mathcal{S}_z^{\triangle}\tilde{\mathrm{H}}_y^{\triangle}(t) - \mathcal{S}_y^{\triangle}\tilde{\mathrm{H}}_z^{\triangle}(t)\Big)_i \\ &= \Big(\vec{\mathcal{S}}^{\triangle}\times\tilde{\vec{\mathrm{H}}}^{\triangle}\Big)_{x,\,i}. \end{aligned} \tag{3.27}$$

Please note that $\tilde{\vec{\mathrm{H}}}^{\triangle}$ represents a three-dimensional vector with x-, y-, and z-components, where each component is a vector of expansion coefficients (compare Table 3.1). Hence, the vector product also leads to a three-dimensional vector whose components are vectors of expansion coefficients. Thus, the index $(x,\,i)$ selects the i-th expansion coefficient in the x-component of this vector product.

Face Mass Matrices

Finally, we have to deal with the remaining integral $I_{3,\,i}^{\triangle}$. Conceptually, there is no big difference to the steps necessary to simplify $I_{1,\,i}^{\triangle}$ and $I_{2,\,i}^{\triangle}$. Due to the number of terms in $\hat{\mathrm{n}}\cdot(\vec{\mathrm{F}}-\vec{\mathrm{F}}^*)$, however, writing down the necessary steps is rather tedious. Since there is little new to be learned, we simply state the end result:

$$I_{3,\,i}^{\triangle,\,E_x} = \sum_f \left[\mathcal{F}_f^{\triangle}\left(\frac{\alpha\Big[\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}}(\hat{\mathrm{n}}\cdot\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle})\Big] + Z^{+}\hat{\mathrm{n}}\times\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}}{\bar{Z}}\right)_x\right]_i. \tag{3.28}$$

[1] The terms "mass matrix" and "stiffness matrix" were originally coined in the context of finite element analysis of continuum mechanics.

Here, we have introduced face mass matrices

$$\left(\mathcal{F}_f^{\triangle}\right)_{ij} = \int_{f^{\triangle}} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\, \mathrm{d}^2 r, \tag{3.29}$$

where we integrate over the f-th face of element $\triangle$. Due to the special property (3.15) of the Lagrange polynomials, a Lagrange polynomial L_j identically vanishes on a face f if the associated node $\vec{\mathrm{r}}_j$ does not lie on this face.[2] Thus, many matrix entries of $\mathcal{F}_f^{\triangle}$ are zero. In addition, $\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle}$ represents a three-dimensional vector (x-, y-, z-components) of differences of expansion coefficients for the electric field across element $\triangle$'s face f. An equivalent statement holds for $\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}$. Consequently, the indices x and i in Eq. (3.28) should be understood in the spirit of Eq. (3.27).

Matrix Formulation of the Time-Dependent Curl Equations

Now that we have understood how the field expansion transforms spatial dependencies into matrix-vector products, we can solve for the time-dependent expansion coefficients to obtain the semi-discrete form of Maxwell's curl equations for linear, isotropic, dispersion-less, piecewise constant material parameters:

$$\begin{aligned}
\partial_t \tilde{\vec{\mathrm{E}}}^{\triangle} &= \frac{1}{\epsilon^{\triangle}} \left(\mathcal{M}^{\triangle}\right)^{-1} \left(\vec{\mathcal{S}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle} + \mathcal{F}_f^{\triangle} \frac{\alpha \left[\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}}(\hat{\mathrm{n}} \cdot \Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle}) \right] + Z^{+} \hat{\mathrm{n}} \times \Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}}{\bar{Z}} \right), \\
\partial_t \tilde{\vec{\mathrm{H}}}^{\triangle} &= \frac{1}{\mu^{\triangle}} \left(\mathcal{M}^{\triangle}\right)^{-1} \left(-\vec{\mathcal{S}}^{\triangle} \times \tilde{\vec{\mathrm{E}}}^{\triangle} + \mathcal{F}_f^{\triangle} \frac{\alpha \left[\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle} - \hat{\mathrm{n}}(\hat{\mathrm{n}} \cdot \Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}) \right] - Y^{+} \hat{\mathrm{n}} \times \Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle}}{\bar{Y}} \right).
\end{aligned} \tag{3.30}$$

Please note that the inverse mass matrix and the face mass matrix are applied to physical vectors with x-, y-, and z-components, where each component itself is a vector of expansion coefficients (see Table 3.1). To such three-dimensional vectors, these matrices appear like scalar coefficients, i.e.,

$$\mathcal{F}_f^{\triangle} \cdot \tilde{\vec{\mathrm{a}}} \equiv \left(\mathcal{F}_f^{\triangle} \tilde{\mathrm{a}}_x,\, \mathcal{F}_f^{\triangle} \tilde{\mathrm{a}}_y,\, \mathcal{F}_f^{\triangle} \tilde{\mathrm{a}}_z \right)^T. \tag{3.31}$$

The semi-discrete form (3.30) determines the time evolution of all expansion coefficients in a *single* element. Since the same equation can be independently applied to all elements, it also determines the time-evolution of the global electromagnetic fields.

This is the key difference to standard finite element methods (FEM). Even though these methods follow a similar approach, they uniquely define fields everywhere in the computational domain. This, in turn, leads to a strong coupling of neighbouring elements. It is not possible to assign a local mass matrix $\mathcal{M}^{\triangle}$ to a single element, since this element effectively shares some degrees of freedom, i.e., expansion coefficients, with its neighbours. The global mass matrix lacks the block-diagonality of

[2] The restriction of a polynomial of order p in three dimensions on a face is still a polynomial of at most order p, albeit a two-dimensional one. At the same time, each face contains a number of nodes which corresponds to the degrees of freedom of such a polynomial. Hence, the values at the node positions uniquely determine the restriction of the original polynomial on the face. If these values are all zero, the restriction must be the trivial polynomial.

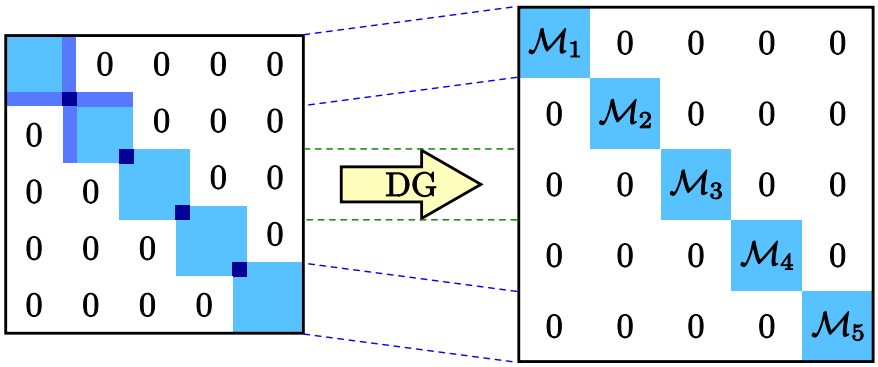


Figure 3.8: *Mass matrices for standard finite elements (left panel) and the discontinuous Galerkin method (right panel) for a one-dimensional system. Uniquely defined field values lead to an interwoven pattern in the case of standard FEM. Since the DG method uses two degrees of freedom to describe a field on both sides of an element-element interface, we obtain a block-diagonal mass matrix, whose inverse retains this convenient property. However, this comes at the price of an increased number of degrees of freedom, especially for three-dimensional calculations.*

the DG method (Fig. 3.8), which follows from assigning different field values to both sides of an element-element interface. Where DG's global mass matrix can be easily inverted by inverting the local mass matrices, this is considerably more difficult for FEM: There, the inverse global mass matrix will be a dense matrix in general, i.e., a matrix where most elements are nonzero. As the number of entries in this matrix quadratically depends on the number of elements used, inverting (and storing) it is impractical even for small systems. With DG, we are not subject to such restrictions. More details on matrix handling can be found in section 3.2.3.

To summarise, in this section we have absorbed the spatial dependence in Maxwell's equations in a set of expansion coefficients and matrix operators. The resulting semi-discrete curl equations (3.30) present a set of coupled, meanwhile ordinary differential equations with respect to t. To solve these equations, chapters 4 and 5 provide complementary approaches in the time- and frequency-domain, respectively.

3.2 Technicalities

In section 3.1, we have presented the key ideas behind the discontinuous Galerkin method. Now it is time we provided additional information which might have been found distracting if mentioned earlier. First, we motivate the result for the numerical flux as given by Eq. (3.8). Subsequently, instructions on generating Lagrange polynomials in higher dimensions are provided. Then, we reduce all matrix operations in the semi-discrete form to operations on a reference element. This is the foundation to a memory-efficient implementation. In the last part, we describe a set of assembly-based routines which lead to significant performance gains as compared to standard libraries and C++ implementations.

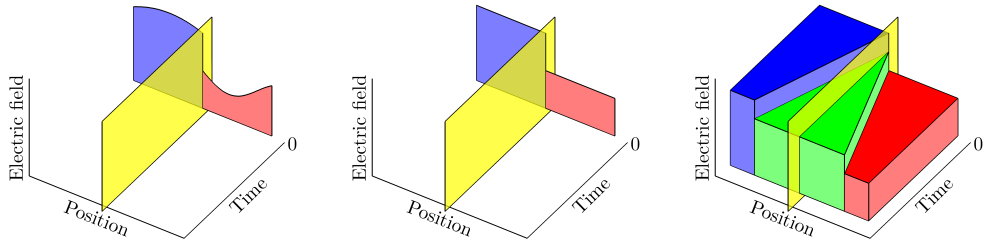


Figure 3.9: *Idea behind the upwind flux. Consider two one-dimensional elements with a common interface (yellow plane in the figures). At some initial time $t = 0$, we have a smooth distribution of the electric field on both sides of the interface (left panel). We are interested in the electric field near the interface after evolving the fields for an infinitesimally short time. To this end, we approximate the fields on both elements by constants which match the respective values of the electric field at the interface (central panel). This constitutes a Riemann problem. Solving this, we obtain propagating waves whose light cones enclose a central region (depicted in green). The amplitude in this region is determined by the material parameters in both elements (see Fig.* 3.11*).*

3.2.1 Derivation of the Numerical Flux

The numerical flux, introduced in Eq. (3.7) and defined in Eq. (3.8), is one of the central elements of the discontinuous Galerkin discretisation. Fortunately, the concept of numerical flux is borrowed from the related finite volume methods for hyperbolic differential equations [37]. Experience gathered in this area of research is easily transferred to the DG method.

As already mentioned, the choice of the numerical flux is not unique. A viable numerical flux must have the following properties:

- It mediates coupling between adjacent elements.
- It enforces the matching conditions of the electromagnetic fields at material interfaces. In particular, it reconciles the field values on both sides of an element-element interface, if both elements are filled with the same material.
- It leads to a numerically stable and convergent scheme.

A suitable (and somewhat natural) choice is given by the upwind flux [27, 28].

To this end, let us consider an element, which we associate with the index "$-$", and its neighbour, which we associate with the index "$+$". The interface between both elements is defined by the outwardly directed normal vector $\hat{\mathrm{n}}$, i.e., $\hat{\mathrm{n}}$ points from element $\triangle^-$ to element $\triangle^+$. In both elements, we have two independent field distributions $\mathbf{q}^-(\vec{\mathrm{r}})$ and $\mathbf{q}^+(\vec{\mathrm{r}})$, respectively.

Intuitively, the numerical flux $\vec{\mathrm{F}}^*$ represents those fields which propagate from the neighbour to the local element at some given time t. To determine this propagation, we evolve the electromagnetic fields according to Maxwell's curl equations for an infinitesimally short time dt. Since the speed of light is finite, only field values in an infinitesimally small volume around the interface must be considered. Thus, we can safely approximate the fields $\mathbf{q}^-(\vec{\mathrm{r}})$ and $\mathbf{q}^+(\vec{\mathrm{r}})$ by *constant* field values taken on the interface. This situation is illustrated in Fig. 3.9 and constitutes a Riemann problem [37]: We investigate the time evolution of a hyperbolic system (Maxwell's equations) with the initial

condition being a step function.

As a first step, we rewrite Maxwell's equations in conservation form (3.3) in terms of matrices $\mathcal{A}_k$:

$$\mathcal{Q}\partial_t \mathbf{q} + \mathcal{A}_x \partial_x \mathbf{q} + \mathcal{A}_y \partial_y \mathbf{q} + \mathcal{A}_z \partial_z \mathbf{q} = 0. \tag{3.32}$$

For example, $\mathcal{A}_x$ is given by

$$\mathcal{A}_x = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.33}$$

The dynamics of the Riemann problem depends on the speed with which the electromagnetic fields propagate in the direction normal to the interface. This velocity is given by the eigenvalues of

$$\mathcal{Q}^{-1} \cdot \Big(n_x \mathcal{A}_x + n_y \mathcal{A}_y + n_z \mathcal{A}_z \Big), \tag{3.34}$$

where n_k denotes the k-th component of the unit vector $\hat{\mathrm{n}}$. Depending on the material constants in the elements $\triangle^-$ and $\triangle^+$, we obtain three distinct eigenvalues

$$c_1^{\pm} = 0, \quad c_2^{\pm} = +c^{\pm}, \quad \text{and} \quad c_3^{\pm} = -c^{\pm} \quad \text{with} \quad c^{\pm} = \frac{1}{\sqrt{\epsilon^{\pm}\mu^{\pm}}}. \tag{3.35}$$

Wave propagation in each medium can only occur with the respective velocities. In particular, the discontinuity in the fields $\mathbf{q}^{\pm}$ at the interface can only propagate along three characteristic lines, also called characteristics [37]:

- It can propagate in negative direction with respect to the interface normal, i.e., into element $\triangle^-$. The corresponding velocity is given by $-c^-$.
- Propagation is also possible along the normal direction, i.e., into element $\triangle^+$. Consequently, the respective velocity is given by $+c^+$.
- Alternatively, the discontinuity can remain at the interface, which corresponds to a vanishing velocity.

These characteristic lines divide the $\vec{\mathrm{r}}$-t space into four domains, in each of which the respective values of $\mathbf{q}$ are constant. Since the discontinuity only propagates at finite speed, the values in two of these domains are already known and given by $\mathbf{q}^-$ and $\mathbf{q}^+$. The other two regions, represented by indices "$*$" and "$**$", introduce new, currently unknown field values $\mathbf{q}^*$ and $\mathbf{q}^{**}$, respectively. This setup is sketched in Fig. 3.10.

On crossing one of the characteristics, the component of $\mathbf{q}$ which corresponds to the eigenvector associated with the given characteristic exhibits a jump. The latter is related to the respective components of the fields on either side of the characteristic. This observation is known as the Rankine-Hugoniot jump condition [37]. Since we have three characteristics to cross, the Rankine-Hugoniot

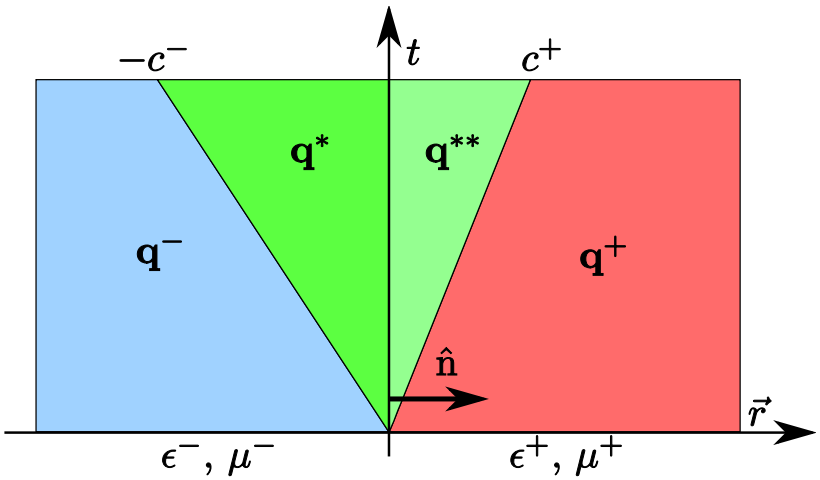


Figure 3.10: *Fields in the vicinity of an interface between two elements. In essence, this figure is a top-view of Fig.* 3.9*). The colours indicate different field values. Note that the discontinuity propagates along characteristic lines with speeds determined by the material parameters.*

condition leads to the equations [27]

$$c^- \mathcal{Q}^- \cdot \left(\mathbf{q}^* - \mathbf{q}^-\right) + \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}^* - \vec{\mathrm{F}}^-\right) = 0, \tag{3.36a}$$

$$\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}^* - \vec{\mathrm{F}}^{**}\right) = 0, \tag{3.36b}$$

$$-c^+ \mathcal{Q}^+ \cdot \left(\mathbf{q}^{**} - \mathbf{q}^+\right) + \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}^{**} - \vec{\mathrm{F}}^+\right) = 0. \tag{3.36c}$$

Herein, we have used the obvious abbreviations

$$\vec{\mathrm{F}}^{\pm} = \vec{\mathrm{F}}(\mathbf{q}^{\pm}), \quad \vec{\mathrm{F}}^* = \vec{\mathrm{F}}(\mathbf{q}^*), \quad \text{and} \quad \vec{\mathrm{F}}^{**} = \vec{\mathrm{F}}(\mathbf{q}^{**}). \tag{3.37}$$

Starting from this ansatz, we can solve for $\hat{\mathrm{n}} \cdot \vec{\mathrm{F}}^*$, and subsequently determine $\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}}^- - \vec{\mathrm{F}}^*)$. To this end, we use Eq. (3.36b) to eliminate $\hat{\mathrm{n}} \cdot \vec{\mathrm{F}}^{**}$ in Eq. (3.36c). Then, we multiply (3.36a) by $c^+ \mathcal{Q}^+$ and (3.36c) by $c^- \mathcal{Q}^-$. Adding the last two equations and inserting the definitions (3.9) through (3.11), we obtain

$$\begin{aligned} &Y^+ Y^- \left(\vec{\mathrm{E}}^- - \vec{\mathrm{E}}^+\right) + Y^+ Y^- \left(\vec{\mathrm{E}}^{**} - \vec{\mathrm{E}}^*\right) \\ &\qquad - \hat{\mathrm{n}} \times \left(Y^+ \vec{\mathrm{H}}^- + Y^- \vec{\mathrm{H}}^+\right) + \bar{Y} \hat{\mathrm{n}} \times \vec{\mathrm{H}}^* = 0. \end{aligned} \tag{3.38}$$

Taking the scalar product of Eq. (3.38) and $\hat{\mathrm{n}}$ yields

$$\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{E}}^{**} - \vec{\mathrm{E}}^*\right) = \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{E}}^+ - \vec{\mathrm{E}}^-\right). \tag{3.39}$$

From (3.36b) we know that

$$\hat{\mathrm{n}} \times \left(\vec{\mathrm{E}}^{**} - \vec{\mathrm{E}}^*\right) = 0. \tag{3.40}$$

Since the identity

$$\vec{\mathrm{X}} = \hat{\mathrm{n}} \cdot \left(\hat{\mathrm{n}} \cdot \vec{\mathrm{X}}\right) - \hat{\mathrm{n}} \times \left(\hat{\mathrm{n}} \times \vec{\mathrm{X}}\right) \tag{3.41}$$

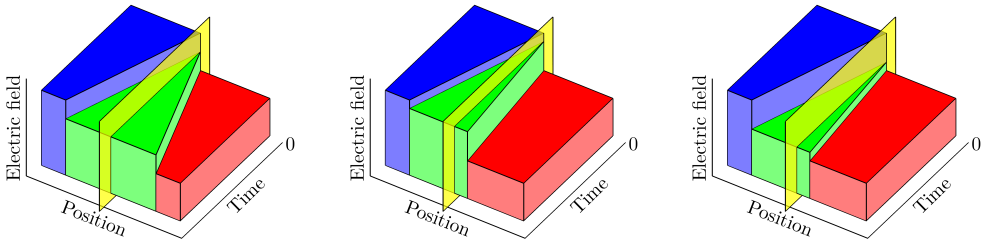


Figure 3.11: *Solutions of the Riemann problem as seen in Fig.* 3.9 *for different material parameters. In all panels, the left-hand side elements represent vacuum, i.e.,* $\epsilon = \mu = 1$*. The right-hand side elements feature different parameters:* $\epsilon = \mu = 1$ *(left panel),* $\epsilon = 1$*,* $\mu = 12$ *(central panel), and* $\epsilon = 12$*,* $\mu = 1$ *(right panel). Please note the different amplitudes of the electric field in the central region and the varying apex angle in element 2, which corresponds to the speed of light in the respective medium.*

holds for any vector $\vec{\mathrm{X}}$, this leads to

$$\vec{\mathrm{E}}^{**} - \vec{\mathrm{E}}^{*} = \Delta\vec{\mathrm{E}} + \hat{\mathrm{n}} \times \left(\hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}\right), \tag{3.42}$$

where we have employed field differences as defined in Eq. (3.12). Then, we insert (3.42) into (3.38) and solve for $\vec{\mathrm{H}}^{*}$:

$$-\hat{\mathrm{n}} \times \vec{\mathrm{H}}^{*} = \frac{1}{Z} \cdot \left[\hat{\mathrm{n}} \times \left(\hat{\mathrm{n}} \times \Delta E\right) - \hat{\mathrm{n}} \times \left(Z^{-}\vec{\mathrm{H}}^{-} + Z^{+}\vec{\mathrm{H}}^{+}\right)\right]. \tag{3.43}$$

This, however, is one of the unknown terms in

$$\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}^{-} - \vec{\mathrm{F}}^{*}\right) = \begin{pmatrix} -\hat{\mathrm{n}} \times \left(\vec{\mathrm{H}}^{-} - \vec{\mathrm{H}}^{*}\right) \\ \hat{\mathrm{n}} \times \left(\vec{\mathrm{E}}^{-} - \vec{\mathrm{E}}^{*}\right) \end{pmatrix}. \tag{3.44}$$

Similarly, we can find an expression for $\hat{\mathrm{n}} \times \vec{\mathrm{E}}$. Finally, we identify $\vec{\mathrm{F}}^{-} \equiv \vec{\mathrm{F}}$ and obtain the result (3.8). Figure 3.11 visualises the values of the electric field in various regions for a one-dimensional problem with various material parameters.

3.2.2 Creation of Basis Functions

In section 3.1.7, we have defined matrix operators which involve volume integrals over a kernel of Lagrange polynomials $L_i(\vec{\mathrm{r}})$. Obviously, we need to know these basis functions explicitly. According to Eq. (3.15), the Lagrange polynomials are uniquely defined once the interpolation nodes $\vec{\mathrm{r}}_i$ are specified.

As it turns out, the choice of the node positions heavily influences the accuracy with which an arbitrary function can be interpolated *in between* the nodes [27]. A suitable choice of nodes for one-dimensional elements is given by the Legendre-Gauss-Lobatto (LGL) quadrature points [27]. Simple construction rules are available for both the quadrature points and Lagrange polynomials.

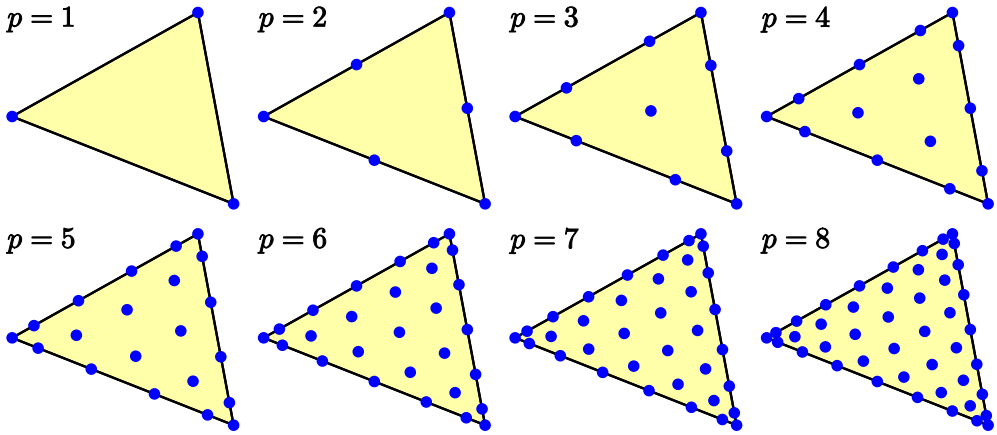


Figure 3.12: *Node positions in two dimensions for various polynomial orders p. The optimal node distribution on the edges is given by an analytical formula. Using the Warp&Blend method, nodes are smoothly arranged within the element. Please note that the nodes are not equally spaced. In particular, higher orders feature very small distances near the corners which heavily influences the maximum time step for the discontinuous Galerkin time-domain method (see section* 4.3*).*

The generalisation to higher dimensions, however, is not as straightforward as one would hope for. For two- and three-dimensional problems, the empirical Warp&Blend method allows to generate two- and three-dimensional point sets starting from LGL nodes on an element's edges [58]. In principle, more optimised point sets might slightly improve the interpolation error [59], but for most practical calculations the nodes generated via the Warp&Blend method perform sufficiently well. Figure 3.12 shows the distribution of nodes in a two-dimensional element.

Once the $\vec{\mathrm{r}}_i$ are given, we can use Eqs. (3.15) and (3.17) to obtain linear systems of equations, whose solutions determine the Lagrange polynomials. Unfortunately, solving these systems suffers from bad conditioning and low precision, especially for higher polynomial orders p [27]. A more viable alternative to compute the L_j employs an intermediate basis of polynomials $\psi_j(\vec{\mathrm{r}})$ [28].

In analogy to the field expansion (3.14) in terms of Lagrange polynomials, we expand the fields in our intermediate basis:

$$\mathrm{E}_x(\vec{\mathrm{r}}) \approx \sum_{j=1}^{n} \tilde{\mathrm{E}}'_{x,\,j} \cdot \psi_j(\vec{\mathrm{r}}) \equiv \sum_{j=1}^{n} \tilde{\mathrm{E}}_{x,\,j} \cdot L_j(\vec{\mathrm{r}}). \tag{3.45}$$

While $\tilde{\mathrm{E}}_x$ denotes the set of expansion coefficients for the L_j basis, $\tilde{\mathrm{E}}'_x$ is the corresponding set for the ψ_j basis. Please note that both function spaces describe the same space of polynomials; Thus, any approximation of $\mathrm{E}_x(\vec{\mathrm{r}})$ in terms of ψ_j can also be represented in terms of L_j without introducing errors. In particular, we can represent the field at a node position $\vec{\mathrm{r}}_i$:

$$\mathrm{E}_x(\vec{\mathrm{r}}_i) \approx \sum_{j=1}^{n} \tilde{\mathrm{E}}'_{x,\,j} \cdot \psi_j(\vec{\mathrm{r}}_i) \equiv \sum_{j=1}^{n} \tilde{\mathrm{E}}_{x,\,j} \cdot L_j(\vec{\mathrm{r}}_i) \equiv \tilde{\mathrm{E}}_{x,\,i}. \tag{3.46}$$

This implies the relation

$$\tilde{\mathrm{E}}_x = \mathcal{V}\tilde{\mathrm{E}}'_x, \tag{3.47}$$

where we have introduced the generalised[3] Vandermonde matrix [28]

$$\mathcal{V}_{ij} \equiv \psi_j(\vec{\mathrm{r}}_i). \tag{3.48}$$

The Vandermonde matrix facilitates an easy transition between the two bases. Obviously, it can be used to express the basis functions in terms of each other, i.e.,

$$\psi_i(\vec{\mathrm{r}}) = \sum_{j=1}^{n} \psi_i(\vec{\mathrm{r}}_j) \cdot L_j(\vec{\mathrm{r}}) = \sum_{j=1}^{n} \mathcal{V}_{ji} \cdot L_j(\vec{\mathrm{r}}) = \sum_{j=1}^{n} (\mathcal{V}^T)_{ij} \cdot L_j(\vec{\mathrm{r}}) \tag{3.49}$$

and, vice versa,

$$L_i(\vec{\mathrm{r}}) = \sum_{j=1}^{n} \left(\mathcal{V}^T\right)^{-1}_{ij} \cdot \psi_j(\vec{\mathrm{r}}). \tag{3.50}$$

Hence, the computation of Lagrange polynomials is essentially reduced to the inversion of the transposed Vandermonde matrix. As compared to the solution of the linear system of equations mentioned earlier, here we have actually retained some freedom. The condition of $\mathcal{V}^T$, basically an indicator of linear independence, strongly affects the precision of the matrix inversion process. The condition number, though, is entirely determined by the choice of basis functions $\psi_j(\vec{\mathrm{r}})$. Experience shows that a choice of orthonormal[4] basis functions yields a sufficient degree of linear independence, such that the inverse is easily and accurately calculated [28]. In our computer code, we employ the Koornwinder–Dubiner polynomials [27, 28, 60], which are constructed from products of classical orthogonal Legendre polynomials.

We conclude that the Warp&Blend method together with the Koornwinder–Dubiner polynomials enable us to efficiently construct Lagrange polynomials of arbitrary spatial orders p in a generic way. This is an important difference to conventional finite element methods, where the construction of higher-order vector elements is not trivial.

3.2.3 Operations on the Reference Element

In section 3.1.7, we have derived the semi-discrete form of Maxwell's curl equations (3.30) on a single element $\triangle$. It appears that for each element we have to construct an individual set of basis functions $L_i(\vec{\mathrm{r}})$ and perform the required integrations. This, however, would be impractical regarding either CPU time, memory consumption, or even both. In the following, we will introduce an important technique which allows for an extremely efficient computation of mass, stiffness, and face mass matrices.

The key idea has already been illustrated in Figs. 3.1 to 3.3: For each element in the mesh, we define an affine transform which maps the element $\triangle$ (and the associated set of nodal points) to a common reference element $\triangle_{\mathrm{ref}}$. In general, such a transform is given by

$$\vec{\mathrm{r}}^{\triangle} = \vec{\mathrm{r}}_0^{\triangle} + \mathcal{J}^{\triangle}\vec{\mathrm{s}}. \tag{3.51}$$

[3] We omit this attribute in the remainder of this thesis.

[4] Orthonormality is defined with respect to an appropriate scalar product

Given a point $\vec{\mathrm{s}} = (s_1,\, s_2,\, s_3)^T$ in the reference element, Eq. (3.51) constructs the corresponding point in an element of the mesh. The specific values of the offset $\vec{\mathrm{r}}_0^{\triangle}$ and the Jacobian matrix

$$\mathcal{J}^{\triangle} = \frac{\partial \vec{\mathrm{r}}}{\partial \vec{\mathrm{s}}} = \begin{pmatrix} \dfrac{\partial x}{\partial s_1} & \dfrac{\partial y}{\partial s_1} & \dfrac{\partial z}{\partial s_1} \\ \dfrac{\partial x}{\partial s_2} & \dfrac{\partial y}{\partial s_2} & \dfrac{\partial z}{\partial s_2} \\ \dfrac{\partial x}{\partial s_3} & \dfrac{\partial y}{\partial s_3} & \dfrac{\partial z}{\partial s_3} \end{pmatrix} = \begin{pmatrix} \mathcal{J}_{1x}^{\triangle} & \mathcal{J}_{1y}^{\triangle} & \mathcal{J}_{1z}^{\triangle} \\ \mathcal{J}_{2x}^{\triangle} & \mathcal{J}_{2y}^{\triangle} & \mathcal{J}_{2z}^{\triangle} \\ \mathcal{J}_{3x}^{\triangle} & \mathcal{J}_{3y}^{\triangle} & \mathcal{J}_{3z}^{\triangle} \end{pmatrix} \tag{3.52}$$

depend on the shapes of both elements, particularly on the respective vertices. If we assume both $\triangle$ and $\triangle_{\mathrm{ref}}$ to be straight-sided simplices, the Jacobian matrix is constant and depends on neither $\vec{\mathrm{s}}$ nor $\vec{\mathrm{r}}$.

Mass Matrix

With this knowledge, we can express the mass matrix of element $\triangle$ in terms of an integral over the reference element:

$$\left(\mathcal{M}^{\triangle}\right)_{ij} = \int_{V_{\triangle}} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\, \mathrm{d}^3 r = \int_{V_{\triangle_{\mathrm{ref}}}} L_i'(\vec{\mathrm{s}}) L_j'(\vec{\mathrm{s}}) \cdot \det(\mathcal{J}^{\triangle}(\vec{\mathrm{s}}))\, \mathrm{d}^3 s. \tag{3.53}$$

It is important to realise that the Lagrange polynomials $L_i'(\vec{\mathrm{s}})$ are defined on the reference element; Nevertheless, they are connected to the Lagrange polynomials on element $\triangle$ by

$$L_i(\vec{\mathrm{r}}_j) \equiv L_i'(\vec{\mathrm{s}}_j), \tag{3.54}$$

with $\vec{\mathrm{r}}$ and $\vec{\mathrm{s}}$ related via Eq. (3.51). In the following, we will omit this explicit distinction. In the important special case of straight-sided simplices (straight-sided triangles and tetrahedrons), we can remove the Jacobian determinant

$$J^{\triangle} = \det(\mathcal{J}^{\triangle}) \tag{3.55}$$

from Eq. (3.53)'s integrand. With the help of Eq. (3.50), we obtain

$$\left(\mathcal{M}^{\triangle}\right)_{ij} = J^{\triangle} \left(\mathcal{V}^T\right)^{-1}_{ik} \left(\mathcal{V}^T\right)^{-1}_{jl} \int_{V_{\triangle_{\mathrm{ref}}}} \psi_k(\vec{\mathrm{s}}) \psi_l(\vec{\mathrm{s}})\, \mathrm{d}^3 s. \tag{3.56}$$

The integral collapses to a mere Kronecker delta δ_{kl} due to the orthogonality of the chosen polynomial set on the reference element (see section 3.2.2 and Ref. [28]). Acknowledging

$$\left(\mathcal{V}^T\right)^{-1} \equiv \left(\mathcal{V}^{-1}\right)^T, \tag{3.57}$$

we obtain the result

$$\mathcal{M}^{\triangle} = J^{\triangle} \mathcal{M} \quad \text{with} \quad \mathcal{M} \equiv \left(\mathcal{V}\mathcal{V}^T\right)^{-1}. \tag{3.58}$$

Thus, the mass matrix of any straight-sided element $\triangle$ is conveniently reduced to the mass matrix $\mathcal{M}$ of the *reference* element scaled by a scalar factor $J^{\triangle}$. This drastically reduces the memory consumption: For three-dimensional elements with a fourth order ($p = 4$) basis, the mass matrix consists of $35 \cdot 35 = 1225$ entries (cf. Table 3.3).[5] Obviously, a single number $J^{\triangle}$ is much easier to store.

[5] Actually, we would only need to store the upper half of the matrix due to symmetry.

Stiffness and Derivative Matrices

Motivated by this success, we strive to obtain a similar result for the stiffness matrix. Let us start by reformulating the definition of the stiffness matrix (3.25):

$$\left(\mathcal{S}_k^\triangle\right)_{ij} = \int_{V_\triangle} L_i(\vec{\mathrm{r}})\Big(\partial_k L_j(\vec{\mathrm{r}})\Big)\,\mathrm{d}^3 r = \int_{V_\triangle} L_i(\vec{\mathrm{r}})\Big(\partial_k L_j(\vec{\mathrm{r}}_m)\Big) L_m(\vec{\mathrm{r}})\,\mathrm{d}^3 r. \tag{3.59}$$

Here, we have made use of the fact that the derivative of any polynomial of order p is a polynomial of order $p-1$. As such, there is a unique and exact representation of the derivative in terms of Lagrange polynomials, with the coefficients given by the values of $\partial_k L_j$ at the nodal positions. More concisely, the previous equation then reads

$$\left(\mathcal{S}_k^\triangle\right)_{ij} = \left(\mathcal{D}_k^\triangle\right)_{mj} \int_{V_\triangle} L_i(\vec{\mathrm{r}}) L_m(\vec{\mathrm{r}})\,\mathrm{d}^3 r = \left(\mathcal{M}^\triangle \mathcal{D}_k^\triangle\right)_{ij} \tag{3.60}$$

with the derivative matrices $\mathcal{D}_k^\triangle$ defined by

$$\left(\mathcal{D}_k^\triangle\right)_{mj} = \partial_k L_j(\vec{\mathrm{r}}_m). \tag{3.61}$$

For later use, we also define the vector of derivative matrices

$$\vec{\mathcal{D}}^\triangle = \left(\mathcal{D}_x^\triangle,\, \mathcal{D}_y^\triangle,\, \mathcal{D}_z^\triangle\right)^T. \tag{3.62}$$

Interestingly, in the semi-discrete form of Maxwell's curl equations (3.30), the stiffness matrices only occur in combination with the inverse mass matrix. Thus, we seek for a representation of

$$\left(\mathcal{M}^\triangle\right)^{-1} \mathcal{S}_k^\triangle \equiv \mathcal{D}_k^\triangle \tag{3.63}$$

in terms of matrices on the reference element. Such a connection is easily established using the inverse Jacobian matrix

$$\left(\mathcal{J}^\triangle\right)^{-1} = \frac{\partial \vec{\mathrm{s}}}{\partial \vec{\mathrm{x}}}, \tag{3.64}$$

with which we find

$$\begin{aligned} \left(\mathcal{D}_k^\triangle\right)_{mj} &= \sum_d \left(\mathcal{J}^\triangle\right)^{-1}_{kd} \cdot \partial_d L_j(\vec{\mathrm{s}}_m) \\ &= \sum_d \left(\mathcal{J}^\triangle\right)^{-1}_{kd} \cdot \Big[\left(\mathcal{V}^T\right)^{-1}_{jn} \cdot \partial_d \psi_n(\vec{\mathrm{s}}_m)\Big]. \end{aligned} \tag{3.65}$$

Here, the summation index d runs from 1 to 3 for three-dimensional systems. We identify the term

$$\left(\mathcal{D}_d\right)_{mj} = \left(\mathcal{V}^T\right)^{-1}_{jn} \cdot \partial_d \psi_n(\vec{\mathrm{s}}_m) \tag{3.66}$$

with the derivative matrix $\mathcal{D}_d$ on the reference element, and employ this definition to find a compact notation on the element $\triangle$:

$$\mathcal{D}_k^\triangle = \sum_{d=1}^{3} \left(\mathcal{J}^\triangle\right)^{-1}_{kd} \cdot \mathcal{D}_d. \tag{3.67}$$

We conclude that for an efficient computation of the derivative matrix on each element of the mesh, we need to store the elements of the inverse Jacobi matrix plus a set of derivative matrices on the reference element.

Face Mass Matrix

The remaining component of the semi-discrete form, the face mass matrix, can be computed in analogy to the mass matrix:

$$\left(\mathcal{F}_f^{\triangle}\right)_{ij} = J_f^{\triangle} \cdot \mathcal{F}_f \quad \text{with} \quad \mathcal{F}_f = \int_{f^{\triangle}} L_i(\vec{\mathrm{s}}) L_j(\vec{\mathrm{s}})\, \mathrm{d}^2 s, \tag{3.68}$$

As expected, the face mass matrices on each element $\triangle$ can be easily computed from face mass matrices on a reference element, provided we store the respective face Jacobian determinants $J_f^{\triangle}$.

The Semi-Discrete Form on the Reference Element

Let us now summarise the efforts made in the previous paragraphs. After replacing the matrices with their counterparts on the reference element, the semi-discrete form of Maxwell's curl equations on straight-sided elements reads:

$$\begin{aligned}
\partial_t \tilde{\vec{\mathrm{E}}}^{\triangle} &= \frac{1}{\epsilon^{\triangle}} \left(\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle} + \frac{J_f^{\triangle}}{J^{\triangle}} \mathcal{M}^{-1} \mathcal{F}_f \frac{\alpha \left[\Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}} (\hat{\mathrm{n}} \cdot \Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle}) \right] + Z^{+} \hat{\mathrm{n}} \times \Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle}}{\bar{Z}} \right), \\
\partial_t \tilde{\vec{\mathrm{H}}}^{\triangle} &= \frac{1}{\mu^{\triangle}} \left(-\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{E}}}^{\triangle} + \frac{J_f^{\triangle}}{J^{\triangle}} \mathcal{M}^{-1} \mathcal{F}_f \frac{\alpha \left[\Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle} - \hat{\mathrm{n}} (\hat{\mathrm{n}} \cdot \Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle}) \right] - Y^{+} \hat{\mathrm{n}} \times \Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle}}{\bar{Y}} \right).
\end{aligned} \tag{3.69}$$

Thus, Eq. (3.69) determines the time-evolution of the field expansion coefficients solely with matrix operators formulated on a reference element.[6] For an efficient computation, we need the following ingredients:

- On the reference element, we need to precalculate and store the inverse mass matrix $\mathcal{M}^{-1}$, the derivative matrices $\mathcal{D}_i$, and the face mass matrices $\mathcal{F}_f$.
- For each element in the computational domain, we need to store the Jacobian determinant of the affine transform (3.51). Furthermore, we need to store the components of the *inverse* Jacobian matrix. Finally, the face Jacobian determinants must be kept in memory.

For three-dimensional calculations, this leads to a total of 14 numbers per element. In double precision, this amounts to an insignificant $112\,\mathrm{bytes}$ of memory. This is independent of the order of the Lagrange polynomials.

3.2.4 Efficient Implementation of Matrix-Vector Products

In the previous section we have learned that the time-dependence of the electromagnetic fields is given by the right-hand sides of Eq. (3.69). Therein, the spatial dependence of the fields is represented by matrix operators acting on vectors of expansion coefficients. Since the time-stepping schemes presented in chapter 4 and the iterative solvers in chapter 5 rely on repeated evaluations of

[6] For typesetting reasons, we have refrained from replacing $\vec{\mathcal{D}}^{\triangle}$ by Eq. (3.67).

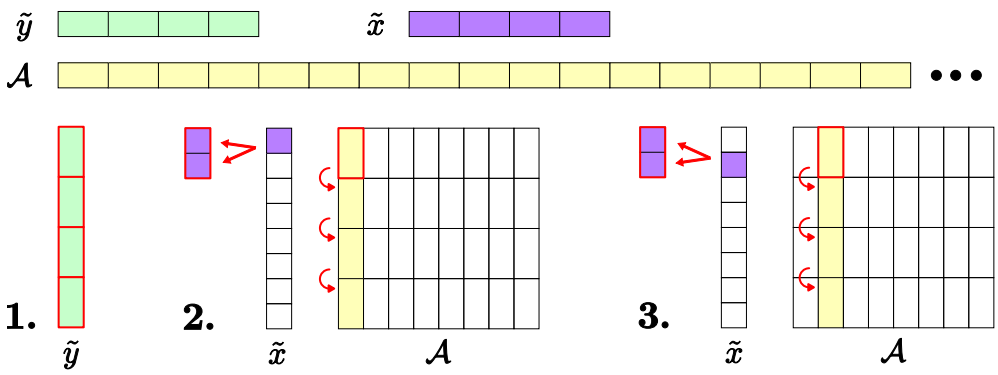


Figure 3.13: *Optimised matrix-vector product $\tilde{\mathrm{y}} = \tilde{\mathrm{y}} + \mathcal{A}\tilde{\mathrm{x}}$ for a small matrix with even size. Red boxes mark registers of 16 bytes. Large/small black boxes indicate 16/8 byte blocks in memory.* ***1.*** *Vector $\tilde{\mathrm{y}}$ is loaded from memory into a set of registers.* ***2.*** *From $\tilde{\mathrm{x}}$ we take the first component, and store it in another register. At the same time, we duplicate it so that the register contains $(\tilde{\mathrm{x}}_1, \tilde{\mathrm{x}}_1)$. Matrix $\mathcal{A}$, stored as a contiguous block of memory, is interpreted as a two-dimensional array. With a single load instruction we fill a register with* two *different values which correspond to different rows in $\mathcal{A}$. We then multiply it with the $(\tilde{\mathrm{x}}_1, \tilde{\mathrm{x}}_1)$ register and add it to the corresponding $\tilde{\mathrm{y}}$-register. This step is repeated for all remaining rows.* ***3.*** *We switch to the next column and proceed analogously to the first one. Once all columns are completed, we overwrite the original vector $\tilde{\mathrm{y}}$ in memory with the contents of the $\tilde{\mathrm{y}}$-registers.*

the right-hand sides of the semi-discrete form, fast matrix-vector multiplications are crucial for the overall performance.

First of all, let us consider the dimensions of the matrices involved. The (inverse) mass matrix $\mathcal{M}^{\Delta}$ is an $n \times n$ matrix, and so are the derivative matrices $\mathcal{D}_k^{\Delta}$. We recall that n is the number of nodes per element in d dimensions for order p. Omitting zeros, the face mass matrices are $n \times n'$ matrices, where n' is the number of points on a face. This number coincides with the total number of nodes for dimension $d-1$ and order p. Typical values of n and n' for various orders and dimensions can be found in Table 3.3. In most cases, n and n' will be approximately 20. Hence, the matrices typically contain just a few hundred entries.

Numerous libraries, usually some implementations of the standard Basic Linear Algebra System (`BLAS`), provide highly optimised matrix-vector products. However, the optimisations are usually performed with large matrices with tens of thousands of entries in mind. Standard implementations of matrix-vector products written in `C++` yield unsatisfactory results. Thus, we have developed a custom library of hand-written assembly routines, which is dedicated to matrices with the typical DG sizes.

Our library heavily leans on the streaming single-instruction multiple-data extensions (`SSE`), which are built-in instruction sets provided by most modern desktop and server processors [61]. In particular, we use features of a certain iteration called `SSSE3`. CPUs which support this instruction set provide 16 registers (extremely fast memory) with a size of 16 bytes each. This size is sufficient to store two floating point numbers in double precision. We will refer to such numbers as “doubles”. `SSE` provides useful instructions which allow for concurrent handling of two numbers. For example,

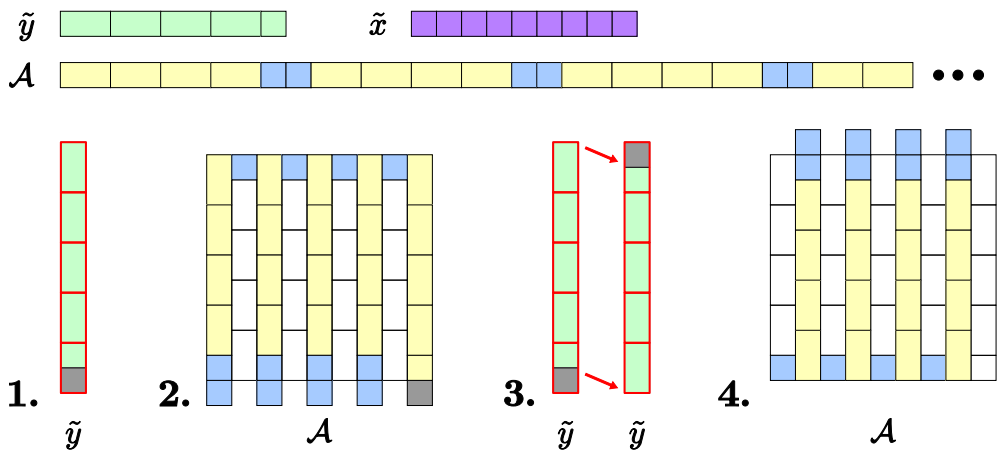


Figure 3.14: *Optimised matrix-vector product $\tilde{y} \leftarrow \tilde{y} + \mathcal{A}\tilde{x}$ for a small matrix with odd size. Here, blue blocks indicate conflicts in the 16 byte pattern, i.e., blocks where the two values belong to* different *columns of $\mathcal{A}$. **1.** As in Fig.* 3.13*, $\tilde{y}$ is loaded into registers. Due to the odd size, one register is padded with a randomly read number (grey boxes). **2.** We proceed as in Fig.* 3.13*, but only with odd columns. For these, we can always load 16 bytes starting with the first value. The last value, however, will be accompanied by another value, which will be added to the pad double in the $\tilde{y}$-registers. This value is either an actual matrix entry from the next, even column (blue), or a pad value behind the matrix (grey). **3.** We shuffle the values in the $\tilde{y}$-registers such that the pad double is at first position. **4.** We repeat step 2 for even columns. Note that we load data starting with the last value of the* previous *column to guarantee alignment. The additional values will be added to the pad number in $\tilde{y}$. Once we have completed all columns, we store the contents of the $\tilde{y}$-registers back to main memory, carefully avoiding storing the pad number.*

two registers a and b may hold two doubles $(a_1,\ a_2)$ and $(b_1,\ b_2)$, respectively. Using the appropriate instruction, we can calculate $a_1 \cdot b_1$ and $a_2 \cdot b_2$ in parallel, and thus twice as fast as with conventional instructions.

However, we can only benefit from such instructions if we can efficiently transfer data from memory to the registers. Since register space is limited and other memory types are comparatively slow, we have to minimise unnecessary or repeated memory access. Furthermore, memory access is considerably faster if we access blocks of two doubles in size *and* whose base addresses in memory are integer multiplies of 16 bytes. Such memory blocks are called to be aligned.

The ideal situation for the operation[7] $\tilde{y} \leftarrow \tilde{y} + \mathcal{A}\tilde{x}$ is illustrated in Fig. 3.13. Here, the matrix is small enough such that we can hold the entire target vector $\tilde{y}$ in registers. Furthermore, the size is an even number. Thus, blocks located at addresses not divisible by 16 bytes are avoided.

Conversely, Fig. 3.14 depicts the situation where $\mathcal{A}$ has an odd number of rows. Care has to be taken that only blocks with an address divisible by 16 bytes are accessed. We use an odd-column/even-column scheme to guarantee fast memory access. The excess operations on the subse-

[7] The left arrow means: Overwrite left-hand side with result from right-hand side.

p	Matrix size	Aligned	C++	MKL BLAS	Optimised	Speed-Up	
1	3×3	✓	10.00	30.78	6.67	1.50	4.61
		—	10.00	30.89	7.44	1.34	4.15
	3×6	✓	7.50	17.06	4.56	1.64	3.74
		—	7.50	17.50	4.56	1.64	3.83
2	6×6	✓	5.83	8.94	2.08	2.80	4.29
	6×9	✓	5.41	6.39	1.80	3.01	3.55
3	10×10	✓	4.80	3.97	1.42	3.38	2.80
	10×12	✓	4.75	3.38	1.31	3.63	2.58
4	15×15	✓	4.47	2.53	1.27	3.52	1.99
		—	4.47	2.73	1.27	3.52	2.15
5	21×18	✓	4.30	1.96	1.05	4.10	1.87
		—	4.29	2.22	1.05	4.09	2.11
	21×21	✓	4.29	1.84	1.00	4.29	1.84
		—	4.29	2.07	1.00	4.29	2.07
6	28×28	✓	4.15	1.20	0.81	5.12	1.48
	28×21	✓	4.20	1.39	0.85	4.94	1.64

Table 3.4: *Performance of optimised matrix-vector products for two-dimensional systems. The first three columns list the polynomial order p, the size of the matrix, and the correct alignment of $\tilde{\mathrm{y}}$. The next three columns feature the ratio of CPU cycles and the number of elements in the matrix for three different implementations: A standard `C++` implementation (compiled with `GNU C++ Compiler 4.3.2` with full optimisations), a call to Intel's Math Kernel Library (`MKL`) routines, and our optimised functions. Finally, the last two columns provide speed-up factors achieved with our assembly code as compared to the `C++` and `MKL` implementations, respectively. All measurements were performed on an Intel® Core™2 Quad Q9300 CPU running at* $2.5\,\mathrm{GHz}$.

quently introduced pad numbers are more efficient than unaligned memory access.

More difficulties arise due to potentially unaligned $\tilde{\mathrm{y}}$-vectors, and for vectors too large to be fully held in the registers. In the first case, one has to switch the order of odd and even columns, while in the second case the columns have to be split into multiple sections.

We have created optimised matrix-vector products for orders up to $p = 6$ for two-dimensional systems. For three-dimensional computations, we provide matrix-vector products up to $p = 5$. In our code, we combine $\mathcal{M}^{-1}\mathcal{F}_f$ for all faces to one large matrix. Hence, this matrix has the size $n \times (d+1)n'$.

Tables 3.4 and 3.5 compare the performance of our custom matrix-vector products against a standard `C++` implementation and the `BLAS` implementation included with the Intel's Math Kernel Library (`MKL`). The average numbers of CPU cycles per matrix entry listed in these tables are a measure of speed. Each CPU instruction requires a certain number of cycles of the clock frequency generator.[8]

[8] For a CPU running at $2.5\,\mathrm{GHz}$, each cycle corresponds to a time of $0.4\,\mathrm{ns}$.

p	Matrix size	Aligned	C++	MKL BLAS	Optimised	Speed-Up	
1	4×4	✓	7.50	18.25	3.75	2.00	4.87
	4×12	✓	5.62	7.02	2.02	2.78	3.48
2	10×10	✓	4.80	3.97	1.42	3.38	2.80
	10×24	✓	4.50	2.22	1.15	3.91	1.93
3	20×20	✓	4.29	1.59	0.88	4.88	1.81
	20×40	✓	4.21	1.24	0.81	5.20	1.53
4	35×35	✓	4.14	1.33	0.92	4.50	1.45
		—	4.14	1.54	0.93	4.45	1.66
	35×60	✓	4.22	1.23	0.88	4.80	1.40
		—	4.13	1.42	0.89	4.64	1.60
5	56×56	✓	4.07	0.86	0.76	5.36	1.13
	56×84	✓	4.09	1.15	0.83	4.93	1.39

Table 3.5: *Performance of optimised matrix-vector products for three-dimensional systems. Compare Fig.* 3.4 *for an explanation of the columns.*

For a standard matrix-vector product, we require a number of multiplications which corresponds to the number of elements in the matrix. Since modern CPUs execute additions and multiplications in parallel [62], the additions should not increase the execution time. As `SSSE3` allows two multiplications in parallel, the theoretical minimum for the number of cycles per element is 0.5.

In practice, this number cannot be reached for the small matrices presented here due to the overhead of function calls. Furthermore, all implementations actually compute

$$\tilde{\mathrm{y}} \leftarrow \beta\tilde{\mathrm{y}} + \alpha\mathcal{A}\tilde{\mathrm{x}}\,. \tag{3.70}$$

The additional multiplications by two real-valued numbers α and β lead to additional effort. For increasing matrix sizes, however, these operations become less dominant since they only scale linearly with the number of matrix rows, not with the number of elements.

Our results demonstrate a significant speed-up when using our implementation of matrix-vector products. The performance gains are most pronounced for two-dimensional systems and low orders. This is expected, as Intel's `BLAS` implementation is optimised for larger matrices. Nevertheless, we can save 30% and more time when using our custom library. This allows to simulate more time steps and larger systems in the same time.

3.3 Discussion

To conclude our presentation of the discontinuous Galerkin method, we want to highlight a few key points. Section 3.3.1 briefly recapitulates a few computational techniques and emphasises why DG is an efficient method. Section 3.3.2 provides us with a result on the accuracy of the spatial discretisation. At the same time, it motivates to use higher-order basis functions. Finally, section 3.3.3 summarises the considerable benefits DG holds for nanophotonic systems.

3.3.1 Computational Efficiency

The semi-discrete form, Eqs. (3.30), looks rather involved, especially in comparison with expressions obtained by standard finite-difference discretisations (see appendix A). It contains a number of matrices and even the inverse of the mass matrix. This section will explain why the discontinuous Galerkin discretisation is an efficient choice after all.

As we have seen in section 3.2.4, the occurring matrices are quite small and comprise a few hundred numbers. Storage and inversion are not a problem at all. When following the derivation, we find that small matrices are an immediate consequence of explicitly allowing discontinuous fields. In contrast, conventional finite element methods yield large, sparse matrices.[9] The inverse of a sparse matrix, however, is not necessarily sparse itself; In the case of standard finite elements, we would have to store large (10^{10} numbers for 10^5 degrees of freedom) inverse matrices. This is not possible even for medium-sized problems, since the size of the mass matrix reflects the number of elements in the computational domain.

Furthermore, we have learned in section 3.2.3 that—in most cases—there is no need to actually store mass and derivative matrices for each element separately. Using simple affine transformations, straight-sided elements (for curvilinear elements please refer to section 6.3) are mapped on a reference element. Thereby, all matrices can be recovered by scaling and combining matrices defined on the reference element. Instead of storing a number of matrices for each element, we merely store a few parameters associated to the affine transformation.

This overhead to the actual degrees of freedom, the expansion coefficients, is comparatively small and negligible for higher orders p. Hence, we have a memory-efficient scheme at hand. Furthermore, the right-hand side of the semi-discrete form is explicitly known because we deduce the required matrices on any element following simple rules. Given an arbitrary set of expansion coefficients, we can immediately determine their time-evolution without solving a system of linear equations first. It is this property which renders the DG method so attractive for solving Maxwell's equations in the time-domain (see Chapter 4).

The price we pay is that on each interface between neighbouring elements we store *two* distinct sets of field values; one for each element involved. Table 3.3 compares n, the total number of nodal points, with b, the number of nodal points on the element's boundary. In addition, the number of internal, non-boundary points i is listed. For two- and three-dimensional systems, most nodes are boundary nodes. Thus, we store most of the points twice. Higher orders reduce this overhead, but yield larger matrices. Nevertheless and especially in nanophotonics, higher orders are often favourable as will be discussed in the upcoming sections.

3.3.2 Error of the Spatial Discretisation

For numerical simulations it is essential to control the error of the approximate solution with respect to the correct solution of the original equation. In our case, we can increase the accuracy of the simulations by either decreasing the element size h or by increasing the polynomial degree p. For

[9] A sparse matrix is a matrix where almost all elements are zero. With this we mean that there is a fixed maximum number of nonzero elements per matrix row, which does not depend on the size of the matrix. Sparse matrices are extremely important in numerics, because only the values and positions of nonzero entries need to be stored. Furthermore, specialised algorithms which make use of the sparsity are available (see chapter 5).

sufficiently smooth solutions and an upwind flux ($\alpha \equiv 1$), the deviation between the numerical and the analytical solution is given by

$$\left\| \mathbf{q}^{\mathrm{ref}} - \mathbf{q}^{\mathrm{num}} \right\| = \mathcal{O}\Big(h^{p+1}\Big). \tag{3.71}$$

A rigorous proof can be found in Ref. [28]. Here, $\mathbf{q}^{\mathrm{ref}}$ represents the (not necessarily known) exact reference solution of the original, non-discretised equation, while $\mathbf{q}^{\mathrm{num}}$ is the numerical one. Given some arbitrary norm $\| \cdot \|$, the convergence is algebraic in h and exponential in p. Via p, we can choose how fast the error shall decrease with the mesh size, with the cost of additional operations per element. In principle, both h and p can be different for each element (hp-adaptivity). In practice, it is convenient to use the same p for all elements and vary h locally. To control the error, one can globally increase p (p-refinement) or reduce h either globally or locally (h-refinement). Finding a balance between accuracy and computational effort is not always easy. However, in most of our calculations we find that orders of $p \in [3, 4, 5, 6]$ offer a good compromise. For the mesh-size h we then typically choose a maximal edge length which corresponds to at most half the wavelength of interest.

3.3.3 The Discontinuous Galerkin Method and Nanophotonics

The spatial discretisation via the discontinuous Galerkin method has a number of merits of particular interest to nanophotonics. First of all, the method does not rely on regular grids such as the FDTD method, but on flexible meshes with elements of various sizes and shapes instead. This freedom basically allows us to "adapt" the simulation to a specific system. In particular, it is straightforward to handle inclined or curved material interfaces (compare Fig. A.2). Locally increasing the spatial resolution (via h- or p-refinement) allows to accurately model strong local variations of electromagnetic fields, e.g., near metallic surfaces. At the same time, one can save a substantial amount of degrees of freedom in regions with field variations on long length scales. This is usually the case in the area surrounding a nanophotonic device with features often much smaller than the wavelength of interest. Therefore, the DG method lends itself for multi-scale analysis.

Secondly, since the method employs discontinuous basis functions, discontinuities in the electromagnetic fields are inherently dealt with. Accuracy and convergence issues due to inhomogeneous material distributions as present in FDTD simulations are absent in DG calculations. At the same time, the special choice of basis functions also allows us to treat the elements separately. As a result, the DG discretisation lends itself for parallel computations on multiple cores, multiple computers, and graphic processors [63]. Especially higher polynomial orders p result in excellent parallel performance, since the ratio of local-element operations and inter-element communication is more favourable.

Simultaneously, high interpolation orders also lead to very faithful representations of both magnitudes and phases of propagating electromagnetic fields [28]. Resonators, interferometers, and other, potentially nonlinear, phase-sensitive systems often encountered in nanophotonics greatly benefit from the accuracy of higher-order methods [38].

Chapter 4

4 The Discontinuous Galerkin Time-Domain Method

Starting from Maxwell's curl equations, chapter 3 explained how to discretise the electromagnetic fields via the discontinuous Galerkin method. After a number of steps, the final result is given by the semi-discrete form (3.30), which presents a set of coupled, first-order, *ordinary* differential equations of time-dependent expansion coefficients.

In this chapter, we will regard the semi-discrete form as an initial value problem. Starting from some initial state $\mathbf{q}(\vec{\mathrm{r}},\, t_0) \equiv (\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t_0),\, \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t_0))$, we will apply an iterative procedure to evolve the electromagnetic fields time step by time step. Thus, we will obtain the Discontinuous Galerkin *Time-Domain* (DGTD) method.

The upcoming discussion is prefixed by a general introduction to time-domain methods (section 4.1). Afterwards, section 4.2 presents our time stepping algorithm of choice, the low-storage Runge-Kutta method. The issue of picking an appropriate time step is covered in section 4.3. Based on insight gained there, we briefly motivate the use of customised Runge-Kutta schemes in section 4.4, before we eventually compare the DGTD method with other numerical methods (section 4.5)

4.1 General Characteristics of Time-Domain Simulations

Before we come to the details of the DGTD method, let us outline a few general characteristics, advantages, and shortcomings of time-domain methods first. Time is a quantity human beings have an intuitive feeling for. Being accustomed to the concept that the present translates into the future, time-domain visualisations may help us distinguish cause and effect.

For example, we might be able to recognise the dominant scatterer among a set of particles just by looking at what happens to a pulse which propagates in a scattering region. We could study a light pulse in a photonic crystal waveguide bend and see *where* undesirable reflections occur. In phase-sensitive experiments, we can track interfering light pulses. Their phase differences will often be visible as spatial displacements. Researchers can benefit from such knowledge, since it helps to improve designs of devices and scatterers. In the extreme, animations of time-dependent field distributions most easily illustrate the behaviour of a physical system even for non-experts in the

field (or even the general area of science).[1]

At the end of the day, however, visualisation is not everything. To elevate a method above the status of a fancy toy, we must be able to predict the outcome of experiments. To this end, we are usually interested in spectral information about a system. The most trivial time-domain approach, i.e., simulating the same system over and over again with plane waves of different frequencies, leads to inadequately long simulation times. Instead, one launches ultra-short pulses into the system. Such pulses possess a very broad frequency spectrum (see section 6.1.3). Using the on-the-fly Fourier transform (section 6.4), one can obtain a spectrum over the complete bandwidth in a single time-domain simulation. Hence, obtaining spectra with time-domain methods is not a conceptual problem.

Nevertheless, time-domain simulations may have difficulties with some systems. Resonators with high quality factors, for example, often require excessively long simulation times in order to obtain reliable results. The inclusion of dispersive materials in the time-domain is also a non-trivial task (compare section 2.3), and may require a considerable amount of computational effort. Furthermore, for some systems it is challenging to obtain eigenmodes which often prove invaluable to understand the physics of a system.

On the other hand, time-domain solvers have a number of fundamental advantages. In particular, they are not limited to the treatment of linear equations. Classical nonlinearities, for example, lead to the generation of sum and difference frequencies and other wave mixing effects [64]. While frequency-domain methods (chapter 5) cannot treat such phenomena directly, time-domain methods do not have conceptual problems with nonlinear physics, since the equations themselves are given in the time-domain. Last but not least, time-domain methods are usually quite memory-efficient. In most cases, it is sufficient to store one or two vectors of expansion coefficients plus information about the physical system. Both the memory requirements and the computational time scale linearly with the number of degrees of freedom. Thus, time-domain methods can handle large systems—even on computers with limited resources.

4.2 Time-Stepping and the Runge-Kutta Method

To simplify the notation, we summarise Eqs. (3.30) as

$$\partial_t \begin{pmatrix} \tilde{\vec{\mathrm{E}}}(t) \\ \tilde{\vec{\mathrm{H}}}(t) \end{pmatrix} = \mathcal{H} \cdot \begin{pmatrix} \tilde{\vec{\mathrm{E}}}(t) \\ \tilde{\vec{\mathrm{H}}}(t) \end{pmatrix} \tag{4.1a}$$

or, more concisely,

$$\partial_t \tilde{\mathbf{q}}(t) = \mathcal{H}\tilde{\mathbf{q}}(t). \tag{4.1b}$$

The system operator $\mathcal{H}$ acting on the current expansion coefficients $\tilde{\mathbf{q}}(t)$ represents the right-hand sides of Eqs. (3.30). Please note that $\tilde{\mathbf{q}}(t)$ comprises *all* elements in the mesh, and not just a single one. Furthermore, we add source terms

$$\tilde{\mathbf{q}}^{\mathrm{source}}(t) = \left(\tilde{\vec{\mathrm{E}}}^{\mathrm{source}}(t),\, \tilde{\vec{\mathrm{H}}}^{\mathrm{source}}(t)\right)^T \tag{4.2}$$

[1] As a side note, visualisations are also most helpful for testing numerical implementations. In many cases, one can track down bugs by identifying regions where instabilities occur, for example.

to the equation:

$$\partial_t \tilde{\mathbf{q}}(t) = \mathcal{H}\tilde{\mathbf{q}}(t) + \tilde{\mathbf{q}}^{\text{source}}(t). \tag{4.3}$$

Section 6.1 will explain how sources are included in the simulation.

Equation (4.3) represents a set of ordinary differential equations of first order in time. To solve it numerically, we divide the time axis into a number of (not necessarily equidistant) time steps. The fields at the first time step t_0 define the initial condition. Usually, one assumes vanishing fields but other initial conditions, e.g., cavity modes, are not uncommon. Using a time stepping scheme, we successively evolve the fields from t_n to t_{n+1}, where n indicates the time step. Time stepping concludes once t_n exceeds a predetermined maximum simulation time T.

Let us consider the equation

$$\partial_t \tilde{\mathbf{q}}(t) = \tilde{\mathbf{f}}(\tilde{\mathbf{q}}(t),\, t), \tag{4.4}$$

which is a generalisation of Eq. (4.3). The choice of the time stepping scheme crucially affects the performance. In essence, one can either choose explicit or implicit methods.

For explicit methods, we need to evaluate the right-hand side $\tilde{\mathbf{f}}(\tilde{\mathbf{q}}(t),\, t)$, depending on the scheme usually multiple times per time step. Prominent solvers are multistep and Runge-Kutta methods [27]. However, if the time step is too large, then explicit solvers yield numerically unstable simulations, i.e., the electromagnetic fields undergo unphysical exponential growth. In contrast, implicit methods require the solution of a system of equations for each time step, but in turn provide unconditionally stable results. Nevertheless, the error increases with the time step size. As solving a system of equations is computationally expensive, time stepping with implicit solvers is often less efficient than using explicit solvers.

In the case of DGTD, the time evolution of the electromagnetic fields is most conveniently accomplished using explicit low-storage Runge-Kutta (LSRK) methods [65] for a number of reasons. First of all, the evaluation of the right-hand side of Eq. (4.3) is extremely efficient due to the techniques presented in sections 3.2.3 and 3.2.4. Hence, we expect good performance of explicit solvers. Secondly, it is desirable to accompany the higher-order accurate DG discretisation in space with a higher-order accurate time stepping scheme. Suitable LSRK schemes are available for up to fourth order. Finally, given a total number of N expansion coefficients, LSRK methods require a total of $2N$ values stored in two registers $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$.[2] This represents a moderate increase in memory requirements as compared to FDTD's leapfrog scheme, which only requires the storage of one set of field values at a time (see appendix A.3). However, FDTD is restricted to second-order accuracy in time and requires the electric and magnetic fields to be interleaved by half a time step.

To evolve the electromagnetic fields one time step from t_n to t_{n+1}, LSRK methods in the formulation of Williamson [66] comprise the following steps:

$$\tilde{\mathbf{q}}_0 := \tilde{\mathbf{q}}(t_n), \tag{4.5a}$$

$$\tilde{\mathbf{p}}_i := A_i \cdot \tilde{\mathbf{p}}_{i-1} + \Delta t \cdot \tilde{\mathbf{f}}(\tilde{\mathbf{q}}_{i-1},\, t_n + c_i \Delta t), \tag{4.5b}$$

$$\tilde{\mathbf{q}}_i := \tilde{\mathbf{q}}_{i-1} + B_i \cdot \tilde{\mathbf{p}}_i, \tag{4.5c}$$

$$\tilde{\mathbf{q}}(t_{n+1}) := \tilde{\mathbf{q}}_s. \tag{4.5d}$$

Steps (4.5b) and (4.5c) define the stages $i = 1 \,..\, s$, where s is the number of stages. Together with this parameter, the coefficients A_i ($A_1 \equiv 0$), B_i, and c_i define the properties (order of accuracy,

[2] These registers should not be confused with registers in the CPUs as discussed in section 3.2.4.

stability contour) of the LSRK scheme. It is noteworthy, however, that the storage requirements of the algorithm neither depend on the order of accuracy nor the number of stages. The five-stage, fourth-order accurate LSRK scheme by Carpenter and Kennedy [65] is most commonly used for the field evolution in DGTD [28,34,67]. Nevertheless, other choices [68–70] are possible and potentially advantageous. Hence, they are discussed in section 4.4.

4.3 Eigenvalues, Conditional Stability and Maximum Time Steps

Being explicit methods, LSRK schemes are subject to conditional stability, also known as the Courant-Friedrichs-Lewy condition (CFL condition) [21]. As soon as the time step Δt exceeds a critical time step $\Delta t_{\max}$, the numerical solution is subject to unphysical exponential growth. The critical time step, for which the numerical solution just does not grow exponentially, depends both on the time stepping scheme and the system of equations to be integrated.

Let us first consider the influence of a specific LSRK scheme, which is characterised by the characteristic polynomial (or amplification factor)

$$R(z) = 1 + \gamma_1 z + \gamma_2 z^2 + \cdots + \gamma_s z^s, \quad z \in \mathbb{C}. \tag{4.6}$$

Here, the coefficients γ_i can be directly related to the coefficients A_j, B_j, and c_j of the LSRK scheme [69, 70]. In the absence of sources, the discretised physical system can be expressed as a matrix-vector product of a system matrix $\mathcal{H}$ and a number of unknowns, compare Eq. (4.3). We can only obtain a stable time-integration if the necessary condition

$$\left| R(\Delta t \cdot \lambda_i) \right| \leq 1 \tag{4.7}$$

holds for *all* eigenvalues $\lambda_i = -\mathrm{i}\omega_i$ of $\mathcal{H}$, where ω_i is the frequency associated with this mode (see section 5.2). If there is a single eigenvalue for which $|R(\Delta t \lambda_i)| > 1$, then the amplitude of the corresponding mode will be unphysically amplified each time step. As a consequence, the simulation will yield spurious results.

Geometrically speaking, the curve $|R(z)| = 1$ defines a stability contour in the complex plane. The complex eigenvalue spectrum of $\mathcal{H}$ can be scaled by the time step Δt until it is completely enclosed by the stability contour. This is illustrated in Fig. 4.1. The largest time step for which this condition holds is called the maximum stable time step $\Delta t_{\max}$. It should be noted that this time step does not *guarantee* stability, since its derivation only relies on a necessary condition. Nevertheless, it serves as a good approximation to the time step obtained from a sufficient condition [27, 71].

Depending on the shape of the eigenvalue spectrum, the scaling factor required to squeeze it within the stability contour is different. Hence, it is worth to investigate the typical eigenvalue spectrum of our discretised Maxwell operator $\mathcal{H}$ as defined by Eq. (3.30). The spectrum is subject to a number of influences and can be tuned by changing the upwind parameter α [27, 70]. As discussed in section 3.1.4, $\alpha = 0$ reduces the upwind flux to the energy-conserving central flux. Consequently, all eigenfrequencies ω_i should be real numbers, provided the physical system does not include dissipative materials or absorbing boundary conditions. This translates to an eigenvalue spectrum of purely imaginary λ_i. As α increases, the eigenvalues gain a negative real part. This effectively damps modes which do not fulfil the vanishing divergence condition or are not sufficiently

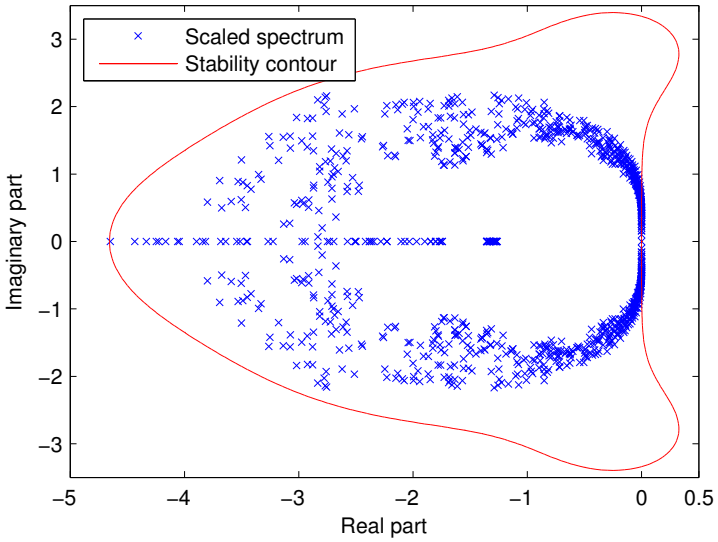


Figure 4.1: *Low-storage Runge-Kutta stability contour and eigenvalues of a two-dimensional cavity. The red contour encloses the region of stability of the five-stage, fourth-order accurate low-storage Runge-Kutta scheme as proposed by Carpenter and Kennedy [65]. The blue crosses indicate eigenvalues λ_i of the system operator $\mathcal{H}$ rescaled by the maximum time step $\Delta t_{\max}$. If the time step were any larger, at least one rescaled eigenvalue would lie outside of the stability contour. As a result, the numerical simulation would be numerically unstable. The data for this plot was kindly provided by Richard Diehl [70].*

resolved by the spatial discretisation. Auxiliary fields, for example to implement dispersive materials (see chapter 6) or perfectly matched layers (chapter 7), further modify the spectrum.

The size of the time step is strongly related to the quality of the spatial discretisation as well. Similarly to the popular FDTD algorithm (see appendix A), where the maximum time step is given by

$$\Delta t_{\max} = \frac{\Delta x}{c\sqrt{\dim}}, \tag{4.8}$$

the smallest length scale in the spatial discretisation influences the DGTD time step. The smallest length scale is given by the smallest node-to-node distance (measured within each element). The higher the polynomial order p (cf. Fig. 3.12) and the less regular[3] an element is, the smaller the maximum time step will be. In practice, one can use the estimate

$$\Delta t_{\max} = f \cdot d_{\min}(p) \cdot \min_{\triangle}(r_{\text{in}}^{\triangle}), \tag{4.9}$$

where $r_{\text{in}}^{\triangle}$ is the radius of the insphere of element $\triangle$, $d_{\min}(p)$ is the smallest distance between two nodes on an edge of the reference element, and f is a constant factor of the order of 1. This heuristic approach works for a wide range of applications. However, it rarely provides the optimal time step. Especially for higher orders $p > 4$, the maximally allowed time step is often underestimated

[3] An element is regular if all edges have the same length.

and, therefore, performance is lost. The estimate (4.9) can be improved by making the factor f order-dependent and use a fitting to empirical data to improve the performance. As an example, in Ref. [72], a p-dependence of

$$f(p) = 0.8 + 0.27p - 0.011p^2 \tag{4.10}$$

was proposed for Carpenter and Kennedy's LSRK scheme and the upwind parameter $\alpha = 1$. An extensive study of the maximum time step and its dependence on the polynomial order p, choice of numerical flux, element shape, and more was recently presented in Ref. [73].

For a more advanced, system-dependent estimate of the optimal time step, one can calculate a few extremal eigenvalues of the system operator using an iterative eigenvalue solver such as `ARPACK` [74]. From these eigenvalues, one can infer a scaling factor which ensures all of them are found within the stability contour of some given time step algorithm. Unfortunately, the computational cost associated with finding a sufficient number of eigenvalues can be quite considerable for large systems. Therefore, this approach is only feasible for medium-sized problems.

4.4 Optimised Runge-Kutta Schemes

The dependence of the time step on the shape matching between the eigenvalue spectrum and the Runge-Kutta stability contour suggests that performance gains can be expected from optimised Runge-Kutta schemes. For memory efficiency we restrict ourselves to LSRK schemes. The parameters A_i, B_i, and c_i are subject to a number of conditions stemming from the desired order of accuracy and the low-storage property. Increasing the number of stages s above the LSRK's order of accuracy introduces additional free parameters. As they affect the characteristic polynomial, they can be used to modify the shape of the stability contour [69, 70].

To compare the performance of different schemes, it is worthwhile to consider the normalised time step

$$\Delta\bar{t} = \frac{1}{s} \cdot \Delta t. \tag{4.11}$$

It represents the fraction of the total time step which can be attributed to a single stage. This is a reasonable measure because each stage requires a new evaluation of the right-hand side $\tilde{\mathbf{f}}(\tilde{\mathbf{q}}_{i-1},\, t_n + c_i\Delta t)$, see Eq. (4.5b). Hence, the normalised time step also represents a measure for time step per computational effort (CPU time). Diehl *et al.* have compared more than 50 established RK schemes to the commonly used Carpenter-Kennedy scheme [68]. For fourth-order accurate schemes, only marginal performance improvements could be found for the upwind parameter α close to one. For $\alpha \approx 0.6$, the five-stage third-order scheme by Spiteri and Ruuth [75] yields a speed-up factor of up to 1.6.

Very recently, Diehl *et al.* have developed a numerical optimisation method to generate LSRK schemes with tailored stability regions [69]. With this approach, a fourth-order 14-stage scheme has been generated which outperforms the 5-stage scheme by Carpenter and Kennedy by around 40-50%. It is remarkable that these performance gains can be achieved by merely changing a couple of numbers in an existing DGTD implementation, which typically features several thousand lines of code.

4.5 Advantages and Drawbacks

We conclude the discussion of DGTD by a short comparison with other existing time-domain methods. Let us start with the most established one. Compared to FDTD (appendix A), DGTD requires significantly more computations per degree of freedom because we have a number of—albeit small—matrix-vector products within each element. At the same time, DGTD computations usually require less degrees of freedom to obtain results of similar accuracy. This is an immediate consequence of the superior h- and p-refinement capabilities of DGTD. Small elements can be used to resolve small geometrical features. Even curvilinear elements (see section 6.3) can be employed where necessary. Depending on the system, DGTD can be considerably more efficient concerning both memory and CPU time than FDTD [38].

The finite volume (FVTD) and finite element time-domain (FETD) methods are conceptually quite similar to DGTD. In fact, the DGTD method discussed here reduces to a classical FVTD scheme for zeroth order ($p = 0$). However, in contrast to traditional finite volume methods, the DGTD method also offers a systematic approach to increase the order of the spatial discretisation to an arbitrarily high polynomial degree. This is also a big advantage over FETD, where the generation of higher order Nédélec elements [25, 26] is certainly not trivial. In addition, as mentioned before, a FETD approach requires the solution of a large (albeit sparse) system of linear equations each time step, since it is not possible to store and invert the global mass matrix. The main disadvantage of the DGTD method is that one has to deal with considerably more degrees of freedom due to the discontinuities across element interfaces. These discontinuities, however, also allow us to treat elements individually with a set of small matrix-vector multiplications. For this reason, DGTD is an ideal candidate for parallelisation on multiple cores, computers, and graphics cards [63].

To summarise, we find that DGTD is an explicit time-domain method which combines the efficiency of FDTD with unstructured meshes and higher order accuracy as known from finite element methods. Moreover, we are free to choose the time stepping algorithm. Customised low-storage Runge-Kutta algorithms appear particularly efficient. As a consequence, DGTD can accurately simulate systems which involve fine geometric features, strongly varying local fields, and different length and time scales. Hence, DGTD is certainly an excellent method for the time-domain simulation of nanophotonic systems.

5 Chapter 5
Discontinuous Galerkin Frequency-Domain Methods

In the previous chapter, we have introduced low-storage Runge-Kutta methods to discretise the time dependence in

$$\partial_t \tilde{\mathbf{q}}(t) = \mathcal{H}\tilde{\mathbf{q}}(t) + \tilde{\mathbf{q}}^{\text{source}}(t). \tag{4.3}$$

This chapter follows a different approach. Instead of looking at the full dynamics of the system, we restrict ourselves to time-harmonic solutions in the frequency domain. Thus, we employ the Fourier transformation rules (2.48) to obtain

$$-\mathrm{i}\omega\tilde{\mathbf{q}}(\omega) = \mathcal{H}\tilde{\mathbf{q}}(\omega) + \tilde{\mathbf{q}}^{\text{source}}(\omega). \tag{5.1}$$

Please recall that ω denotes the angular frequency[1] and that $\tilde{\mathbf{q}}$ consists of expansion coefficients for all field components and *all* elements in the mesh. As opposed to the time-domain quantities used in chapters 3 and 4, $\tilde{\mathbf{q}}(\omega)$ is a vector of frequency-dependent expansion coefficients, where each expansion coefficient is a *complex* number. The corresponding time-domain solution is given by

$$\tilde{\mathbf{q}}(t) = \Re\Big\{\tilde{\mathbf{q}}(\omega)\cdot\exp(-\mathrm{i}\omega t)\Big\}. \tag{5.2}$$

We would like to emphasise that in this thesis we want to solve the very same *first-order* system as in the time-domain. In particular, we will *not* derive a wave equation for one of the electromagnetic fields as, for example, done in Ref. [76]. Therefore, our approach will not lead to optimal performance, but to maximum consistency with an existing time-domain code.

The remainder of this chapter is structured as follows: First, we present general properties of frequency-domain methods and compare them against time-domain methods in section 5.1. Section 5.2 briefly deals with eigenvalue problems and illustrates problems which arise due to the DG discretisation. Source-driven problems, subsequently presented in section 5.3, are more suitable for the DG method. Among other topics, we discuss various techniques to solve the system of linear equations which the problem is eventually reduced to. These techniques are applied to three model systems in section 5.4. These systems allow us to compare the performance of various solvers in terms of CPU and memory consumption. Finally, this chapter concludes with section 5.5, which summarises the merits and disadvantages of discontinuous Galerkin frequency-domain (DGFD) methods.

[1] For brevity, we will simply refer to ω as "frequency".

5.1 General Characteristics of Frequency-Domain Simulations

Even though Maxwell's equations are formulated in terms of time-dependent electromagnetic fields, most experiments in the nanophotonic regime cannot resolve the time scales involved. Instead, such experiments usually yield time-averaged frequency-domain quantities, i.e., spectral information.

Obviously, such data is the natural output of frequency-domain solvers. Furthermore, it is straightforward to include frequency-dependent quantities as input parameters. This facilitates, for example, direct inclusion of material dispersion (section 2.1.1). In particular, we do not have to rely on analytical models for dispersion as presented in section 2.3; we can immediately employ $\epsilon(\omega)$ and $\mu(\omega)$ for a given frequency ω.

If we want to obtain a complete spectrum, we have to run separate simulations of the same system for each frequency of interest. This may or may not be an advantage depending on how many simulations we have to perform. If we are only interested in a few frequencies[2] or coarse spectra, frequency-domain solvers usually provide corresponding results faster than time-domain simulators (chapter 4). If we are interested in highly refined spectra, time-domain solvers may prove more efficient. Efficiency, however, strongly depends on the system under investigation. Time-domain methods require extremely long simulation times to obtain reliable results for high-quality resonators. In such cases, frequency-domain methods are often favourable. Once we have detected a resonance, frequency-domain solvers allow to refine the spectrum in this region simply by performing additional simulations. These simulations may be trivially run in parallel on multiple cores: There is no need to implement a parallelised computer program.

In terms of numerics, frequency-domain methods generally yield eigenvalue problems or systems of linear equations which need to be solved. Both possibilities are computationally challenging. Available resources, in particular main memory, often limit the applicability of frequency-domain solvers to medium-sized problems. More fundamentally, frequency-domain methods are limited to linear equations, since nonlinearities prohibit the superposition of solutions.

For linear materials, however, time-harmonic solvers provide an alternative route to obtain numerical results for a multitude of systems. Depending on the situation, either frequency-domain or time-domain simulations are better suited. In some cases, a time-domain simulation is used to calculate a broad-band spectrum. Once the resonances are identified, time-harmonic simulations are employed to evaluate mode profiles. In this sense, both approaches complement each other to the user's benefit.

5.2 Eigenvalue Problems

Let us first consider a system (characterised by the system operator $\mathcal{H}$ introduced in section 4.2) in the absence of external sources, i.e.,

$$\tilde{\vec{\mathrm{E}}}^{\mathrm{source}}(\omega) \equiv \tilde{\vec{\mathrm{H}}}^{\mathrm{source}}(\omega) \equiv 0. \tag{5.3}$$

In this case, Eq. (5.1) reduces to the eigenvalue problem

$$\mathcal{H}\tilde{\mathbf{q}}(\omega) = -\mathrm{i}\omega\tilde{\vec{\mathrm{q}}}. \tag{5.4}$$

[2] For example, we might be interested in the field distribution at a resonance frequency previously extracted from a time-domain simulation.

The solution to this problem consists of complex eigenvalues $-\mathrm{i}\omega$ and corresponding eigenvectors of expansion coefficients. Thus, ω is the result of an eigenvalue problem.

Though this problem looks simple enough, its numerical solution using iterative eigenvalue solvers such as `ARPACK` [74] is rather challenging. In particular, the spectrum of the discretised Maxwell operator $\mathcal{H}$ features a large number of spurious (i.e., unphysical) modes. For example, Fig. 4.1 shows the eigenvalues $\lambda = -\mathrm{i}\omega$ of $\mathcal{H}$ for a two-dimensional rectangular cavity. In principle, all eigenvalues should lie on the imaginary axis, i.e., they should correspond to real-valued frequencies ω. In particular, there should be a minimum frequency which translates to the largest possible wavelength in the system.

Instead, we observe eigenvalues with negative real parts which appear due to the artificial damping introduced by the upwind flux (see section 3.1.4). It can be difficult to distinguish the associated unphysical modes from physical modes if absorbing materials are present in the system. Furthermore, we even find eigenvalues on the real axis which correspond to purely complex ω. In particular, we find a very large number of zero eigenvalues which correspond to the null space of the DG operator and belong to stationary solutions of Maxwell's curl equations with non-vanishing divergence.

As a result, iterative eigenvalue solvers have difficulties to find physical eigenvalues. Eigenvalues with small frequencies are particularly difficult to find because of the large spurious null space. On the other hand, eigenvalues with large magnitudes often correspond to unphysical modes introduced by the spatial discretisation. Even when using the shift and invert transformation [77] to find eigenvalues near a complex value σ, great care on the choice of σ has to be taken in order to obtain physical eigenvalues. The underlying reason for the formation of spurious modes is that the DG discretisation does not enforce the *global* divergence conditions (2.8a) and (2.8b). To reduce the number of spurious modes, one could use a *locally* divergence-free basis on each element [78]. Combining such a scheme with a suitable numerical flux significantly reduces the null space and helps to distinguish physical modes from spurious ones for certain systems [27]. Nevertheless, calculating and identifying relevant eigenvalues still remains a non-trivial problem, for which the DG method is not ideally adapted. Thus, we will not study eigenvalue problems in more detail.

5.3 Driven Problems

The eigenvalue problem presented in the previous section investigates "natural" states of the electromagnetic fields, i.e., field patterns which are sustained even in the *absence* of external irradiation. On the other hand, the response of a system to incident light of a specific frequency ω is studied in the majority of experiments.

In this section we want to calculate field distributions of arbitrary systems illuminated by general illumination patterns $\tilde{\mathbf{q}}^{\mathrm{source}}(\omega)$. To this end, we rewrite Eq. (5.1) as

$$\Big(\mathcal{H} + \mathrm{i}\omega\Big) \cdot \tilde{\mathbf{q}}(\omega) = -\tilde{\mathbf{q}}^{\mathrm{source}}(\omega). \tag{5.5}$$

Apparently, the desired field distribution $\tilde{\mathbf{q}}(\omega)$ is the solution of a system of linear equations (SLE). Its right-hand side is given by the sources while the equations are characterised by the DG system operator plus a diagonal shift. In contrast to the eigenvalue problem, ω enters the calculation as an *input* parameter which determines the oscillation frequency of the external source. Thus, we will refer to Eq. (5.5) as the driven time-harmonic problem. For brevity's sake, we will often omit the

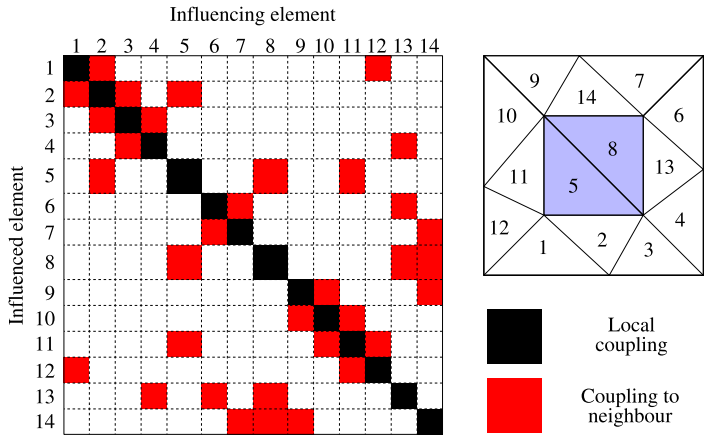


Figure 5.1: *Sketch of the system matrix for a two-dimensional cavity as a sample system. The matrix $\mathcal{H}'$ on the left-hand side consists of a number of rectangular blocks, each of which couples degrees of freedom of a certain element (column) to those of another element (row). However, just a few of these blocks actually feature non-zero entries. Blocks with at least one non-zero entry are coloured black (if the block lies on the diagonal of $\mathcal{H}'$) or red (the block is off-diagonal). The right-hand side features the corresponding mesh used to create the system matrix. Of the fourteen numbered elements, elements 5 and 8 support more degrees of freedom due to being filled with a dispersive material (see section* 6.2*).*

word "driven" in the future.[3] Consequently, strategies to solve Eq. (5.5) will be called time-harmonic solvers.

5.3.1 The Structure of the System Matrix

Before we discuss different strategies of how to solve Eq. (5.5), it is worthwhile to have a closer look at the properties of the system matrix

$$\mathcal{H}' = \mathcal{H} + \mathrm{i}\omega. \tag{5.6}$$

The shape of this matrix immediately follows from the semi-discrete form (3.30) derived in section 3.1.7. We identify the following contributions to $\mathcal{H}'$:

- The degrees of freedom associated with an element $\triangle$ are coupled to each other via the curl operator discretised in terms of derivative matrices $\mathcal{D}_i$. These matrices are square $n \times n$ matrices, where n is the number of nodes per element (Table 3.3). Please note that these matrices are not symmetric. This is an immediate consequence of the first-order form of Maxwell's equations we want to solve.
- Vectors of differences of expansion coefficients across the faces of an element, $\Delta\tilde{\vec{\mathrm{E}}}$ and $\Delta\tilde{\vec{\mathrm{H}}}$, occur in combination with matrices $\mathcal{M}^{-1}\mathcal{F}_f$ due to the numerical flux. This introduces coupling to the finite number of *neighbouring* elements. The matrices $\mathcal{M}^{-1}\mathcal{F}_f$ are of size $n \times n'$,

[3] Even though the eigenvalue problem Eq. (5.4) is a time-harmonic problem as well, confusion should be limited to a minimum because the discussion of eigenvalue problems has already been finished.

where n' is the number of points per face (Table 3.3).

- The term $\mathrm{i}\omega$ translates to a diagonal shift in the system matrix.

Some extensions, for example dispersive materials (section 6.2) and perfectly matched layers (chapter 7), introduce auxiliary fields which are coupled to the electromagnetic fields. This leads to additional, often diagonal $n \times n$ matrices in $\mathcal{H}'$.

What is common to all these contributions is that they connect degrees of freedom of each element with themselves or with those of its neighbours. There are no contributions which connect elements which do not share a common interface. For a d-dimensional system, each element has up to $d + 1$ neighbours. Since this number does not depend on the total number of elements, almost all entries of $\mathcal{H}'$ will be zero. Thus, $\mathcal{H}'$ is called a sparse matrix. The system matrix is visualised in Fig. 5.1 at hand of a small test system.

Properties of Sparse Matrices

Sparse matrices are an important concept in numerical analysis and often occur when discretising partial differential equations using finite differences or finite elements [26, 79]. In comparison to conventional dense matrices, storage and matrix-vector products are more efficient. For a dense matrix of size $N \times N$, N being the total number of unknowns, we need to store N^2 matrix entries. Likewise, the product of such a matrix and a vector of dimension N requires N^2 scalar multiplications and $N \cdot (N - 1)$ scalar additions. Such quadratic scaling impedes the use of dense matrices for larger systems.

In contrast, sparse matrices of the same size feature a limited, constant number of at most $n_{\mathrm{nz}} \ll N$ non-zero entries per row. Thus, the total number of non-zero entries in a sparse $N \times N$ matrix is given by $N \cdot n_{\mathrm{nz}}$. It is sufficient to store only these numbers plus their respective positions, i.e., their row and column indices. Similarly, matrix-vector products require at most $N \cdot n_{\mathrm{nz}}$ scalar multiplications and $N \cdot (n_{\mathrm{nz}} - 1)$ additions. It is important to appreciate that n_{nz} is independent of N, since it is essentially a measure of finite coupling. Increasing the number of elements, i.e., the problem size N, leads to an optimal *linear* scaling behaviour of both the storage requirements and the costs of matrix-vector products. This property makes sparse matrices extremely attractive for numerics, since it renders larger systems computationally accessible.

Sparse Block Matrices

In addition, we note that all contributions to the system matrix $\mathcal{H}'$ can be written as special matrices with n rows and either n or $n' < n$ columns. Depending on the type of the coupling, those matrices are either dense (derivatives), diagonal (auxiliary fields, diagonal shift), or have a column-shaped structure (numerical flux). This is demonstrated in Fig. 5.2, which provides a zoomed view on those matrix elements which correspond to the rows that belong to element 8 in Fig. 5.1. We observe that the numerical flux leads to a structurally non-symmetric system matrix, i.e., the distribution of non-zero entries of $\mathcal{H}'$ and $\mathcal{H}'^{,T}$ differs.

In our code, we exploit this structure in an adaptive matrix generation algorithm. Instead of storing $(\mathrm{row}, \mathrm{column}, \mathrm{value})$ triples for each matrix entry, we create a sparse *block* matrix: The $N \times N$ matrix is divided into a set of smaller $n \times n$ matrices. For each matrix we need to store a single

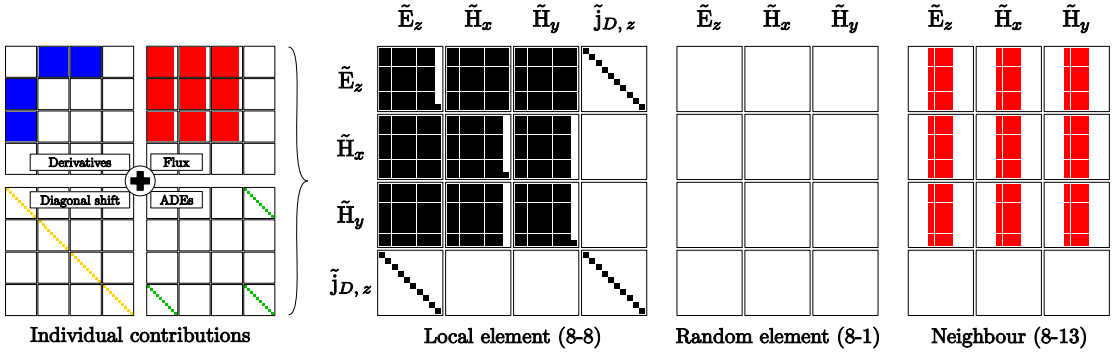


Figure 5.2: *Details on the structure of the system matrix $\mathcal{H}'$. In principle, this figure is a more detailed view of row 8 of Fig.* 5.1 *for $p = 3$. In contrast to the latter figure, coloured rectangles indicate* individual *non-zero matrix entries here. Spatial derivatives, the numerical flux, auxiliary differential equations (ADEs) for a Drude model (section* 6.2*) and the diagonal shift contribute matrix entries to the diagonal block, which corresponds to the black 8-8 block in Fig.* 5.1*. Neighbours influence the chosen element via the numerical flux across the mutual interface. As an example, the red 8-13 block has been chosen. All other elements do not contribute non-zero elements to a particular row of the system matrix $\mathcal{H}'$.*

(row, column) pair. The indices of the remaining entries in each matrix block are easily derived from the respective reference indices of the blocks. Our implementation supports automatic and dynamic creation of blocks, i.e., we do not have to specify the positions of non-zero entries upfront. Instead, we merely add values at certain positions of the matrix. Existing values are modified while new blocks are created to store values at newly populated positions. Moreover, our storage scheme distinguishes three different types of blocks (dense blocks, diagonal blocks, column-shaped blocks) to suit the requirements of the contributions to $\mathcal{H}'$. The resulting data structure is very efficient in terms of storage and allows for maximum flexibility and expendability, which is an essential feature when combining the standard DG discretisation with auxiliary fields (section 6.2, chapter 7).

Storage Order of Expansion Coefficients

A last remark concerns the ordering of the field expansion coefficients in $\tilde{\mathbf{q}}$. It appears natural to store all expansion coefficients of a single field component for a single element in a contiguous memory area. Doing so leaves us with the task of mixing field components and elements (see Fig. 5.3). In principle, one can either group expansion coefficients of one field component for all elements, i.e.,

$$\tilde{\mathbf{q}} = \left(\tilde{\mathrm{E}}_x^{\Delta 1}, \tilde{\mathrm{E}}_x^{\Delta 2}, \ldots, \tilde{\mathrm{E}}_y^{\Delta 1}, \tilde{\mathrm{E}}_y^{\Delta 2}, \ldots, \tilde{\mathrm{E}}_z^{\Delta 1}, \tilde{\mathrm{E}}_z^{\Delta 2}, \ldots \right), \tag{5.7}$$

or one can group all field components of a single element, i.e.,

$$\tilde{\mathbf{q}} = \left(\tilde{\mathrm{E}}_x^{\Delta 1}, \tilde{\mathrm{E}}_y^{\Delta 1}, \tilde{\mathrm{E}}_z^{\Delta 1}, \ldots, \tilde{\mathrm{E}}_x^{\Delta 2}, \tilde{\mathrm{E}}_y^{\Delta 2}, \tilde{\mathrm{E}}_z^{\Delta 2}, \ldots \right). \tag{5.8}$$

The second version seems preferable since it keeps element-local data together. This improves CPU cache usage and eases data handling for computations on graphics cards. In addition, it simplifies

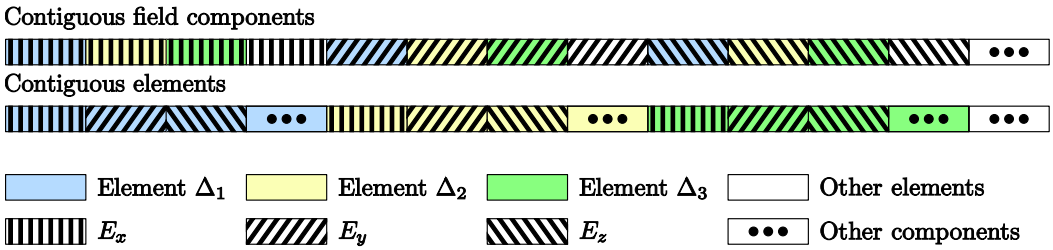


Figure 5.3: *Storage order of expansion coefficients. Tasked with storing expansion coefficients for a number of field components on a number of elements, one can either store individual* field components *(top display) or all components of individual* elements *(bottom display) in contiguous blocks of memory. Usually, the latter is preferable (see section* 5.3.1*).*

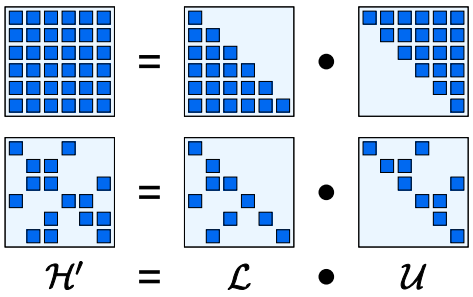


Figure 5.4: *Decomposition of the matrix $\mathcal{H}'$ into lower and upper triangular matrices. The top panel illustrates the result for a dense matrix, while the lower panel shows the decomposition for a sparse matrix. Each small square represents a non-zero matrix entry. Please note that the matrices used in this sketch bear little similarity to the actual system matrix.*

the use of block Jacobi preconditioners later presented in section 5.3.3. Figures 5.1 and 5.2 employ this storage order.

5.3.2 Direct and Iterative Methods

Let us now return to the actual solution of Eq. (5.5). From the discussion in the previous section we know that the system matrix is sparse, structurally non-symmetric, and complex-valued. For such matrices, efficient SLE solvers are available [79–83]. These solvers can be roughly divided into two categories: direct and iterative solvers.

Direct Solvers

Direct solvers attempt to decompose the system matrix into a product of a lower triangular matrix $\mathcal{L}$ and an upper triangular matrix $\mathcal{U}$ in a process called LU factorisation:

$$\mathcal{H}' = \mathcal{L}\mathcal{U}. \tag{5.9}$$

Figure 5.4 illustrates sample factorisations for small matrices. The main challenge is not to lose the sparsity property of $\mathcal{H}'$, i.e., to minimise the number of non-zero entries in both $\mathcal{L}$ and $\mathcal{U}$. To this end, sophisticated column-reordering strategies are employed [80, 81]. Once the triangular matrices are known, the problem

$$\mathcal{H}'\tilde{\mathbf{q}}(\omega) = \tilde{\mathbf{q}}^{\text{source}}(\omega) \tag{5.10}$$

reduces to the—then trivial—forward and backward substitutions

$$\begin{aligned} \mathcal{L}\tilde{\mathbf{p}}(\omega) &= \tilde{\mathbf{q}}^{\text{source}}(\omega), \\ \mathcal{U}\tilde{\mathbf{q}}(\omega) &= \tilde{\mathbf{p}}(\omega). \end{aligned} \tag{5.11}$$

A key issue with direct solvers is storing the triangular matrices, which suffer from a large fill-in of entries as compared to the original matrix $\mathcal{H}'$. Larger systems can only be simulated on computers equipped with dozens of gigabytes of memory. Once these resources are available, direct solvers provide fast and highly accurate results. In this thesis we employ the popular direct solver libraries `PARDISO` [81][4] and `UMFPACK` [80].

Iterative Methods

Iterative methods follow a different approach to find a solution to Eq. (5.10) [79, 82–84]. Instead of factorising $\mathcal{H}'$, they start with an initial guess $\tilde{\mathbf{q}}_0$ and try to reduce the residuum

$$\mathcal{H}'\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}^{\text{source}} \tag{5.12}$$

in an iterative procedure. Here, the index "i" represents the i-th iteration step. The iterative procedure stops once the condition

$$\left\| \mathcal{H}'\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}^{\text{source}} \right\| \leq \text{tol} \cdot \left\| \tilde{\mathbf{q}}^{\text{source}} \right\| \tag{5.13}$$

is met. Here, the symbol $\| \cdot \|$ denotes the Euclidean norm of a vector and the tolerance tol is a positive real number.

To update the temporary solution $\tilde{\mathbf{q}}_i$, iterative solvers rely on matrix-vector products between the system matrix $\mathcal{H}'$ and (auxiliary) vectors of expansion coefficients. This is conceptually similar to the low-storage Runge-Kutta time stepping algorithm presented in section 4.2. In both cases it is not necessary to store $\mathcal{H}'$ explicitly, as long as we can efficiently evaluate its product with an arbitrary vector. This is indeed the case as the discussion in section 3.3.1 shows. There are, however, two differences compared to the matrix-vector product which is used in the Runge-Kutta scheme: By splitting the matrix-vector product with $\mathcal{H}'$ into two parts, i.e.,

$$\mathcal{H}'\tilde{\mathbf{q}}(\omega) = \mathcal{H}\tilde{\mathbf{q}}(\omega) + \mathrm{i}\omega\tilde{\mathbf{q}}, \tag{5.14}$$

we see that we have to multiply $\mathcal{H}$ with a *complex*-valued vector $\tilde{\mathbf{q}}(\omega)$. Furthermore, we have to take care of the diagonal shift, which translates to a simple scaling of the original vector $\tilde{\mathbf{q}}(\omega)$. Nevertheless, the DG method provides us with a very efficient scheme to perform matrix-vector products.

In our computer program we employ a custom implementation[5] of the stabilised biconjugate gradients method `BiCGstab(l)` [82] and an implementation of the restarted generalised minimal

[4] The version of `PARDISO` we use is included in Intel's Math Kernel Library (`MKL`).

[5] Courtesy of Jens Niegemann.

residual method (GMRES) [83] as provided by Intel's Math Kernel Library (MKL). Unfortunately, the number of iterations, and thus matrix-vector products, required to reach the target tolerance strongly depends on the physical system, the size of the elements in the mesh, and the polynomial order p. In section 5.3.3 we will introduce techniques which reduce the influence of the system. Furthermore, the desired level of accuracy tremendously influences the simulation times, though we are free to specify what we understand by "accurate simulations".

Finally, iterative solvers are quite sensitive to the initial guess $\tilde{\mathbf{q}}_0$. The naïve choice of all initial expansion coefficients being zero often represents a poor guess. For scattering problems, it can prove advantageous to use the fields as created by the time-harmonic source in the absence of scatterers as an initial guess. This is especially the case in the limit of weak scattering. For example, a plane wave can be a good start when one is interested in Mie scattering of a sphere.

Multigrid Methods

Due to the impact of the initial set of expansion coefficients on the solution time of iterative solvers, it is highly desirable to obtain a system-dependent, automated, and educated initial guess. This can be achieved using multigrid methods.[6] The idea is to simulate the very same system using a set of grids, where each grid is a refined version of the previous one. The problem on the coarsest grid can be solved using direct methods. Here, we have considerably less degrees of freedom than in the case of the finest grid. Consequently, the memory-consumption of direct solvers is usually acceptable. In the simplest version of multigrid methods, the solution of the coarser grid is interpolated (or prolongated) to the next finer one. There, an iterative solver is applied to reduce the residual error. The procedure is reapplied until the desired level of accuracy is reached.

Finer grids can be created in two ways. First, one can refine the mesh, i.e., one can use more and smaller elements (h-refinement). Secondly, one can increase the number of degrees of freedom per element by using higher-order basis functions (p-refinement). Since h-refinement requires additional algorithms and higher polynomial orders are one of DG's strengths, p-refinement is often the favourable choice.

5.3.3 Preconditioning

Usually, large time-harmonic problems cannot be treated by direct solvers because of vast memory requirements. Iterative solvers, on the other hand, have very low memory requirements at the cost of increased run times. Besides the number of unknowns N and the quality of the initial guess, the run time essentially depends on the condition number $\kappa(\mathcal{H}')$ of the system matrix $\mathcal{H}'$ [79]. For the spectral norm, it can be expressed as

$$\kappa(\mathcal{H}') = \frac{\max\limits_i \sigma_i}{\min\limits_i \sigma_i}, \tag{5.15}$$

the ratio between the maximum and the minimum absolute values of the system operator's singular values σ_i. Intuitively, the condition number measures how much the solution of an SLE changes if

[6] It should be noted that the techniques presented here are by no means as sophisticated as those presented in Ref. [79], for example.

the system matrix or the right-hand side are perturbed [79]. The larger the condition number, the more iterations it takes to reach a desired level of accuracy. The idea behind preconditioning is not to solve Eq. (5.10), but the equivalent problem

$$\mathcal{P}^{-1}\mathcal{H}'\tilde{\mathbf{q}} = \mathcal{P}^{-1}\tilde{\mathbf{q}}^{\text{source}}. \tag{5.16}$$

Please note that Eq. (5.16) has the very same solution $\tilde{\mathbf{q}}$ as Eq. (5.10). The $N \times N$ matrix $\mathcal{P}$ is called a preconditioner. If

$$\kappa(\mathcal{P}^{-1}\mathcal{H}') \ll \kappa(\mathcal{H}') \tag{5.17}$$

and the cost of multiplying $\mathcal{P}^{-1}$ with a vector is sufficiently small, than the reduced number of iterations outweighs the additional computational costs per iteration. As a consequence, iterative solvers combined with preconditioners may need considerably less time to obtain an approximate solution to the system of linear equations than iterative solvers for the bare matrix $\mathcal{H}'$.

The minimum condition number is obtained for $\mathcal{P} = \mathcal{H}'$. However, in this case one would require the inverse of $\mathcal{H}'$—which would be the solution of the actual problem (5.10) for any source vector. If such an inverse was readily available, one would not have to resort to iterative solvers in the first place. Instead, preconditioners should *approximate* the system operator in a way that allows an easy computation (and probably storage) of its inverse, while the preconditioned matrix $\mathcal{P}^{-1}\mathcal{H}'$ should be as close to the unit matrix as possible.

Fulfilling and balancing these requirements—especially for non-symmetric complex problems like the present one—is not easy and sometimes more art than science. In the following, we will present a few simple preconditioners. Corresponding comparisons of performance characteristics are later presented in section 5.4.

Jacobi and Block Jacobi Preconditioners

The most simple preconditioner, the so-called Jacobi preconditioner, is given by [79]

$$\mathcal{P}_{ij} = \begin{cases} \mathcal{H}'_{ii} & \text{for} \quad i = j \\ 0 & \text{for} \quad i \neq j \end{cases}. \tag{5.18}$$

Unfortunately, the system operator of the DG method is not diagonally dominant, which limits the effectiveness of the Jacobi preconditioner. On the other hand, it is simple to implement and memory-efficient, since it only requires the storage of $N_\triangle \cdot n$ numbers, where $N_\triangle$ is the total number of elements in the mesh.

A block Jacobi preconditioner extends this approach by dividing the available indices into mutually distinct contiguous index sets I_k:

$$\mathcal{P}_{ij} = \begin{cases} \mathcal{H}'_{ij} & \text{for} \quad i \in I_k,\, j \in I_k \\ 0 & \text{for} \quad i \in I_k,\, j \in I_l,\, l \neq k \end{cases}. \tag{5.19}$$

Thus, $\mathcal{P}$ represents those values of $\mathcal{H}'$ which lie on square blocks on its diagonal. The size of the blocks is given by the number of elements in each index set I_k. Naturally, the block size influences the performance of the preconditioner. For our DG scheme, there are three obvious choices.

Choice of Block Size

For a start, the blocks might comprise the n expansion coefficients of a single field component in a single element. In this case, the entries in each block typically correspond to coupling of a field component with itself as introduced by the numerical flux (compare Fig. 5.2). Our experience shows that this choice only leads to minor speed-ups for iterative solvers.

A more advanced choice involves index sets which consist of *all* degrees of freedom of *single* elements, respectively. These blocks are $[n \cdot (N_c + N_{\mathrm{ADE}})] \times [n \cdot (N_c + N_{\mathrm{ADE}})]$ matrices, where N_c denotes the number for electromagnetic field components[7] and N_{ADE} represents the number of auxiliary fields in the respective element (see section 6.2 and chapter 7). Seen from a physical perspective, this preconditioner corresponds to the solution of Maxwell's equations in an element-local fashion. Hence, it is very similar to the original idea of the DG discretisation. We will call this choice an element Jacobi preconditioner.

Finally, we can choose index sets which merely comprise the *electromagnetic* field expansion coefficients of single elements, even if auxiliary fields are present. As a result, all blocks have the same size and represent $N_c n \times N_c n$ matrices. The remaining indices which correspond to auxiliary fields can be preconditioned using simple Jacobi preconditioners or the inverse diagonal shift $(\mathrm{i}\omega)^{-1}$. We will call this choice a Maxwell block Jacobi preconditioner.

Block Jacobi preconditioners with suitably chosen block sizes significantly speed up iterative solvers (see section 5.4). It should be pointed out, though, that storing these blocks may be quite memory-consuming. For example, a $p = 3$ discretisation in three dimensions ($n = 20$, $N_c = 6$) leads to 120×120 blocks for the Maxwell block Jacobi preconditioner. These numbers require $225\,\mathrm{kB}$ of memory to store the complex entries in double precision.[8] This severely reduces the practicability of block Jacobi preconditioners for larger systems.

Incomplete Factorisations

Incomplete factorisation preconditioners follow a different approach. For example, incomplete LU (ILU) factorisations approximate the system matrix by a product of lower and upper triangular matrices $\mathcal{L}'$ and $\mathcal{U}'$, respectively:

$$\mathcal{H}' = \mathcal{L}'\mathcal{U}' + \mathcal{R}. \tag{5.20}$$

Because these matrices are *not* the complete factors $\mathcal{L}$ and $\mathcal{U}$ from Eq. (5.9), there remains a residual matrix $\mathcal{R}$ when reconstructing $\mathcal{H}'$. There are several ways to compute ILU factorisations. The basic algorithm ILU0 uses a predetermined sparsity pattern for both $\mathcal{L}'$ and $\mathcal{U}'$: These matrices only have non-zero entries where $\mathcal{H}'$ is non-zero as well. Other implementations store more elements to achieve a better approximation of $\mathcal{H}'$ at the cost of additional memory. A more detailed description together with algorithms can be found in Ref. [79].

Once an ILU factorisation is available, we use it as a preconditioner:

$$\mathcal{P} = \mathcal{L}'\mathcal{U}'. \tag{5.21}$$

[7] For two-dimensional simulations we typically have $N_c = 3$, while for three-dimensional systems $N_c = 6$.
[8] Half the memory is needed if we work with single precision preconditioners.

A preconditioner's basic operation is the application of its inverse to an arbitrary vector $\tilde{\mathbf{x}}$. The result of this application is given by

$$\tilde{\mathbf{y}} \equiv \mathcal{P}^{-1}\tilde{\mathbf{x}}. \tag{5.22}$$

This, however, is equivalent to solving the SLE

$$\mathcal{P}\tilde{\mathbf{y}} = \mathcal{L}'\mathcal{U}'\tilde{\mathbf{y}} = \tilde{\mathbf{x}}. \tag{5.23}$$

In analogy to Eq. (5.11), $\tilde{\mathbf{y}}$ is easily obtained via forward and backward substitutions. Thus, ILU preconditioners can be efficiently applied to any vector of expansion coefficients $\tilde{\mathbf{x}}$.

5.4 Performance Comparison

Having introduced the numerical techniques, it is time to demonstrate the computational capabilities of the time-harmonic solvers. Eigenvalue problems are not considered due to the issues discussed in section 5.2.

In the following, we consider three test cases. The scattering of light by an infinite cylinder is investigated in the first one. Mie theory provides an analytical reference solution for this system. In a second test, we investigate a slotted microresonator coupled to two waveguides. Finally, we examine the scattering of light by a sphere as a three-dimensional model problem.

5.4.1 Scattering of Light by an Infinite Cylinder (2D)

As a first test system we consider an infinite dielectric cylinder of radius $r = 150\,\mathrm{nm}$ embedded in vacuum (Fig. 5.5). The cylinder is made of a dielectric material with permittivity $\epsilon = 2.25$ and permeability $\mu = 1$. Due to the translational invariance of the system along the cylinder axis we reduce it to a two-dimensional problem (see section 2.1.5). Of the two available polarisations we choose to investigate the transverse-electric (TE) polarisation. The cylinder is illuminated by a monochromatic plane wave with wavelength λ, which is injected into the system by means of the total-field/scattered-field (TF/SF) technique presented in section 6.1.1.[9] To this end, a TF/SF interface is included in the system and divides the computational domain into a total-field (TF) and a scattered-field (SF) region, respectively.

On the boundary of the computational domain we apply a Silver-Müller boundary condition to absorb outgoing radiation. In addition, a uniaxial perfectly matched layer (UPML, see chapter 7) surrounds the physical region of interest. UPMLs are implemented in the DG method by adding auxiliary equations and degrees of freedom to the standard system of equations given by the discretised version of Maxwell's curl equations. Using the meshing tool `NETGEN`, we generate a triangular mesh which consists of exactly 2,000 straight-sided elements. Thus, the cylinder is not represented perfectly, but well enough to serve as a system for performance tests.

Spectrum

Before we present an actual comparison of performance characteristics, we want to establish the correctness of our code. To this end we calculate the scattering cross section spectrum of the di-

[9] To preserve the translational invariance, the incidence is normal with respect to the cylinder's axis of symmetry.

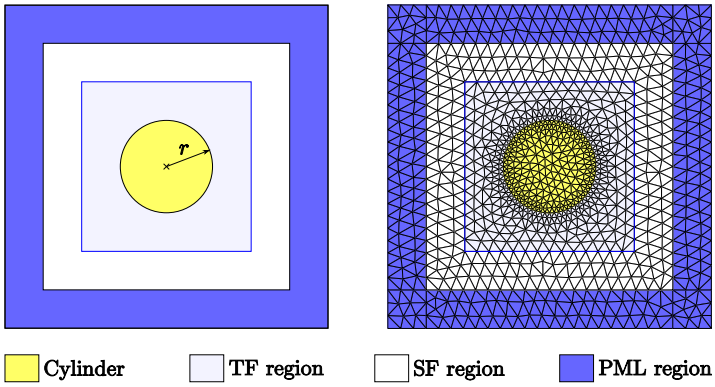


Figure 5.5: *The setup used for the calculations in section* 5.4.1. *The left-hand panel shows the computational domain. A cylinder of radius r made of a material with permittivity ϵ (yellow shade) is embedded in vacuum. The total-field/scattered-field interface (blue contour) divides the domain into the total-field (TF region, light blue shade) and the scattered-field (SF region, white) region (see section* 6.1.1*). A uniaxial perfectly matched layer (blue shade) surrounds the physical region and absorbs outgoing radiation (see chapter* 7*). The right-hand panel shows the mesh we use for our simulations, which has been generated with the aid of* `NETGEN`*. It only features straight-sided elements.*

electric cylinder for wavelengths between 60 and 600 nm. It is easily obtained by integrating the time-averaged scattered-field Poynting vector (2.75) on the TF/SF contour and normalising it to the incident power flux according to Eq. (2.77). Please note that in this two-dimensional system we can only specify the scattering cross section per unit length of the cylinder. We employ the direct solver `UMFPACK` to solve the occurring SLEs for the unknown field expansion coefficients up to machine precision. Figure 5.6 presents the resulting spectra for polynomial orders $p \leq 5$. Furthermore, the figure includes an analytical reference solution obtained using Mie theory [50, 52].

We observe that the high-wavelength part of the spectrum is well reproduced for all p. Visible deviations start to emerge for $\lambda \lesssim 200\,\mathrm{nm}$. In this range, the wavelength is relatively small compared to the size of the elements in the mesh (Fig. 5.5). As a result, lower order polynomials, in particular $p = 2$, cannot accurately resolve the electromagnetic fields. A closer inspection of the relative error

$$\text{rel. error}(\lambda) = \frac{C_{\text{scat}}^{\text{num},\,p}(\lambda)}{C_{\text{scat}}^{\text{ref}}(\lambda)} - 1 \tag{5.24}$$

is facilitated by Fig. 5.7, where $C_{\text{scat}}^{\text{num},\,p}$ is the scattering cross section calculated using DGFD for order p and $C_{\text{scat}}^{\text{ref}}$ is the corresponding value obtained by Mie theory.

Higher polynomial orders provide significantly more accurate results in the lower wavelength regime. For larger wavelengths, all orders roughly produce the same, but non-vanishing deviation. This residual error is caused by the misrepresentation of the cylinder using straight-sided elements. Smaller elements or curvilinear ones (section 6.3) will reduce this effect. Nevertheless, it is justified to conclude that our frequency-domain implementation works and produces valid results.

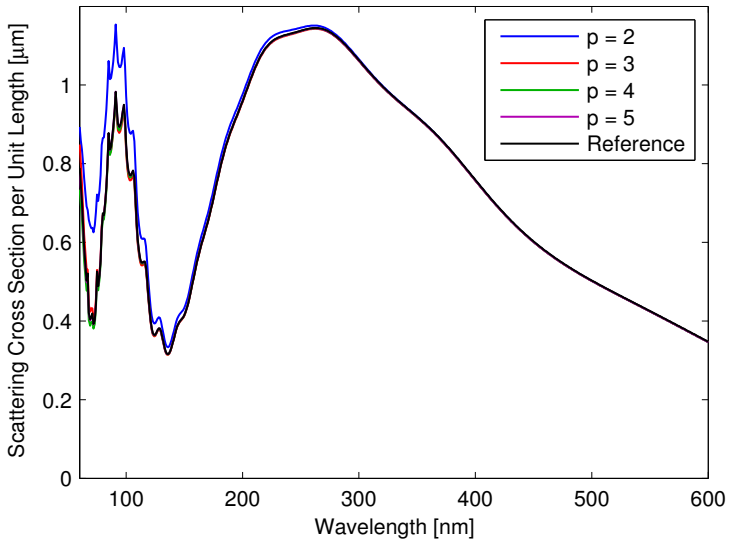


Figure 5.6: *Scattering cross section spectra of a cylinder calculated for various polynomial orders p. The figure also includes a reference solution calculated using Mie theory.*

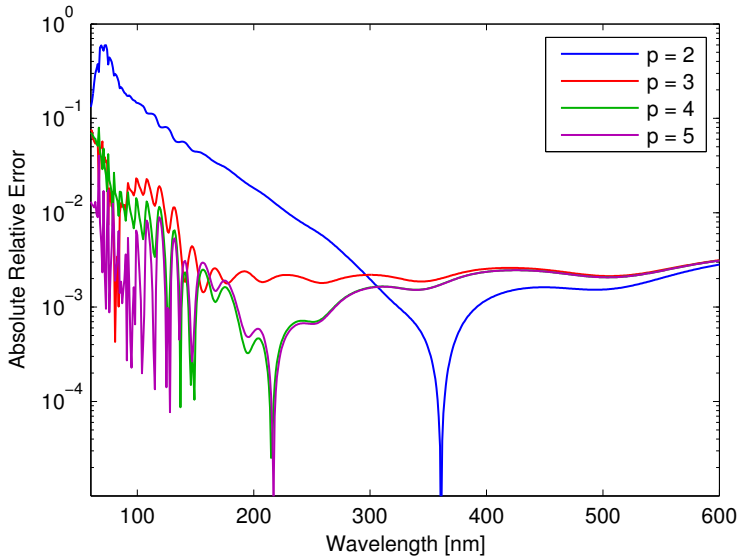


Figure 5.7: *Relative error of the spectra Spectrum of a cylinder for various polynomial orders. An analytical reference solution is given by Mie theory. The relative error is defined according to Eq.* (5.24)*. Please note the semi-logarithmic scale.*

Direct vs. Iterative Solvers

Let us now come to a first series of tests in which we want to determine the performance of various solvers. Specifically, we compare the performance of two direct solvers, `UMFPACK` [80] and `PARDISO` [81], with the iterative solvers `BiCGstab(l)` ($l = 8$) [82] and `GMRES` (restarted after 40 iterations) [83]. For the iterative solvers we choose a tolerance of $\mathrm{tol} = 10^{-4}$ in Eq. (5.13). `BiCGstab(l)` comes in four variations. First, we test the pure algorithm without a preconditioner. In addition, we combine `BiCGstab(l)` with three different preconditioners (element Jacobi, Maxwell block Jacobi, ILU0). `GMRES` is accompanied by a Maxwell block Jacobi preconditioner.

For each solver, we measure the pure time needed to create and solve the system of equations for a *single* frequency which corresponds to $\lambda = 80\,\mathrm{nm}$. This frequency will be investigated in all subsequent studies. The time required for initialisation (mesh loading, building the physical system, precalculations, etc.) is not considered and is below $2\,\mathrm{s}$ for this test case. Furthermore, we record the total memory consumption of the simulation program. In contrast to the time measurements, this includes "overhead" not immediately related to the actual system of equations (material parameters, mesh data, etc.). For the iterative solvers we monitor the number of iterations and the number of matrix-vector products with the (preconditioned) system matrix $\mathcal{H}'$ as well. For all computations we used a computer equipped with an Intel® Core™ 2 Quad Q9300 CPU running at $2.5\,\mathrm{GHz}$ and $8\,\mathrm{GB}$ main memory. The results of our measurements for various polynomial orders p (see section 3.1.6) are listed in Table 5.1.

Comparing `UMFPACK` and `PARDISO` with `BiCGstab(l)` without preconditioner, we observe that direct and iterative solvers represent different extrema of the time/memory spectrum. While direct solvers are very fast, they require enormous amounts of memory. For example, `PARDISO` calculates the solution to the $p = 7$th order problem in $170\,\mathrm{s}$, but requires about $4.5\,\mathrm{GB}$ of memory. On the other hand, `BiCGstab(l)` solves the same problem with just $114\,\mathrm{MB}$ of memory in about 14 minutes. Comparing both direct solvers with each other, `UMFPACK` appears to be slightly more memory-efficient than `PARDISO`. At the same time, it is considerably faster. Thus, `UMFPACK` is our direct solver of choice.

Addressing the question whether we can combine the speed of direct solvers with the memory-efficiency of iterative solvers, we find a trade-off is possible with preconditioners. While the iteration count increases with p for standard `BiCGstab(l)`, it is approximately stable if we use element Jacobi (around 35) or Maxwell block Jacobi (around 38) preconditioners. The combination with an incomplete LU factorisation (ILU0) shows erratic behaviour for orders above $p = 3$. For $p \geq 5$ it even fails to converge within 20,000 iterations. Thus, we focus our discussion on the more reliable element Jacobi and Maxwell block Jacobi preconditioners. Even though the iteration count—and consequently the number of matrix-vector products—is a little lower for the element Jacobi preconditioner, it is less efficient than the Maxwell block Jacobi preconditioner. Apparently, the advantage of better preconditioning (fewer iterations) is outweighed by the increased computational costs of larger matrix-vector products due to the presence of auxiliary fields in some elements (cf. Fig. 5.2).

Remarkably, `BiCGstab(l)` with a Maxwell block Jacobi preconditioner can even compete with the performance of direct solvers. In particular, it outperforms `PARDISO` for all orders except $p = 2$. Compared to `UMFPACK` for $p = 7$, it is just 30% slower, but requires only a seventh of its memory. Hence, `BiCGstab(l)` with a Maxwell block Jacobi preconditioner seems an excellent compromise between speed and memory usage. Finally, we compare `GMRES` combined with a Maxwell

		Polynomial order					
		2	3	4	5	6	7
Degrees of freedom		40,296	67,160	100,740	141,036	188,048	241,776
UMF-PACK	Memory [MB]	347	793	1,300	1,960	2,921	3,413
	Time [s]	3.6	11.0	19.8	31.9	50.1	75.0
PAR-DISO	Memory [MB]	357	729	1,268	2,080	3,158	4,685
	Time [s]	5.7	13.7	30.3	57.6	98.8	170.1
BiCGstab(l)	Memory [MB]	38	49	62	77	94	114
	Time [s]	24.1	63.3	137.4	257.3	467.4	814.7
	Iterations	183	300	428	564	739	907
	MV products	2,933	4,806	6,854	9,031	11,831	14,520
BiCGstab(l) +El. Jacobi	Memory [MB]	52	86	144	238	379	583
	Time [s]	6.7	11.2	22.3	37.3	70.2	105.4
	Iterations	33	29	32	33	40	38
	MV products	533	468	516	532	644	612
BiCGstab(l) +Max. Jacobi	Memory [MB]	49	78	125	200	313	474
	Time [s]	6.3	11.6	20.8	35.0	65.6	96.4
	Iterations	33	33	34	35	43	40
	MV products	534	532	547	564	693	644
BiCGstab(l) +ILU0	Memory [MB]	91	170	312	523	827	1,281
	Time [s]	10.5	71.4	2,985.5	—	—	—
	Iterations	18	60	1,483	—	—	—
	MV products	292	965	23,735	—	—	—
GMRES +Max. Jacobi	Memory [MB]	62	99	158	246	373	551
	Time [s]	16.2	38.5	74.7	128.3	204.1	331.0
	Iterations	1,131	1,410	1,442	1,446	1,499	1,645
	MV products	1,160	1,446	1,479	1,483	1,537	1,687

Table 5.1: *Performance of direct and iterative solvers for the scattering of light by an infinite dielectric cylinder. For all solvers we measure the time required to construct and solve the SLE. Memory consumption is measured for the entire program including overhead for meshes, precached values, and more.* `BiCGstab(l)` *and* `GMRES` *are combined with various types of preconditioners (element Jacobi, Maxwell block Jacobi, ILU0). For these, we also list the number of iterations and matrix-vector products required until the solution is sufficiently converged.*

block Jacobi preconditioner with this successful combination. However, we find that GMRES requires significantly more matrix-vector products than BiCGstab(l) to obtain sufficiently converged results.[10] Consequently, this leads to longer simulation times. In addition, GMRES requires more main memory. We have tested GMRES with other preconditioners as well, but find that BiCGstab(l) is consistently faster and more memory-efficient. Thus, it is our iterative solver of choice.[11]

Influence of the Initial Guess

As pointed out earlier, a good initial guess leads to rapid convergence of iterative solvers. Since we have identified BiCGstab(l) together with the Maxwell block Jacobi preconditioner as the most efficient combination, we restrict the following study to this solver.

Let us consider two different initial guesses which we would like to use for the standard cylinder setup. The first guess is the trivial one, i.e., we assume all electromagnetic and auxiliary fields to be zero *everywhere*. The second choice is more involved and was tacitly employed for the calculations presented in the previous section: In the scattered-field region (including the PML boundary), all fields remain zero. In the total-field region, however, the electric and the magnetic field components assume the respective values of the unperturbed plane wave which illuminates the cylinder. Disregarding errors introduced by the spatial discretisation, this initial guess is the solution of the "scattering" problem of a vacuum cylinder. Thus, we expect this "TF/SF" initial guess to be a fairly good choice for the scattering of light by cylinders with low refractive indices $n = \sqrt{\epsilon}\,(\mu \equiv 1)$. Both initial guesses along with the scattered field and the final solution of the problem are visualised in Fig. 5.8.

To verify this expectation we perform additional simulations for the standard cylinder setup for both types of initial guesses. This time, however, we vary the cylinder's refractive index. Figure 5.9 compares the corresponding numbers of iterations. We observe the tendency that more iterations are required for higher refractive indices. The reasons behind this behaviour include the following effects:

- The differences between the initial guess and the actual solution of the SLE are more pronounced for higher n.
- The condition number of $\mathcal{H}'$ changes with n.

In this simple experiment we cannot distinguish between both effects. Furthermore, it is likely that the fixed wavelength $\lambda = 80\,\mathrm{nm}$ crosses some resonances of the cylinder as we vary its refractive index. This may influence the iteration count beyond $n = 2.2$.

Comparing the performance of both initial guesses with each other, we find that the TF/SF guess usually requires fewer iterations than the zero initial guess, in particular for $n \leq 2.1$. For higher refractive indices we cannot predict which guess leads to better performance. Across the entire

[10] There is not too much point in comparing the iteration count of different solvers because iterations require different amounts of work, i.e., matrix-vector products per iteration.

[11] It should be noted that GMRES in principle leads to ideal convergence, i.e., it converges in at most N steps, where N is the number of unknowns in the system [79]. This, however, requires that we store an individual vector of expansion coefficients for *each* iteration. In this study we restart the GMRES solver after 40 iterations, and thus we have to store 40 vectors of expansion coefficients. While this limits the memory consumption, it also impedes the performance. The number of iterations performed until the algorithm is restarted is a free parameter which allows a trade-off between memory and speed. Forty iterations, however, already seems a rather large number.

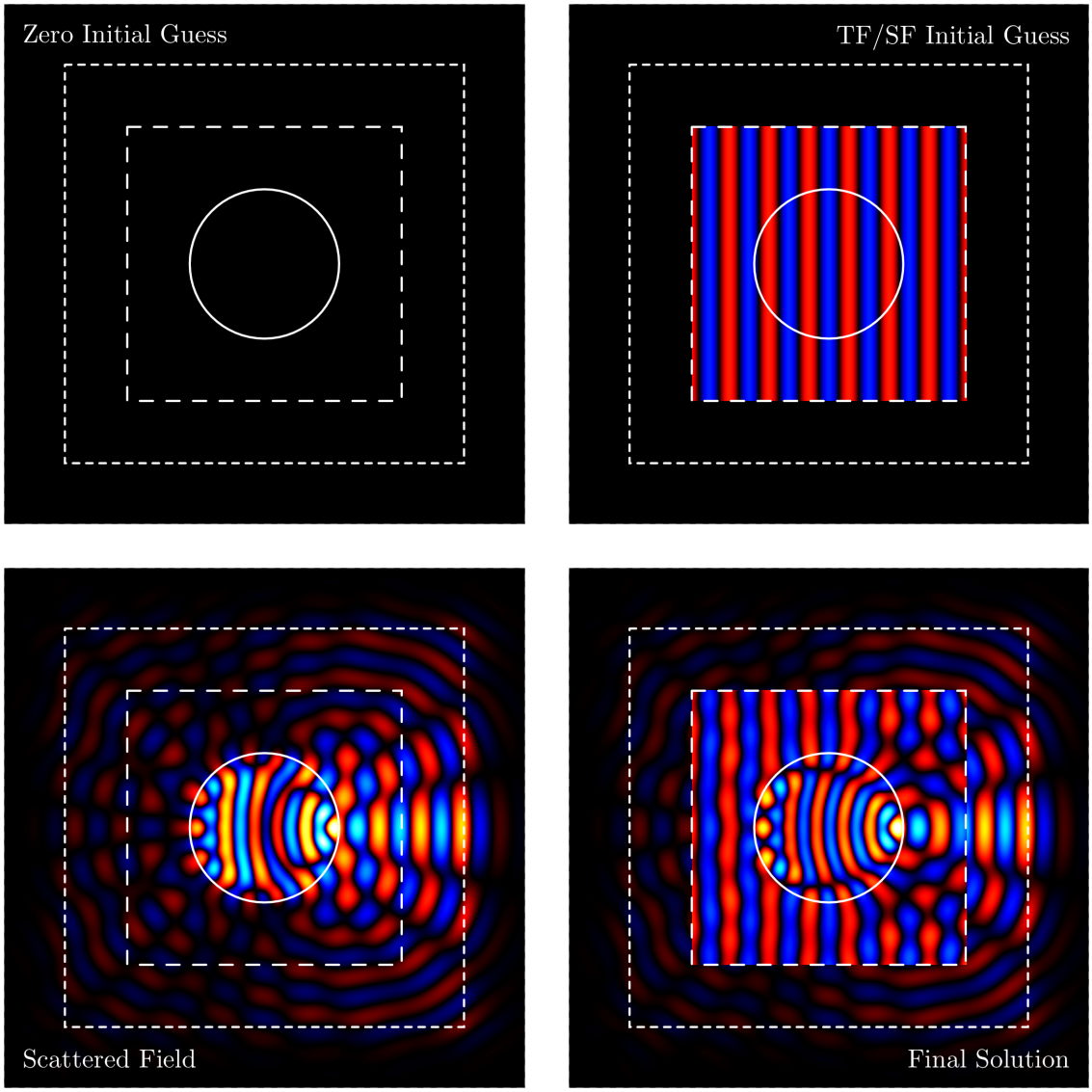


Figure 5.8: *Initial guesses, scattered field, and final solution for a cylinder with* $n = 1.5$ *illuminated by a plane wave with* $\lambda = 80\,\mathrm{nm}$*. All plots show the* z*-component of the magnetic field on the same, but arbitrary colourbar. The top left panel shows the trivial initial guess with all fields being zero. The top right panel shows the TF/SF guess with field values in the TF region determined by the incident wave. The bottom left panel shows the field scattered by the cylinder. This is the difference between the TF/SF guess and the final solution, which is plotted in the bottom right panel. Dashed white lines indicate TF/SF and PML interfaces (see Fig.* 5.5*).*

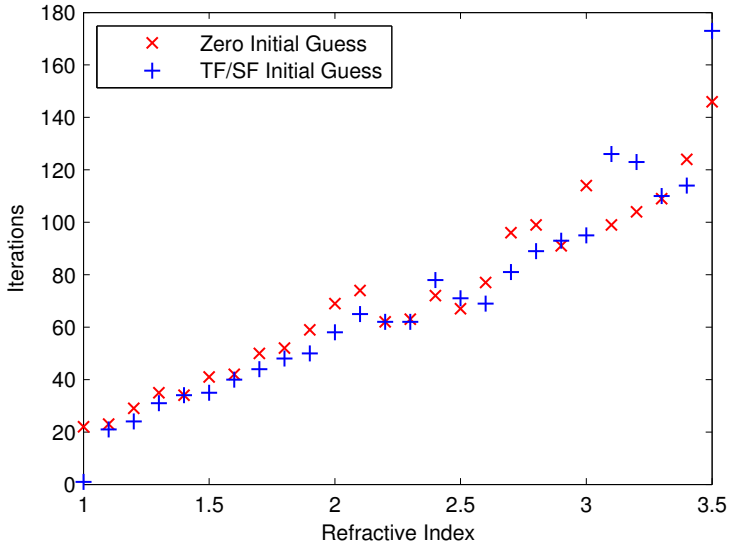


Figure 5.9: *Influence of the initial guess on the performance of iterative solvers. We used the same iterative solver (`BiCGstab(l)` with a Maxwell block Jacobi preconditioner) in combination with two initial guesses. The zero initial guess initialises all fields with zeros. The TF/SF initial guess consists of vanishing fields in the SF region and the incident electromagnetic wave in the TF region. The abscissa shows the refractive index of the cylinder in the setup described in section* 5.4.1.

range, the ratio between both iteration counts differs by up to 20%. This immediately translates to the required CPU time, since the costs associated with an iteration only depend on the solver, not on the initial guess.

We conclude that a good initial guess may save some time. It is, however, difficult to predict which of these generic initial guesses will perform better.

Influence of the Precision Threshold

Iterative solvers do not provide the exact solution of an SLE. Instead, they provide approximations which are accurate up to some predefined precision threshold, often called tolerance. In our case, we refine the numerical solution $\tilde{\mathbf{q}}_i$ until the condition (5.13) is satisfied. Using `BiCGstab(l)` and the Maxwell block Jacobi preconditioner, we perform additional simulations of the cylinder setup with $\epsilon = 2.25$. We investigate the number of iterations required until various levels of precision are achieved. Results for various polynomial orders p are compiled in Fig. 5.10. We notice an *approximately* logarithmic connexion between the number of iterations and the tolerance:

$$\#\text{Iterations} \propto \log(\text{tol}). \tag{5.25}$$

Improving the precision threshold by one order of magnitude roughly leads to an additional 10 to 20 iterations in this example. For other systems and frequencies the behaviour may be different.

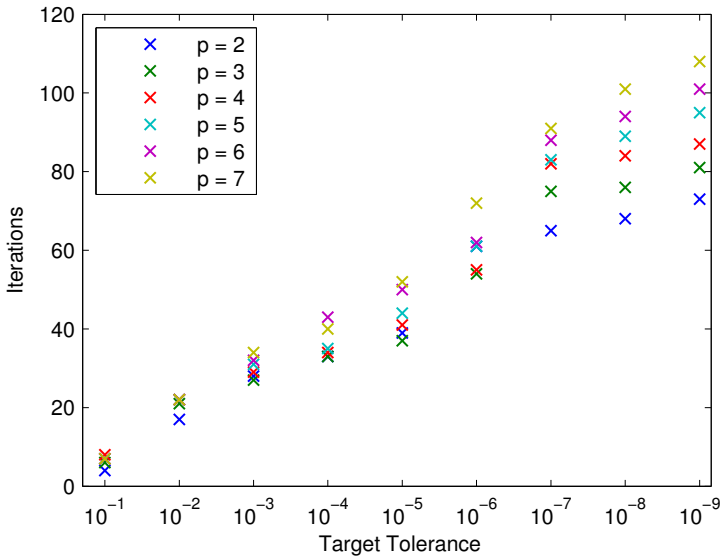


Figure 5.10: *Influence of the precision threshold on the number of iterations required by* `BiCGstab(l)` *combined with a Maxwell block Jacobi preconditioner for various polynomial orders p. Please note the logarithmic scale on the abscissa.*

Obviously, performance comes at the cost of accuracy and vice versa. It is important to know how inaccuracies in the solution of the SLE affect quantities we extract from the simulation. To answer this question we compare the results of iterative solvers with different thresholds against `UMFPACK` calculations of corresponding orders.[12] More precisely, we define the relative error as

$$\text{rel. Error} = \left| \frac{C_{\text{scat}}^{\texttt{BiCGstab(l)},\,p,\,\text{tol}}(\lambda = 80\,\text{nm})}{C_{\text{scat}}^{\texttt{UMFPACK},\,p}(\lambda = 80\,\text{nm})} - 1 \right|. \tag{5.26}$$

In this definition, $C_{\text{scat}}^{\texttt{BiCGstab(l)},\,p,\,\text{tol}}(\lambda = 80\,\text{nm})$ represents the scattering cross section of the cylinder for $\lambda = 80\,\text{nm}$ when calculated using `BiCGstab(l)` with a precision threshold of tol and Lagrange polynomials of order p. Similarly, $C_{\text{scat}}^{\texttt{UMFPACK},\,p}(\lambda = 80\,\text{nm})$ describes the scattering cross section obtained with `UMFPACK`.

The dependence of the relative error on p and tol is plotted in Fig. 5.11. These results indicate that the tolerance of iterative solvers roughly translates one-to-one to the relative deviation of physical quantities. It should be noted that "relative" refers to the exact solution of the SLE, which features spatial discretisation errors as well. This is an important point: Errors due to inaccurately solved SLEs are often shadowed by discretisation errors, in particular for round objects and small wavelengths. Thus, there is often little benefit in extreme tolerances for iterative solvers. In practice, a tolerance of 10^{-4} is sufficient in many cases.

[12] If we compared results obtained with `BiCGstab(l)` against the analytical solution, artifacts due to the inaccurate solution of the SLE would mix with errors due to the spatial discretisation. Since `UMFPACK` solves the very same SLE as `BiCGstab(l)` does, it provides a better reference for our purpose.

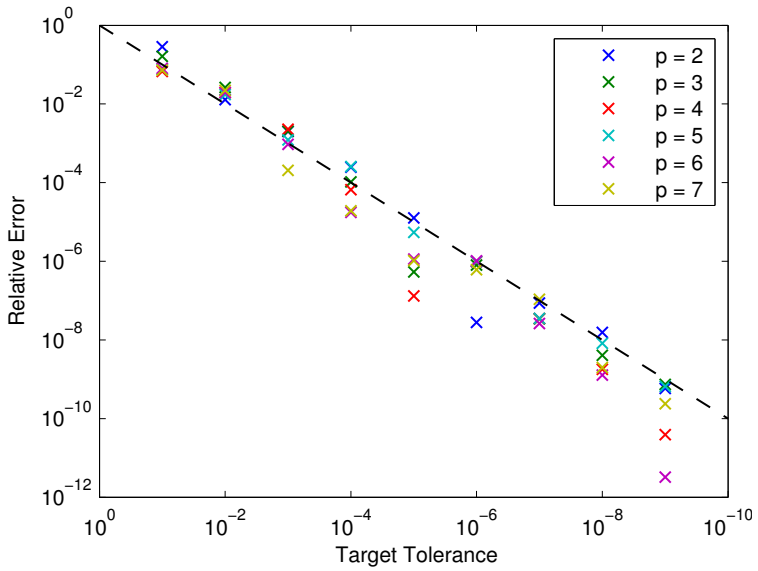


Figure 5.11: *Relative error of the scattering cross section as a function of the target tolerance used when solving an SLE with the iterative solver* `BiCGstab(l)` *(combined with a Maxwell block Jacobi preconditioner). For a definition of the relative error please refer to Eq.* (5.26).

Dispersion

In a final series of simulations, we investigate the influence of dispersion on the performance of numerical solvers. In our code, two different techniques are used to implement dispersive materials in the frequency domain:

- One may keep real-valued material parameters ϵ (and μ), but introduce auxiliary differential equations (ADEs) and associated auxiliary fields. This increases the size of the SLE and the vector of expansion coefficients $\tilde{\mathbf{q}}$.
- Alternatively, one may choose complex values for the material parameters. This does not introduce additional degrees of freedom.

A more detailed explanation of both approaches can be found in section 6.2. It is worthwhile to investigate whether the ADE method has any benefits over the use of complex material parameters.

To this end, we conduct additional simulations for a metallic cylinder illuminated by a plane wave with $\lambda = 169\,\mathrm{nm}$ and monitor the performance of three different solvers (`UMFPACK` and `BiCGstab(l)` combined with element Jacobi and Maxwell block Jacobi preconditioners). The metallic behaviour is modelled by a standard Drude model with a single pole (section 2.3.3). The material parameters are given by Eq. (2.69) and model the dispersion of gold.[13] Both implementations of dispersion were used in combination with each solver for various polynomial orders p. The results are compiled in Table 5.2.

[13] Even though this approximation of gold is everything but accurate in the considered wavelength regime, it is sufficient as a test case. No conclusions concerning the actual physics of gold cylinders are drawn.

		Polynomial order				
		2	3	4	5	
Degrees of freedom		45,024	75,040	112,560	157,584	ADEs
		40,296	67,160	100,740	141,036	$\epsilon \in \mathbb{C}$
UMFPACK	Memory [MB]	349	767	1,294	2,143	ADEs
		341	757	1,417	2,010	$\epsilon \in \mathbb{C}$
	Time [s]	3.7	10.3	20.7	33.9	ADEs
		3.8	10.6	22.7	32.7	$\epsilon \in \mathbb{C}$
BiCGstab(l) +Element Jacobi	Memory [MB]	58	99	170	286	ADEs
		53	87	145	239	$\epsilon \in \mathbb{C}$
	Time [s]	17.8	57.7	140.7	306.0	ADEs
		16.6	45.2	100.0	195.2	$\epsilon \in \mathbb{C}$
	Iterations	77	131	182	243	ADEs
		58	80	98	113	$\epsilon \in \mathbb{C}$
	MV products	1,237	2,103	2,917	3,897	ADEs
		932	1,284	1,573	1,814	$\epsilon \in \mathbb{C}$
BiCGstab(l) +Maxwell block Jacobi	Memory [MB]	51	81	130	206	ADEs
		50	78	126	201	$\epsilon \in \mathbb{C}$
	Time [s]	100+	180+	300+	500+	ADEs
		33.3	93.5	220.5	448.9	$\epsilon \in \mathbb{C}$
	Iterations	500+	500+	500+	500+	ADEs
		119	176	231	281	$\epsilon \in \mathbb{C}$
	MV products	8,000+	8,000+	8,000+	8,000+	ADEs
		1,909	2,822	3,702	4,501	$\epsilon \in \mathbb{C}$

Table 5.2: *Performance of direct and iterative solvers for the scattering of light by an infinite cylinder. In contrast to Table* 5.1*, the values presented here are measured for a cylinder made of a Drude metal. To model this kind of dispersion we may either use auxiliary differential equations (rows with trailing "ADEs", see section* 6.2*) or employ complex values of the permittivity ϵ (rows with trailing "$\epsilon \in \mathbb{C}$"). For each solver/preconditioner combination and each polynomial order p, we list performance data (according to Table* 5.1*) for both implementations. Values with a "+" indicate that convergence has not been achieved.*

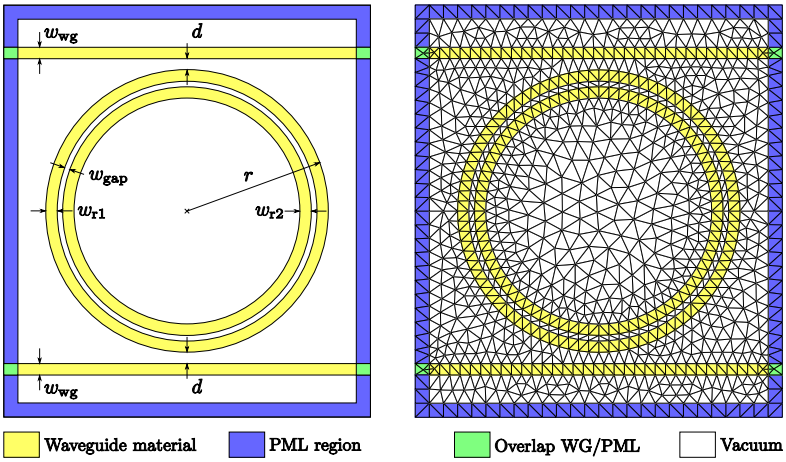


Figure 5.12: *Setup and mesh of a slotted ring resonator coupled to two waveguides. The left panel shows a sketch of the computational domain together with relevant parameters. The waveguides and the resonator are made of the same material with $n = 2.1$ (yellow regions). A uniaxial perfectly matched layer (blue region, see chapter 7) surrounds the physical region and absorbs outgoing radiation. Regions in which the waveguides and the boundary layer overlap are coloured green. The remainder of the computational domain is "filled" with vacuum ($n = 1.0$). The right panel shows the mesh generated by* `NETGEN`*. Curvilinear elements are employed to model the rings.*

It appears that even though the number of degrees of freedom—and thus the size of the SLE—is considerably larger if we use ADEs, the performance of `UMFPACK` is largely unaffected by the choice of how to implement dispersion. Both implementations require roughly the same amount of memory and CPU time, although the performance seems to be slightly worse than in the dielectric case presented in Table 5.1. In contrast, `BiCGstab(l)` benefits from using complex material parameters. The solver requires less memory and is faster as it is when using ADEs. Unfortunately, the combination of `BiCGstab(l)` and the Maxwell block Jacobi preconditioner, which proved excellent in the dielectric case, fails to converge within 500 iterations when using ADEs to model the cylinder's Drude-type dispersion. For complex ϵ, it is significantly slower than `BiCGstab(l)` with the element Jacobi preconditioner. In comparison to the dielectric case, the general performance of `BiCGstab(l)` is worse. This may be attributed to the different frequency (a resonance of the metallic cylinder) and the system itself. We conclude, however, that it is best to avoid ADEs if possible.

5.4.2 Light Propagation in Waveguides Coupled to a Slotted Ring Resonator (2D)

As a second test system we consider a two-dimensional slotted ring resonator which is coupled to two waveguides (Fig. 5.12) in transverse-electric polarisation (section 2.1.5). This system has been investigated recently in Ref. [85], to which the reader is referred to for a more extensive study.[14]

[14] In the referenced paper, the authors use coupled-mode theory and compare it to DGTD results.

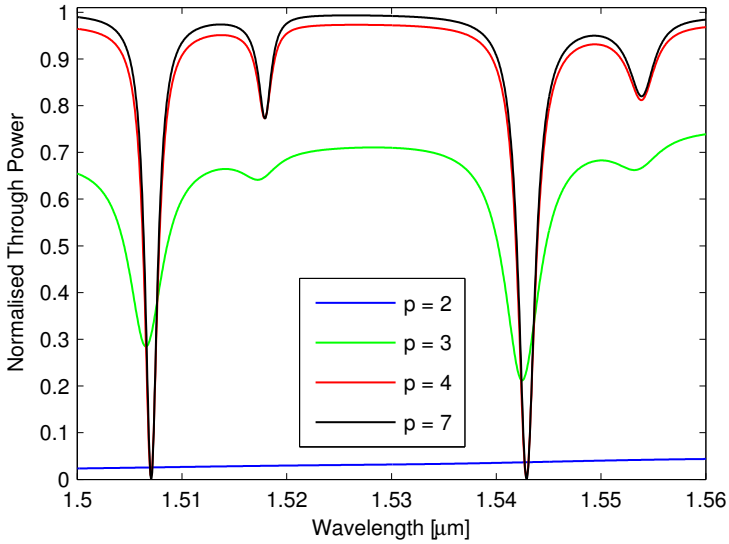


Figure 5.13: *Through power spectrum of a ring resonator coupled to two waveguides. The colours represent calculations performed with different polynomial orders p. For $p = 2$, almost no energy is transmitted to the through port, since insufficiently resolved fields lead to dissipation due to the numerical flux. For $p = 3$ we already observe the ring resonator's resonances. The spectrum for $p = 4$ still shows some dissipation, but otherwise nicely reproduces the converged result for $p = 7$. Calculations for $p = 5$ and $p = 6$ are hardly distinguishable from $p = 7$ and are thus not shown.*

The ring resonator has an outer radius of $r = 5\,\mu\mathrm{m}$ and consists of two concentric rings. Both rings have a width of $w_{\mathrm{r1}} = w_{\mathrm{r2}} = 0.4\,\mu\mathrm{m}$. The gap between both rings has a width of $w_{\mathrm{gap}} = 0.2\,\mu\mathrm{m}$. Both waveguides have a width of $w_{\mathrm{wg}} = 0.4\,\mu\mathrm{m}$. The distance between the waveguides and the resonator is $d = 0.4\,\mu\mathrm{m}$. In the upper left port (Fig. 5.12), the input port, we inject the fundamental waveguide mode. If the ring resonator were absent, this mode would propagate along the waveguide towards the upper right port, the through port. Uniaxial perfectly matched layers (chapter 7) surround the computational domain and absorb energy which passes through the ports. A mesh of this system is generated using `NETGEN` and is shown in Fig. 5.12 as well. Curvilinear elements (section 6.3) are used to model the slotted ring resonator's circular structure as accurately as possible. In total, the mesh consists of 2,220 elements, out of which 632 are curvilinear.

Spectrum

We first investigate the through power spectrum of the resonator-waveguide system. To this end, we record the power transmitted to the through port and normalise it by the power injected in the input port. The quantity thus obtained is called the normalised through power and takes values between 0 and 1. The calculations are repeated for several orders, where we use the direct solver `UMFPACK` to solve the SLE up to machine precision. The resulting spectra are plotted in Fig. 5.13.

It is important to sufficiently resolve the electromagnetic fields, in particular in the waveguides and

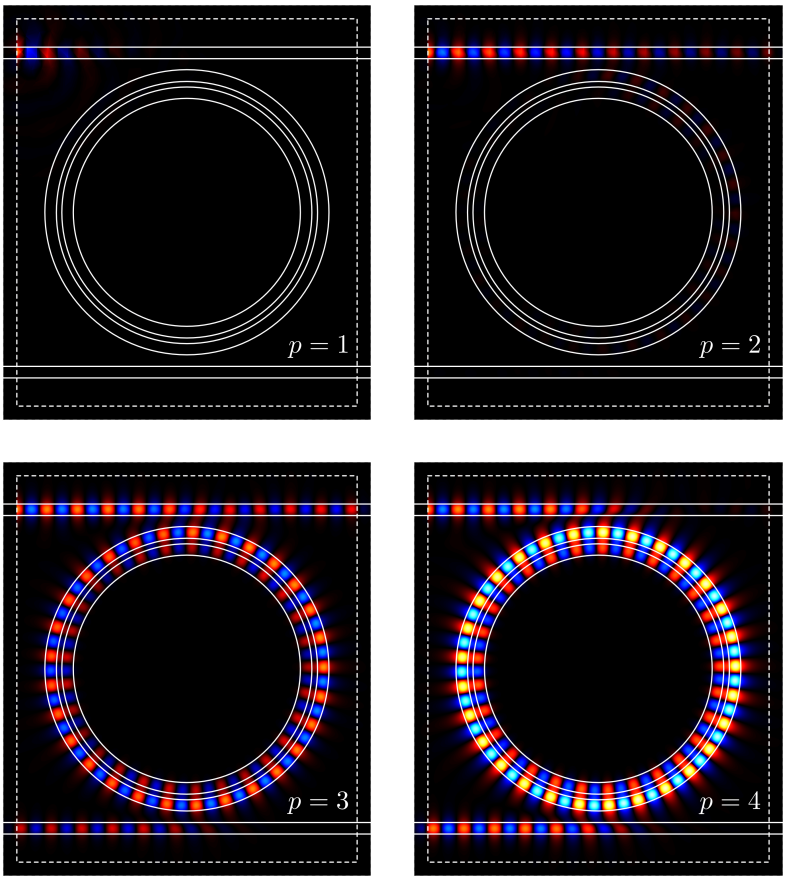


Figure 5.14: *Magnetic field distribution (H_z) for the resonator-waveguide system at $\lambda = 1.54285\,\mu\mathrm{m}$ for various orders p. All plots use the same pseudo-colour scale. Modes for higher orders are very similar to the $p = 4$ mode.*

the resonators. For low orders ($p = 2, 3$), there is strong dissipation and only little energy reaches the through port. This effect is caused by the upwind flux (section 3.1.4), which does not conserve energy and damps modes of high spatial frequency [27]. This effect is still visible in the results for $p = 4$. Starting with $p = 3$, however, the resolution is sufficiently high so that light noticeably couples to the resonator. Consequently, resonances appear in the spectrum. Higher orders ($p = 5$ and above) show nearly full transmittance in between resonances and virtually no energy reaches the through port at the main resonances. The field distributions at resonance calculated for various orders are plotted in Fig. 5.14.

Direct vs. Iterative Solvers

Once the spectrum is available, we focus on the resonance at $\lambda = 1.54285\,\mu\mathrm{m}$ in order to compare the performance of direct and iterative solvers. For this wavelength we simulate the system for various orders p using `UMFPACK` as a direct solver and `BiCGstab(l)` combined with a Maxwell block Jacobi preconditioner as an iterative solver. The precision threshold for `BiCGstab(l)` is set

		Polynomial order					
		2	3	4	5	6	7
Degrees of freedom		41,448	69,080	103,620	145,068	193,424	248,688
UMFPACK	Memory [MB]	294	746	1,275	2,048	2,984	3,496
	Time [s]	3.7	9.9	19.2	31.8	50.7	71.5
BiCGstab(l) +Max. Jacobi	Memory [MB]	56	88	143	230	360	548
	Time [s]	15.1	70.3	145.1	264.5	426.3	601.1
	Iterations	52	150	188	221	234	232
	MV products	835	2,405	3,012	3,540	3,749	3,716

Table 5.3: *Performance of* `UMFPACK` *and* `BiCGstab(l)` *combined with a Maxwell block Jacobi preconditioner at a resonance of the resonator-waveguide system. For an explanation of the rows please refer to Table* 5.1.

to 10^{-4}. Table 5.3 lists the results of our performance tests.

The results for `UMFPACK` are very similar to those of the *dielectric* cylinder system (Table 5.1). On the other hand, `BiCGstab(l)` is not as efficient as expected. The number of iterations is much higher and significantly increases with p, even though this effect seems to saturate for $p \geq 5$. This behaviour is similar to that of the spectrum, for which we obtain accurate results for the same polynomial orders. This shows that the performance of iterative solvers—even if preconditioners are applied—strongly depends on the system.

Multigrid Methods

In an attempt to speed up iterative solvers, we return to the question of a good initial guess. Following the discussion of section 5.3.2, we want to use multigrid methods to obtain a system-dependent initial guess. The general strategy is as following:

1. We choose a start order p_s for the Lagrange polynomials (section 3.1.6) and a target order $p_t > p_s$. For all $p \in [p_s, p_t]$, we create corresponding sets of nodes on all elements of the *same* mesh.
2. We construct an SLE for $p = p_s$. This SLE is either solved directly using `UMFPACK` or using `BiCGstab(l)` in combination with a Maxwell block Jacobi preconditioner. The (intermediate) tolerance of the iterative solver in this step is given by tol_i. We will have to study the influence of this parameter later.
3. The solution of the previous SLE is used as the initial guess for the subsequent order p. Since the SLE for this order has more degrees of freedom than the previous one, it is necessary to interpolate the fields. To this end, we reconstruct the fields obtained for $p - 1$ at the node positions of order p. These field values correspond to the expansion coefficients due to the Lagrange polynomials' defining property (3.15).
4. Starting from this initial guess, we use `BiCGstab(l)` with a Maxwell block Jacobi preconditioner to solve the SLE up to the intermediate tolerance tol_i.

UMFPACK	Time [s] for $\mathrm{tol}_i =$				**BiCGstab(l)**, $\mathrm{tol}_i = 10^{-1}$	
Start order p_s	10^{-1}	10^{-2}	10^{-3}	Memory [MB]	Time [s]	Memory [MB]
1	351.1	454.3	449.7	703	—	—
2	372.8	435.1	471.2	862	366.7	640
3	308.6	393.7	457.1	1,127	362.8	633
4	286.6	292.9	340.5	1,454	388.4	621
5	282.3	289.4	294.7	2,089	425.0	604
6	102.5	102.2	102.4	3,017	418.1	581

Table 5.4: *Performance of multigrid methods for a ring resonator coupled to two waveguides for target order $p_t = 7$ and target tolerance $\mathrm{tol}_t = 10^{-4}$. The left-hand table presents data obtained for multigrid methods which employ `UMFPACK` in the first step. Furthermore, we have used different values for the intermediate tolerance tol_i at step 4. We also list the memory requirements of the calculations, which are independent of the intermediate tolerance. The right-hand table lists results of multigrid strategies which apply `BiCGstab(l)` in step 1. For these results the intermediate tolerance is fixed to $\mathrm{tol}_i = 10^{-1}$. The rows of both tables represent different start orders p_s.*

5. Steps 3 and 4 are repeated until the target order p_t is reached. For the target order, the tolerance of the iterative solver is the target tolerance tol_t.

Please note that one is not restricted to this particular choice of solvers and preconditioners. There is no point, however, in using a direct solver at step 4. We perform a few performance tests to find a good combination of solvers and parameters. For all calculations, we assume $p_t = 7$ and $\mathrm{tol}_t = 10^{-4}$. The start order p_s and the intermediate tolerance tol_i are varied. Corresponding measurements of CPU time and memory consumption are compiled in Table 5.4.

Using `UMFPACK` at step 1 of the multigrid strategy, we observe that the total memory consumption is dominated by the direct solver, especially for higher start orders p_s. Even though higher orders lead to reduced simulation times (the direct solver already does most of the job), the memory overhead is pronounced. The best compromise between CPU time and memory consumption seems to be achieved for $p_s = 3$. This is the first polynomial order for which we observe resonances in the spectrum (Fig. 5.13) and non-negligible fields in the ring resonator (Fig. 5.14). Another important observation is the fact that smaller intermediate tolerances do not speed up the multigrid process at all. The overhead of additional matrix vector products for lower orders outweighs the potential gain of having a slightly better initial guess for the next order.

In the right-hand side of Table 5.4, we show results for multigrid strategies which employ preconditioned `BiCGstab(l)` in step 1. Since we have found that low intermediate tolerances do not lead to performance improvements, we fix $\mathrm{tol}_i = 10^{-1}$. The fastest result is obtained for $p_s = 3$. Please note that for lower start orders we incur a higher memory overhead due to the additional storage of node positions and degrees of freedom for intermediate orders. Comparing the performance of `UMFPACK` and `BiCGstab(l)` used in step 1 of the multigrid algorithm for $p_s = 3$, we find that `UMFPACK` is slightly (15%) faster, but requires almost twice of `BiCGstab(l)`'s memory.

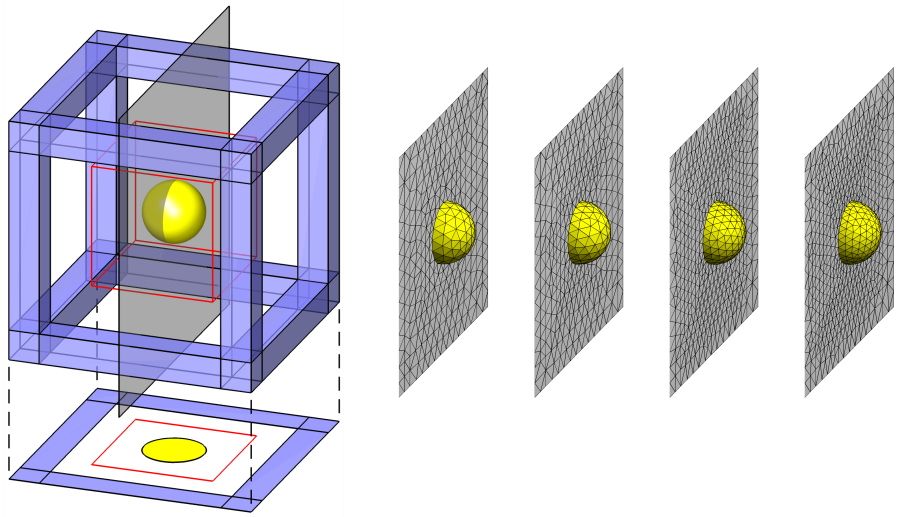


Figure 5.15: *The sphere system used for the calculations in section* 5.4.3*. The left-hand panel shows the computational domain. A sphere of radius* $r = 100\,\mathrm{nm}$ *made of a material with refractive index* $n = 1.5$ *(yellow shade) is embedded in vacuum. A total-field/scattered-field interface (red contour) injects a plane wave into the system (see section* 6.1*). A uniaxial perfectly matched layer (blue shade) surrounds the physical region and absorbs outgoing radiation (see chapter* 7*). The grey plane indicates the plane of symmetry which is exploited to reduce the number of elements by a factor of two. The right-hand panel shows surface meshes of the symmetry plane and the sphere for four different element sizes (cf. Eq.* (5.24)*). From left to right, the three-dimensional meshes consist of* 13,073*,* 16,236*,* 22,515*, and* 26,245 *tetrahedrons.*

5.4.3 Scattering of Light by a Sphere (3D)

As a final example we investigate the scattering of light by a sphere in fully three-dimensional computations. Essentially, this system is a generalisation of the two-dimensional cylinder system presented in section 5.4.1: A sphere of radius $r = 100\,\mathrm{nm}$ and refractive index $n = 1.5$ is surrounded by vacuum (Fig. 5.15). A plane wave is injected into the system via the TF/SF technique (section 6.1). The physical region is surrounded by a layer of UPMLs (chapter 7) to absorb scattered radiation. Except for the higher dimensionality of the sphere system, the only other difference to the cylinder setup is the inclusion of a symmetry plane, on which we apply perfect electric conductor boundary conditions (section 3.1.5). This reduces the size of the computational domain—and consequently the number of degrees of freedom—by 50%. The symmetry plane's normal vector is parallel to the electric field of the incident plane wave.

As before, `NETGEN` is used to generate meshes of the sphere system. Specifically, we create four different meshes which consist of 13,073 to 26,245 straight-sided tetrahedral elements. Surface meshes of the sphere and the symmetry plane are depicted in Fig. 5.15. We perform a series of simulations for Lagrange polynomials of order $p = 2$ using a `BiCGstab(1)`-based multigrid solver. The target tolerance is set to 10^{-4}, the $p = 1$ step employs the same iterative solver and an intermediate tolerance of 10^{-1}. As usual, we also employ a Maxwell block Jacobi preconditioner.

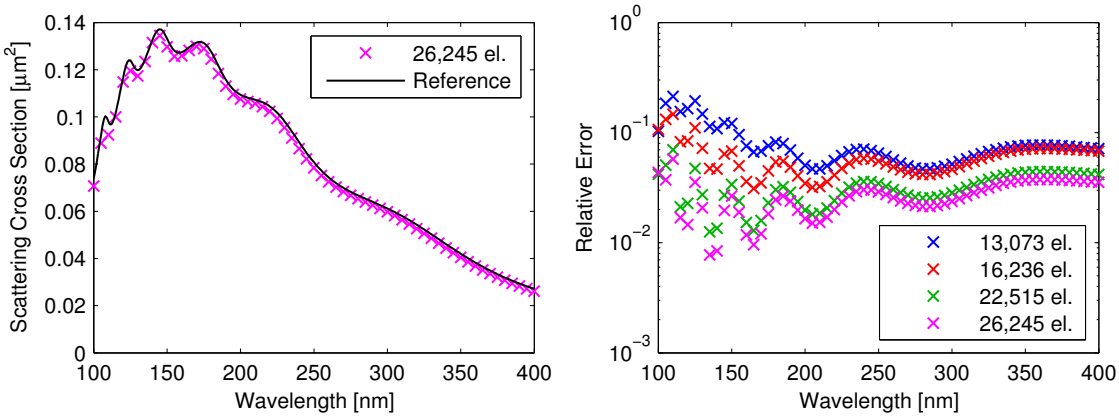


Figure 5.16: *Scattering cross section of a sphere. The left panel shows the cross section as calculated with the finest mesh for $p = 2$ and a reference solution obtained via Mie theory. The right-hand panel shows the absolute value of the relative deviation between the numerically calculated scattering cross section and the reference solution for four different meshes on a semi-logarithmic scale. The finer the mesh, the lower the errors are.*

From these simulations, we extract the scattering cross section by integrating the scattered Poynting vector over the area of the TF/SF interface according to Eq. (2.77). Figure 5.16 compares the resulting spectra against a reference obtained via Mie theory [50, 52]. We observe that the relative error strongly depends on the spatial resolution, where finer meshes reduce artificial damping due to insufficiently resolved modes and enhance the geometrical accuracy of the sphere mesh.

Concerning the computational resources, we record a memory consumption of $2.6\,\mathrm{GB}$ for the finest mesh. The corresponding SLE comprises $2{,}376{,}120$ unknowns. Figure 5.17 presents the time required to solve the SLEs depending on the mesh and the wavelength of interest. Finer meshes obviously require more time to find a solution. In addition, the solution time increases for higher wavelengths. For $\lambda < 300\,\mathrm{nm}$ we obtain a frequency point within a few minutes on a single core of an Intel® XEON® X5670 CPU with a clock speed of $2.93\,\mathrm{GHz}$. .

5.5 Advantages and Drawbacks

Let us conclude this chapter with a short discussion regarding the frequency-domain methods. The approach presented here relies on a semi-discrete version of Maxwell's curl equations, i.e., we solve for both the electric and the magnetic fields. This is in contrast to conventional finite element methods (FEM), which usually solve a discretised version of a wave equation for *either* the electric *or* the magnetic field. This reduces the degrees of freedom by a factor of two as compared to our DG approach.

Furthermore, FEM commonly employs curl-conforming edge elements [25, 26] which explicitly enforce the divergence condition on each element. In contrast, DG's nodal basis (section 3.1.6) does not enforce the divergence conditions. As a consequence, we are burdened with more degrees of freedom than necessary and eigenvalue computations are extremely difficult.

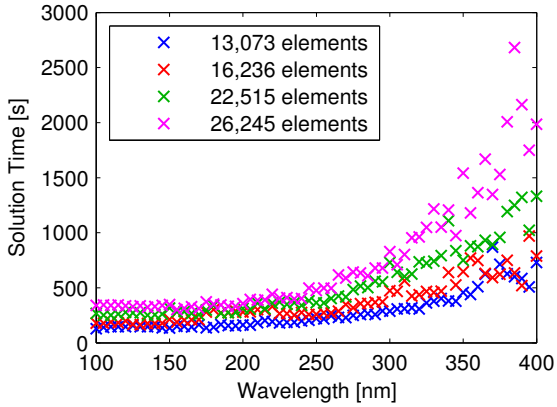


Figure 5.17: *Time required to solve an SLE for various meshes and frequencies.*

Finally, we are challenged by the non-symmetric nature of the system matrix $\mathcal{H}'$, which leads to issues regarding preconditioning and solving the related system of linear equations for driven problems. Symmetric DG discretisations of the second-order wave equation, as discussed in Ref. [76], for example, seem preferable from an efficiency point of view.

The advantage of our approach is its simplicity and that it exploits the key strength of the DG method, the fast matrix-vector product. This is the same property which renders the time-domain algorithm so efficient (chapter 4). We identify two key modifications to an existing code base:

1. The code must support complex field expansion coefficients in addition to real-valued ones. Appropriate routines to multiply real-valued matrices (derivative matrices, etc.) with complex-valued vectors must be provided as well.
2. Solvers for systems of linear equations must be provided.

We have presented an extensive study regarding the second issue. We have found that direct solvers are fast, but very memory-consuming. `UMFPACK` outperforms `PARDISO` and is our direct solver of choice. Iterative solvers are slow in comparison, but require moderate amounts of memory only. Preconditioners applied to iterative solvers are useful techniques to trade speed versus memory. Best results are obtained for `BiCGstab(l)` combined with an element Jacobi or Maxwell block Jacobi preconditioner. The accuracy of iterative solvers has been investigated as well. In most cases moderate tolerances on the deviation to the correct solution are acceptable, since the corresponding errors are shadowed by modelling imperfections. The initial guess heavily influences the performance of iterative solvers. Multigrid methods provide a better initial guess at reasonable costs.

We conclude that—equipped with these techniques—our time-harmonic solver is efficient enough to complement the time-domain algorithm in two and three spatial dimensions. In typical situations, time-domain simulations will be used to obtain finely resolved frequency spectra. A closer investigation of resonances is then facilitated by additional frequency-domain computations. In certain situations, frequencies of interest are known in advance because a laser of certain frequency provides the illumination, for example. In such cases, one may even get along with just the frequency-domain techniques presented in this chapter.

Chapter 6
6 Essential Extensions for Practical Use

The previous chapters have shown us how to discretise Maxwell's equations using the discontinuous Galerkin approach in combination with time stepping schemes or frequency-domain solvers. Along the way, however, we have made a few assumptions which considerably limit the range of applications of these methods. For example, we have eliminated current sources from Maxwell's equations. The material description has been restricted to linear, isotropic, non-dispersive dielectrics. Straight-sided (linear) elements have been assumed to efficiently deduce mass, face mass, and derivatives matrices from matrices calculated on a reference element. The truncation of an infinitely extended system to a finite computational domain has been discussed half-heartedly at best.

The reasons behind these assumptions do not lie in fundamental issues with the DG method. Rather, these simplifications have been introduced for notational convenience, to highlight the key ideas of the method, and to provide as concise a discussion as possible.

In this and the following chapters we present a few extensions to recover a great deal of flexibility. This particular chapter mainly focusses on well-established techniques. First, we describe how to include sources via the total-field/scattered-field technique (section 6.1). Based on the auxiliary differential equation method discussed in chapter 2, we generalise the DG method to support dispersive materials (section 6.2). Afterwards, differences between curved and linear elements are briefly discussed (section 6.3). On a final note, we describe how to obtain frequency-domain quantities in the course of time-domain simulations (section 6.4). Chapters 7 and 8 focus on *novel* techniques to implement absorbing boundaries and anisotropic materials. These developments have been achieved during the research which led to this thesis, and thus have been moved to dedicated chapters.

6.1 Sources

When modelling a physical system, one usually thinks in terms of materials and geometrical structures. Equally important, though, is how the system is illuminated, as the properties of the incident light might decide whether we will observe interesting physics or not. Consequently, control over amplitude, polarisation, propagation direction, and frequency (spectrum) of an incident wave is desired for a number of possible illumination patterns. These patterns include plane waves (section 2.2), waveguide modes [86], focussed laser beams [87], point sources [88], shaped beams [89], and many more.

In principle, almost arbitrary radiation patterns can be included via properly set initial conditions $\vec{\mathrm{E}}(\vec{\mathrm{r}},\, t_0)$ and $\vec{\mathrm{H}}(\vec{\mathrm{r}},\, t_0)$. Unfortunately, even short pulses show significant field amplitudes in rather large volumes compared to the wavelength. Hence, additional computational effort is needed just to model the illumination. Furthermore, some radiation patterns such as a dipole source in close vicinity to a scatterer cannot be easily represented using initial conditions. As soon as the initial fields are non-zero within the scatterer, wrong results will be obtained unless the scatterer is incorporated during the calculation of the initial field pattern. If this were the case, however, there would be no need for a simulation in the first place. Finally, frequency-domain simulations do not support initial conditions. Thus, we need another way to deal with incident fields. The method of choice is the total-field/scattered-field technique, which is efficiently implemented via the numerical flux.

6.1.1 The Total-Field/Scattered-Field Technique

The total-field/scattered-field (TF/SF) technique is very popular for incorporating plane wave sources in FDTD simulations [21, 90]. It relies on the linearity of Maxwell's equations, which allows us to split the electromagnetic fields into two contributions (section 2.4):

$$\begin{aligned} \vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t) &= \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}},\, t) + \vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t), \\ \vec{\mathrm{H}}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t) &= \vec{\mathrm{H}}_{\mathrm{inc}}(\vec{\mathrm{r}},\, t) + \vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t). \end{aligned} \tag{6.1}$$

The subscripts "tot" indicate the *total*, i.e., physical fields, whereas "inc" and "scat" represent *incident* and *scattered* fields, respectively. The incident field is given by a predetermined analytical expression for the desired illumination pattern for arbitrary time and position, e.g., the field distribution of a plane wave (section 2.2.1). With its help, one can calculate scattered fields from total fields and vice versa.

Acknowledging this, we exploit that elements only couple to their immediate neighbours and divide the computational domain—and thus the mesh—into two regions, the total-field (TF) and the scattered-field (SF) region (Fig. 6.1a). Due to linearity, we can solve Maxwell's equations in both regions for either the scattered or the total electromagnetic fields. Hence, we solve the following equations in the respective domains:

$$\begin{aligned} \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_{\mathrm{tot}}) &= 0, \qquad \text{(TF region)} \\ \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_{\mathrm{scat}}) &= 0. \qquad \text{(SF region)} \end{aligned} \tag{6.2}$$

From the steps involved in the DG discretisation (section 3.1), it is clear that within both regions no changes to the standard scheme are necessary. In particular, it is straightforward to evaluate the field differences

$$\begin{aligned} \Delta\vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t) &= \vec{\mathrm{E}}^{+}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{E}}^{-}_{\mathrm{tot}}(\vec{\mathrm{r}},\, t), \\ \Delta\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t) &= \vec{\mathrm{E}}^{+}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{E}}^{-}_{\mathrm{scat}}(\vec{\mathrm{r}},\, t). \end{aligned} \tag{6.3}$$

As previously introduced in section 3.1.4, "$+$" represents the value of the electric field given by the neighbouring element while "$-$" represents the corresponding value from the local element.

Right at the interface between both regions, however, we cannot adhere to the procedure of subtracting field values from both sides of the interface. As illustrated in Fig. 6.1b, the elements involved

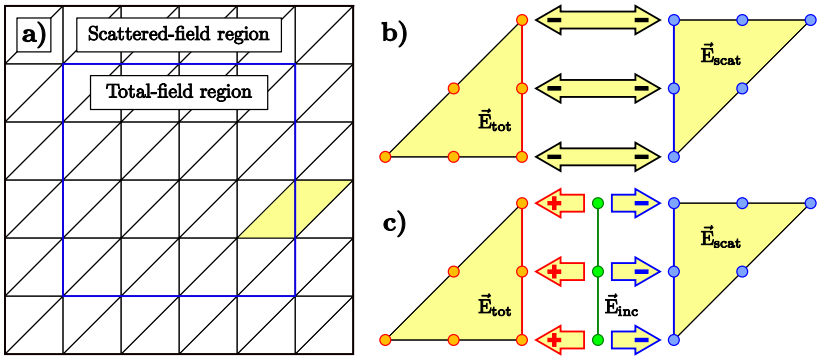


Figure 6.1: *Illustration of the total-field/scattered-field technique as discussed in section* 6.1. ***a)*** *The computational domain is split into a total-field and a scattered-field region. The blue line indicates their mutual interface.* ***b)*** *Zoom on the elements highlighted in a). One element stores total field components* $\vec{\mathrm{E}}_{\mathrm{tot}}$ *while the other stores scattered field components* $\vec{\mathrm{E}}_{\mathrm{scat}}$*. According to section* 3.1.4*, the field differences across element interfaces are given by the fields in the neighbouring element minus the fields in the local one, as indicated by the arrows.* ***c)*** *After the field differences for all elements have been calculated according to b), the field differences on the total-field/scattered-field interface are modified. Using the incident field* $\vec{\mathrm{E}}_{\mathrm{inc}}$*, we can convert* $\vec{\mathrm{E}}_{\mathrm{scat}}$ *to* $\vec{\mathrm{E}}_{\mathrm{tot}}$ *and vice versa.*

store *different* fields: The neighbour of an element in the TF region stores $\vec{\mathrm{E}}_{\mathrm{scat}}$, and not $\vec{\mathrm{E}}_{\mathrm{tot}}$ as would be needed. Obviously, we can use relation (6.1) to obtain

$$\begin{aligned}\Delta\vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}},\,t) &= \vec{\mathrm{E}}^{+}_{\mathrm{scat}}(\vec{\mathrm{r}},\,t) - \vec{\mathrm{E}}^{-}_{\mathrm{tot}}(\vec{\mathrm{r}},\,t) + \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}},\,t),\\ \Delta\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}},\,t) &= \vec{\mathrm{E}}^{+}_{\mathrm{tot}}(\vec{\mathrm{r}},\,t) - \vec{\mathrm{E}}^{-}_{\mathrm{scat}}(\vec{\mathrm{r}},\,t) - \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}},\,t).\end{aligned} \tag{6.4}$$

Analogous statements hold for the differences of the magnetic field. To implement TF/SF sources in an existing framework, there is no need to explicitly distinguish between total fields and scattered fields—or even implement independent discretisations of Maxwell's equations in both regions. Instead, it is sufficient to use the standard scheme on the whole computational domain. The only necessary modification relates to the field differences right at the TF/SF interface, where we either add or subtract the incident electromagnetic fields according to Eq. (6.4). This procedure is illustrated in Fig. 6.1c.

Finally, we have to derive an expression for $\tilde{\mathbf{q}}^{\mathrm{source}}$ in Eqs. (4.3) and (5.1). To this end, we assume that the system operator $\mathcal{H}$ employs the standard calculation of field differences across interfaces between neighbouring elements. Tracking the modifications introduced by the TF/SF method eventually leads to an additional contribution to the numerical flux. Taking the expansion of the fields in terms of Lagrange polynomials into account yields the electric field's source term

$$\tilde{\vec{\mathrm{E}}}^{\mathrm{source},\,\Delta} = \pm(\mathcal{M}^{\Delta})^{-1}\sum_{f\in\mathrm{IF}}\mathcal{F}^{\Delta}_{f}\frac{\alpha\left[\tilde{\vec{\mathrm{E}}}^{\Delta}_{\mathrm{inc},\,f} - \hat{\mathrm{n}}(\hat{\mathrm{n}}\cdot\tilde{\vec{\mathrm{E}}}^{\Delta}_{\mathrm{inc},\,f})\right] + Z^{+}\hat{\mathrm{n}}\times\tilde{\vec{\mathrm{H}}}^{\Delta}_{\mathrm{inc},\,f}}{\bar{Z}}. \tag{6.5}$$

An analogous expression holds for $\tilde{\vec{\mathrm{H}}}^{\mathrm{source},\,\Delta}$. The summation runs over all faces f which lie on

the TF/SF interface IF and vanishes if no such faces are present in element $\triangle$. The plus sign in Eq. (6.5) is valid if $\triangle$ is part of the total-field region. Conversely, the minus sign is valid if $\triangle$ is part of the scattered-field region. Please note that $\tilde{\vec{\mathrm{E}}}_{\mathrm{inc},\,f}$ and $\tilde{\vec{\mathrm{H}}}_{\mathrm{inc},\,f}$ represent expansion coefficients of the known incident fields. They are easily determined by calculating the respective field values at the node positions on the TF/SF interface.

6.1.2 Spatial Profiles

Typically, the mathematical trick of the TF/SF technique presented in the last section is used for various types of plane waves, as for example discussed in section 2.2. As long as the field distribution is known analytically, we have perfect control over polarisation, angle of incidence, intensity, and other parameters of the incident beam.[1]

A less intuitive application of the TF/SF technique is modelling oscillating dipoles. Instead of adding an electric current to one of the expansion coefficients of the electric field, it is favourable to calculate the field distribution using a standard Green's function approach [34, 54]. This altogether avoids the problem of resolving diverging fields near the location of the source. We will employ this implementation of point sources in section 7.3.1 to test the performance of absorbing boundary conditions.

6.1.3 Time Dependencies

In order to employ the TF/SF technique we need to know the incident field distribution at any position $\vec{\mathrm{r}}$ for any time t. In section 2.2 we have learned that the time dependence of a plane wave at any point reduces to the time dependence of the electric field at some reference point in combination with position-dependent delay times, e.g., compare Eqs. (2.32), (2.38), and (2.43). In the case of frequency-domain computations, it is fairly obvious that the time-dependence of this reference point must be harmonic, i.e.,

$$\vec{\mathrm{E}}(t) = \vec{\mathrm{E}}_0 \cdot \exp(-\mathrm{i}\omega t). \tag{6.6}$$

Here, $\vec{\mathrm{E}}_0$ represents a constant polarisation vector. The situation is more involved for time-domain simulations, to which the remainder of this section is dedicated.

Let us first reconsider the harmonic time dependence a plane wave is usually associated with:

$$\vec{\mathrm{E}}(t) = \vec{\mathrm{E}}_0 \cdot \sin(\omega_0 t), \tag{6.7}$$

Even though this equation is very familiar, it is disadvantageous for time-domain simulations for a number of reasons:

- The plane wave exists at all times, it has neither a well-defined beginning nor an end. As the sum of incident and scattered fields must match the total field, the initial condition in the TF region must match the incident wave. If it does not, artificial, non-physical scattering occurs at the TF/SF interface. However, modifying the initial condition is not trivial, especially in the presence of scatterers. If we knew how an initial field in the presence of a scatterer could be calculated, we would not need the simulation anyway.

[1] In the case of a focussed laser beam, for example, we can control the laser mode and the numerical aperture of the focussing objective [87].

- A Fourier transform is often employed for data acquisition and analysis (see section 6.4 for a detailed discussion). Accurate results require the fields of interest to decay until the end of the simulation. The plane wave as given by Eq. (6.7) does not decay at all.
- The plane wave is monochromatic. For such computations frequency domain simulations seem more appropriate.

A more sophisticated time dependence is given by

$$\vec{\mathrm{E}}(t) = \vec{\mathrm{E}}_0 \cdot \sin\Big(\omega_0(t-t_0)\Big) \cdot \exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right). \tag{6.8}$$

It represents a harmonic oscillation with frequency ω_0 with a Gaussian envelope centred around t_0 with width σ. Due to this envelope, all problems which we have discussed earlier are accounted for:

- For $t = 0$, the electromagnetic fields are sufficiently close to zero for all positions $\vec{\mathrm{r}}$ within the TF region, provided that t_0 is sufficiently large. Hence, artificial scattering introduced by the mismatch of the trivial initial condition (vanishing fields) and the incident field is minimised.
- Similarly, for large values of t the Gaussian envelope will exponentially suppress the amplitude of the incident field. Only a finite amount of energy is injected into the system. Loss channels in the system, for example dispersion (sections 2.3 and 6.2) or absorbing boundary conditions (section 3.1.5 and chapter 7), eventually lead to a decay of the electromagnetic fields. Thus, errors due to residual fields in subsequent Fourier transformats are minimised.
- Last but not least at all, the Fourier transform of the time-dependence (6.8) is given by

$$\begin{aligned}\vec{\mathrm{E}}(\omega) = \vec{\mathrm{E}}_0 \cdot \mathrm{i}\sigma\sqrt{\frac{\pi}{2}} \\ \times \left[\exp\left(-\frac{\sigma^2(\omega-\omega_0)^2}{2}\right) - \exp\left(-\frac{\sigma^2(\omega+\omega_0)^2}{2}\right)\right] \cdot \exp(\mathrm{i}\omega t_0).\end{aligned} \tag{6.9}$$

  Please note that the minus sign between the exponential functions in parentheses is a consequence of the sine in Eq. (6.8). Evidently, the energy of the incident pulse is distributed across a frequency *band* instead of just a single frequency. There is no contribution for $\omega = 0$, thus we avoid the creation of static fields. The larger σ, i.e., the longer the duration of the pulse in the time-domain, the narrower the band of significantly contributing frequencies will be. Conversely, low values of σ, which correspond to ultra-fast pulses, lead to very broad frequency spectra.

The last point is particularly important for time-domain methods. Using appropriate signal processing techniques (section 6.4), a single time-domain simulation yields spectral information equivalent to the outcome of many frequency-domain simulations. Depending on the desired spectral resolution and the system, either calculation can be faster.

For spatial profiles other than a plane wave, dispersion is often a problem. For example, the shape of a focussed laser beam heavily depends on its frequency. Similar statements hold for waveguide modes and other profiles. In this case, one may calculate the spatial field pattern for a central frequency ω_0 and use a slowly ramped (large σ) Gaussian pulse (6.8) for the time-dependence. This leads to a very sharp, well-defined peak in the frequency spectrum while at the same time avoiding the problems otherwise introduced by a standard harmonic time dependence. Alternatively, one

can construct the pulse correctly by superimposing spatial profiles of individual frequencies with time-dependent phase shifts.

Other time dependencies might be interesting to match experiments involving ultra-fast phenomena or to achieve coherent control [67, 89, 91]. In particular, the application of time-reserved pulses to achieve spatio-temporal localisation of radiation is most appealing [92]. The higher-order spatio-temporal discretisation of the DGTD method appears to be ideally suited for such problems.

6.2 Dispersive Media

Dispersive materials, in particular metals such as silver and gold, play a prominent role in nanophotonics. Consequently, there is significant demand for accurately modelling these materials within the DG framework. As we have already seen in section 2.1.1, dispersion enters Maxwell's equations via frequency-dependent material parameters.

Obviously, frequency-domain methods as presented in chapter 5 should have little problems handling such parameters. Indeed, if a system is excited by a time-harmonic source (section 5.3), it is sufficient to initialise both permittivity and permeability with the corresponding values at the respective frequency. Thus, each frequency is associated with a specific set of (complex) material parameters $\epsilon(\omega)$ and $\mu(\omega)$.

On the other hand, eigenvalue problems (section 5.2) yield eigenfrequencies as a solution; they are no *input* parameters. Thus, it is difficult to determine the frequency for which material parameters should be taken. Reliable results are only obtained in a narrow band around this frequency. Modelling the actual dispersion, i.e., the *change* of material parameters with frequency, seems preferable.

A suitable option to model dispersion for both frequency- and time-domain simulations is given by the auxiliary differential equation (ADE) method presented in section 2.3. In particular, the Drude-Lorentz model, Eq. (2.68), leads to a good description of plasmonic systems.

Implementation and Efficiency Considerations

For convenience, we collect the ADEs derived in section 2.3 for the Drude model:

$$\partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) = \epsilon_\infty^{-1}(\vec{\mathrm{r}}) \cdot \Big(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}},\, t) - \vec{\mathrm{j}}_D(\vec{\mathrm{r}},\, t) \Big), \tag{2.52'}$$

$$\partial_t \vec{\mathrm{j}}_D(\vec{\mathrm{r}},\, t) = \omega_D^2(\vec{\mathrm{r}}) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}},\, t) - \gamma_D(\vec{\mathrm{r}}) \cdot \vec{\mathrm{j}}_D(\vec{\mathrm{r}},\, t). \tag{2.60'}$$

Repeating the steps undertaken in section 3.1, we first append the polarisation current $\vec{\mathrm{j}}_D(t)$ to the state vector $\mathbf{q}$. Since the ADE for $\vec{\mathrm{j}}_D(t)$ does not involve spatial derivatives, the components of the flux vector are extended by zeros to match the size of the state vector. In complete analogy to the electromagnetic fields, we also expand the auxiliary current in terms of Lagrange polynomials. Due to the convenient absence of spatial derivatives, the only ADE-related matrices which appear in the discretisation are mass matrices $\mathcal{M}^\Delta$ (section 3.1.7):

$$\begin{aligned}
\mathcal{M}^\Delta \partial_t \tilde{\vec{\mathrm{E}}}^\Delta(t) &= \epsilon_\infty^{-1}(\vec{\mathrm{r}}) \cdot \Big(\vec{\mathcal{S}}^\Delta \times \tilde{\vec{\mathrm{H}}}^\Delta(\vec{\mathrm{r}},\, t) - \mathcal{M}^\Delta \tilde{\vec{\mathrm{j}}}_D^\Delta(\vec{\mathrm{r}},\, t) + \mathcal{F}_f^\Delta \cdot \Big[\hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}} - \vec{\mathrm{F}}^* \Big) \Big]_{\vec{\mathrm{E}},\, f} \Big), \\
\mathcal{M}^\Delta \partial_t \tilde{\vec{\mathrm{j}}}_D^\Delta(t) &= (\omega_D^\Delta)^2 \cdot \mathcal{M}^\Delta \tilde{\vec{\mathrm{E}}}^\Delta(t) - \gamma_D^\Delta \cdot \mathcal{M}^\Delta \tilde{\vec{\mathrm{j}}}_D^\Delta(t).
\end{aligned} \tag{6.10}$$

Please note that we have assumed constant material parameters $\omega_D^{\triangle}$ and $\gamma_D^{\triangle}$ on the element $\triangle$. The numerical flux components which correspond to the electric field are not modified with respect to Eq. (3.8). Multiplying Eqs. (6.10) with the inverse mass matrix yields

$$\begin{aligned}\partial_t \tilde{\vec{\mathrm{E}}}^{\triangle}(t) &= \epsilon_\infty^{-1}(\vec{\mathrm{r}}) \cdot \left(\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle}(\vec{\mathrm{r}},\, t) - \tilde{\vec{\mathrm{j}}}_D^{\triangle}(\vec{\mathrm{r}},\, t) + \left(\mathcal{M}^{\triangle}\right)^{-1} \mathcal{F}_f^{\triangle} \cdot \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^{*} \right) \right]_{\vec{\mathrm{E}},\, f} \right), \\ \partial_t \tilde{\vec{\mathrm{j}}}_D^{\triangle}(t) &= (\omega_D^{\triangle})^2 \cdot \tilde{\vec{\mathrm{E}}}^{\triangle}(t) - \gamma_D^{\triangle} \cdot \tilde{\vec{\mathrm{j}}}_D^{\triangle}(t).\end{aligned} \tag{6.11}$$

These equations do not feature any additional matrices[2] as compared to the semi-discrete curl equations (3.30). Thus, each expansion coefficient of the auxiliary current couples to exactly one expansion coefficient of the electric field. As a result, this coupling spawns diagonal blocks in the system operator $\mathcal{H}$ (cf. section 5.3.1 and Fig. 5.2).

The discussion of the Drude model is also valid in the case of the Lorentz model. Again, the absence of spatial derivatives leads to node-local coupling between auxiliary and electromagnetic fields, this time, however, with two auxiliary fields instead of one.

At first glance it seems quite expensive to store additional field components. A single Lorentz pole, for example, doubles the number of unknowns in three-dimensional simulations.[3] It is, however, important to note that—in most simulations—only a few elements (typically below 10% of the mesh) are filled with metals. A little book keeping allows us to restrict the necessary ADEs to these elements only. As a consequence, the storage of auxiliary fields can be avoided on all non-metallic elements. Furthermore, memory is usually not an issue for time-domain simulations, where the CPU time consumption generally is the limiting factor. However, evaluating the Drude-Lorentz ADEs is very fast because of the absence of matrix operations. Consequently, the inclusion of dispersion in a DGTD framework merely leads to slight performance impairments.

In conclusion, adding these ADEs to an existing DG framework requires two simple steps:

1. The expansion coefficients for the auxiliary fields must be integrated into the vector of expansion coefficients $\tilde{\mathbf{q}}$. As discussed previously in section 5.3.1, we suggest to store all degrees of freedom of a single element in a contiguous block of memory.
2. For most elements, the system operator $\mathcal{H}$ without ADEs can be immediately applied. For all other elements, one first applies the existing operator. Then, one applies corrections due to the ADEs in a "post-processing" step.

With the vector $\tilde{\mathbf{q}}$ and the system operator $\mathcal{H}$ thus modified, the existing time stepping scheme or SLE solver can be used to solve both physical and auxiliary fields at the same time.

6.3 Curvilinear Elements

One of the key strengths of the DG discretisation is that it does not rely on structured grids, but employs geometry-adapted meshes instead. This allows to accurately represent thin films, small features, tilted planes and more. However, curved objects remain challenging to model, particularly since we strive for meshes with as few elements as possible. Specifically, we wish to avoid very

[2] Besides "invisible" unit matrices, of course.

[3] We have to store $\tilde{\vec{\mathrm{j}}}_L$ and $\tilde{\mathrm{q}}_L$ in addition to $\tilde{\vec{\mathrm{E}}}$ and $\tilde{\vec{\mathrm{H}}}$.

small elements which drastically limit the maximum stable time step. For this reason, it makes sense to introduce curvilinear elements [35, 93] as a means to reduce the number of elements while at the same time increasing the accuracy of the geometrical representation.

When revisiting the derivation of the semi-discrete system (3.30), we find that no assumptions were made on the specific shape of the elements. The only place where the shape of an element actually enters is in the construction of the matrices, i.e., in Eqs. (3.23), (3.25), and (3.29). As detailed in section 3.2.3, for straight-sided elements we can generate the matrices of each element from only five template matrices evaluated on a straight-sided reference element $\triangle_{\mathrm{ref}}$.

If we wish to represent a curved element, this necessarily leads to a non-affine mapping between the coordinates $\vec{\mathrm{r}}$ of the curved element and the coordinates $\vec{\mathrm{s}}$ of the straight-sided reference element. This mapping can be described by a *spatially dependent*[4] Jacobian matrix $\mathcal{J}^{\triangle}(\vec{\mathrm{s}})$. Details on how to obtain this mapping and related topics can be found in Refs. [25, 27, 35, 94].

For a given non-affine mapping, we then have to evaluate the matrices individually for each curved element. In practice, one typically calculates the volume and surface integrals numerically by employing suitable cubature rules [95]. Then, the mass, stiffness and face-mass matrices need to be stored individually for every curved element. This leads to a dramatically increased memory consumption of the calculation, but also significantly improves the accuracy of the geometrical description. Fortunately, in most systems few elements actually need to be curvilinear. Therefore, even rather large three-dimensional systems can be treated without running into memory constraints. An elegant formulation which altogether avoids this memory overhead was recently proposed by Tim Warburton [96]. However, it requires some post-processing if we are interested in the electromagnetic fields.

6.4 Data Recording and Analysis

The natural output of DGTD simulations are time-dependent expansion coefficients. Most experiments, however, cannot resolve the rapid oscillations of the electromagnetic fields in the optical regime, i.e., frequencies around $100\,\mathrm{THz}$. Instead, time-averaged frequency spectra of integrated quantities like the transmittance of a material slab are often desired. Evidently, such quantities are naturally obtained from frequency-domain computations. For time-domain computations it is obvious to extract spectral information using some discretised analogon of the Fourier transform (2.4).

More precisely, from a series of τ time steps we infer the discrete Fourier transform

$$F(\omega) \approx \sum_{i=1}^{\tau} F(t_i) \cdot \exp(\mathrm{i}\omega t_i) \cdot \Delta t_i \tag{6.11}$$

of an arbitrary time-domain quantity $F(t)$. Here, t_i represents the simulated time after the i-th time step, t_0 the time of initialisation, and $\Delta t_i = t_i - t_{i-1}$ is the size of the i-th time step. The popular fast Fourier transform (FFT) [97, 98] is not the ideal choice to evaluate Eq. (6.11). Though it is arguably the fastest method to extract the complete information from a given time series, it is subject to a few limitations:

[4] This is the crucial difference to the discussion in section 3.2.3.

- All time steps must be equally large. Time stepping schemes with adaptive time steps cannot be implemented if we want to use conventional FFT. In principle, however, generalisations to non-equidistant time steps are available, e.g., see Ref. [99].
- Due to the divide and conquer strategy used by FFT, it is essential that the *complete* time series is available at the time we want to calculate the Fourier transform. For a single data point this is hardly a problem in most cases. If one is, however, interested in recording fields on a plane[5] or in a volume, one often ends up with tens of thousands of data points, each accompanied by a distinct series of time steps. As it turns out, storing these series can be a severe issue even if the simulation itself easily fits into main memory.

Furthermore, obtaining the full spectral information is usually not desired. Rather, one is often interested in a certain frequency range with a certain resolution. For example, a metamaterial might show an interesting behaviour for wavelengths between 800 and $1800\,\mathrm{nm}$. A resolution of $5\,\mathrm{nm}$ in this range leaves us with a total of 201 wavelengths and, thus, frequencies of interest. As the number of frequencies is orders of magnitudes smaller than the number of time steps in typical simulations[6], it is convenient to perform an on-the-fly Fourier transform.

To this end, we specify a set of frequencies ω_j for which the field F should be recorded. After each time step, one updates the Fourier coefficients according to

$$F(\omega_j) := F(\omega_j) + F(t_c) \cdot \exp(\mathrm{i}\omega_j t_c) \cdot \Delta t_c. \tag{6.12}$$

Here, the subscript "c" denotes the current time step. Using this iterative update procedure, it is not necessary to store the complete time series, as we only need one value a time. In particular, the memory consumption of the on-the-fly Fourier transform does not depend on the number of time steps, but rather on the number of frequencies.

However, its accuracy does. The transition from continuous time, i.e., Eq. (2.4), to discrete time, i.e., Eq. (6.11), truncates the domain of integration. If the conditions

$$\begin{aligned} F(t) &\equiv 0 \quad \text{for} \quad t \in (-\infty,\, 0\,], \\ F(t) &\equiv 0 \quad \text{for} \quad t \in [\, t_f,\, \infty) \end{aligned} \tag{6.13}$$

are satisfied, the error of the discrete Fourier transform only depends on Δt, which is usually sufficiently small due to the explicit time stepping scheme. The first condition is easily fulfilled by setting all initial expansion coefficients to zero. The second condition states that the fields must decay during the simulation, where t_f is the final simulation time. Even though this can never be achieved perfectly, the system needs to be simulated long enough to allow for a sufficient decay. Problems arise especially for systems with high quality factors because they require extremely long simulation times for he fields to decay. If it is too short, oscillations appear in the spectrum.

In essence, the Fourier transform—whichever way implemented—allows us to obtain a spectrum from just a single time-domain simulation. This is an important point, as we do not have to launch multiple simulations with incident waves of slightly different frequencies. It is often cited as a key benefit of time-domain methods.

[5] For example, this is necessary to obtain transmittance and reflectance spectra (see section 9.3.2).

[6] Several ten thousand time steps are quite common. High quality resonators may even require hundreds of thousands of time steps in order to obtain reliable results.

Chapter 7

7 Absorbing Boundaries: Perfectly Matched Layers

Being a volume method, the discontinuous Galerkin technique cannot inherently solve infinitely extended systems. In contrast to Green's functions techniques, for example, the basis functions are spatially localised. In particular, there are no dedicated basis functions for radiation into the infinity of space. Though this property eventually leads to the explicit time-stepping scheme and the sparsity of the system matrix, it complicates the simulation for many physical situations.

In nanophotonics, one is often interested in studying isolated particles, i.e., particles which are surrounded by free space and often lie on a substrate. One might also be interested in the transmittance spectrum of a periodic array of building blocks. There, incident illumination is both transmitted and reflected into free space. Obviously, the boundary of the computational domain requires special treatment which goes beyond reflecting boundary conditions such as perfect electric or magnetic conductor boundary conditions (see section 3.1.5). The issues arising with these reflecting boundary conditions are illustrated in Fig. 7.1.

All energy which is radiated into infinity should be lost for the actual system of interest because infinity does not contain any scatterers. Hence, if we have to terminate our computational domain, its boundary should perfectly absorb incident light irrespective of its angle of incidence, wavelength, and polarisation. Boundary conditions which approximate this behaviour are called absorbing boundary conditions (ABCs, see Fig. 7.1). Two complementary approaches exist.

First, one can directly enforce physical boundary conditions which support outgoing radiation modes but suppress reflected ones. Strategies which follow this route are called analytical ABCs (AABCs). Even though higher-order schemes are available [100], in practice one usually employs lower order approximations of AABCs. While relatively straightforward to implement, they suffer from mediocre performance for oblique incidence and short distances between the radiation source and the boundary [21]. With larger distances, however, the number of unknowns significantly increases while most of them essentially simulate free space. For this reason, AABCs often do not suffice in practice.

As an alternative, Bérenger developed a novel technique in the mid 90s [101]. His idea was to divide the computational domain into a physical region and a surrounding, specially tailored, absorbing boundary layer. To this end, he introduced an unphysical splitting of the fields in Maxwell's

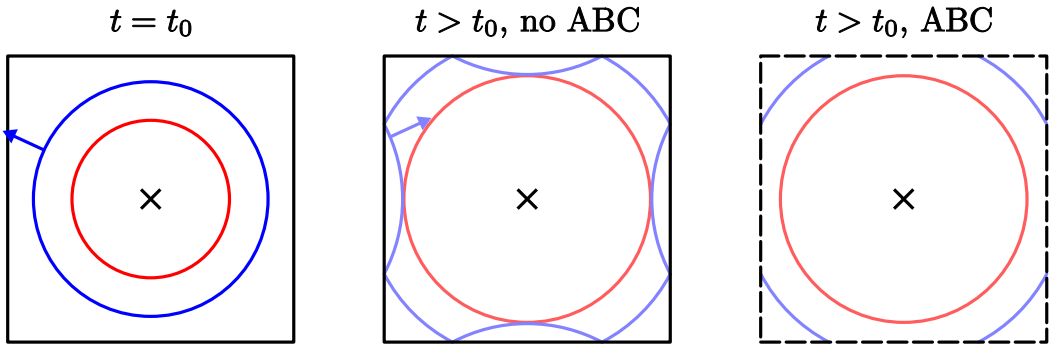


Figure 7.1: *Effect of absorbing boundaries. Left panel: A dipole source emits radiation (red and blue circles) at the centre of the computational domain (black box). The cylindrical waves propagate towards the domain's boundary (blue arrow). Central panel: Once the wave impinges on a non-absorbing boundary it is reflected. Right panel: An absorbing boundary, however, does not reflect incident waves. Thus, it appears like a transparent boundary to an infinitely extended, but not simulated domain.*

equations. By cleverly defining additional material parameters in the boundary layer, he was able to match its impedance with that of the material in the physical region. Thus, the interface between both domains is perfectly transparent for electromagnetic waves. Hence, he called the boundary layer a perfectly matched layer (PML). Within the PML, propagating waves are attenuated. Even though they are eventually reflected at the boundary of the computational domain, by then they have been sufficiently attenuated to provide a merely insignificant contribution to the electromagnetic fields in the physical region (see Fig. 7.2).

Over the years, Bérenger's original formulation was refined and generalised [21, 102–104]. Today, one usually implements PMLs either as an uniaxial anisotropic absorber (UPML) or interprets them in terms of a complex stretching of the coordinate axes. Instructions how to include UPMLs in a DGTD framework have been known for some time [34, 105] and will be reviewed in section 7.1. Section 7.2 describes a novel stretched-coordinate based implementation of PMLs in the discontinuous Galerkin framework, which has been published recently in Ref. [39]. In particular, we present a numerical flux which yields numerically stable and accurate results. Finally, a study of numerical and computational performance concludes the discussion of absorbing boundaries in section 7.3.

7.1 Uniaxial Perfectly Matched Layers

The uniaxial perfectly matched layer (UPML) approach is very convenient for the DG method [34, 105], since it requires only minor modifications to the basic algorithm. The idea is to replace scalar, non-dispersive material properties ϵ and μ by material *tensors* [21, 102]

$$\underline{\epsilon}' \equiv \underline{\Lambda}\,\epsilon \quad \text{and} \quad \underline{\mu}' \equiv \underline{\Lambda}\,\mu, \quad \text{where} \quad \underline{\Lambda} \equiv \begin{pmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{pmatrix}. \tag{7.1}$$

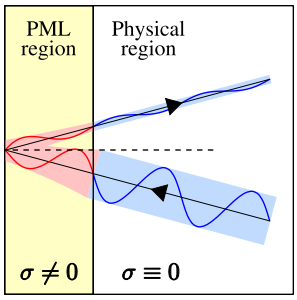


Figure 7.2: *Working principle of perfectly matched layers. Incident radiation is attenuated in the PML region. At the boundary of the computational domain the light wave is reflected and undergoes continued attenuation. Once it reenters the physical region, its amplitude (typically suppressed by multiple orders of magnitude) no longer represents a significant perturbation to the physical fields. Please note the absence of reflections at the interface between the physical and the PML region.*

The PML parameters $s_i, i = x,\ y,\ z$ are chosen as

$$s_i(\omega) \equiv 1 - \frac{\sigma_i}{\mathrm{i}\omega}, \tag{7.2}$$

where the frequency-independent parameter σ_i controls the damping of a wave propagating along the i-direction. We observe that s_i is both complex-valued and dispersive. Please note that Eq. (7.2) also obeys the Fourier transform's sign convention (2.4).

From the discussion of section 2.3.1, we are already familiar with how to handle dispersive material parameters in the time-domain.[1] In the spirit of the auxiliary differential equation (ADE) technique presented there, we aim at an easy time-domain transformation of

$$\begin{aligned} -\mathrm{i}\omega\vec{\mathrm{D}}(\omega) &= -\mathrm{i}\omega\underline{\epsilon}'(\omega)\vec{\mathrm{E}}(\omega), \\ -\mathrm{i}\omega\vec{\mathrm{B}}(\omega) &= -\mathrm{i}\omega\underline{\mu}'(\omega)\vec{\mathrm{H}}(\omega). \end{aligned} \tag{7.3}$$

In the following, we will only consider the x-component of the electric field to simplify the notation. Analogous procedures lead to expressions for the other two components and the magnetic field. Inserting (7.1) yields

$$-\mathrm{i}\omega D_x(\omega) = -\mathrm{i}\omega\epsilon E_x + j_x(\omega), \tag{7.4a}$$

$$j_x(\omega) = -\mathrm{i}\omega\epsilon \cdot \left(\frac{s_y s_z}{s_x} - 1\right) \cdot E_x(\omega). \tag{7.4b}$$

Here, we have introduced an auxiliary field $j_x(\omega)$, which should not be confused with the electric source current in Maxwell's equations (2.18). Using definition (7.2), we can rewrite (7.4b) to obtain

$$-\sigma_x j_x(\omega) = -\mathrm{i}\omega\Big[j_x(\omega) + \epsilon \cdot (\sigma_x - \sigma_y - \sigma_z) \cdot E_z(\omega)\Big] - \epsilon\sigma_y\sigma_z E_x(\omega). \tag{7.5}$$

[1] The following discussion is also valid and useful for the frequency-domain, since it essentially eliminates the necessity to modify the numerical flux due to tensorial material parameters.

Motivated by the Fourier rules (2.48), we introduce

$$p_x(\omega) = j_x(\omega) + \epsilon \cdot (\sigma_x - \sigma_y - \sigma_z) \cdot E_z(\omega) \tag{7.6}$$

to replace $j_x(\omega)$ as the auxiliary field. Doing so, Eqs. (7.4) read

$$\begin{aligned} -\mathrm{i}\omega D_x(\omega) &= -\mathrm{i}\omega\epsilon E_x + p_x(\omega) - \epsilon \cdot (\sigma_x - \sigma_y - \sigma_z) \cdot E_z(\omega), \\ -\mathrm{i}\omega p_x(\omega) &= \quad (\sigma_x^2 - \sigma_x\sigma_y - \sigma_x\sigma_z + \sigma_y\sigma_z) \cdot \epsilon E_x(\omega) - \sigma_x p_x(\omega). \end{aligned} \tag{7.7}$$

Applying the—then trivial—Fourier transform finally leads to the time-domain formulation

$$\begin{aligned} \partial_t E_x(t) &= \epsilon^{-1}[\vec{\nabla} \times \vec{\mathrm{H}}(t)]_x + (\sigma_x - \sigma_y - \sigma_z) \cdot E_x(t) - \epsilon^{-1} p_x(t), \\ \partial_t p_x(t) &= (\sigma_x^2 - \sigma_x\sigma_y - \sigma_x\sigma_z + \sigma_y\sigma_z) \cdot \epsilon E_x(t) - \sigma_x p_x(t). \end{aligned} \tag{7.8}$$

Similar equations follow for the other electromagnetic field components.

Apparently, UPMLs introduce one additional field component per electromagnetic field component. Please note that in the region of interest, i.e., the domain which is surrounded by the PML, we have $\sigma_x = \sigma_y = \sigma_z = 0$. In this case, we do not need to store the auxiliary fields since both the initial conditions and the time derivatives of the auxiliary fields vanish. Following the DG discretisation procedure (section 3.1), one obtains the matrix form

$$\begin{aligned} \partial_t \tilde{\mathrm{E}}_x^{\triangle}(t) = \; & \frac{1}{\epsilon^{\triangle}} \Big[\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle}(t) \Big]_x + (\sigma_x^{\triangle} - \sigma_y^{\triangle} - \sigma_z^{\triangle}) \cdot \tilde{\mathrm{E}}_x^{\triangle}(t) - \frac{1}{\epsilon^{\triangle}} \tilde{\mathrm{p}}_x^{\triangle}(t) \\ & + (\mathcal{M}^{\triangle})^{-1} \mathcal{F}_f^{\triangle} \cdot \Big[\hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}} - \vec{\mathrm{F}}^* \Big) \Big]_{E_x,\, f}, \\ \partial_t \tilde{\mathrm{p}}_x^{\triangle}(t) = \; & \Big((\sigma_x^{\triangle})^2 - \sigma_x^{\triangle}\sigma_y^{\triangle} - \sigma_x^{\triangle}\sigma_z^{\triangle} + \sigma_y^{\triangle}\sigma_z^{\triangle} \Big) \cdot \epsilon^{\triangle} \tilde{\mathrm{E}}_x^{\triangle}(t) - \sigma_x^{\triangle} \tilde{\mathrm{p}}_x^{\triangle}(t). \end{aligned} \tag{7.9}$$

Here, we have assumed constant PML parameters $\sigma_i^{\triangle}$ within the element $\triangle$. These equations must be solved concurrently in order to implement UPMLs in a DG framework. In particular, the state vector $\tilde{\mathbf{q}}$ must be expanded to contain the auxiliary fields as well.

Please note that the operation $\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)$ returns a vector with the same size as the state vector $\tilde{\mathbf{q}}^{\triangle}$. Hence, the subscript "$E_x$" of this term refers to those components which correspond to $\tilde{\mathrm{E}}_x^{\triangle}$ in $\tilde{\mathbf{q}}^{\triangle}$. The additional subscript "f" again refers to the faces of the element $\triangle$. Also note the convenient absence of additional matrix-vector products in Eq. (7.9). Instead, each auxiliary expansion coefficient solely depends on the corresponding expansion coefficient of an electromagnetic field, which allows for an extremely fast implementation. This is an immediate consequence of the locality of the UPML ADEs which do not feature spatial derivatives.

Modifications for Dispersive Materials

Though UPMLs are fairly easily formulated for simple dielectric materials, the derivation is more complicated for dispersive materials. In this case, it is useful to write

$$\begin{aligned} -\mathrm{i}\omega D_x(\omega) &= -\mathrm{i}\omega \Big[\epsilon_\infty + \chi(\omega) \Big] \cdot \left[1 + \left(\frac{s_y s_z}{s_x} - 1 \right) \right] \cdot E_x(\omega) \\ &= -\mathrm{i}\omega\epsilon_\infty E_x(\omega) - \mathrm{i}\omega\epsilon_\infty \left(\frac{s_y s_z}{s_x} - 1 \right) \cdot E_x(\omega) \\ &\quad - \mathrm{i}\omega\chi(\omega) E_x(\omega) - \mathrm{i}\omega \left(\frac{s_y s_z}{s_x} - 1 \right) \cdot \chi(\omega) E_x(\omega). \end{aligned} \tag{7.10}$$

The first three terms represent dielectric material properties, UPMLs for dielectric materials, and the dispersion as already known from previous discussions, respectively. The last term can be generically written as

$$\begin{aligned} k_x(\omega) &= -\mathrm{i}\omega \left(\frac{s_y s_z}{s_x} - 1 \right) \cdot \chi(\omega) E_x(\omega) \\ &= \left(\frac{s_y s_z}{s_x} - 1 \right) \cdot j_x(\omega), \end{aligned} \tag{7.11}$$

where $j_x(\omega)$ is the polarisation current as given by Eq. (2.50b), i.e., it models the dispersion of a material.

Inserting the definitions of s_i and applying some algebraic transformations eventually leads to an additional auxiliary field for each electric field component as soon as we want to combine UPMLs and dispersive materials. Since Eq. (7.11) holds for *any* polarisation current $j_x(\omega)$, it is also valid if $j_x(\omega)$ is the result of a combination of multiple Drude and Lorentz poles (see section 2.3.3). Hence, the combination of dispersion and UPMLs always, i.e., irrespective of the number of Drude and Lorentz poles, leads to *one* additional ADE per electromagnetic field component.

7.2 A Novel Stretched-Coordinate Implementation

As an alternative to the established UPML formulation, we have adapted the coordinate stretching approach of Chew and Weedon [103] to the DG method. Though their technique is often used in combination with the FDTD algorithm [21], its implementation within a DGTD framework is less straightforward. In this section we present our recently published results concerning this issue [39].

7.2.1 Stretched Coordinates in Maxwell's Equations

In a homogeneous medium, plane waves propagate according to $\exp(\mathrm{i}[\vec{\mathrm{k}} \cdot \vec{\mathrm{r}} - \omega t])$, where $\vec{\mathrm{k}}$ is the wave vector and ω the frequency of the wave (see section 2.2.1). To achieve spatial damping, either $\vec{\mathrm{k}}$ or $\vec{\mathrm{r}}$ has to be a complex-valued vector. A coordinate transformation, proposed by Chew and Weedon in 1994 [103], allows us to map real position vectors $\vec{\mathrm{r}}$ to the complex space. We can include such a stretching into Maxwell's equations via the substitutions [21]

$$\frac{\partial}{\partial x} \to \frac{1}{s_x(\omega)} \frac{\partial}{\partial x}, \quad \frac{\partial}{\partial y} \to \frac{1}{s_y(\omega)} \frac{\partial}{\partial y}, \quad \frac{\partial}{\partial z} \to \frac{1}{s_z(\omega)} \frac{\partial}{\partial z}, \tag{7.12}$$

where we have introduced complex, dispersive stretching factors $s_i(\omega)$. A choice commonly found in the FDTD literature [21] is given by

$$s_i(\omega) \equiv \kappa_i - \frac{\sigma_i}{\mathrm{i}\omega - \alpha_i}. \tag{7.13}$$

This particular choice includes three real-valued parameters which can be tuned for optimum performance. The main control parameter for the imaginary part of s_i is σ_i. For all $\sigma_i = 0$ we retrieve the non-absorbing physical space as indicated in the right-hand side of Fig. 7.2. A non-zero value of α_i shifts the pole from $\omega = 0$ to the complex frequency $\omega = -\mathrm{i}\alpha_i$. Hence, PMLs with this particular choice of the stretching factor are referred to as complex frequency-shifted PMLs (CFS-PMLs).

Finally, κ_i is the main contribution to the real part. Instead of adding absorption to the coordinate transform, $\kappa_i > 1$ effectively increases the width of the PML layer. This parameter plays an important role for FDTD, where one often resorts to an equidistant grid. For DGTD, however, this parameter is less relevant, since we can stretch the PML layer and the elements it consists of during the generation of the mesh. Thus, we can assume $\kappa_i \equiv 1$ without losing generality as compared to FDTD:

$$s_i(\omega) \equiv 1 - \frac{\sigma_i}{\mathrm{i}\omega - \alpha_i}. \tag{7.14}$$

In the following, we will show how the inclusion of the stretching factor (7.14) modifies Maxwell's curl equations (2.18). For brevity, we again restrict our discussion to the necessary changes to the E_x component. The relevant equation in the frequency-domain reads

$$-\mathrm{i}\omega\epsilon E_x(\omega) = \frac{1}{s_y}\partial_y H_z(\omega) - \frac{1}{s_z}\partial_z H_y(\omega). \tag{7.15}$$

Similarly to the derivation of the UPML formulation (section 7.1), we will introduce auxiliary differential equations which absorb the dispersive nature of the stretching factors. By splitting

$$\frac{1}{s_i} = 1 + \left(\frac{1}{s_i} - 1\right) \tag{7.16}$$

and introducing two auxiliary fields $G^E_{xy}(\omega)$ and $G^E_{xz}(\omega)$, we obtain

$$\begin{aligned} -\mathrm{i}\omega\epsilon E_x(\omega) = &\quad \partial_y H_z(\omega) - \partial_z H_y(\omega) - G^E_{xy}(\omega) - G^E_{xz}(\omega), \\ G^E_{xy}(\omega) = &-\left(\frac{1}{s_y} - 1\right) \cdot \partial_y H_z(\omega), \\ G^E_{xz}(\omega) = &\quad \left(\frac{1}{s_z} - 1\right) \cdot \partial_z H_y(\omega). \end{aligned} \tag{7.17}$$

Now we insert the identity

$$\frac{1}{s_i} - 1 = \frac{\sigma_i}{\mathrm{i}\omega - (\sigma_i + \alpha_i)} \tag{7.18}$$

and multiply by the denominator, which yields

$$\begin{aligned} -\mathrm{i}\omega G^E_{xy}(\omega) = &\quad \sigma_y \partial_y H_z(\omega) - \Big(\alpha_y + \sigma_y\Big) \cdot G^E_{xy}(\omega), \\ -\mathrm{i}\omega G^E_{xz}(\omega) = &-\sigma_z \partial_z H_y(\omega) - \Big(\alpha_z + \sigma_z\Big) \cdot G^E_{xz}(\omega). \end{aligned} \tag{7.19}$$

Applying the Fourier transformation rules (2.48) finally leads to the time-domain formulation

$$\begin{aligned} \epsilon\partial_t E_x(t) = &\quad \partial_y H_z(t) - \partial_z H_y(t) - G^E_{xy}(t) - G^E_{xz}(t), \\ \frac{1}{\sigma_y}\partial_t G^E_{xy}(t) = &\quad \partial_y H_z(t) - \frac{\alpha_y + \sigma_y}{\sigma_y} \cdot G^E_{xy}(t), \\ \frac{1}{\sigma_z}\partial_t G^E_{xz}(t) = &-\partial_z H_y(t) - \frac{\alpha_z + \sigma_z}{\sigma_z} \cdot G^E_{xz}(t). \end{aligned} \tag{7.20}$$

Similar equations for the other components of the electric field follow from cyclic permutations of the coordinate axes. Expressions for the magnetic field can be obtained by straightforward substitutions. Hence, Eq. (7.20) represents Maxwell's curl equations in a stretched coordinate system.

7.2.2 Spatial Discretisation via the Discontinuous Galerkin Method

The derivation of the DG discretisation for the stretched-coordinate formulation does not provide any conceptual difficulties and follows closely the procedure detailed in section 3.1. We start by recasting Maxwell's equations with stretched-coordinate ADEs into the conservation form

$$\mathcal{Q}\partial_t \mathbf{q} + \vec{\nabla}\cdot\vec{\mathrm{F}} + \mathbf{S} = 0. \tag{7.21}$$

The state vector $\mathbf{q}$ of the standard Maxwell case, Eq. (3.1), must be extended to include the auxiliary fields, i.e.,

$$\mathbf{q} = \Big(E_x,\, E_y,\, E_z, H_x, H_y, H_z,\, G^E_{xy},\, G^E_{xz},\, \ldots,\, G^E_{zy},\, G^H_{xy},\, G^H_{xz},\, \ldots,\, G^H_{zy}\Big)^T. \tag{7.22}$$

$G^{E/H}_{ij}$ denotes the auxiliary field which results from the modification of the j-derivative for the i-component of E or H, respectively. The material matrix $\mathcal{Q}$ is modified as well and reads

$$\mathcal{Q} = \mathrm{diag}\left(\epsilon,\, \epsilon,\, \epsilon,\, \mu,\, \mu,\, \mu,\, \frac{1}{\sigma_y},\, \frac{1}{\sigma_z},\, \frac{1}{\sigma_x},\, \ldots,\, \frac{1}{\sigma_y}\right). \tag{7.23}$$

Finally, we define the components

$$\begin{aligned}
\mathbf{F}_x(\mathbf{q}) &= \Big(0,\, H_z,\, -H_y,\, 0,\, -E_z,\, E_y,\, 0,\, 0,\, H_z,\, 0,\, 0,\, -H_y,\, 0,\, 0,\, -E_z,\, 0,\, 0,\, E_y\Big)^T,\\
\mathbf{F}_y(\mathbf{q}) &= \Big(-H_z,\, 0,\, H_x,\, E_z,\, 0,\, -E_x,\, -H_z,\, 0,\, 0,\, 0,\, H_x,\, 0,\, E_z,\, 0,\, 0,\, 0,\, -E_x,\, 0\Big)^T,\\
\mathbf{F}_z(\mathbf{q}) &= \Big(H_y,\, -H_x,\, 0,\, -E_y,\, E_x,\, 0,\, 0,\, H_y,\, 0,\, -H_x,\, 0,\, 0,\, 0,\, E_y,\, 0,\, -E_x,\, 0,\, 0\Big)^T
\end{aligned} \tag{7.24}$$

of the flux vector $\vec{\mathrm{F}} = (\mathbf{F}_x,\, \mathbf{F}_y, \mathbf{F}_z)^T$ and specify the newly introduced source vector [28] as

$$\mathbf{S} = \left(G^E_{xy} + G^E_{xz},\, \ldots,\, G^H_{zx} + G^H_{zy},\, \frac{\alpha_y+\sigma_y}{\sigma_y}G^E_{xy},\, \ldots,\, \frac{\alpha_y+\sigma_y}{\sigma_y}G^H_{zy}\right)^T. \tag{7.25}$$

After conducting the usual steps, we obtain the following semi-discrete form for the x-component of the electric field and related ADEs:

$$\begin{aligned}
\epsilon^{\triangle}\partial_t \tilde{\mathrm{E}}^{\triangle}_x &= \mathcal{D}^{\triangle}_y\cdot\tilde{\mathrm{H}}^{\triangle}_z - \mathcal{D}^{\triangle}_z\cdot\tilde{\mathrm{H}}^{\triangle}_y - \left(\tilde{\mathrm{G}}^{\triangle}_{xy} + \tilde{\mathrm{G}}^{\triangle}_{xz}\right) + \left(\mathcal{M}^{\triangle}\right)^{-1}\mathcal{F}^{\triangle}_f\cdot\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{E_x,\,f},\\
\frac{1}{\sigma^{\triangle}_y}\partial_t \tilde{\mathrm{G}}^{\triangle}_{xy} &= \mathcal{D}^{\triangle}_y\cdot\tilde{\mathrm{H}}^{\triangle}_z - \left(\alpha^{\triangle}_y + \sigma^{\triangle}_y\right)\cdot G^{\triangle}_{xy} + \left(\mathcal{M}^{\triangle}\right)^{-1}\mathcal{F}^{\triangle}_f\cdot\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xy},\,f},\\
\frac{1}{\sigma^{\triangle}_z}\partial_t \tilde{\mathrm{G}}^{\triangle}_{xz} &= -\mathcal{D}^{\triangle}_z\cdot\tilde{\mathrm{H}}^{\triangle}_y - \left(\alpha^{\triangle}_z + \sigma^{\triangle}_z\right)\cdot G^{\triangle}_{xz} + \left(\mathcal{M}^{\triangle}\right)^{-1}\mathcal{F}^{\triangle}_f\cdot\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xz},\,f}.
\end{aligned} \tag{7.26}$$

In comparison to the semi-discrete form (7.9) of the UPML formulation, we note the presence of flux terms in the ADEs. The origin of these terms are the spatial derivatives in the continuous equations (7.20). For this reason we have to find a modified expression for the numerical flux.

7.2.3 The Split-Flux Formulation

The numerical flux (compare sections 3.1.4 and 3.2.1) is the remaining key ingredient of the spatial discretisation. It is responsible for the exchange of information between adjacent elements and, as a result, crucial for the accuracy and stability of the whole numerical scheme. The upwind flux (3.8) is our choice for the standard Maxwell equations.

Ultimately, our new flux (including the components for the ADEs) should be compatible with the established formulation. In particular, the flux components relevant for the electric and magnetic field components should correspond to the upwind case. Proposing a new flux is particularly easy once we note that

$$\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}}-\vec{\mathrm{F}}^{*}\right)\right]_{\vec{\mathrm{E}}}=\frac{1}{\bar{Z}}\Big(\alpha\Big[\Delta\vec{\mathrm{E}}-\hat{\mathrm{n}}(\hat{\mathrm{n}}\cdot\Delta\vec{\mathrm{E}})\Big]+Z^{+}\hat{\mathrm{n}}\times\Delta\vec{\mathrm{H}}\Big) \qquad \text{(from (3.8))}$$

$$\equiv\frac{1}{\bar{Z}}\hat{\mathrm{n}}\times\Big(Z^{+}\Delta\vec{\mathrm{H}}-\alpha\hat{\mathrm{n}}\times\Delta\vec{\mathrm{E}}\Big). \tag{7.27}$$

The upwind parameter α should not be confused with the frequency shifts α_i.[2] Apparently, the curl operator (which motivated the flux definition in section 3.1.1) translates to a vector product with the normal vector $\hat{\mathrm{n}}$ on a face of an element. It seems natural to split the vector product into two contributions once we deal with just *one* spatial derivative[3] of the curl operator. Hence, we propose to use

$$\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}}-\vec{\mathrm{F}}^{*}\right)\right]_{G_{ij}^{E}}=\sum_{k=1}^{3}\varepsilon_{ijk}\frac{n_j}{\bar{Z}}\Big(Z^{+}\Delta\vec{\mathrm{H}}-\alpha\hat{\mathrm{n}}\times\Delta\vec{\mathrm{E}}\Big)_{k} \tag{7.28}$$

as the numerical flux for the stretched-coordinate formulation of PMLs. Please note the use of the Levi-Civita symbol

$$\varepsilon_{ijk}=\begin{cases}1 & \text{for } (ijk)\equiv(123) \text{ and cyclic permutations}\\ -1 & \text{for } (ijk)\equiv(213) \text{ and cyclic permutations}\\ 0 & \text{for all other triples } (ijk)\end{cases}. \tag{7.29}$$

Analogue expressions hold for the magnetic field components and the respective auxiliary equations.

An interesting analogy to Bérenger's original split field formulation [101] surfaces in the identity

$$\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}}-\vec{\mathrm{F}}^{*}\right)\right]_{E_x}\equiv\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}}-\vec{\mathrm{F}}^{*}\right)\right]_{G_{xy}}+\left[\hat{\mathrm{n}}\cdot\left(\vec{\mathrm{F}}-\vec{\mathrm{F}}^{*}\right)\right]_{G_{xz}}. \tag{7.30}$$

Instead of splitting the electromagnetic fields, our stretched-coordinate formulation relies on splitting the numerical flux into two contributions. Hence, the term "split-flux formulation" seems appropriate.

[2] Unfortunately, the use of α in both contexts is established in the literature.
[3] In this case $\partial_y H_z$ and $\partial_z H_y$.

7.2.4 Stability

Since we do not provide an explicit proof of the stability of our stretched-coordinate formulation, we perform numerical time-domain experiments to demonstrate stability. To this end, we continuously illuminate a spherical object ($\epsilon = 2.25$) of diameter 2 with a plane wave of wavelength $\lambda = 2.5$ (all lengths are given in dimensionless units, see section 2.1.4). Scattered radiation is absorbed by a single layer of stretched-coordinate PMLs. The computational domain is terminated by a perfect electric conductor to exclude absorption of the boundaries as a means of extracting energy from the computational domain. We record the time-dependent electric field at a few points within the computational domain and monitor the amplitudes of the oscillations. After simulating a few time units, a steady state is reached and the amplitudes do not change any more. In particular, the amplitudes do not grow exponentially with time. After simulating for $100,000$ time units, i.e., $40,000$ optical cycles, we conclude that our implementation is numerically stable. Similarly, numerical experiments will demonstrate the correctness of our formulation in section 7.3.2.

7.2.5 Some Differences to UPMLs

Apparently, the inclusion of CFS-PMLs in Maxwell's equations (7.26) leads to the rather significant overhead of two auxiliary fields per electromagnetic field component. In comparison, the UPML formulation Eq. (7.9) requires only one auxiliary field per field component. Please note, however, that the auxiliary fields in both formulations can be omitted in regions where no PMLs are present, i.e., where $\sigma_i = 0$ for all i.

Furthermore, the stretched-coordinate formulation features matrix-vector products in the auxiliary equations. Hence, the computational effort to calculate the right-hand sides of Eq. (7.26) is significantly higher than that of the corresponding scalar operations in Eq. (7.9).

On the other hand, during the derivation of the stretched-coordinate formulation we have not made any assumptions besides isotropy concerning the materials. Thus, the stretched-coordinate formulation intrinsically supports dispersive materials (and in principle even nonlinear ones), which are usually implemented via auxiliary polarisation currents. The UPML formulation, in contrast, requires additional auxiliary fields to combine dispersive materials and PML regions. Hence, the memory consumption of both PML formulations is equal in such overlapping metal/PML regions.

Last but not least, it is worthwhile to remark that CFS-PMLs provide additional free PML parameters α_i. In some situations, these may help us to improve the performance of the absorbing boundary layer, but more on that matter in the next section.

7.3 Parameter Choice and Performance Comparison

In the previous sections we have presented two implementations of PMLs. Together with Silver-Müller absorbing boundary conditions introduced in section 3.1.5, we have a total of three methods which essentially serve the same purpose. Hence, it is interesting to investigate and compare their respective numerical performance. To this end, we study the influence of the PML parameters on the numerical error with the help of a simple test case.

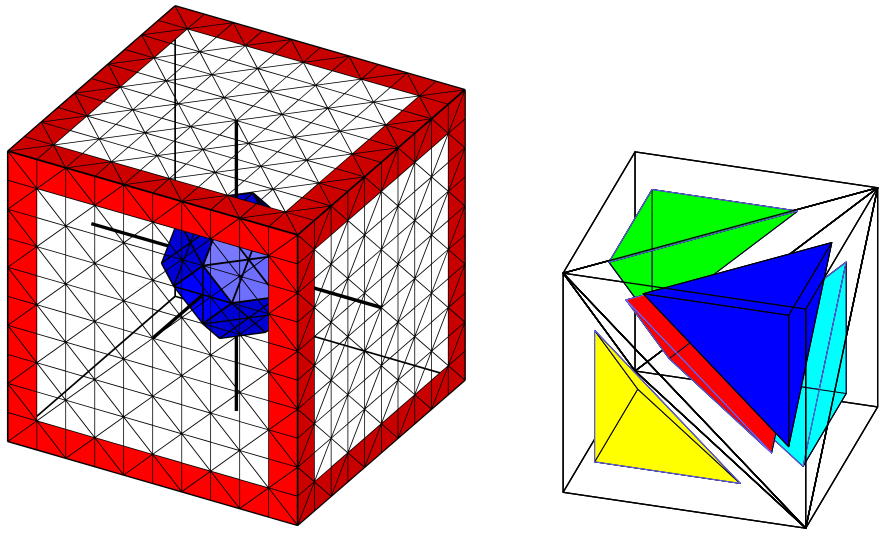


Figure 7.3: *Setup of the test system. The left panel shows a surface mesh of the computational domain. The red triangles indicate the extent of the boundary layer. The blue surface is used for the injection of radiation as generated by an oscillating dipole located at the centre of the system. The volume mesh consists of a number of tetrahedrons, a few of which are depicted in the right panel. As outlined in the text, five tetrahedrons make up a cube. For better visibility, all tetrahedrons have been coloured and shrunken.*

7.3.1 Test Configuration

Our physical test system consists of vacuum with a radiating point dipole located at the origin and was used in similar form to optimise UPML parameters in the literature [34,54]. As no scatterers are present, outgoing radiation should propagate towards the infinity of space. We model this physical problem by a cubic computational domain centred around the origin. The computational domain consists of the actual domain of interest and a surrounding boundary layer. The domain of interest is given by $\vec{r} \in [-2.0,\ 2.0]^3$ in dimensionless units. The boundary layer has a variable thickness. A sketch of the computational domain is presented in Fig. 7.3.

Manually generated, regular meshes are used to avoid mesh-induced perturbations of outgoing waves.[4] To this end, we divide the whole computational domain into small cubes of edge length $\Delta x = 0.5$. In turn, each of these cubes is made up of five tetrahedrons (see Fig. 7.3). We further minimise the mesh anisotropy by dividing the computational domain into eight sectors separated by the xy-, yz-, and xz-planes. Within each sector all cubes share the same configuration of tetrahedrons. The respective configurations in the other sectors follow by mirror symmetry operations (cf. Fig. 7.3).

We include the dipole source by means of the total-field/scattered-field method (section 6.1.1). The injection surface surrounds the origin and was chosen to be as spherical as possible with the

[4] More specifically, we use a custom `MATLAB` script to generate the meshes in a well-defined and orderly fashion.

given mesh (Fig. 7.3) to minimise numerical errors. The time-dependence of the dipole is given by

$$\vec{\mathrm{j}}(t) = \begin{cases} \hat{\mathrm{e}}_z \cdot \sin(\omega_0 t) \cdot \exp\Big(- \frac{(t-t_0)^2}{2w^2} \Big), & 0 \leq t \leq 2t_0 \\ 0, & \text{otherwise} \end{cases} \tag{7.31}$$

where $\hat{\mathrm{e}}_z$ represents the unit vector in z-direction. The excitation parameters in dimensionless units are given by $\omega_0 = 0.4 \cdot 2\pi$, $w = 1$, and $t_0 = 5$. We simulate this system for a total of 15 time units using the DGTD method. During the simulation, the time-dependence of E_z is recorded for each time step t_i at three different observation points

$$\vec{\mathrm{r}}_A = (1.8,\, 0.0,\, 0.0)^T, \qquad \vec{\mathrm{r}}_B = (1.8,\, 1.8,\, 0.0)^T, \qquad \vec{\mathrm{r}}_C = (1.8,\, 1.8,\, 1.8)^T. \tag{7.32}$$

These points lie in close vicinity to the boundary layer. Hence, potential distortions of the outgoing waves due to the PMLs should be quite pronounced.

The accuracy of the absorbing boundary can be obtained by comparing the numerical solution with a reference field distribution. Even though an analytical reference for dipole radiation is readily available, it is not suitable for our purpose. Each numerical simulation introduces discretisation errors which might overshadow the influence of the boundary condition.

Instead, we perform another DGTD simulation on the much larger computational domain $\vec{\mathrm{r}} \in [-10.0,\, 10.0]^3$ terminated by a perfect electric conductor (PEC boundary condition). The mesh is generated in the same fashion as outlined earlier. This system, too, is simulated for 15 time units with the same time step as the smaller system. During this time, the dipole source emits radiation which propagates towards the boundaries of the computational domain. Since the system is terminated by PECs, all radiation is reflected. As the speed of light in dimensionless units is given by 1, a distance of 10 units from the centre to the boundary guarantees that within the simulation time the field at the observation points is not influenced by the reflected waves. Thus, the enlarged system provides a reference solution which incorporates the *very same* spatial and temporal discretisation errors as our performance test system. Consequently, it is perfectly suited to isolate the errors due to the truncation of the domain with absorbing boundaries from other sources.

Finally, we define the error of the absorbing boundary implementation as

$$\mathrm{Error}\,(\vec{\mathrm{r}}_p) = \frac{\max\limits_{t_i \leq 15} \Big| E_z^{\mathrm{abs}}(\vec{\mathrm{r}}_p,\, t_i) - E_z^{\mathrm{ref}}(\vec{\mathrm{r}}_p,\, t_i) \Big|}{\max\limits_{t_i \leq 15} \Big| E_z^{\mathrm{ref}}(\vec{\mathrm{r}}_p,\, t_i) \Big|}, \qquad p = A,\, B,\, C. \tag{7.33}$$

This error measure represents the largest relative deviation between the reference simulation on the larger system (index "ref") and the one with an absorber (index "abs"), where we take the maximum over all time steps t_i. To get a representative error measure for the whole computational domain, we define the average error

$$\mathcal{E} = \frac{1}{3}\Big[\mathrm{Error}\,(\vec{\mathrm{r}}_A) + \mathrm{Error}\,(\vec{\mathrm{r}}_B) + \mathrm{Error}\,(\vec{\mathrm{r}}_C) \Big]. \tag{7.34}$$

7.3.2 Numerical Results

Using the setup described in the previous section we perform a number of simulations for various parameters. For a boundary layer of size Δx, i.e., for one layer of cubes as described in the previous

section, parameter scans for five different configurations of absorbing boundaries are conducted. These configurations are classified as

- Only a Silver-Müller boundary condition (SMBC) without PMLs,
- UPMLs combined with a PEC boundary condition,
- UPMLs combined with a Silver-Müller boundary condition,
- Stretched-coordinate PMLs with a PEC boundary condition,
- Stretched-coordinate PMLs with a Silver-Müller boundary condition.

With the exception of the first (parameter-free) configuration, we vary the parameter $\sigma \equiv \sigma_x \equiv \sigma_y \equiv \sigma_z$ in 51 steps. More precisely, σ is obtained via the formula

$$\sigma = \frac{R}{2d}, \tag{7.35}$$

where d is the thickness of the PML layer and R is a parameter which was varied in the range $[0.0,\ 20.0]$ in steps of $\Delta R = 0.4$. Equation (7.35) is also used in the FDTD community, where it is usually combined with a polynomial grading [21]. Such a grading, however, was previously shown to have no beneficial impact on the PML error in a DGTD code [34, 54]. Thus, we do not employ one in our study. The stretched-coordinate formulation also allows us to tune the complex frequency-shift[5] $\alpha \equiv \alpha_x \equiv \alpha_y \equiv \alpha_z$, which we vary in steps of $\Delta\alpha = 0.05$ in the ranges $[0.0,\ 1.0]$ (PEC) and $[0.0,\ 2.0]$ (Silver-Müller), respectively.

These calculations are repeated for Lagrange polynomials of orders $p = 3,\ 4,\ 5$, where higher orders represent a finer spatial discretisation (see section 3.1.6). Additional simulations are performed for $p = 3$ with SMBCs, where we investigate the influence of additional layers of PMLs, i.e., two and three layers. The results are depicted in Figs. 7.4 to 7.7.

Figure 7.4 illustrates the performance of the stretched-coordinate PML formulation for $p = 3$. The left panel shows the results for the computational domain being terminated by PEC boundary conditions, while an SMBC was used in the right panel. The averaged error $\mathcal{E}(R,\ \alpha)$ is mapped on a logarithmic pseudo-colour scale. For better visibility each order of magnitude has been assigned a distinct colour. Moreover, black lines represent isocontours of the error in the R-α-plane. Thick lines indicate the orders of magnitude while thin lines are isocontours defined by

$$\log_{10} \mathcal{E}(R,\ \alpha) \equiv 0.1 \cdot n, \tag{7.36}$$

where n is an arbitrary integer. The parameter set with the minimum error is marked by a black cross. For comparison with the UPML formulation we added another isocontour (thick blue contour). Its level is given by the minimum error obtained from corresponding UPML simulations with *optimised* σ. We do not present individual plots for the UPML results, since this has been done earlier in Refs. [34, 54], for example. Due to the poor performance of SMBCs without PMLs, we refrain from including a corresponding isocontour in the system. Please note that we have kept the size of the computational domain the same for all simulations in Fig. 7.4. This guarantees a fair comparison of the different techniques. Finally, the error of the SMBC and the minimum errors of the UPML and CFS-PML calculations are included in the respective colour bars. This facilitates a quick quantitative overview on the performance of the various methods. The panels in Figs. 7.5 to 7.7 are to be understood in this fashion, too.

[5] The complex frequency shift α should not be confused with the upwind parameter α of the numerical flux. Throughout this study, the upwind parameter is set to 1.

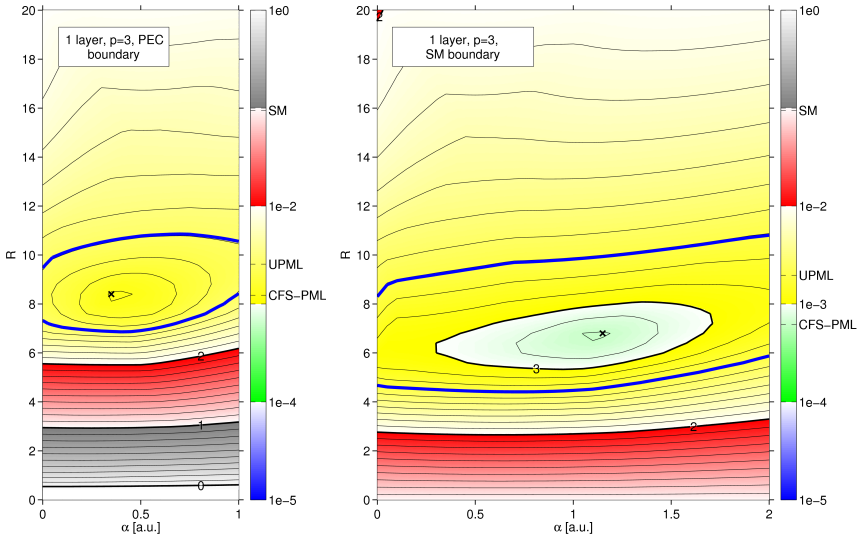


Figure 7.4: *Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 3$ and one layer of PMLs. The left panel shows the average error $\mathcal{E}$ for a test system terminated using the PEC boundary condition. The right-hand side shows corresponding results with the PEC replaced by a Silver-Müller boundary condition. Please note the logarithmic scales of the pseudo-colour plots. The blue isocontour represents the level of accuracy achieved with UPMLs with optimal parameters. These levels together with the best value for stretched-coordinate PMLs are included in the colour bar as well.*

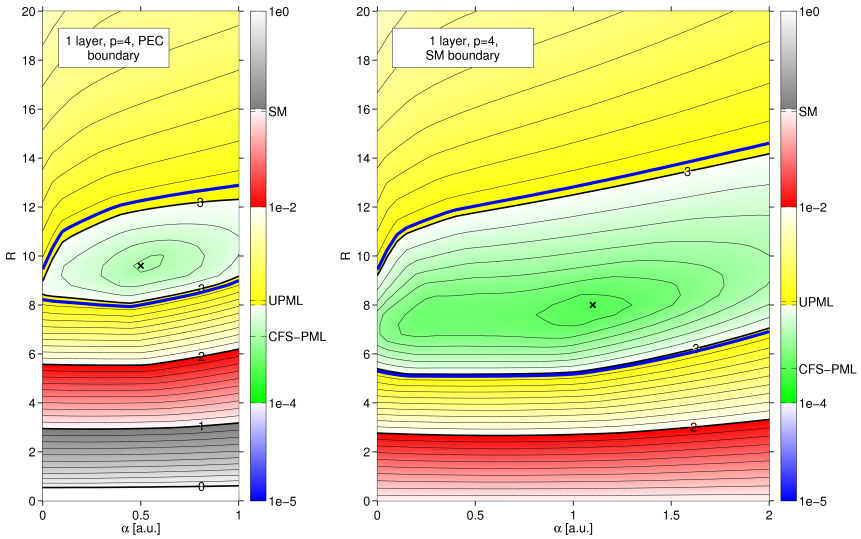


Figure 7.5: *Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 4$ and one layer of PMLs. For a more detailed explanation please refer to the caption of Fig.* 7.4.

Order p	**Optimal** R	**Minimal error** $\mathcal{E}$
3	6.8	$6.2 \cdot 10^{-4}$
4	8.0	$2.2 \cdot 10^{-4}$
5	8.4	$6.6 \cdot 10^{-5}$

Table 7.1: *Optimal R-parameters for one layer of stretched-coordinate PMLs for various orders p. The computational domain is terminated using a Silver-Müller boundary condition. Please note that the optimal value of the complex frequency-shift α depends on the excitation and, thus, cannot be easily tabularised.*

The results for one layer of PMLs and $p = 3,\ 4,\ 5$ (Figs. 7.4 to 7.6) show quite a few similarities. For fixed α, the variation of R leads to strong variations of the error. If R is too small, the PMLs do not sufficiently absorb radiation: Eventually, it is (partly) reflected at the outer boundary of the computational domain (see Figs. 7.1 and 7.2) and reenters the physical region. If R is too high, then the PMLs absorb radiation too fast, i.e., the mesh cannot properly resolve the abrupt spatial change of the electromagnetic fields. For extreme values of R we would just observe an almost immediate drop from a finite value to zero. This situation would be comparable to a perfect electric conductor, inside of which electromagnetic fields cannot exist. Of such a conductor, however, we know that it presents a mirror to incoming radiation. Hence, high values of R lead to numerically induced reflection from the PML interface. The optimum value of R balances both effects.

We observe that both UPMLs and our stretched-coordinate formulation considerably outperform the simple SMBC. Nevertheless, combining the Silver-Müller condition with PMLs helps to reduce the error. At the same time, the error becomes less sensitive on the PML parameters. Since the SMBC is easily implemented via the numerical flux, its usage is essentially for free and does not introduce any computational overhead over PEC boundaries. Hence, we recommend to always accompany PMLs with the Silver-Müller boundary condition. For this case, Table 7.1 lists optimum parameter values R for various polynomial orders p.

It is also obvious that the additional freedom in choosing α significantly reduces the error. For $p = 5$ and Silver-Müller termination, the best error for the stretched-coordinate formulation is one order of magnitude smaller than the corresponding UPML error (see Fig. 7.6). It should be noted, though, that this impressive result strongly depends on the ratio between the wavelength and the thickness of the PML. In experiments with other pulse parameters (increased ω_0) we observe that $\alpha \neq 0$ is only advantageous if the PML is small compared to the wavelength of the incident light. This result is consistent with the FDTD literature [21]. We also observe this behaviour in Fig. 7.7, where we have increased the size of the boundary region to two and three layers. The optimum value for α is very close to 0 and the accuracy gains of the stretched-coordinate formulation over the UPML formulation are quite small. Since the optimum value of α crucially depends on the incident wavelength to PML width ratio, it is difficult to give general recommendations.

A final lesson we can learn from Figs. 7.4 to 7.7 is that we can reduce spurious reflections by improving the spatial discretisation. Either we use higher-order Lagrange polynomials or we increase the thickness of the PML layer. Both approaches effectively distribute more degrees of freedom in the PML region. As a result, the evanescent decay of the incident radiation is much better resolved,

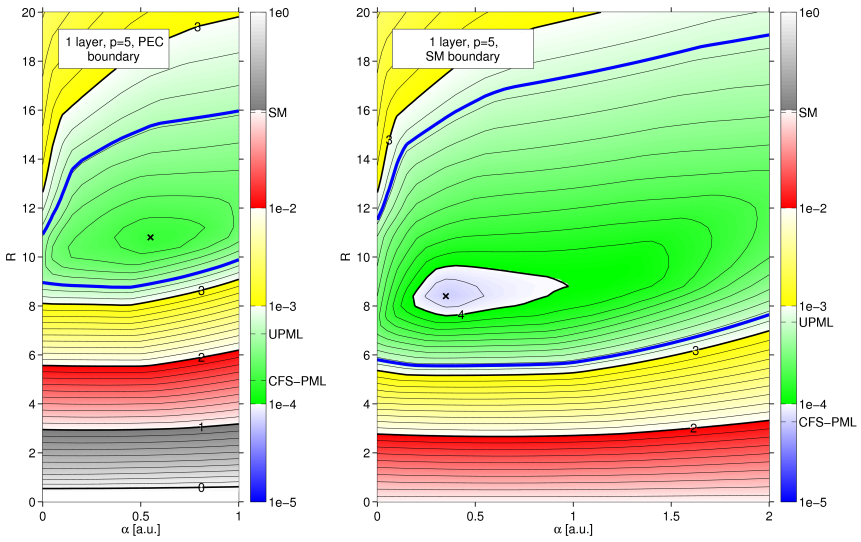


Figure 7.6: *Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 5$ and one layer of PMLs. For a more detailed explanation please refer to the caption of Fig.* 7.4.

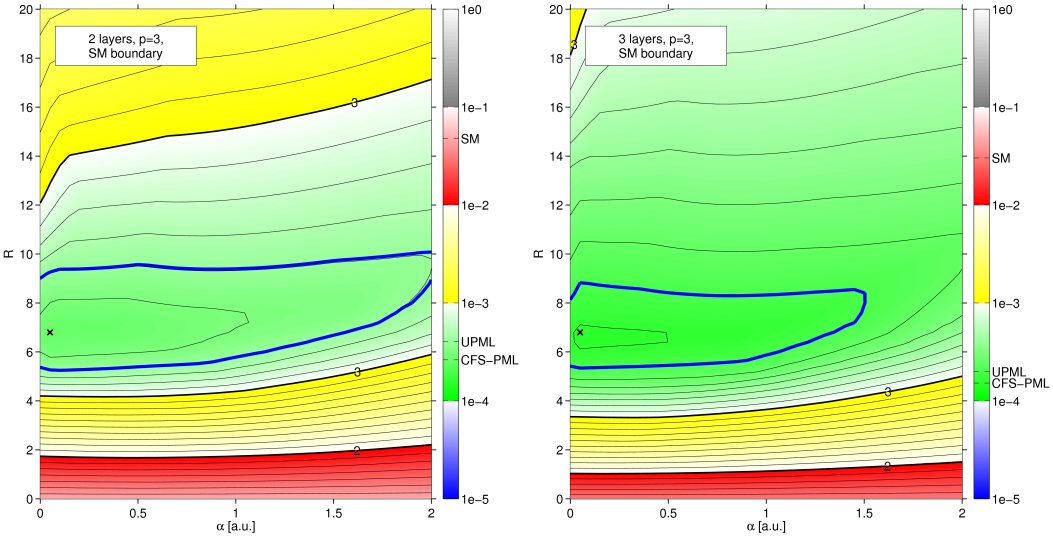


Figure 7.7: *Comparison of the PML performance for two and three layers and $p = 3$. For a more detailed explanation please refer to the caption of Fig.* 7.4.

System	$\mathbf{N_{tot}}$	$\mathbf{N_{PML}}$	Time	Relative time	Error
No PMLs	5,000	—	132 s	100% / —	$9.5 \cdot 10^{-2}$
UPML, 1 layer	5,000	2,440	155 s	117% / 100%	$2.0 \cdot 10^{-3}$
CFS-PML, 1 layer	5,000	2,440	185 s	140% / 119%	$6.2 \cdot 10^{-4}$
UPML, 2 layers	8,640	6,080	279 s	— / 180%	$4.0 \cdot 10^{-4}$

Table 7.2: *Computational effort required for the simulation of various systems. The table features a short system description (all systems terminated by SMBCs), the total number of elements N_{tot}in the mesh, the number of PML elements N_{PML} in the mesh, and the CPU time required to evolve the respective system for 15 time units (see section* 7.3.2*). The next column compares the CPU time against the system without PMLs and against the system with one layer of UPMLs. Finally, the last column provides an error comparison for the different boundary conditions.*

and thus reflections due to insufficient resolution are diminished.

7.3.3 Computational Efficiency

As we have seen in the previous section, stretched-coordinate PMLs can significantly suppress the numerical errors introduced by the boundary of the computational domain. In this section we provide some numbers regarding the associated computational overhead.

To this end, we measure the time required to simulate the test systems of the previous section. All systems are terminated by Silver-Müller boundary conditions. All simulations are performed with third-order polynomials ($p = 3$) on a single core of an Intel® Core™ 2 Quad CPU (Q9300) running at $2.50\,\mathrm{GHz}$. CPU times are recorded for four systems. The first one does not comprise any PMLs. The next ones feature single layers of UPMLs and stretched-coordinate CFS-PMLs, respectively. The last system is surrounded by two layers of UPMLs. The PML parameter values correspond to the best values in Figs. 7.4 and 7.7. Please note that for these optimal parameters the residual error when using one layer of CFS-PMLs is approximately the same as the error caused by two layers of UPMLs.

The resulting CPU times can be found in Table 7.2. As compared to the single layer UPML formulation, the stretched-coordinate formulation requires 19% more time for the same physical system. However, it provides more accurate results. On the other hand, two layers of UPMLs require 80% more time than the single layer computation, while providing only slightly more accurate results than the CFS-PML simulations. It should be noted, though, that in these systems an unusually large fraction of the mesh elements is filled with PMLs. In practice, the relative number of PML elements is often below 10% if one resorts to one layer of PMLs.

7.4 Summary

In this chapter, we have presented two implementations of perfectly matched layers. UPMLs are easy to implement. Stretched-coordinate PMLs lead to significant computational overhead and re-

quire a modified numerical (split-) flux, but are material-independent and provide an additional free parameter. Performance comparisons indicate that stretched-coordinate PMLs may lead to faster calculations than UPMLs in situations where

1. the ratio of wavelength to PML thickness is large, i.e., if we have a setup where the complex frequency-shift α provides an advantage, and
2. one actually requires low PML errors.

In most other situations UPMLs should be sufficient. In any case, PMLs should always be accompanied by Silver-Müller boundary conditions to reduce numerical errors and the sensitivity of the PML's performance on the PML parameters.

Chapter 8

8 Anisotropic Materials in the Discontinuous Galerkin Framework

The discussion of the discontinuous Galerkin method, so far, was limited to the case of isotropic permittivities. This excludes a wide class of interesting materials and related physics. For example, anisotropic materials provide a way to achieve so-called invisibility cloaking using carpets, cylindrical structures, or more general geometries [7, 106, 107].

The treatment of anisotropic materials within a DGTD approach was discussed in Ref. [108], where the authors employed a central flux (see Eq. (3.8) with $\alpha = 0$) to interconnect neighbouring elements. However, a central flux may reduce the accuracy because spurious modes are not sufficiently damped. Furthermore, it may introduce instabilities when used in combination with perfectly matched layers [68]. Thus, it is highly desirable to generalise on previous results reported in the literature.

In the following, we will present how to extend the standard DG method to anisotropic materials using an upwind flux as in section 3.1.4. These results were recently published in Ref. [40]. Section 8.1 quickly derives the semi-discrete form of Maxwell's equations with anisotropic materials. Section 8.2 solves the Riemann problem to obtain an upwind numerical flux. The algorithm is tested in section 8.3 with a small cavity. An optical cloak serves as a more advanced application and a stress test in section 8.4 before we summarise the results of this chapter in section 8.5.

8.1 The Semi-Discrete Form of the Anisotropic Maxwell Equations

To simplify the problem, we restrict ourselves to the important case of two-dimensional systems in TE-polarisation, i.e., the only non-vanishing components of the electromagnetic fields are E_x, E_y, and H_z (section 2.1.5). Furthermore, we assume an anisotropic permittivity tensor

$$\underline{\epsilon} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{yx} & \epsilon_{yy} \end{pmatrix} \tag{8.1}$$

and an isotropic permeability $\mu \equiv \mu_{zz}$. The corresponding curl equations (2.22) have already been derived in section 2.1.5. Please note that for notational convenience we have reduced the three-dimensional material tensors of Eq. (2.21) to those parts relevant to the TE case. Our first task will

be to discretise the spatial dependencies in Maxwell's equations using the discontinuous Galerkin method. Since conceptual differences to the standard derivation given in section 3.1 are minuscule, we just state the end result:

$$\begin{aligned}
\partial_t \tilde{\mathrm{E}}_x^{\triangle} &= \eta_{xx}^{\triangle} \mathcal{D}_y^{\triangle} \tilde{\mathrm{H}}_z^{\triangle} + \eta_{xx}^{\triangle} \Big(\mathcal{M}^{\triangle}\Big)^{-1} \mathcal{F}_f^{\triangle} \Big[\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)\Big]_{E_x,\, f} \\
&\quad - \eta_{xy}^{\triangle} \mathcal{D}_x^{\triangle} \tilde{\mathrm{H}}_z^{\triangle} + \eta_{xy}^{\triangle} \Big(\mathcal{M}^{\triangle}\Big)^{-1} \mathcal{F}_f^{\triangle} \Big[\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)\Big]_{E_y,\, f}, \\
\partial_t \tilde{\mathrm{E}}_y^{\triangle} &= \eta_{yx}^{\triangle} \mathcal{D}_y^{\triangle} \tilde{\mathrm{H}}_z^{\triangle} + \eta_{yx}^{\triangle} \Big(\mathcal{M}^{\triangle}\Big)^{-1} \mathcal{F}_f^{\triangle} \Big[\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)\Big]_{E_x,\, f} \\
&\quad - \eta_{yy}^{\triangle} \mathcal{D}_x^{\triangle} \tilde{\mathrm{H}}_z^{\triangle} + \eta_{yy}^{\triangle} \Big(\mathcal{M}^{\triangle}\Big)^{-1} \mathcal{F}_f^{\triangle} \Big[\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)\Big]_{E_y,\, f}, \\
\partial_t \tilde{\mathrm{H}}_z^{\triangle} &= \frac{1}{\mu^{\triangle}} \mathcal{D}_y^{\triangle} \tilde{\mathrm{E}}_x^{\triangle} - \frac{1}{\mu^{\triangle}} \mathcal{D}_x^{\triangle} \tilde{\mathrm{E}}_y^{\triangle} + \frac{1}{\mu^{\triangle}} \Big(\mathcal{M}^{\triangle}\Big)^{-1} \mathcal{F}_f^{\triangle} \Big[\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)\Big]_{H_z,\, f}.
\end{aligned} \tag{8.2}$$

Please recall the definition of the inverse permittivity tensor $\underline{\eta} \equiv \underline{\epsilon}^{-1}$. Furthermore, we want to point out that the operation $\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)$ yields a vector with the same size as the state vector $\tilde{\mathbf{q}}^{\triangle} = (\tilde{\mathrm{E}}_x^{\triangle}, \tilde{\mathrm{E}}_y^{\triangle}, \tilde{\mathrm{H}}_z^{\triangle})^T$. Hence, the subscripts of this term refer to the expansion coefficients of the corresponding field component of $\mathbf{q}$. The additional subscript "f" again refers to the faces of the element $\triangle$.

8.2 An Upwind Flux for Anisotropic Materials

To complete our extension to anisotropic materials, we have to obtain an upwind expression for $\hat{\mathrm{n}} \cdot (\vec{\mathrm{F}} - \vec{\mathrm{F}}^*)$. It is clear that Eq. (3.8) cannot be the correct solution in the anisotropic case, since it only involves a *scalar* permittivity ϵ.

In search of an upwind numerical flux, we have to solve the Riemann problem [37]

$$c^- \mathcal{Q}^- \cdot \Big(\mathbf{q}^* - \mathbf{q}^-\Big) + \hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}}^* - \vec{\mathrm{F}}^-\Big) = 0, \tag{3.36a}$$

$$\hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}}^* - \vec{\mathrm{F}}^{**}\Big) = 0, \tag{3.36b}$$

$$-c^+ \mathcal{Q}^+ \cdot \Big(\mathbf{q}^{**} - \mathbf{q}^+\Big) + \hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}}^{**} - \vec{\mathrm{F}}^+\Big) = 0 \tag{3.36c}$$

previously introduced in section 3.2.1. In comparison to the Riemann problem for isotropic materials, however, the speed with which a wave travels along the direction of the unit normal is different:

$$c^{\pm} = \sqrt{\frac{\hat{\mathrm{n}}^T \underline{\epsilon}^{\pm} \hat{\mathrm{n}}}{\mu^{\pm} \det(\underline{\epsilon}^{\pm})}}. \tag{8.3}$$

After some tedious algebraic manipulations we find

$$H_z^- - H_z^* = \frac{1}{d} \cdot \Big[c^- \det(\underline{\epsilon}^-) \cdot \Big(\hat{\mathrm{n}}^T \underline{\epsilon}^+ \hat{\mathrm{n}}\Big) \cdot \Delta H_z - c^- c^+ \det(\underline{\epsilon}^-) \det(\underline{\epsilon}^+) \cdot \Big(\hat{\mathrm{n}} \times \Delta \vec{\mathrm{E}}\Big)_z\Big], \tag{8.4}$$

where the denominator d is given by

$$d = c^- \det(\underline{\epsilon}^-) \cdot \Big(\hat{\mathrm{n}}^T \underline{\epsilon}^+ \hat{\mathrm{n}}\Big) + c^+ \det(\underline{\epsilon}^+) \cdot \Big(\hat{\mathrm{n}}^T \underline{\epsilon}^- \hat{\mathrm{n}}\Big). \tag{8.5}$$

The E_x and E_y components of the numerical flux follow immediately from Eq. (8.4). Furthermore, we obtain

$$\Big[\hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}}^{-} - \vec{\mathrm{F}}^{*}\Big)\Big]_{H_z} = \frac{1}{c^{-}\mu^{-} + c^{+}\mu^{+}} \Big[- c^{-}c^{+}\mu^{-}\mu^{+} \cdot \Delta H_z + c^{-}\mu^{-} \cdot \Big(\hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}\Big)_z \Big]. \tag{8.6}$$

The intermediate results (8.4) and (8.6) can be simplified by eliminating $\det(\underline{\epsilon}^{\pm})$ and $\hat{\mathrm{n}}^T \underline{\epsilon}^{\pm} \hat{\mathrm{n}}$ using the square of (8.3). Finally, the numerical flux reads

$$\hat{\mathrm{n}} \cdot \Big(\vec{\mathrm{F}} - \vec{\mathrm{F}}^{*}\Big) = \begin{pmatrix} -n_y(H_z^{-} - H_z^{*}) \\ n_x(H_z^{-} - H_z^{*}) \\ n_x(E_y^{-} - E_y^{*}) \\ -n_y(E_x^{-} - E_x^{*}) \end{pmatrix} = \begin{pmatrix} \dfrac{-n_y}{Z^{+} + Z^{-}} \Big[Z^{+}\Delta H_z - \Big(n_x \Delta E_y - n_y \Delta E_x \Big) \Big] \\ \dfrac{n_x}{Z^{+} + Z^{-}} \Big[Z^{+}\Delta H_z - \Big(n_x \Delta E_y - n_y \Delta E_x \Big) \Big] \\ \dfrac{1}{Y^{+} + Y^{-}} \Big[Y^{+} \Big(n_x \Delta E_y - n_y \Delta E_x \Big) - \Delta H_z \Big] \end{pmatrix}, \tag{8.7}$$

where we have defined $Z^{\pm} = \mu^{\pm} c^{\pm}$ and $Y^{\pm} = (Z^{\pm})^{-1}$ as before.[1]

A closer examination of Eq. (8.7) reveals that it has the exact same form as the numerical flux (3.8) for two-dimensional TE polarisation and isotropic materials. The only difference is hidden in the speed of light, Eq. (8.3), which eventually enters the numerical flux via the impedances and admittances. Fortunately, Eq. (8.3) is closely connected to the speed of light in isotropic media. If we assume a medium characterised by an effective *scalar* permittivity

$$\epsilon_{\text{eff}}^{\pm} = \frac{\det \underline{\epsilon}^{\pm}}{\hat{\mathrm{n}}^T \underline{\epsilon}^{\pm} \hat{\mathrm{n}}}, \tag{8.8}$$

then the speed of light in this medium will be identical to Eq. (8.3). Hence, replacing $\epsilon^{\pm}$ by $\epsilon_{\text{eff}}^{\pm}$ in the definitions of the impedances and admittances, Eqs. (3.9) and (3.10), accounts for all modifications of the numerical flux due to anisotropic materials. The semi-discrete form (8.2), however, still contains the full inverse permittivity tensor $\underline{\eta}$.

Besides this rigorous derivation of the upwind flux, a physical interpretation is readily available. Consider the numerical flux through an edge of a triangle as shown in Fig. 8.1. The vector normal to this edge is given by

$$\hat{\mathrm{n}} = \Big(\cos(\phi),\, \sin(\phi) \Big)^T \tag{8.9}$$

The upwind flux describes the propagation of fields across an element boundary, i.e., along the direction of $\hat{\mathrm{n}}$. Since for a plane wave we know that $\vec{\mathrm{E}} \cdot \vec{\mathrm{k}} = 0$ (cf. section 2.2.1), we identify the electric field component parallel to the edge, $\vec{\mathrm{E}}_{\parallel}$, as the relevant component for the propagation. After a rotation of the system around the z-axis by the angle $-\phi$, the normal vector is parallel to the x-axis. For the inverse permittivity in the new system we obtain

$$\underline{\eta}' = \mathcal{R}(-\phi)\, \underline{\epsilon}^{-1}\, \mathcal{R}^T(-\phi) \tag{8.10}$$

$$= \frac{1}{\det(\underline{\epsilon})} \cdot \mathcal{R}(-\phi) \begin{pmatrix} \epsilon_{yy} & -\epsilon_{xy} \\ -\epsilon_{yx} & \epsilon_{xx} \end{pmatrix} \mathcal{R}^T(-\phi). \tag{8.11}$$

[1] In the isotropic case, $c^{\pm} = (\epsilon^{\pm}\mu^{\pm})^{-1/2}$, and thus $Z^{\pm} = \mu^{\pm} c^{\pm} \equiv (\mu^{\pm}/\epsilon^{\pm})^{1/2}$.

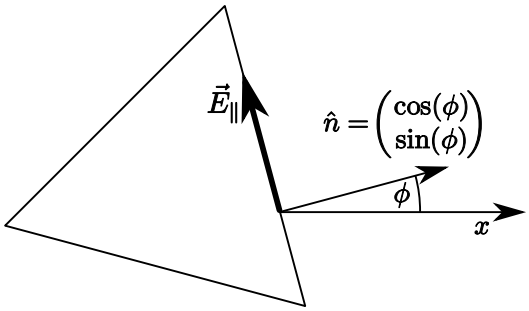


Figure 8.1: *Physical interpretation of the effective permittivity. We consider the electric field component parallel to an edge of a triangle. Rotating the system in a way that the normal vector coincides with the x-axis allows us to connect the projection of $\underline{\epsilon}^{-1}$ on $\vec{\mathrm{E}}_\parallel$ with the effective permittivity we need for the numerical flux.*

Here, $\mathcal{R}(-\phi)$ denotes the two-dimensional rotation matrix for the angle $-\phi$. After evaluating the matrix products, we find the identity

$$\epsilon_{\mathrm{eff}} \equiv \frac{1}{\underline{\eta}'_{22}} \quad \text{or, differently stated,} \quad \epsilon_{\mathrm{eff}}^{-1} \equiv \underline{\eta}'_{22}. \tag{8.12}$$

Thus, the inverse effective permittivity is essentially the projection of the inverse permittivity tensor on the direction of the electric field component parallel to the edge of an element.

Please note that the rigorous derivation can be applied to three-dimensional systems as well. However, one has to account for two distinct polarisations normal to a propagation direction $\hat{\mathrm{n}}$, which significantly complicates matters. In particular, the physical interpretation cannot hold in its current form as polarisation effects are not considered so far. Nevertheless, ongoing investigations hint that these can be accounted for by minor modifications.

8.3 Validation

To validate our method for anisotropic materials, we investigate the square cavity $(x,\, y) \in [0,\, L]^2$ as shown in the left-hand panel of Fig. 8.2. The cavity is filled with an anisotropic material characterised by

$$\underline{\epsilon} = \begin{pmatrix} \epsilon_{xx} & 0 \\ 0 & \epsilon_{yy} \end{pmatrix}. \tag{8.13}$$

The boundary of the cavity is a perfect electric conductor, i.e., the electric field component tangential to the boundary vanishes (see section 3.1.5). For any given time t, the field distribution within the

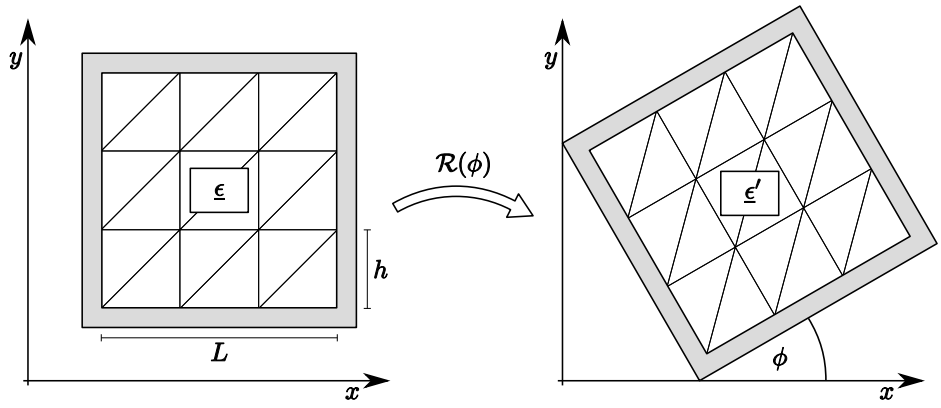


Figure 8.2: *The setup used for the validation of our implementation of anisotropic materials. The left-hand side shows a square cavity of side length L filled with a material with permittivity $\underline{\epsilon}$ which is bounded by a perfect electric conductor. The cavity is tessellated into a number of congruent triangles characterised by a side length h. This system translates into the own shown at the right-hand side by rotating it by an angle ϕ. Thereby, the permittivity tensor changes and acquires off-diagonal components. In turn, this provides us with a robust test case for a general anisotropy.*

cavity for the $(1,\,1)$-mode is analytically given by

$$\begin{aligned} E_x^{\mathrm{ref}}(\vec{\mathrm{r}},\,t) &= \frac{-\pi}{\omega L \epsilon_{xx}} \cdot \cos\left(\frac{\pi}{L}x\right) \cdot \sin\left(\frac{\pi}{L}y\right) \cdot \sin(\omega t), \\ E_y^{\mathrm{ref}}(\vec{\mathrm{r}},\,t) &= \frac{\pi}{\omega L \epsilon_{yy}} \cdot \sin\left(\frac{\pi}{L}x\right) \cdot \cos\left(\frac{\pi}{L}y\right) \cdot \sin(\omega t), \\ H_z^{\mathrm{ref}}(\vec{\mathrm{r}},\,t) &= \qquad \cos\left(\frac{\pi}{L}x\right) \cdot \cos\left(\frac{\pi}{L}y\right) \cdot \cos(\omega t), \end{aligned} \tag{8.14}$$

where

$$\omega = \frac{\pi}{L} \cdot \sqrt{\frac{1}{\epsilon_{xx}} + \frac{1}{\epsilon_{yy}}}. \tag{8.15}$$

By comparing results of a numerical simulation with this analytical reference we can establish the correctness of our algorithm.

However, with this simple test we can only validate our method for diagonal permittivity tensors. For a more general validation, we rotate the entire system by an angle ϕ. In the rotated system, the cavity is filled with a material

$$\underline{\epsilon}' = \mathcal{R}(\phi)\,\underline{\epsilon}\,\mathcal{R}(\phi)^T. \tag{8.16}$$

Again, $\mathcal{R}(\phi)$ denotes the rotation matrix. Thus, in the x-y coordinate system the rotated cavity is filled with an anisotropic material with (in general) non-vanishing, identical off-diagonal elements. The analytical solution of this system is given by a mere rotation of (8.14).

We initialise the fields in our numerical simulation with the respective analytical values for the $(1,\,1)$-mode at $t = 0$. Using the DGTD method (chapter 4), we evolve the fields in time for various

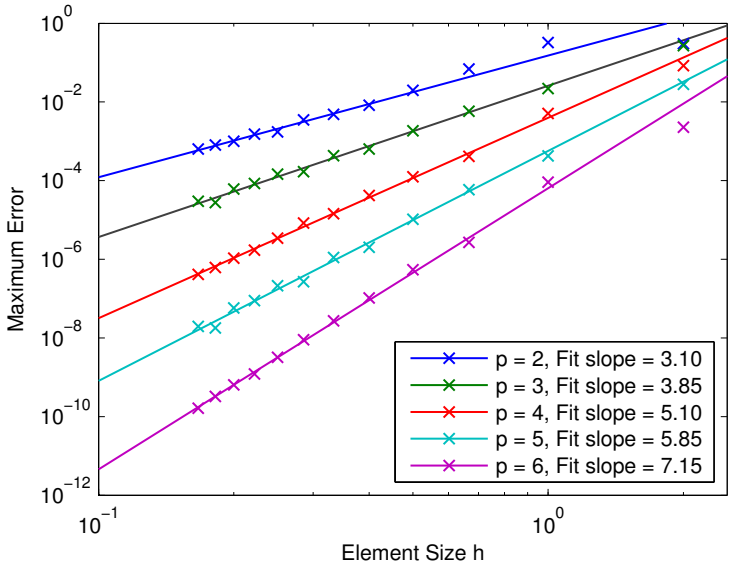


Figure 8.3: *Numerical errors for the H_z component of the $(1,\,1)$-mode in a cavity rotated by $30°$. As material parameters we use $\epsilon_{xx} = 4.0$ and $\epsilon_{yy} = 2.0$ and transfer the permittivity tensor into the rotated system. The crosses indicate the deviations from the numerical solution to the analytical one for different element sizes h (as defined in Fig. 8.2). The colours represent different orders p of Lagrange polynomials. The straight lines are the results of linear fitting procedures which favour smaller values of h.*

polynomial orders p (section 3.1.6) and different element sizes h (Fig. 8.3). For the time stepping, we use a fourth-order low-storage Runge-Kutta solver (cf. chapter 4). Due to the explicit nature of this solver, the time step is bounded by $\Delta t_{\max}$ to maintain numerical stability. To minimise the error of the time integration, we choose a small time step $\Delta t = 0.05 \cdot \Delta t_{\max}$ and simulate about 14 optical cycles. At each time step, we calculate the error as the maximum deviation of the numerically obtained magnetic field and the analytical reference within the cavity. The total error of the simulation is then defined by

$$\text{Error} = \max_i \Big(\max_{\vec{\mathrm{r}}} \Big| H_z^{\text{num}}(\vec{\mathrm{r}},\, i\Delta t) - H_z^{\text{ref}}(\vec{\mathrm{r}},\, i\Delta t) \Big| \Big). \tag{8.17}$$

Figure 8.3 shows the error in dependence of h and p. We observe that the exponential convergence of the isotropic case (compare Eq. (3.71)),

$$\text{Error} \propto h^{p+1}, \tag{8.18}$$

is nicely reproduced even for anisotropic materials. Thus, we conclude that our algorithm works for experimentally relevant symmetric permittivity tensors. Nevertheless, as in the course of the derivation we only assume that $\underline{\epsilon}$ is invertible, the extension should hold for even more general permittivities.

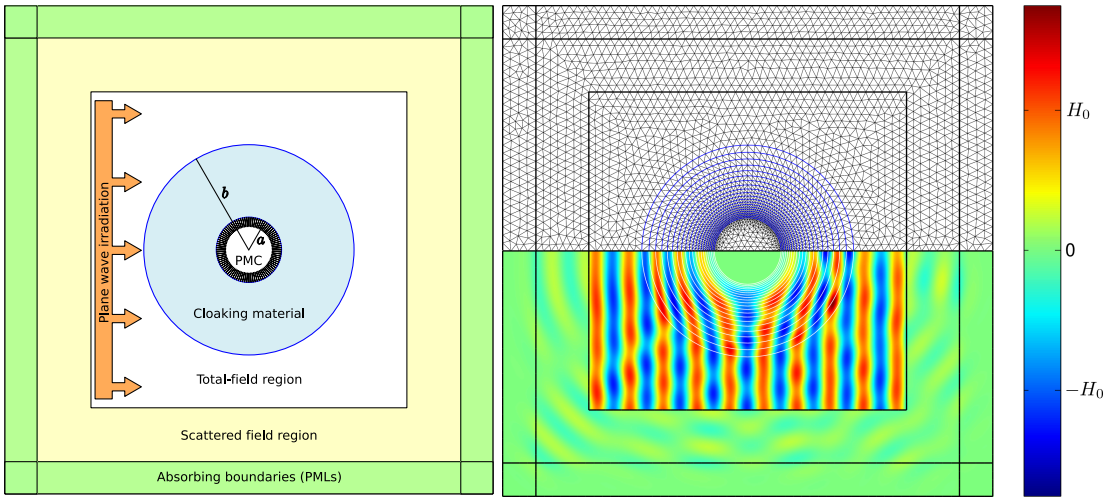


Figure 8.4: *Left panel: Sketch of the simulation setup. Using the total-field/scattered-field technique (section* 6.1*), a plane wave is launched onto a cylindrical cloak which surrounds a perfect magnetic conductor. Perfectly matched layers absorb scattered light (chapter* 7*). Right panel: The upper part shows the triangular mesh used for the simulation. The cloak consists of 20 rings of decreasing width. The lower part shows the magnetic field for a plane wave with* $\lambda = a = 632.8\,\mathrm{nm}$*. The colours indicate the amplitude of the magnetic field, where* H_0 *is the amplitude of the incident plane wave. A good fraction of the wave is reflected into the scattered-field region.*

8.4 Optical Cloaking

As a stress test for our method we want to apply it to optical cloaking [7, 106]. The idea is to coat an obstacle with a specially tailored material layer which guides light around it. Ideally, the shape of the incident electromagnetic wave is recovered behind the scatterer. As an observer cannot distinguish between the recovered wave and the original one, the object is invisible.

In order to achieve optical cloaking, the material layer usually has uncommon properties. In particular, the permittivity of the material continually changes with position. Furthermore, the materials generally show anisotropic behaviour. However, such materials do not exist in nature. Therefore, one relies on artificially created metamaterials whose substructures cannot be resolved by incident radiation and, hence, act as effective media. A variation of the substructure allows for a change in the material parameters, rendering intriguing properties possible. While this problem is experimentally very challenging, it also imposes great problems on numerical simulations.

Let us consider a cylindrical shell of inner radius a and outer radius b as shown in Fig. 8.4. The shell consists of a material with [7]

$$\mu_z = 1,\ \epsilon_\phi = \left(\frac{b}{b-a}\right)^2,\ \text{and}\ \epsilon_r = \left(\frac{b}{b-a}\right)^2\left(\frac{r-a}{r}\right)^2. \tag{8.19}$$

We note that the permittivity is given in a cylindrical coordinate system whose origin coincides with the shell's centre with azimuthal and radial components ϵ_ϕ and ϵ_r, respectively. Most strikingly, ϵ_r

vanishes as the distance r to the centre of the shell approaches a. As a consequence, the phase speed diverges to infinity.

In our simulations, we model the cylindrical cloak by a series of twenty rings. Each ring is approximated by homogeneously filled triangles. Specifically, the material properties of a triangle are determined by the position of its centre according to Eq. (8.19). Though this does not represent ideal, continuously varying parameters, we argue that this is a good approximation to the experimental situation, where one would change the material by varying the material composition of finite-sized unit cells. The outermost ring is five times as thick as the innermost ring to account for the drastic changes near the inner boundary. As an obstacle, we fill the inside of the shell with a perfect magnetic conductor (PMC, see section 3.1.5). We illuminate our system by a transverse-electric (section 2.1.5) polarised plane wave of wavelength $\lambda = 632.8\,\mathrm{nm}$ via the total-field/scattered-field technique (section 6.1). UPMLs (section 7.1) absorb light which is scattered by the cloak/PMC system. For the geometrical parameters we choose $a = \lambda$ and $b = a/0.314$ [7].

To speed up the simulations, we exploit the mirror symmetry of the system. Our mesh is depicted in Fig. 8.4. Third-order Lagrange polynomials are used as local basis functions. To maintain numerical stability, the time step is reduced by the maximum phase speed in the system according to the CFL criterion (cf. Eq. (4.8) or Ref. [21]). For our discretisation, this is a factor of about 50.

Figure 8.4 also features a pseudo-colour plot of the magnetic field. The total-field region shows what an observer would see. The scattered-field region reveals all deviations of the resulting field to an ideal plane wave due to scattering off the optical cloak and the enclosed PMC. In the case of perfect cloaking, an undisturbed plane wave would be expected in the total-field region, while no scattered light should be present in the scattered-field region. Though deviations are obvious, the overall performance of the cloak seems acceptable as the plane wave shape is approximately recovered once the wave has passed the scatterer. To further quantify the quality of the cloak, we calculate its scattering cross section per unit length by integrating the energy flux on a closed contour in the scattered-field region (see Eq. (2.77)) and compare it to that of the PMC alone, i.e., without the cloak. We obtain

$$C_{\mathrm{scat}}^{\mathrm{PMC}} \approx 2890\,\mathrm{nm}, \quad \text{and} \quad C_{\mathrm{scat}}^{\mathrm{cloak}} \approx 464\,\mathrm{nm}. \tag{8.20}$$

Thus, the cloak reduces the amount of scattered light by a factor of approximately 6.2.

The performance of the cloak is fundamentally limited by the chosen material parameters (8.19), which are impedance-mismatched at the outer cloak boundary [7]. Hence, some light is scattered away before the cloak can actually start to guide it around the obstacle. Secondly, the performance of the cloak is further limited by the size of the elements used to generate the desired material parameters. As the permittivity abruptly changes from element to element, inter-element scattering inevitably occurs and impedes the cloaking effect. Finally, the diverging material parameters at the inner boundary cannot be achieved in practice. To assess the dominant contribution to the scattering cross section, we repeat our simulations for 10, 15, and 25 rings. We find that the scattering cross section is nearly independent of the number of rings with the relative difference between the maximum and the minimum cross sections being below 0.3%. We conclude that the impedance mismatch at the outer boundary is the main factor which impairs the performance of the cylindrical cloak. The overall cloaking effect is still visible for surprisingly few rings.

As a last note, we want to mention that time-domain simulations might not be ideal for such kinds of simulations. Tests with radially independent material parameters indicate that it is crucial

to properly resolve the material parameters at the inner boundary. As we increase the resolution near the inner cylinder, the elements' centres approach the boundary as well. Thus, at the same time that we decrease the mesh size h, we also increase the speed of light c according to Eq. (8.19). Since the critical time step $\Delta t_{\max}$ is limited by the smallest element size h and the (inverse) maximum speed of light c, both effects decrease the time step. In turn, this drastically increases the computational time needed for the simulation. For such situations, frequency-domain calculations are more appropriate (see chapter 5). Nevertheless, the cloaking system serves as an excellent stress test for our DGTD implementation of anisotropic materials.

8.5 Summary

We have discussed an extension to the DG method for Maxwell's equations, which allows the inclusion of anisotropic materials in two-dimensional TE polarisation. We have shown that the semi-discrete form undergoes simple modifications. Changes to the numerical flux can be summarised by an effective scalar permittivity, whose numerical value depends on the permittivity tensor and the orientation of the element boundary. Our extension conserves the exponential convergence property of the original method. We have applied our method to an optical cloaking device and have observed pronounced scattering for realistic setups. For the future, a three-dimensional formulation and the application to other physical problems will be interesting.

Chapter 9

9 Numerical Study of Split-Ring Resonators

In the preceding chapters we have introduced a number of techniques and algorithms, the combination of which yields a very flexible and versatile numerical simulation tool in both time- and frequency-domain. In this chapter, we will demonstrate the computational power of DGTD using the example of split-ring resonators (SRRs) [109–114]. An introduction to the topic is given in section 9.1 along with the definition of relevant parameters. Section 9.2 is dedicated to the investigation of individual SRR resonances and their dependence on geometrical parameters. Coupling to other SRRs is included in section 9.3, where we investigate periodic arrays of resonators. In an attempt to understand the formation of array modes, we study the mutual coupling of SRR dimers in section 9.4. Finally, we calculate spectra of finite-sized arrays in section 9.5 and summarise the results of this chapter in section 9.6.

9.1 Split-Ring Resonator Basics

The term "split-ring resonator" in its common use refers to a type of building block of metamaterials, originally proposed by Pendry *et al.* [109]. In essence, an SRR is a metallic structure (often with nanoscopic dimensions) which resembles a circular or rectangular ring with a slot in it. When interacting with incident radiation of appropriate polarisation and frequency, resonances in the collective movement of the metal electrons appear. These resonances lead to an effective magnetic response which may even lead to negative permeabilities μ. In this section we establish the common basis for all subsequent studies and investigations. In particular, we provide parameter definitions and default values for a standard SRR and discuss the physics of the SRR's fundamental mode.

9.1.1 Geometry and Material Parameters

In the remainder of this chapter we will restrict ourselves to a specific shape of SRRs which is depicted in Fig. 9.1. Our SRR can be considered as a rectangular block with a rectangular gap in one of its sides. Alternatively, we can describe it in terms of a rectangular base with two orthogonally attached arms. In either view, this type of SRR is characterised by five independent parameters:

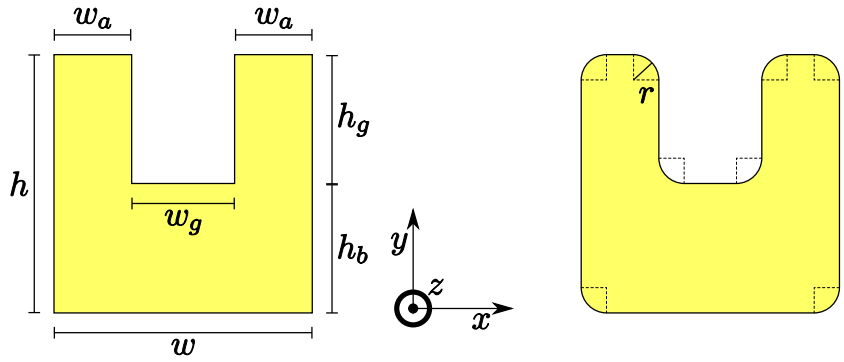


Figure 9.1: *Geometry of SRRs. Each SRR has a width w, a height h, a gap width w_g, and a gap height h_g. Alternatively, the last two parameters may be expressed in terms of the arm width $w_a = \frac{1}{2} \cdot (w - w_g)$ and the base height $h_b = h - h_g$. The right-hand side shows our model for smooth corners with radius r. The SRR's thickness, i.e., its extent in z-direction, is denoted by t (not shown in this sketch).*

- The SRR has a width w of $200\,\mathrm{nm}$. This corresponds to the length of the base.
- The total height of the SRR is $h = 200\,\mathrm{nm}$. This corresponds to the height of the arms.
- The gap has a width w_g of $80\,\mathrm{nm}$. From this, it follows that each arm has a width $w_a = \frac{1}{2} \cdot (w - w_g) = 60\,\mathrm{nm}$.
- The gap height h_g amounts to $100\,\mathrm{nm}$. Consequently, the base height $h_b = h - h_g = 100\,\mathrm{nm}$ as well.
- The SRR's thickness t is $30\,\mathrm{nm}$.

Please note that the resulting SRR is always mirror symmetric with respect to a plane which both includes the SRR's centre and is parallel to the $x - y$ plane (see Fig. 9.1 for the coordinate system). The specific parameter values indicate the *standard* SRR which is used as a reference in all following computations.

In addition, we have to specify the material properties of the SRR. In our study, we assume the investigated SRRs to be made of gold, which is modelled by a single-pole Drude model. For the material parameters we employ those specified in Eq. (2.69) [45]. A typical realisation of an array of gold SRRs using modern electron beam lithography is depicted in Fig. 1.1. From this, it is apparent that even with sophisticated fabrication methods, experimental realisations usually do not feature sharp corners. Hence, we introduce a sixth material parameter:

- The SRR's corners may be replaced by quadrants[1] of radius r to account for the fabrication process (Fig. 9.1). In most cases we use $r = 0\,\mathrm{nm}$, i.e., sharp corners.

Furthermore, SRRs rarely float in space. In most cases, SRRs are fabricated on a planar substrate of refractive index n. Typically, we use $n = 1.5$ which corresponds to glass. Since its thickness is usually in the order of $100\,\mu\mathrm{m}$ and above, it is much thicker than the resonators it supports. Thus, it is included in our simulations as a dielectric half space.

[1] A quarter of a circle

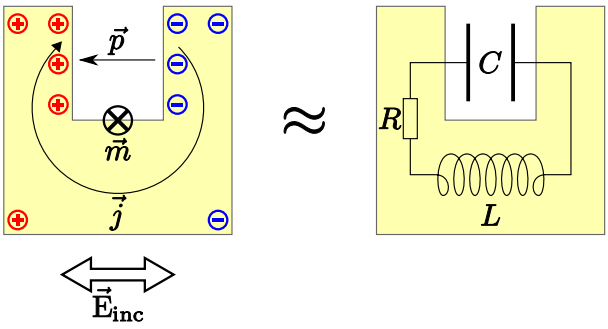


Figure 9.2: *Simple models for the physics of the fundamental resonance of a SRR for horizontal polarisation. The left-hand panel shows a charge current density* $\vec{\mathrm{j}}$ *induced by an externally applied electric field* $\vec{\mathrm{E}}_{\mathrm{inc}}$*. It leads to an accumulation of positive and negative electric charges, in particular at both sides of the gap. The charge distribution in this sketch is motivated by the electric field plot in Fig.* 9.3*. Electric and magnetic dipole moments* $\vec{\mathrm{p}}$ *and* $\vec{\mathrm{m}}$ *are attributed to these charges and currents, respectively. The right-hand panel shows a more abstract notion of the resonator in terms of an RLC circuit. The arms form a capacitor of capacitance* C*, while the entire ring is an inductor of inductance* L*. A resistor with resistance* R *accounts for Ohmic and radiative losses.*

9.1.2 Physical Principles

We are interested in the interaction of SRRs with incident light of frequency ω. In all subsequent studies, we will only consider plane waves at normal incidence with respect to the substrate plane. Furthermore, we will restrict our analysis of SRRs to the important special case of horizontal polarisation, i.e., the incident electric field $\vec{\mathrm{E}}_{\mathrm{inc}} = (E_{\mathrm{inc}},\, 0,\, 0)^T$ is parallel to the base (see Fig. 9.2). In addition, we only describe the physics of the fundamental resonance, i.e. the resonance with the lowest frequency.

The incident electric field interacts with quasi-free gold electrons in the SRR: If the electric field at some time points towards the negative x-axis, i.e., towards the left arm, it exerts a force on the electrons which shuffles them towards the right arm (Fig. 9.2). This leads to an electric charge current $\vec{\mathrm{j}}$ from one arm to the other one. The positively charged ions, in contrast, are too heavy to be moved. The relative excess of ions in the left arm leads to a net positive charge distribution, while the right arm gains a net negative charge. Hence, the incident light wave leads to a separation of charges and in turn to a restoring force which opposes the incident electric field. Eventually, this would lead to a steady state if the incident electric field were static. Since the incident electric field oscillates with frequency ω, however, the charges will follow the external field and move from the right arm to the left one and back again with the same frequency. This collective oscillation is called a plasmon-polariton.

To obtain a physical understanding it is not always necessary to consider the full charge and current distribution in the SRR [112]. Instead, qualitative statements can be obtained by effective electric and magnetic dipole moments $\vec{\mathrm{p}}$ and $\vec{\mathrm{m}}$, respectively (Fig. 9.2). Figure 9.3 illustrates electric and magnetic field distributions for an individual SRR (section 9.2) and motivates to attribute the electric dipole due to the separated charges to the centre of the gap. The magnetic dipole due to the

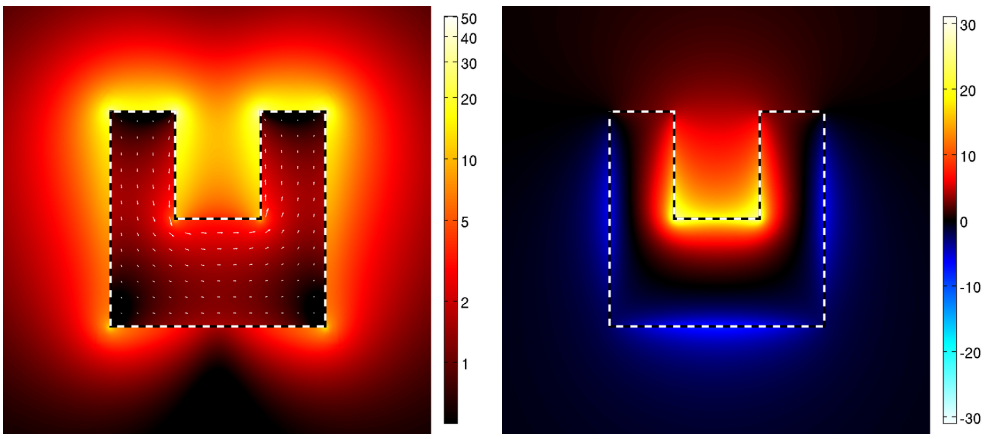


Figure 9.3: *Field distributions at resonance wavelength* $\lambda_{\mathrm{res}} \approx 1465\,\mathrm{nm}$ *for an individual SRR with reference parameters (see section* 9.1.1*). The left-hand side shows the electric field enhancement* $|\vec{\mathrm{E}}(\omega)|/|\vec{\mathrm{E}}_{\mathrm{inc}}(\omega)|$ *as a pseudo-colour plot on a logarithmic colour scale. Within the resonator, arrows indicate direction and relative amplitude of the in-plane components of the Drude model's polarisation current* $\Re\{\vec{\mathrm{j}}_D(\omega)\}$*. The right-hand side shows the normalised out-of-plane component of the magnetic field, i.e.,* $\Re\{H_z(\omega)\}/|\vec{\mathrm{H}}_{\mathrm{inc}}(\omega)|$*, in a pseudo-colour plot. This time, however, the colour scale is linear. For a more detailed explanation see section* 9.2.3*. Please note that the colour bars in this figure are reused for all other mode plots in this chapter.*

charge current is attributed to the centre of the SRR. This effective *induced* dipole model is useful to investigate the coupling of SRRs.

An even simpler and more intuitive description of SRRs is given by an RLC resonator circuit [110]. The aggregation of charges on both sides of the gap is identified with a capacitor of capacitance C. Similarly, the split-ring is considered an incomplete loop of a coil, and thus is assigned a self inductance L. This situation is illustrated in the right-hand panel of Fig. 9.2. To account for Ohmic and radiative losses, we also include a resistor with resistance R into the circuit. When driven by an external AC voltage of frequency ω, its resonance frequency is given by

$$\omega_{\mathrm{res}} = \frac{1}{\sqrt{LC}}. \tag{9.1}$$

As a final model, SRRs can also be considered as antennas whose ends are bent upwards to form the arms of the split-ring (Fig. 9.4). Thus, we can assign an equivalent antenna length l to the SRR, which depends on the geometrical parameters. If we assume the length of the antenna to be determined by a rectangular wire which runs through the centres of the arms and base, we can calculate l by

$$l = (w - w_a) + (2h - h_b) = \frac{1}{2}(w + w_g) + (h + h_g). \tag{9.2}$$

Thus, the length of the antenna is mostly determined by the height h and the gap height h_g. The fundamental resonance of an antenna corresponds to the frequency for which the wavelength in the antenna material is $\frac{l}{2}$ (Fig. 9.4).

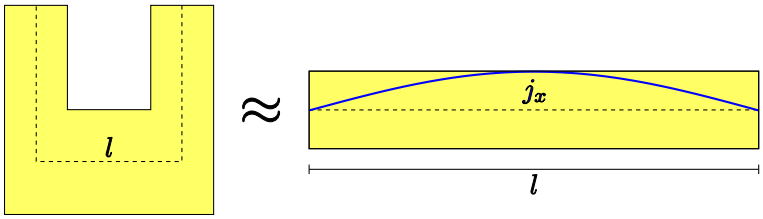


Figure 9.4: *Analogy between SRRs and antennas. Bending the arms of the SRR such that they are parallel to the base approximately yields an antenna of length l, depicted at the right-hand side. The current of a fundamental antenna mode is illustrated by the blue curve.*

9.2 Individual Split-Ring Resonators

We begin our numerical investigation of SRRs by studying individual gold resonators on a glass substrate. The term "individual" means that we do not consider additional objects besides the substrate and the SRR. Section 9.2.1 introduces the basic computational setup used for all computations in this chapter, with the exception of those performed in section 9.3. We subsequently discuss scattering, absorption, and extinction spectra for the standard SRR presented in section 9.1.1. Finally, we investigate the influence of numerous parameters on the resonances of SRRs in section 9.2.4.

9.2.1 Simulation Setup

The simulation setup used in the following is presented in Fig. 9.5. The brick-shaped computational domain consists of an SRR on top of a half space with permittivity $\epsilon = n^2$. The remainder of the computational domain is "filled" with vacuum, i.e., $\epsilon = 1.0$. A layer of uniaxial perfectly matched layers (UPMLs, see chapter 7) with a thickness of $100\,\mathrm{nm}$ surrounds the computational domain. The substrate/vacuum interface divides the PMLs into two halves.

The computational domain is furthermore divided into two regions via a total-field/scattered-field (TF/SF) interface (see section 6.1). The TF/SF interface fully encloses the SRR and also extends into the substrate. The same interface is used to inject a plane wave into the system. For the spatial profile of the incident wave we use Eq. (2.38), i.e., the solution to the half space problem presented in section 2.2.2. This solution acknowledges the presence of the substrate: If the SRR were absent, we would not observe any light outside the TF/SF interface, i.e., in the scattered-field region. Thus, scattered light must originate from the investigated SRR.[2]

The system is perfectly symmetric with respect to the SRR's plane of symmetry in terms of geometry and excitation. Since the incident electric field is normal to the plane of symmetry, any positive charge on one side of the plane is mirrored by a negative charge on the other side. Hence, we can remove one half of the computational domain if we replace the plane of symmetry by a perfect electric conductor (PEC, see section 3.1.5). This is the same concept used in the method of images often encountered in electrostatic boundary problems [43].

The computational domain reduced in such a manner is transformed into a mesh using `NETGEN`.

[2]This situation mirrors the experimental one, where one normalises the signal recorded for a scatterer to a reference signal obtained for the substrate only.

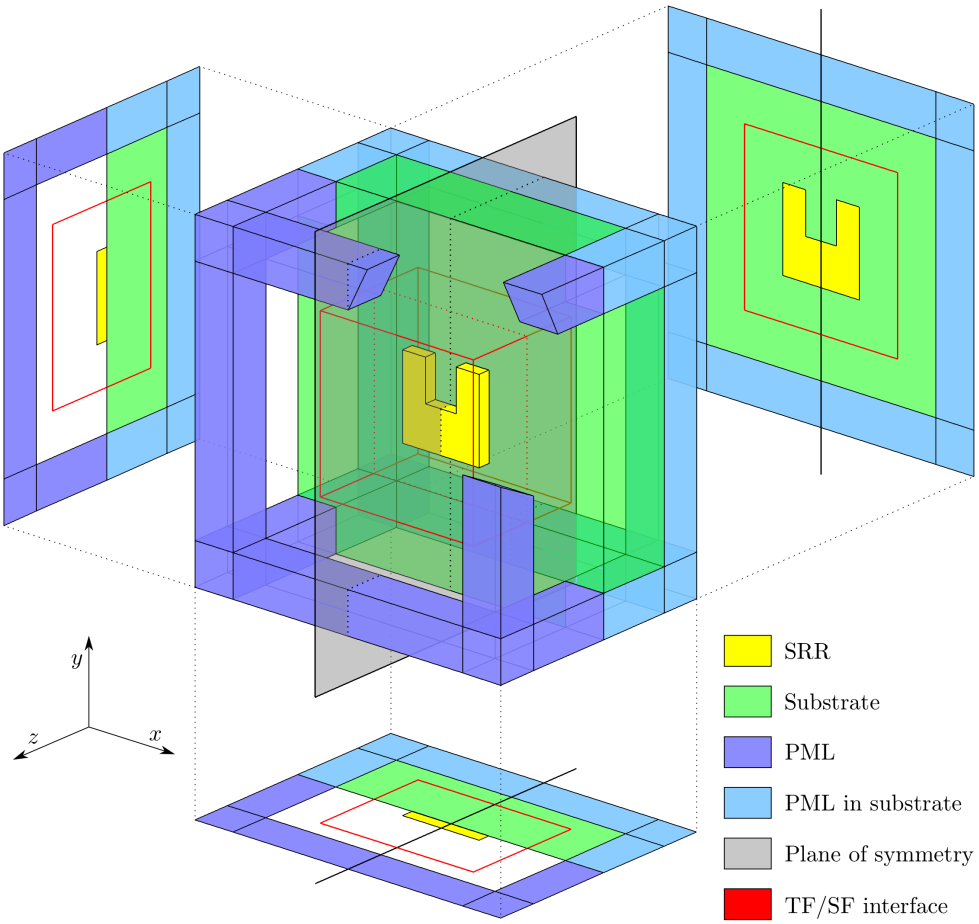


Figure 9.5: *Computational setup for the simulation of individual SRRs on a substrate. The sketch shows a three-dimensional visualisation of the system and corresponding planar projections. The SRR, a more detailed sketch of which can be found in Fig.* 9.1*, is depicted in yellow. The green region indicates the substrate. A layer of UPMLs (section* 7*) surrounds the entire computational domain and is represented by blue blocks. The lighter shade of blue indicates PMLs in the substrate region. For better visibility we have only included regions where at least two layers (terminating the x-, y-, and z-axes, respectively) overlap. Furthermore, we have removed parts of the blocks to allow for an unobstructed view on the SRR. Red lines indicate the TF/SF interface (section* 6.1*) used to inject a plane wave into the system. Finally, we have included the system's plane of symmetry in the image (grey shade in the three-dimensional plot, thick black lines in the projections). We apply PEC boundary conditions (section* 3.1.5*) on this plane. This allows us to simulate the system on just one side of the plane.*

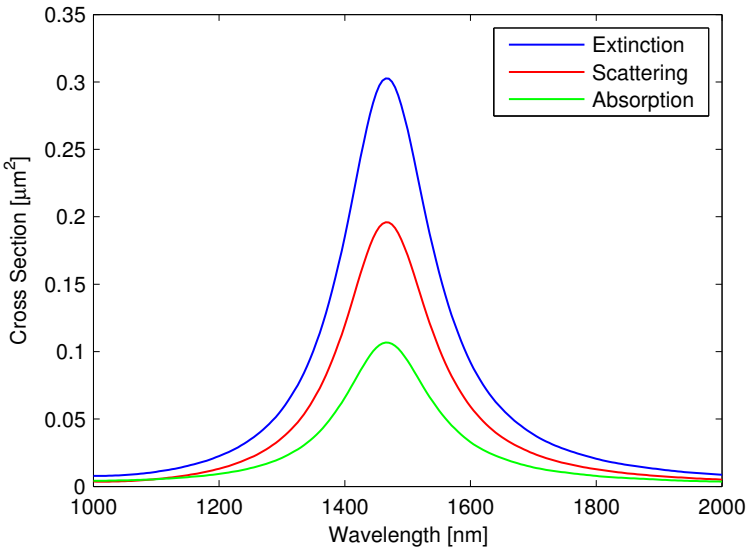


Figure 9.6: *Extinction, scattering, and absorption cross sections for an SRR with standard parameters specified in section* 9.1.1.

We perform DGTD simulations with a 14-stage, fourth-order accurate low-storage Runge-Kutta scheme (chapter 4) for Lagrange polynomials of order $p = 4$ (section 3.1.6). In preliminary studies, this spatial order was previously found to yield sufficiently converged results for all SRRs investigated in this chapter. The dispersion of gold is modelled by auxiliary differential equations as explained in section 6.2.

9.2.2 Scattering, Absorption, and Extinction

For individual particles we are mainly interested in scattering, absorption, and extinction cross sections. These quantities have been discussed in section 2.4. It is straightforward to extract them from simulations using the setup discussed in the previous section.

The TF/SF contour divides the computational domain into regions in which we compute total fields $\vec{\mathrm{E}}_{\mathrm{tot}}, \vec{\mathrm{H}}_{\mathrm{tot}}$ and scattered fields $\vec{\mathrm{E}}_{\mathrm{scat}}, \vec{\mathrm{H}}_{\mathrm{scat}}$, respectively. In particular, we find fields of both types on the TF/SF interface due to the discontinuous nature of our numerical method. With the scattered and total fields available in the time-domain, we employ an on-the-fly Fourier transform (section 6.4) to obtain frequency-domain fields. From the latter we calculate the total and scattered Poynting vectors which are integrated over the TF/SF interface. A subsequent normalisation to the incident Poynting vector, easily constructed using Eq. (6.9), yields the scattering and absorption cross sections C_{scat} and C_{abs} according to Eqs. (2.76) and (2.77). The extinction cross section C_{ext} follows immediately from Eq. (2.78) as the sum of C_{scat} and C_{abs}.

Figure 9.6 shows the resulting spectra for an SRR with standard parameters taken from section 9.1.1. We observe pronounced resonances around $1465\,\mathrm{nm}$ for all three kinds of cross sections. At resonance, the ratio between C_{scat} and C_{abs} is roughly $2:1$, i.e., roughly two thirds of the extinction

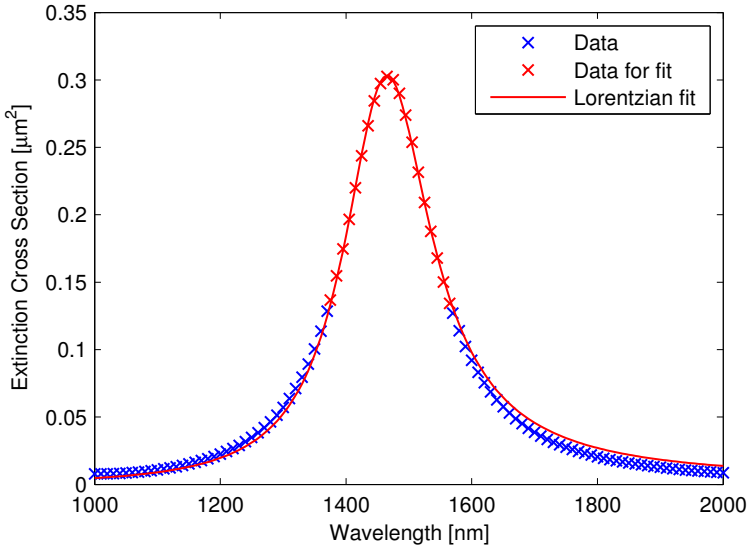


Figure 9.7: *Lorentzian fit for the extinction cross-section of an SRR with standard parameters specified in section* 9.1.1. *Blue crosses indicate data points taken from the simulation which are not used to fit the parameters of the Lorentzian. Red crosses represent data points used for the fit. For better visibility, only every other data point is plotted. The continuous curve represents the resulting Lorentzian from which we extract the resonance wavelength, peak amplitude, and quality factor. Please note that our fitting procedure favours values close to the resonance.*

is due to scattering. This is consistent with results obtained using FDTD (see appendix A) we have previously reported for a similar SRR [111].

Using an automated least-squares fit, we obtain a set of parameters which describe the resonance in terms of a Lorentzian (Fig. 9.7). From these parameters, we determine the resonance wavelength, peak amplitude, and quality factor of a resonance. The quality factor Q is defined as

$$Q = \frac{\omega_{\mathrm{res}}}{\Delta\omega}, \tag{9.3}$$

where ω_{res} is the resonance frequency and $\Delta\omega$ the full width at half maximum [87]. It is a measure of how much energy stored in a resonator is lost per cycle. The higher Q is, the less energy is lost and the longer a resonance can be sustained in the absence of driving fields. Potential loss channels include Ohmic losses in the metal and radiation of electromagnetic waves into the far-field, since the SRRs is essentially a bent antenna (see Fig. 9.4). For our standard resonator we extract $Q \approx 8.6$.

9.2.3 Field Distributions

Resonant field and current distributions of an individual SRR are depicted in Fig. 9.3. To record the data for such plots we run a second DGTD simulation after having identified the frequencies and

wavelengths of interest.[3] During this second simulation we perform on-the-fly Fourier transforms to obtain frequency-domain fields on a plane through the centre of the SRR which is parallel to the x-y plane.

In this fashion we acquire the electric field $\vec{\mathrm{E}}(\omega)$, the magnetic field $\vec{\mathrm{H}}(\omega)$, and the Drude model's polarisation current $\vec{\mathrm{j}}_D(\omega)$. Since the illumination spectrum deviates from a straight line, we have to normalise the fields to the incident field strengths. Thus, we obtain the field enhancement factor

$$\frac{|\vec{\mathrm{E}}(\omega)|}{|\vec{\mathrm{E}}_{\mathrm{inc}}(\omega)|}, \tag{9.4}$$

where $\vec{\mathrm{E}}_{\mathrm{inc}}$ is the amplitude of the incident electric field. The field enhancement factor is plotted in the left-hand panel of Fig. 9.3 in pseudo-colours. The same plot also features the polarisation current $\vec{\mathrm{j}}_D$ within the resonator. Since we are interested in the direction and relative strength of this current, the figure presents the real parts of $\vec{\mathrm{j}}_D(\omega)$. In contrast to absolute values of frequency-domain quantities, taking the real part is an operation sensitive to the complex phase of the field distribution. For the current we use the very same phase as for the normalised[4] out-of-plane (z-)component of the magnetic field as shown in pseudo-colours in the right-hand panel of Fig. 9.3. The latter's phase is determined in a simple procedure: First, we identify the point of maximum $|H_z(\omega)|$. For this point, we determine the angle by which the phase must be adjusted such that $\Re\{H_z(\omega)\} \equiv |H_z(\omega)|$. Both $\vec{\mathrm{j}}_D(\omega)$ and $H_z(\omega)$ are then modified to match this phase.

All mode plots other than Fig. 9.3 are generated in the same fashion, too. Furthermore, the colour scales in all these plots are the same to facilitate an easy comparison for different configurations.

9.2.4 Parameter Studies

Obviously, is is highly desirable to control the resonance of an SRR. Thus, we would like to address the question how the resonance frequency, the quality factor, and the maximum amplitudes of C_{scat}, C_{abs}, and C_{ext} depend on the geometrical parameters of the SRR. For each of the following calculations we vary one of the parameters outlined in section 9.1.1 while we keep the remaining parameters unchanged if possible. This allows for a systematic investigation of the influence of each parameter.

Width

We start by varying the width of the SRR in steps of $10\,\mathrm{nm}$ around $w = 200\,\mathrm{nm}$. Since the relation between the gap width w_g and the arm width w_a depends on w, we have to choose which of these parameters we want to fix. We decide to keep the arm width constant. Corresponding results are presented in Fig. 9.8. We observe an approximately linear dependence of the resonance frequency on w with larger values of w corresponding to larger resonance wavelengths. This is expected from the simple antenna picture summarised in Eq. (9.2), where larger widths correspond to longer antennas and, thus, higher resonance wavelengths. However, for small w the change seems less pronounced and indicates a saturation effect. The reason for this behaviour lies in the dispersion of gold [115, 116].

[3] Obviously, a frequency-domain simulation would be more convenient. Due to the large number of elements, however, a time-domain simulation is more memory-efficient.

[4] With respect to the incident magnetic field $|\vec{\mathrm{H}}_{\mathrm{inc}}(\omega)|$, obviously.

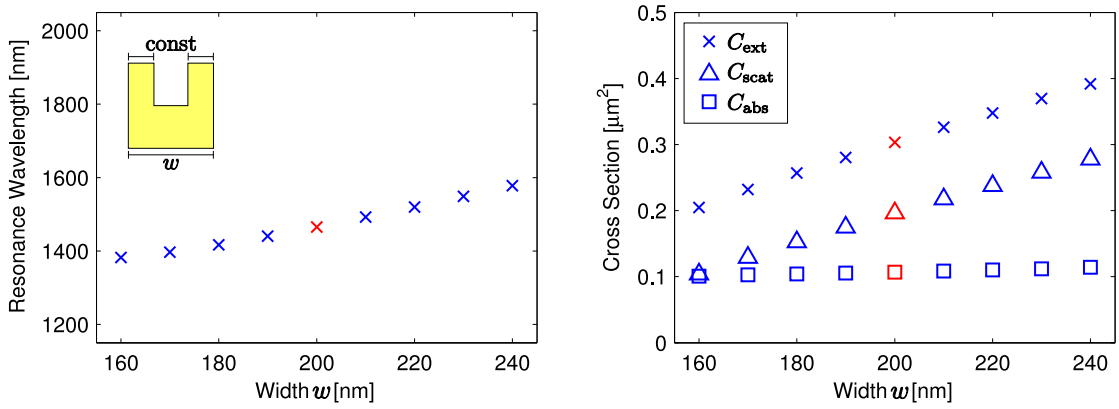


Figure 9.8: *Influence of the width w on a single SRR's resonance wavelength and cross sections. In these calculations, the arm width w_h was fixed to $60\,\mathrm{nm}$. Red data points indicate values which correspond to the standard SRR of section* 9.1.1. *Please note that all corresponding plots in this section show the same ranges on the respective axes of ordinate.*

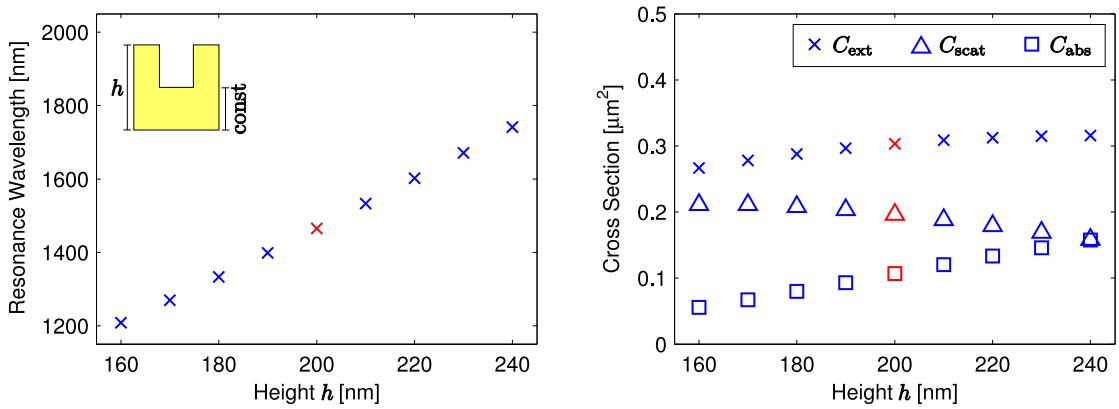


Figure 9.9: *Influence of the height h on a single SRR's resonance wavelength and cross sections. In these calculations, the base height was fixed to $h_b = 100\,\mathrm{nm}$. Red data points indicate values which correspond to the standard SRR of section* 9.1.1.

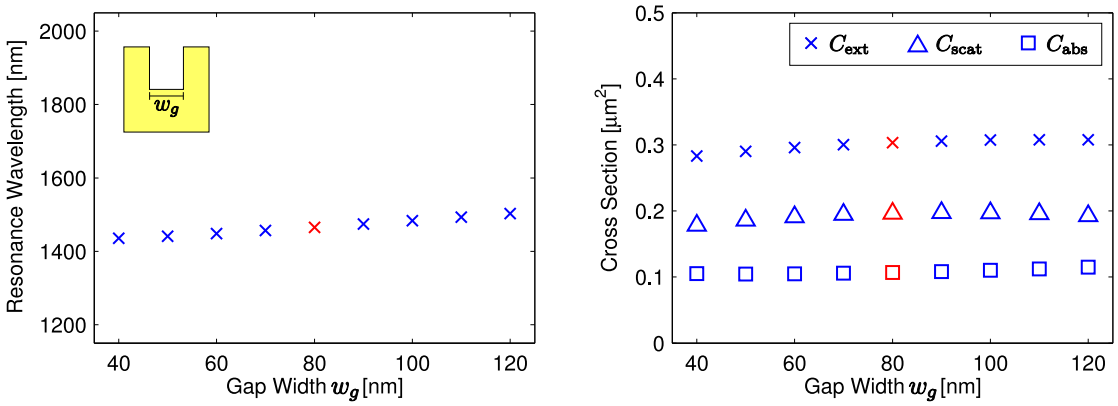


Figure 9.10: *Influence of the gap width w_g on a single SRR's resonance. Red data points indicate values which correspond to the standard SRR of section* 9.1.1.

Concerning the cross sections, we observe that the peak absorption cross section barely depends on the SRR's width. The scattering cross section, however, is strongly affected by this parameter and shows a linear growth in w. This is not only due to the increased *geometrical* cross section

$$C_{\text{geo}} = w \cdot h - w_g \cdot h_g \tag{9.5}$$

of the SRR (which also grows linearly with w), but also due to the increased scattering efficiency $C_{\text{scat}}/C_{\text{geo}}$ (not shown). Effectively, there is a closer resemblance between more elongated SRRs and strongly scattering antennas than there is for SRRs of smaller widths. The quality factor degrades from $Q \approx 11.2$ for $w = 160\,\text{nm}$ to $Q \approx 7.2$ for $w = 240\,\text{nm}$ (not shown).

Height

Similarly to the case of the width, we decide to keep the base height h_b constant when varying the total height h. The SRR's resonance wavelength shows an even more pronounced dependence on h than on w (Fig. 9.9), which is expected from Eq. (9.2). Increasing the height by $40\,\text{nm}$ shifts the resonance another $250\,\text{nm}$ towards higher wavelengths. At the same time, this leads to a significant increase of the absorption cross section. Even though the geometrical cross section C_{geo} grows for larger h, a decreasing scattering efficiency (not shown) leads to a moderate decline of the scattering cross section. Smaller scattering efficiencies are expected since the resonator departs from the ideal antenna shape for larger h. All in all, the extinction cross section slightly grows with h. The quality factor increases from $Q \approx 6.3$ for $h = 160\,\text{nm}$ to $Q \approx 10.2$ for $h = 240\,\text{nm}$ (not shown).

Gap Width

The gap width has only minor influence on the resonance of an SRR (Fig. 9.10). For larger w_g—and consequently smaller arm widths w_a—we observe slightly larger resonance wavelengths. This is consistent with Eq. (9.2), although one would expect shifts similar to the width parameter study.

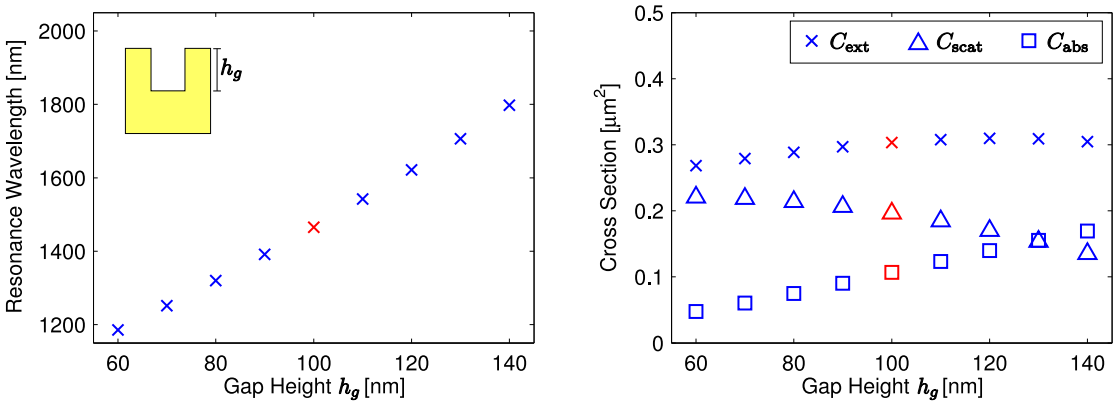


Figure 9.11: *Influence of the gap height h_g on a single SRR's resonance. Red data points indicate values which correspond to the standard SRR of section* 9.1.1.

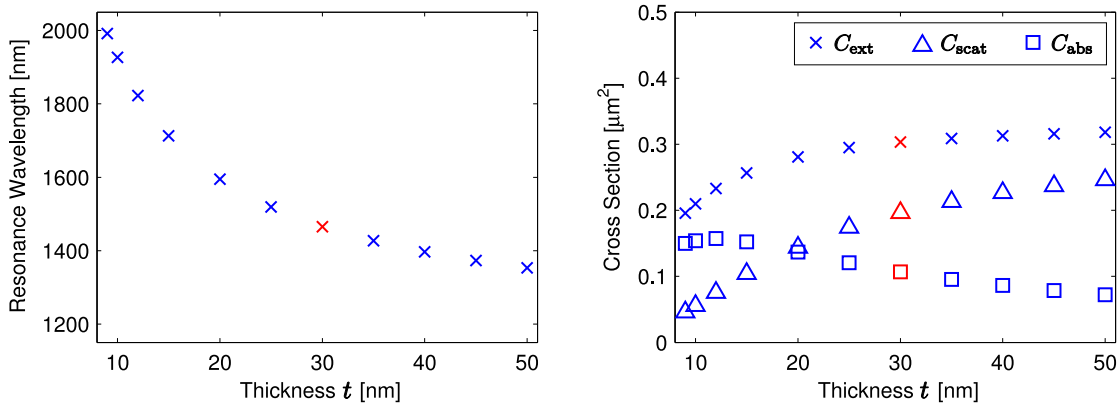


Figure 9.12: *Influence of the thickness t on a single SRR's resonance. Red data points indicate values which correspond to the standard SRR of section* 9.1.1.

This highlights that the antenna model cannot be used for quantitative predictions. A general trend for the cross sections is not apparent; they appear to be roughly constant. Similarly, the quality factor degrades only slightly from $Q \approx 9.0$ for $w_g = 40\,\mathrm{nm}$ to $Q \approx 8.6$ for $w_g \geq 80\,\mathrm{nm}$ (not shown).

Gap Height

The influence of the gap height h_g on the SRR's resonance is easily confused with the results obtained for the total height h. Again, we note that the resonance wavelength is very sensitive on h_g and can be tuned in the range of 1200 to $1800\,\mathrm{nm}$ for $h_g \in [60,\ 140]\,\mathrm{nm}$ (Fig. 9.11). This behaviour is well understood in the antenna model. The quality factor steadily increases from $Q \approx 5.6$ for $h_g = 60\,\mathrm{nm}$ to $Q \approx 10.4$ for $h_g = 140\,\mathrm{nm}$ (not shown). Simultaneously, the absorption cross section grows for larger h_g while C_{scat} decreases.

Please note that large gap heights correspond to small base heights h_b. For small h_b, the base is a bottleneck for the charge transport from one arm to another. Since the resistance of a conductor is inversely proportional to its geometrical cross section, smaller h_b generate higher Ohmic losses, and thus larger absorption cross sections.

Thickness

The thickness t is perfectly suited to tune the SRR's resonance frequency to the desired wavelength. Larger t reduce the resonance wavelength, while thin resonators have very high resonance wavelengths (Fig. 9.12). We see a strong increase of the scattering cross sections for thicker resonators, while the absorption cross section decreases for $t > 15\,\mathrm{nm}$. The quality factor increases for $t \leq 15\,\mathrm{nm}$ and decreases for $t > 15\,\mathrm{nm}$ (not shown). The minimum and maximum values in the considered parameter range are $Q \approx 7.3$ and $Q \approx 9.6$, respectively. Experimentally, the thickness is often controllable up to the level of single atomic layers using electron beam evaporation, and is thus more accessible than the other parameters.

Roundings

Real-world fabrication of nanostructures often tends to smooth corners. Depending on the quality of the fabrication process, the radius r of smoothed corners is in the order of 5 to $25\,\mathrm{nm}$. Including such roundings in our numerical setup, we find that larger radii decrease the resonance wavelength, and thus the effective antenna length l (Fig. 9.13). The cross sections hardly change, but the ratio between scattering and absorption cross sections increases for larger r. The quality factor simultaneously decreases from $Q \approx 8.6$ for $r = 0\,\mathrm{nm}$ to $Q \approx 7.9$ for $r = 25\,\mathrm{nm}$ (not shown).

Material Losses

Material losses are difficult to control in the experiment with the obvious exception of using a different material altogether. It should be noted, however, that Ohmic losses do play a role regarding the shape of SRR resonances. In particular, it is not clear whether the data measured by Johnson and Christy using thin films [45] is applicable for nanostructures. Some papers [110, 117] report

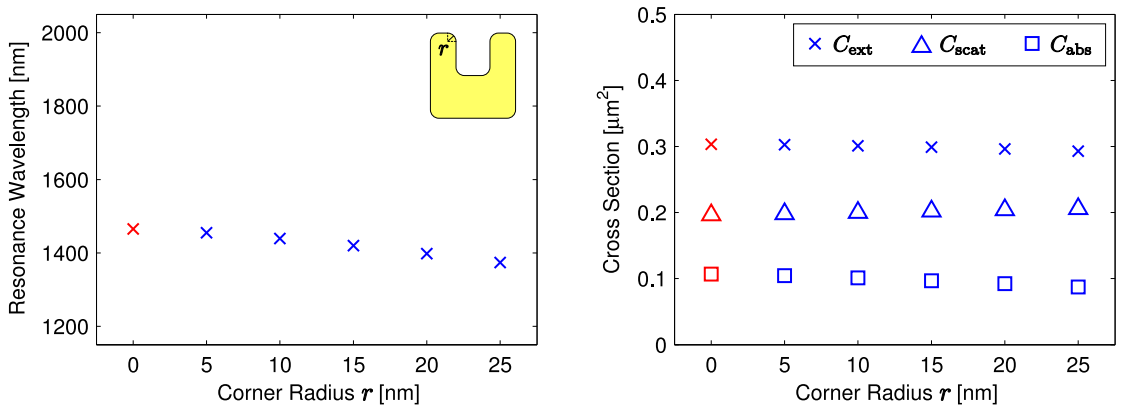


Figure 9.13: *Influence of the corner radius r on a single SRR's resonance. Data points which correspond to a rectangular SRR, i.e., $r = 0$, are plotted in red.*

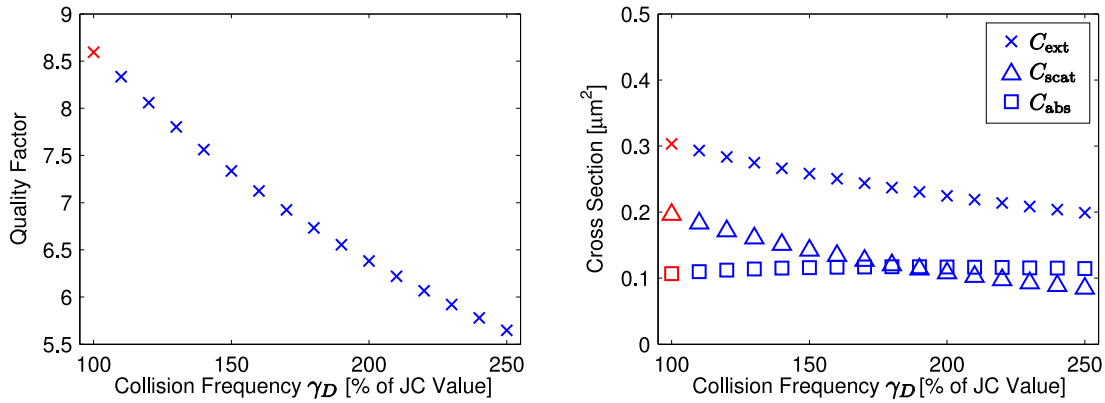


Figure 9.14: *Influence of the collision frequency γ_D on a single SRR's resonance. The Drude model's collision frequency γ_D is given relative to the value determined by Johnson and Christy, Eq.* (2.69)*. Data points for the standard collision frequency are plotted in red.*

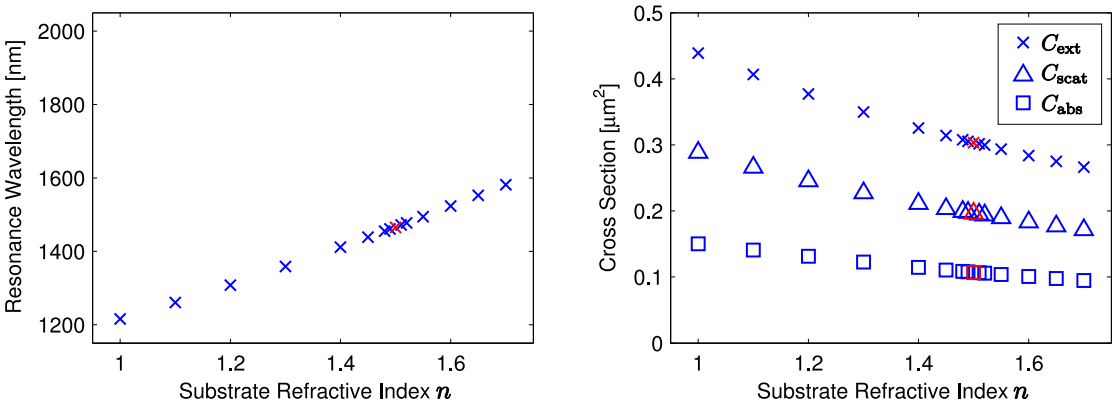


Figure 9.15: *Influence of the substrate's refractive index n on a single SRR's resonance. Data points for a glass substrate with $n = 1.5$ are plotted in red.*

enhanced agreement between simulation and experiment if the Drude model's collision frequency γ_D (section 2.3) is increased as compared to the original Johnson Christy parameters.

To this end, we have studied the influence of γ_D on the resonance of an SRR with standard parameters (Fig. 9.14). To allow for a simple comparison, we specify the collision frequency relative to the original value specified in Eq. (2.69). We find that the resonance frequency hardly changes at all (not shown) and direct our attention to the resonator's quality factor Q. For higher collision frequencies the resonance broadens, which corresponds to lower quality factors. At the same time, the peak scattering cross section decreases. The peak absorption cross section, however, slightly grows. The broadening of the resonance impedes even higher peak values of C_{abs}.

Substrate

Finally, we investigate the influence of the dielectric substrate on the SRR's resonance. Please recall that the wave injected via the TF/SF contour is the solution to the half space problem presented in section 2.2.2. Thus, any scattering stems from the resonator itself. Figure (9.15) shows that the resonance wavelength grows linearly with the substrate's refractive index n, at least for $n \in [1.0,\ 1.7]$. All cross sections decrease for larger n, while the ratio between scattering and absorption is roughly conserved. At the same time, the quality factor decreases from $Q \approx 9.9$ to $Q \approx 8.1$ (not shown).

We furthermore investigate the influence of a different *type* of substrate. For some applications experimentalists cannot use glass substrates as thick as $100\,\mu\mathrm{m}$. Extremely smooth surfaces and a uniform substrate thickness t_s are achieved with ultra-thin substrates of silicon nitride ($\mathrm{Si_3N_4}$, $n \approx 2.0$), which are often used for transmission electron microscopy (TEM). Thus, we investigate the influence of the thickness of such a TEM substrate on the resonance of an SRR with reference parameters.

To this end, we modify the standard scattering setup (section 9.2.1) and replace the substrate half space by a thin substrate film. To eliminate scattering by the substrate itself, we have to incorpo-

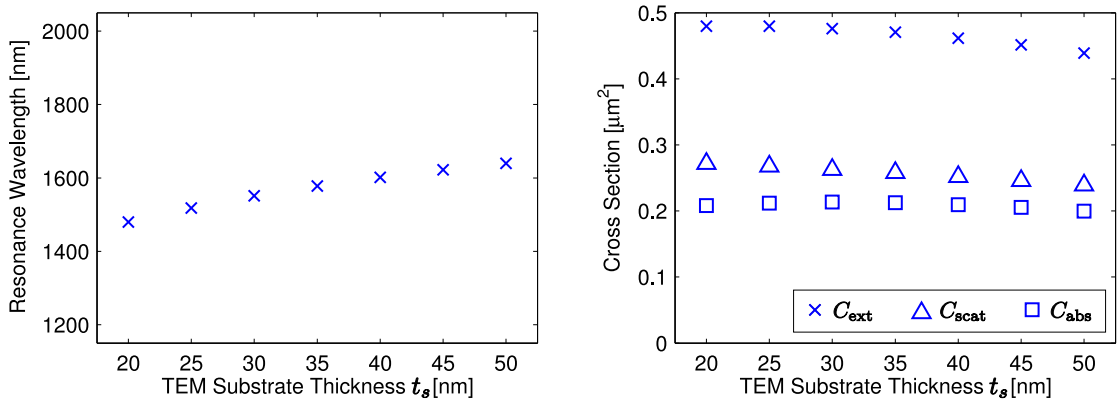


Figure 9.16: *Influence of the TEM substrate's thickness t_s on a single SRR's resonance.*

rate the substrate in the TF/SF source's spatial profile. The time-domain solution of a plane wave impinging on a thin film is given in section 2.2.3.

Results for TEM substrates of various thicknesses t_s are presented in Fig. 9.16. We note that the absorption and extinction cross sections are considerably higher than for the glass half space. For thicker substrates we observe a strong red shift ($150\,\mathrm{nm}$) of the resonance and a slight decrease in the cross sections. Scattering and absorption approach each other. At the same time, the quality factor decreases from $Q \approx 10.3$ for $t_s = 20\,\mathrm{nm}$ to $Q \approx 10.0$ for $t_s = 50\,\mathrm{nm}$ (not shown). This behaviour is somewhat expected. The thin film can be imagined as an imaginary half space with an effective refractive index. The thicker the TEM substrate, the larger the effective index will be, and thus we find behaviour similar to Fig. 9.15.

9.2.5 Summary

The present study indicates that a single SRR's resonance position, cross sections, and quality factor are strongly influenced by geometrical parameters and material constants. Similar results for periodic arrays have been reported in the literature [114,118]. Material constants are not that easily controlled in the experiment, but it shows that the material *model* used in the simulation significantly affects the outcome of the simulations. Smoothed corners introduced by the fabrication process can shift resonance wavelengths by as much as $100\,\mathrm{nm}$ compared to rectangular SRRs. The relative impact of all parameters on the SRR's fundamental resonance is conveniently summarised in Table 9.1.

9.3 Periodic Arrays of Split-Ring Resonators

In the previous section we have learned how geometrical parameters influence the fundamental resonances of individual SRRs. The investigation of individual SRRs, however, is not commonplace. Instead, arrays of SRRs are usually employed in experiments [110, 114, 119, 120]. Consequently, researchers are interested in transmittance, reflectance, and absorption spectra for such arrays. In this section we will investigate the idealised situation of infinitely extended, periodic arrays of identical

Parameter		$\boldsymbol{\lambda}_{\mathrm{res}}$	$\mathbf{C}_{\mathrm{ext}}$	$\mathbf{C}_{\mathrm{scat}}$	$\mathbf{C}_{\mathrm{abs}}$	**Q**
Width	w	++	++	++	0	- -
Height	h	+++	+	-	++	++
Gap width	w_g	+	0	0	0	-/0
Gap height	h_g	+++	+/0	-/- -	++	+++
Thickness	t	- - -	+++/+	+++/++	0/-	++/–
Corner radius	r	-	0	+	-	-
Coll. frequency	γ_D	0	- -	- -	0	- -
Substrate index	n	+++	- -	-	-	-

Table 9.1: *Summary of the results gathered from the parameter study. The table lists the investigated parameters and their effect on the resonance wavelength λ_{res}, the* peak *cross sections C_{ext}, C_{scat}, and C_{abs}, and the quality factor Q. The tag "+" indicates that the respective quantity increases if the value of the parameter increases. Conversely, a "-" represents that a quantity* decreases *with larger parameter values. The number of "+" and "-" indicates the relative influence of a parameter on said quantity. Finally, the symbol "0" indicates constant values or changes hardly worth mentioning. In some cases, slashes separate trends for low and high parameter values.*

SRRs by means of the DGTD method.

9.3.1 Simulation Setup

To study periodic arrays of SRRs we need another setup as the one described in section 9.2.1. Figure 9.17 visualises the computational domain used throughout this section.

The computational domain takes the shape of a cuboid with a geometrical cross section of $a_x \times a_y$ along the z-direction. Periodic boundary conditions (section 3.1.5) are applied along the x- and y-axes, which makes a_x and a_y the lattice constants in x- and y-direction. We place a standard SRR (see section 9.1.1 for parameters) on top of a substrate with refractive index $n = 1.5$. The remainder of the domain is "filled" with vacuum. The vacuum/glass interface is normal to the z-direction and roughly divides the system into two halves.

A plane I in the vacuum region injects a plane wave into the system via the TF/SF technique (section 6.1). In our study we only consider the case of normal incidence, i.e., the wave vector of the incident light is parallel to the z-axis. Thus, we can only excite Γ-point Bloch modes in the periodic lattice [2, 49]. In contrast to the scattering setup in section 9.2.1, this wave is not aware of the material interface, i.e., the spatial profile is given by Eq. (2.32). Thus, the material interface contributes to reflection. UPMLs (chapter 7) terminate the system in both the positive and negative z-direction and absorb transmitted and reflected light.

We use the tool `NETGEN` to create the meshes used in our simulations. All simulations are carried out in the time-domain using a 14-stage, fourth-order accurate low-storage Runge-Kutta scheme (chapter 4) for Lagrange polynomials of order $p = 4$ (section 3.1.6). As before, the dispersion of

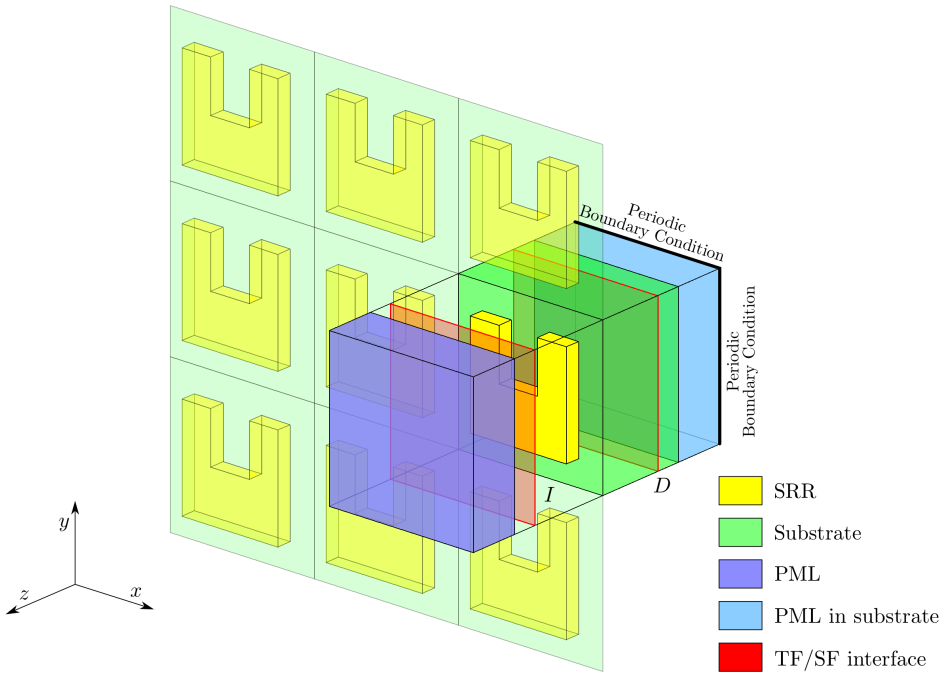


Figure 9.17: *Computational setup for the simulation of a periodic array of SRRs on a substrate. Along the z-axis, we have the following regions: A layer of UPMLs (dark blue) to absorb reflected light, a layer of vacuum (transparent) with the TF/SF injection plane I (red), the SRR (yellow), the substrate (green) with the detection plane D (red contour), and a layer of UPMLs in the substrate region (light blue). We employ periodic boundary conditions along the x- and y-directions.*

gold is modelled by auxiliary differential equations (section 6.2).

9.3.2 Transmittance, Reflectance, and Absorption

As mentioned in the introduction of this section, we are mainly interested in transmittance, reflectance, and absorption spectra. To record such spectra, we introduce a detector plane D behind the resonator (Fig. 9.17). Processing the *total* electromagnetic fields on this plane and the incident fields on the injection plane yields the transmittance

$$T(\omega) = \frac{\int_D \vec{\mathrm{S}}_{\mathrm{tot}}(\vec{\mathrm{r}},\, \omega) \cdot \hat{\mathrm{n}}\, \mathrm{d}^2 r}{\int_I \vec{\mathrm{S}}_{\mathrm{inc}}(\vec{\mathrm{r}},\, \omega) \cdot \hat{\mathrm{n}}\, \mathrm{d}^2 r}. \tag{9.6}$$

Please note that we employ an on-the-fly Fourier transform (section 6.4) to obtain frequency-domain quantities from our time-domain simulations. The reflectance R is calculated similarly to T by integrating the *scattered* field Poynting vector on the injection plane I:

$$R(\omega) = \frac{\int_I \vec{\mathrm{S}}_{\mathrm{scat}}(\vec{\mathrm{r}},\, \omega) \cdot \hat{\mathrm{n}}\, \mathrm{d}^2 r}{\int_I \vec{\mathrm{S}}_{\mathrm{inc}}(\vec{\mathrm{r}},\, \omega) \cdot \hat{\mathrm{n}}\, \mathrm{d}^2 r}. \tag{9.7}$$

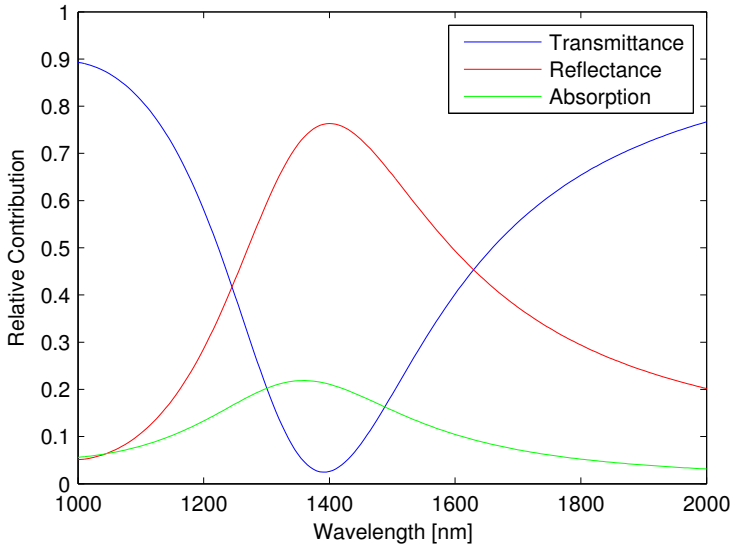


Figure 9.18: *Transmittance, reflectance, and absorption spectra for a periodic array of SRRs with standard parameters (section* 9.1.1*) and lattice constant* $a = 300\,\mathrm{nm}$.

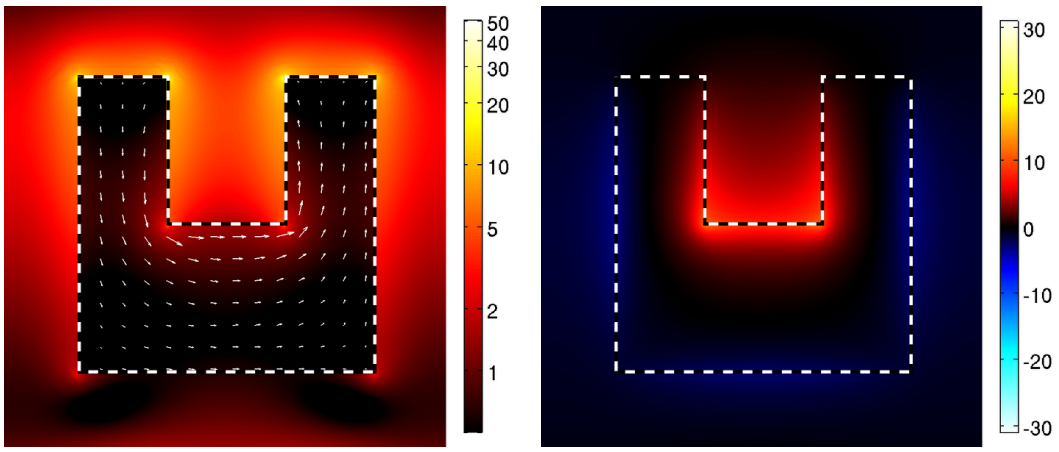


Figure 9.19: *Electric field and charge polarisation current (left panel) and out-of-plane magnetic field component* H_z *(right panel) at resonance wavelength* $\lambda_{\mathrm{res}} \approx 1393\,\mathrm{nm}$ *for one unit cell of a periodic array of SRRs with reference parameters (see section* 9.1.1*) and lattice constant* $a = 300\,\mathrm{nm}$. *For a more detailed explanation please refer to section* 9.2.3 *or Fig.* 9.3.

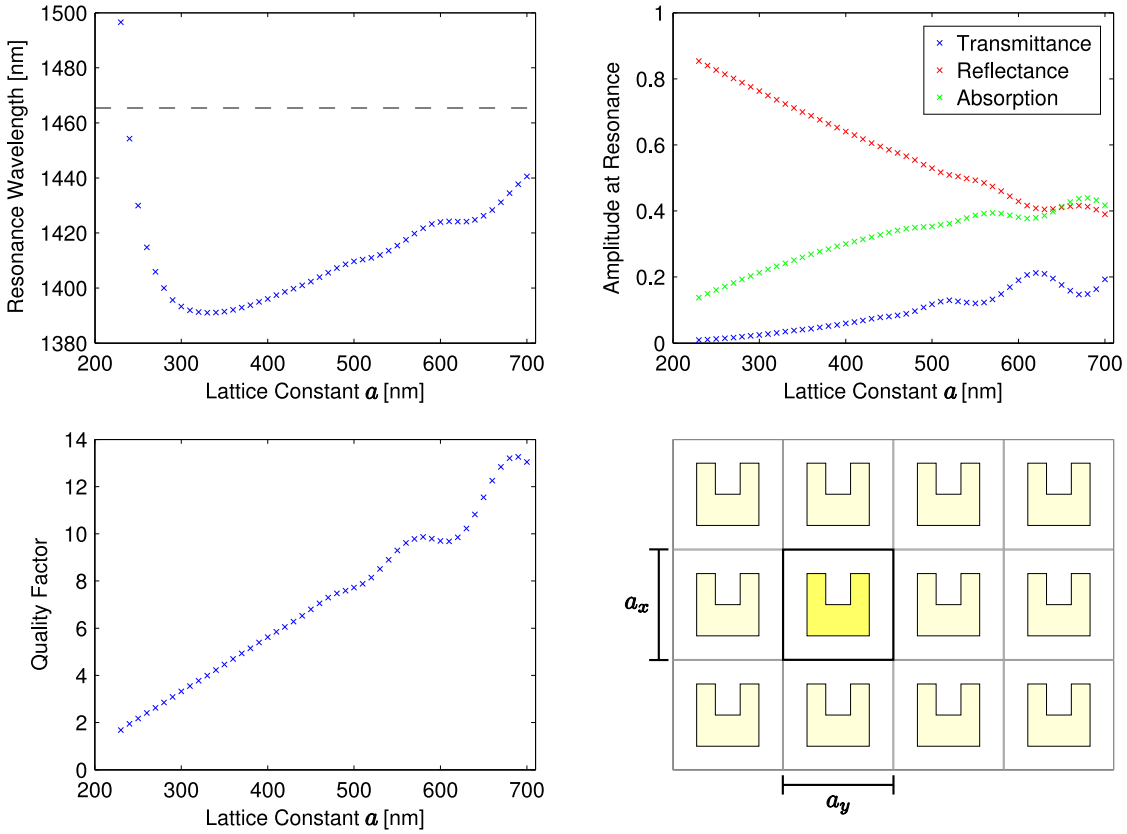


Figure 9.20: *Resonance behaviour of a periodic array of SRRs in dependence of the lattice constant a as indicated in the bottom right figure. The plots show the change in resonance frequency (measured for the transmittance spectrum), extremal values of transmittance (dip), reflectance (peak), and absorption (peak) at said frequency, and the quality factor. The dashed line in the top left figure indicates the resonance position of the individual SRR.*

Due to the discontinuous nature of the DGTD method, both the total and the scattered electromagnetic fields are available on the injection plane at any time during the simulation. Finally, the absorption A is related to T and R via

$$A(\omega) = 1 - T(\omega) - R(\omega). \tag{9.8}$$

Typical spectra can be found in Fig. 9.18. Corresponding field distributions at resonance are presented in Fig. 9.19. Please note that the fields, in particular H_z, are much weaker than those for an individual SRR (Fig. 9.3).

9.3.3 Influence of the Lattice Constant

With all geometrical parameters of the SRRs fixed, there is only one parameter left to investigate: The lattice constant a determines the distance between an SRR and its neighbours. We expect strong

coupling effects if the gaps between SRRs are rather small. For infinitely large distances the resonance frequency of the array should converge to the resonance frequency of the individual resonator investigated in section 9.2.

To quantify the influence of coupling, we perform additional simulations for $a \in [230,\ 700]\,\mathrm{nm}$ in steps of $10\,\mathrm{nm}$ and extract the respective resonance frequencies, quality factors, and peak values of transmittance, reflectance, and absorption (Fig. 9.20).

First of all, it is apparent that the resonance frequency (wavelength) is not as heavily influenced by the lattice constant a as it is by some geometrical parameters, e.g., the height h or the thickness t (see section 9.2.4). Strong changes only appear for small lattice constants near $200\,\mathrm{nm}$. These values of a correspond to very closely placed SRRs ($w = h = 200\,\mathrm{nm}$). For larger values, the resonance wavelength increases with the lattice constant. Similar behaviour has been reported in comparable experimental and theoretical studies [119, 120]. Near-field effects seem to be dominant in this area. For lattice constants above $450\,\mathrm{nm}$, oscillations appear in the resonance frequency, which seem to gain strength for larger a. Such oscillations have not been found in the references mentioned above because of too large variations of the lattice constant. In particular, the periodic array's resonance frequency does not approach that of the individual resonator within the considered parameter range.

Once the resonance frequencies are identified, we have a look at the respective values for the transmittance (a dip at this point), reflectance (peak), and absorption (peak). For small lattice constants, we have nearly perfect suppression of transmittance. For larger unit cells, and consequently smaller fill factors, there is significant transmittance even on resonance. Similarly, reflectance decreases with increasing a. Absorption approaches 40% for larger unit cells. All three quantities show oscillatory behaviour with growing amplitudes for $a \geq 450\,\mathrm{nm}$, which is consistent to observations made for the resonance frequency.

The quality factor strongly depends on the size of the unit cell as well. For small a we have very broad resonances, which get sharper as the lattice constant increases. The familiar oscillations are present as well.

We conclude that there must be a significant degree of coupling between neighbouring split-rings for larger lattice constants. This coupling is probably not limited to next-neighbours. For small lattice constants, nearest-neighbour interaction appears to be dominant as no oscillations are present.

9.4 Split-Ring Resonator Dimers

In the previous section we have tried to investigate the influence of coupling on resonances of SRRs. The setup, however, is not ideal to study the *origin* of the related effects. In particular, the source of the oscillations observed in 9.20 is unclear. To this end, we part with the array setup and focus on the coupling of *two* individual SRRs. Together with our collaborators of the group of Prof. Wegener, we have published similar results with differently shaped SRRs in Ref. [112].

9.4.1 Simulation Setup and Configurations

As usual, we start by specifying the simulation setup. We would like to investigate the following SRR dimer configurations (Fig. 9.21):

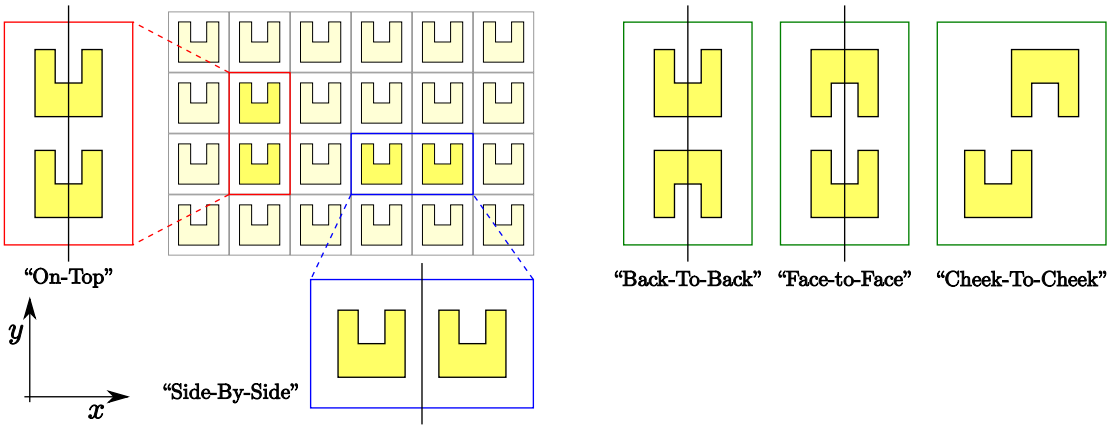


Figure 9.21: *A collection of SRR dimer configurations investigated in section* 9.4. *The "on-top" and "side-by-side" configurations can be found in periodic arrays. The other three configurations allow further analysis of coupling effects. Please note that thick black lines in the configuration sketches indicate the plane of symmetry in the scattering setup of Fig.* 9.5. *The "cheek-to-cheek" configuration does not have a plane of symmetry.*

- Two SRRs with the same orientation and same y-coordinate, but a relative shift along the x-axis are called the "side-by-side" configuration. This configuration occurs in periodic arrays.
- Two SRRs with the same orientation and same x-coordinate, but a relative shift along the y-axis are called the "on-top" configuration. This configuration occurs in periodic arrays as well.

In addition, we investigate three configurations which do not occur in periodic arrays (also see Fig. 9.21):

- The "back-to-back" configuration is similar to the on-top configuration. However, the SRRs are oriented in a way that both gaps face outwards, i.e., away from each other.
- The opposite is true for the "face-to-face" configuration. The SRR positions are the same as in the on-top configuration, but the SRRs are oriented such that the gaps point inwards, i.e., towards the respective other SRR.
- The last configuration is related to the face-to-face configuration. However, we shift the upper SRR by $w_a + w_g = 140\,\mathrm{nm}$ to the right. We will refer to this situation as the "cheek-to-cheek" configuration.

To actually simulate these configurations we rely on the basic scattering setup presented in section 9.2.1 and Fig. 9.5. For the on-top, back-to-back, and face-to-face configurations, we include a second SRR in the system and centre it in x-direction. The mirror symmetry of the single SRR case is conserved. Thus, the simulation of two SRRs is equivalent to the simulation of two halves of two SRRs.

Similarly, the side-by-side configuration boils down to moving the single SRR in Fig. 9.5 off the mirror plane. Thus, we can reduce two coupled SRRs to one SRR plus mirror symmetry. The cheek-to-cheek configuration breaks the mirror symmetry of the standard setup. Thus, we have to remove

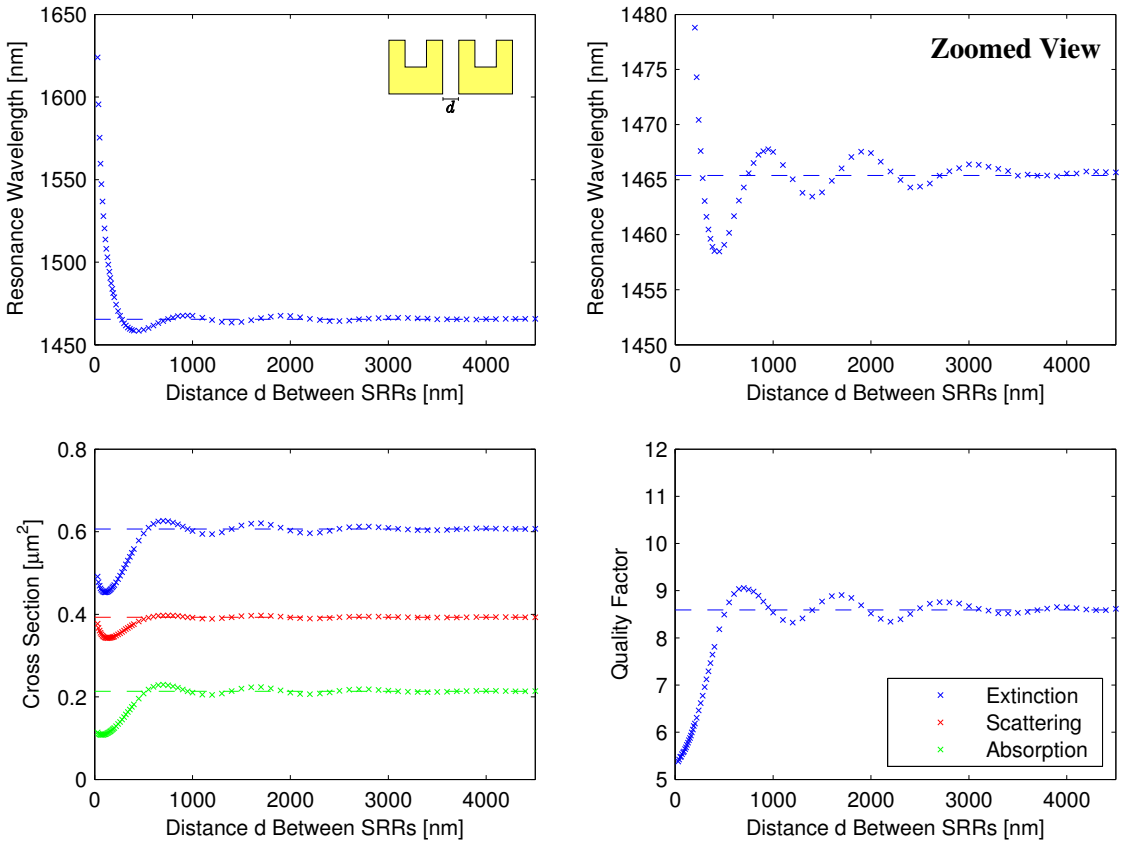


Figure 9.22: *Distance dependence of the resonance frequency, cross sections, and quality factor of two coupled SRRs in side-by-side configuration. The upper right plot is a zoomed view of the upper left plot and highlights the long-range interaction of the SRR dimer. In all plots, dashed lines indicate values expected for infinitely separated SRRs. This corresponds to the case of a single SRR with the cross section amplitudes multiplied by a factor of 2.*

the symmetry plane and simulate the entire computational domain surrounded by PMLs. Planes of symmetries, where applicable, are also included in Fig. 9.21.

For all configurations we perform DGTD simulations for various distances d up to $4.5\,\mu\mathrm{m}$ (!) between both SRRs and record the resulting scattering, absorption, and extinction cross section spectra.[5] Results of such a study[6] have been published in Ref. [112], where they reproduce the results of a beautiful experiment conducted by Nils Feth *et al.*. In all configurations, we expect the dimers to behave like individual, uncoupled particles as d approaches infinity. Thus, the *single* SRR with standard parameters (section 9.2) serves as a reference, from which we immediately take the resonance wavelength and the quality factor. The single SRR's peak cross sections must be multiplied by a factor of two to match the number of SRRs in the dimer configurations.

[5] Please note that d exceeds both the dimensions of the SRRs and the resonance wavelength.
[6] Albeit for smaller d, other SRR parameters, and not for all configurations.

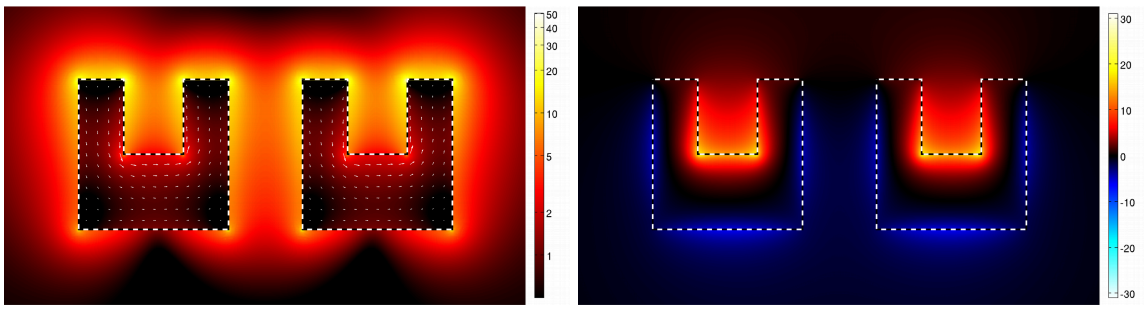


Figure 9.23: *Electric field and charge polarisation current (left panel) and out-of-plane magnetic field component H_z (right panel) at resonance wavelength $\lambda_{\mathrm{res}} \approx 1521\,\mathrm{nm}$ for two SRRs in side-by-side configuration ($d = 100\,\mathrm{nm}$) with reference parameters (see section 9.1.1). For a more detailed explanation please refer to section 9.2.3 or Fig. 9.3.*

9.4.2 Side-By-Side Configuration

For a start, we investigate the distance-dependence of the fundamental resonance in side-by-side configuration. The results of our simulations are compiled in Fig. 9.22, which also contains the definition of the distance d as an inset. For small distances $d < 200\,\mathrm{nm}$, we observe a strong red shift of the resonance with respect to the resonance of the single SRR. Electromagnetic field distributions for a dimer with $d = 100\,\mathrm{nm}$ can be found in Fig. 9.23.

This red shift can be qualitatively understood in a simple picture. Since the excitation of both SRRs is in phase (normal incidence), the charge separation in both SRRs is symmetric. If positive charges accumulate in the left SRR's left arm, positive charges will also accumulate in the right SRR's left arm. At the same time, *negative* charges accumulate in the right arm of the left SRR. These charges exert an attractive force on the positive charges in the left arm of the right resonator and vice versa. This force counteracts the repulsive forces in the charge clouds, and promotes charge separation in each SRR. Seen in the LC circuit model of section 9.1.2, this effectively increases the capacitance C with respect to the individual SRR. Following Eq. (9.1), this leads to a decreased resonance frequency—and thus an *increased* resonance *wavelength*.

An alternative explanation is offered by the dipole model (Fig. 9.24). The electric field caused by the first (induced) dipole reduces the restoring force acting on the second (induced) dipole and thus reduces the resonance frequency. In turn, this corresponds to a higher resonance wavelength. It should be noted, though, that such a simple point dipole model neglects finite size effects undoubtly present in SRRs. Furthermore, our qualitative picture neglects retardation effects and should be valid for $d < \frac{1}{4}\lambda_{\mathrm{res}}$ only [112]. Higher multipole orders are neglected as well.

The red shift is accompanied by a reduced quality factor and a strongly suppressed absorption cross section (Fig. 9.22). Apparently, the vicinity of both SRRs gives rise to an additional, non-dissipative loss channel: The split-rings act as antennas which couple to the evanescent fields of the respective neighbouring resonator and scatter them to the far-field [112]. Scattering becomes dominant at the expense of absorption, although the maximum scattering cross section itself is slightly reduced as compared to the individual SRR, too. To understand this, it is important to realise that the geometrical cross section $C_{\mathrm{geo}} = 0.032\,\mu\mathrm{m}^2$ (see Eq. (9.5)) of a single resonator is much smaller

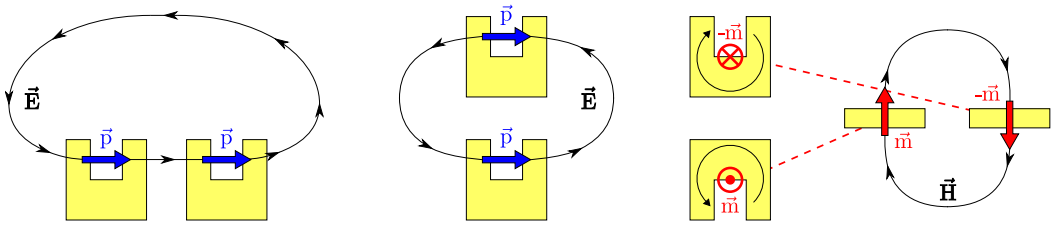


Figure 9.24: *Electric and magnetic dipole-dipole interaction in SRR dimers. Left panel: The electric field caused by the left electric dipole p exerts a force on the second dipole, which is directed to counteract the restoring force. As a result, the resonance frequency is decreased. Central panel: The electric field caused by the first dipole exerts a restoring force on the second dipole, and thus increases the resonance frequency. Right panel: Two antiparallel magnetic dipole moments $\vec{\mathrm{m}}$ and $-\vec{\mathrm{m}}$ interact with each other via the magnetic field. In the side view it is apparent that the magnetic field caused by the magnetic moment $-\vec{\mathrm{m}}$ counteracts the restoring force on $\vec{\mathrm{m}}$. As a result, the resonance frequency decreases. Please note that in this simple model we neglect retardation and finite size effects.*

than the corresponding scattering cross section. Thus, scattering cross sections "overlap" for closely positioned resonators. Since scattered light remains scattered even if it is scattered twice, this overlap effectively reduces the dimer's scattering cross section. Altogether, the peak extinction cross section C_{ext} is reduced for small separations d.

The situation is different for larger separations, where oscillations occur in all investigated quantities. The amplitudes of these oscillations decrease with increasing d; In the large separation limit, the behaviour of two individual SRRs is almost perfectly reproduced. The results for intermediate distances are consistent with corresponding theoretical and experimental studies performed for sphere dimers [121, 122]. Dahmen *et al.* identified constructive and destructive interference of the electromagnetic fields as the physical origin of oscillations in the resonance wavelength and the line width [121]. A model system with two point dipoles led to similar coupling effects as observed for the coupling of two spheres.

9.4.3 On-Top Configuration

The situation is similar, but subtly different for the on-top configuration (Fig. 9.25): For separations d below a few hundred nanometres we observe a blue shift of the dimer's resonance with respect to the reference resonance. Exemplary field distributions for $d = 100\,\mathrm{nm}$ are presented in Fig. 9.26. We will reapply the simple charge model used for the side-by-side configuration to provide an intuitive explanation of this shift.

For reasons of symmetry, both SRRs are excited in phase. At some time, positive charges accumulate in the left arms of *both* resonators, while negative charges are found in the right arms. The charge density is largest near the arms' tips, but excess charges are found in the base to a small degree as well. Since we have charges of equal signs in the tip of the lower SRR's arms and the upper SRR's base, there exists a small, but additional repulsive Coulomb force not present in the single SRR case. This effectively leads to a decreased capacitance C and—according to Eq. (9.1)—to a resonance at

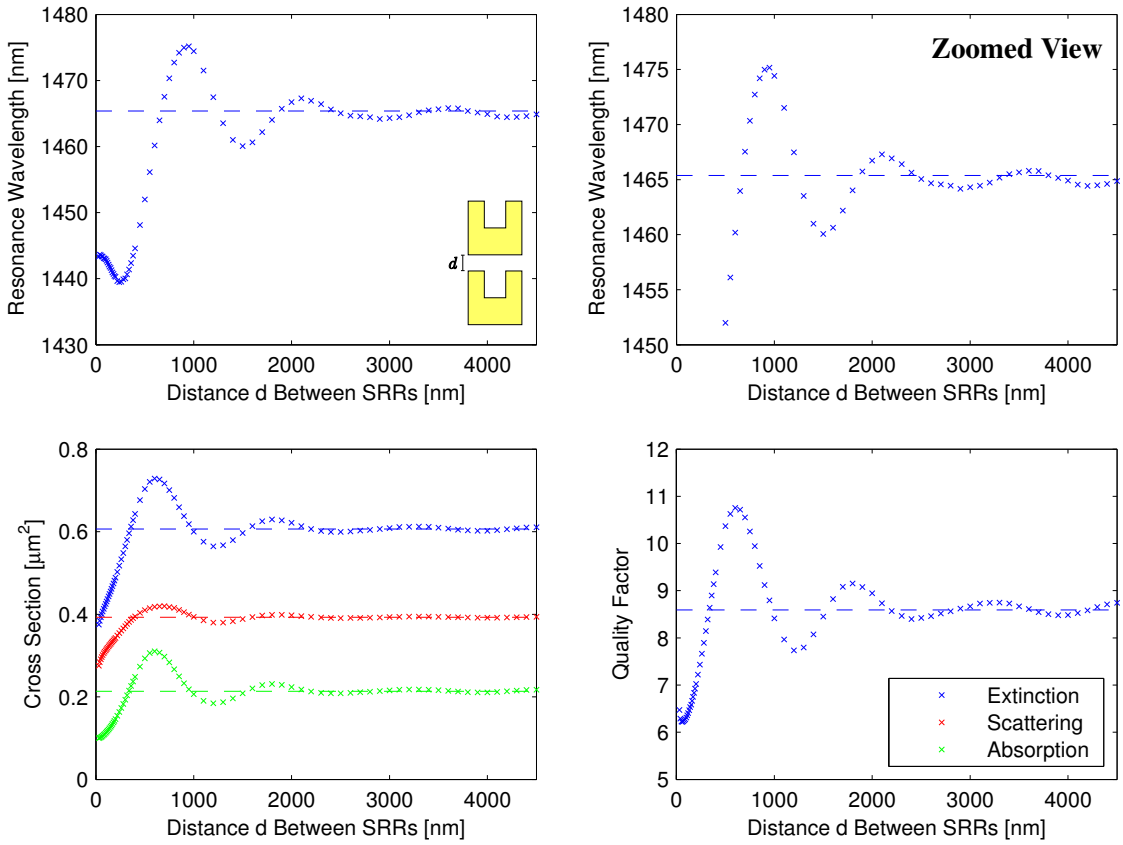


Figure 9.25: *Distance dependence of resonance characteristics for two coupled SRRs in on-top configuration. See Fig.* 9.22 *for a more detailed explanation.*

slightly lower wavelengths. An alternative explanation is again provided by electric dipole-dipole coupling (see Fig. 9.24).

As with the side-by-side configuration, non-negligible far-field interaction is also apparent in the on-top configuration. The oscillations are more pronounced and still influence the dimer's resonance for a separation of $4.5\,\mu\mathrm{m}$, which is more than twenty times the width of a single SRR. The explanation for the stronger coupling along the y-direction lies in the spatial profile of the SRR mode, which is more distinct in the direction of the gap. This behaviour is also consistent with the discussions in Refs. [121, 122].

9.4.4 Back-To-Back Configuration

The back-to-back configuration is similar to the on-top configuration. Thus, we would expect a similar behaviour in the near-field, i.e., a blue shift of the dimer's resonance frequency. We observe, however, the contrary effect (Fig. 9.27).

Even though the physical picture of charges used to explain the resonance shift in both the side-

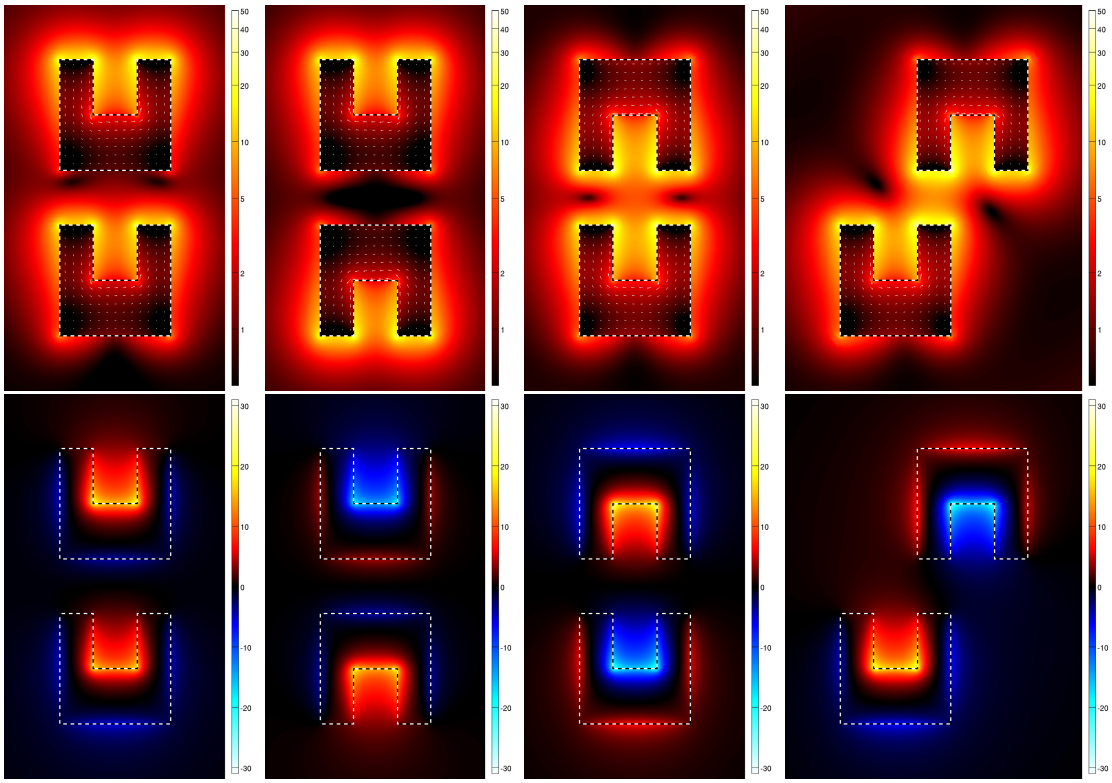


Figure 9.26: *Electric fields and charge polarisation currents (top row) and out-of-plane magnetic field component H_z (bottom row) for two SRRs in on-top, back-to-back, face-to-face, and cheek-to-cheek configuration ($d = 100\,\mathrm{nm}$) with reference parameters (see section* 9.1.1*). All data was recorded at the respective resonance wavelengths. For a more detailed explanation please refer to section* 9.2.3 *or Fig.* 9.3*.*

by-side and on-top configurations remains valid, the involved charge densities are much smaller: In contrast to the earlier examples, the massive accumulation of charges near the tips of the arms is practically irrelevant. These charges hardly interact with charges located on the other SRR because of the rather large distance (see Fig. 9.26). The small clusters of charges at the SRRs' bases interact with each other on a short range, but clearly this effect does not dominate the resonance.

Instead, we direct our attention to the current distribution in the SRRs. We observe that the currents are directed in opposite directions at any time (Fig. 9.26). Thus, the magnetic dipole moments of both SRRs are antiparallel. As illustrated in the right-hand panel of Fig. 9.24, the magnetic field caused by the first (induced) magnetic dipole counteracts the restoring force which acts on the second (induced) dipole. Thus, the resonance frequency (wavelength) is decreased (increased). We conclude that in this configuration magnetic dipole-dipole interaction dominates the resonance for small separations d, because the electric dipoles located at the gaps of both SRRs are too far apart [112, 113].

For larger distances we observe the by-now familiar oscillations in the investigated quantities. There are little differences compared to the on-top configuration, even though the amplitudes may be a bit smaller.

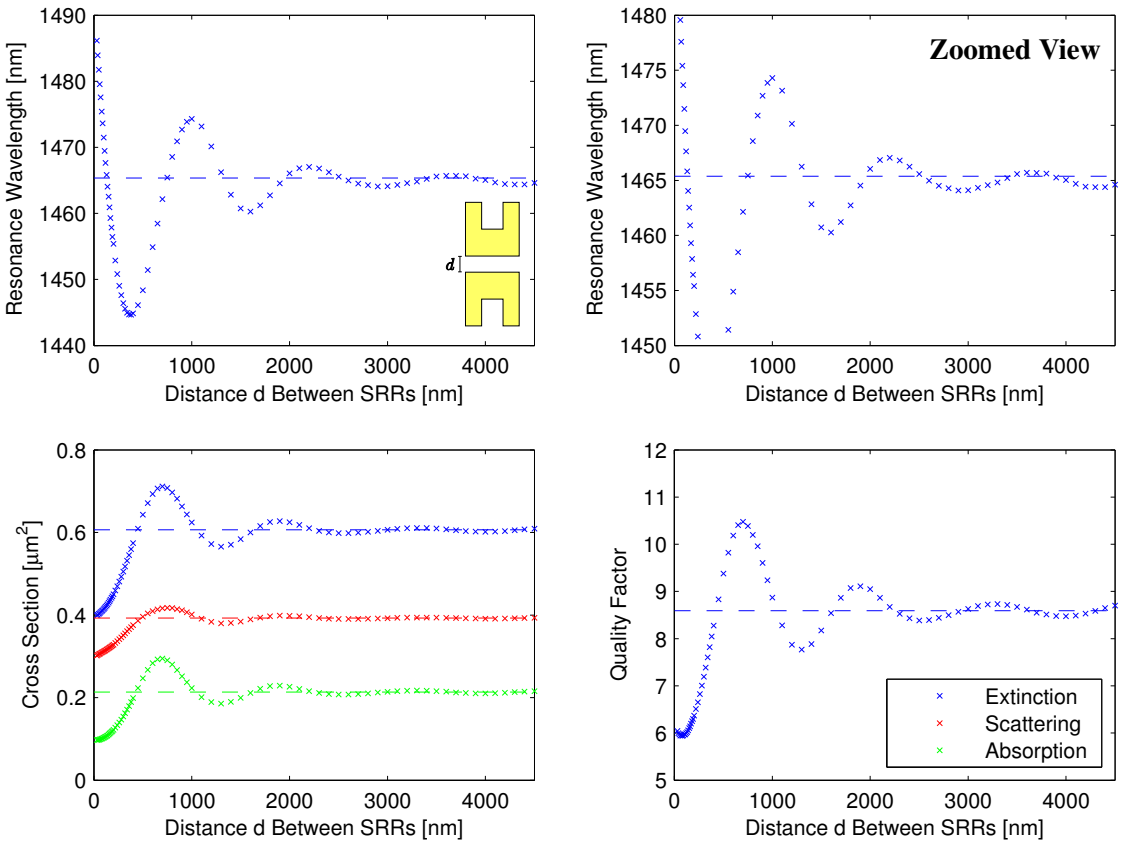


Figure 9.27: *Distance dependence of resonance characteristics for two coupled SRRs in back-to-back configuration. See Fig. 9.22 for a more detailed explanation.*

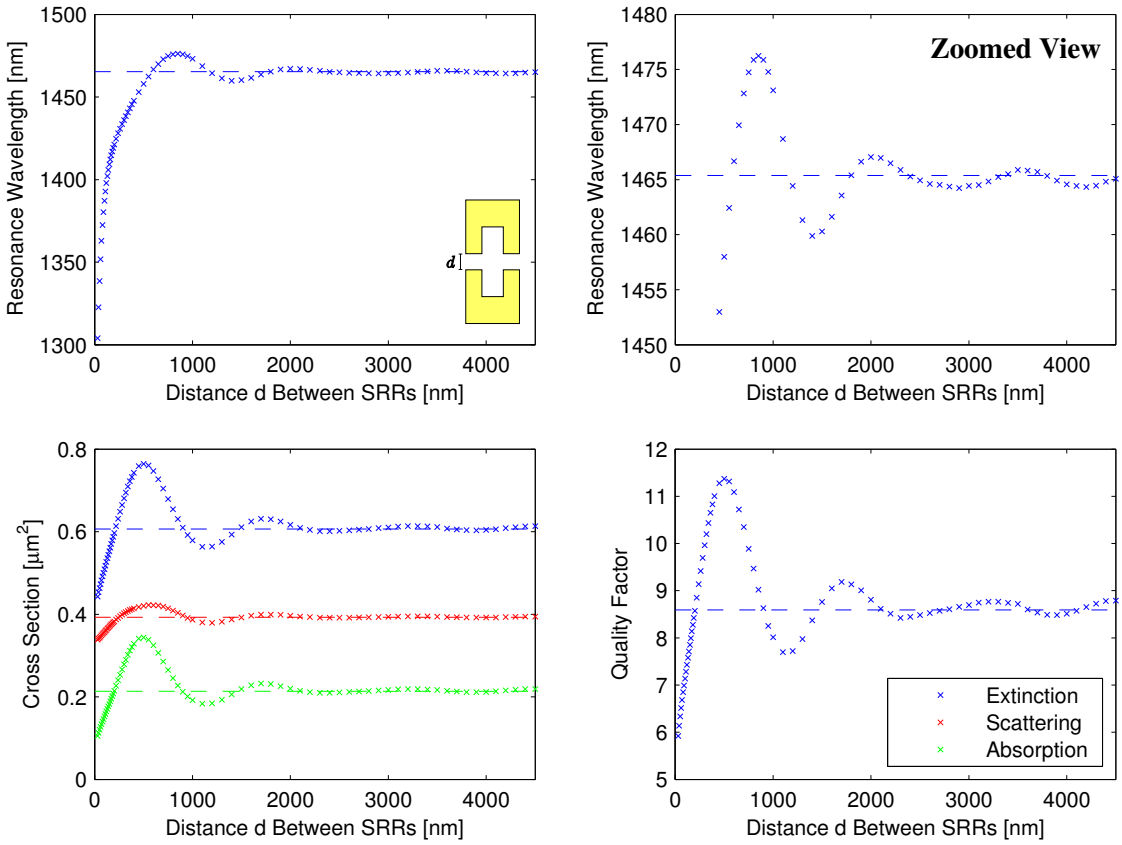


Figure 9.28: *Distance dependence of resonance characteristics for two coupled SRRs in back-to-back configuration. See Fig.* 9.22 *for a more detailed explanation.*

9.4.5 Face-To-Face Configuration

The face-to-face configuration is closely related to the previous two configurations. For closely-spaced SRRs, the resonance strongly shifts towards lower wavelengths as compared to the individual SRR (Fig. 9.28).

This shift is easily explained by the presence of electric charges in the tips of the opposing arms of both SRRs. The repulsive force between charges with equal signs reduces the effective capacitance, and thus reduces the resonance wavelength. In particular, we observe a small area where the electric fields created by the arms almost cancel each other (Fig. 9.26). As compared to the on-top configuration, we expect a stronger shift due to the close vicinity of the arms. For large distances we observe oscillations again very similar to the on-top and back-to-back configurations.

9.4.6 Cheek-To-Cheek Configuration

As a final test for the simple charge model, we would like to investigate the more uncommon cheek-to-cheek configuration. Since the SRRs are excited by the same plane wave, the opposing arms of

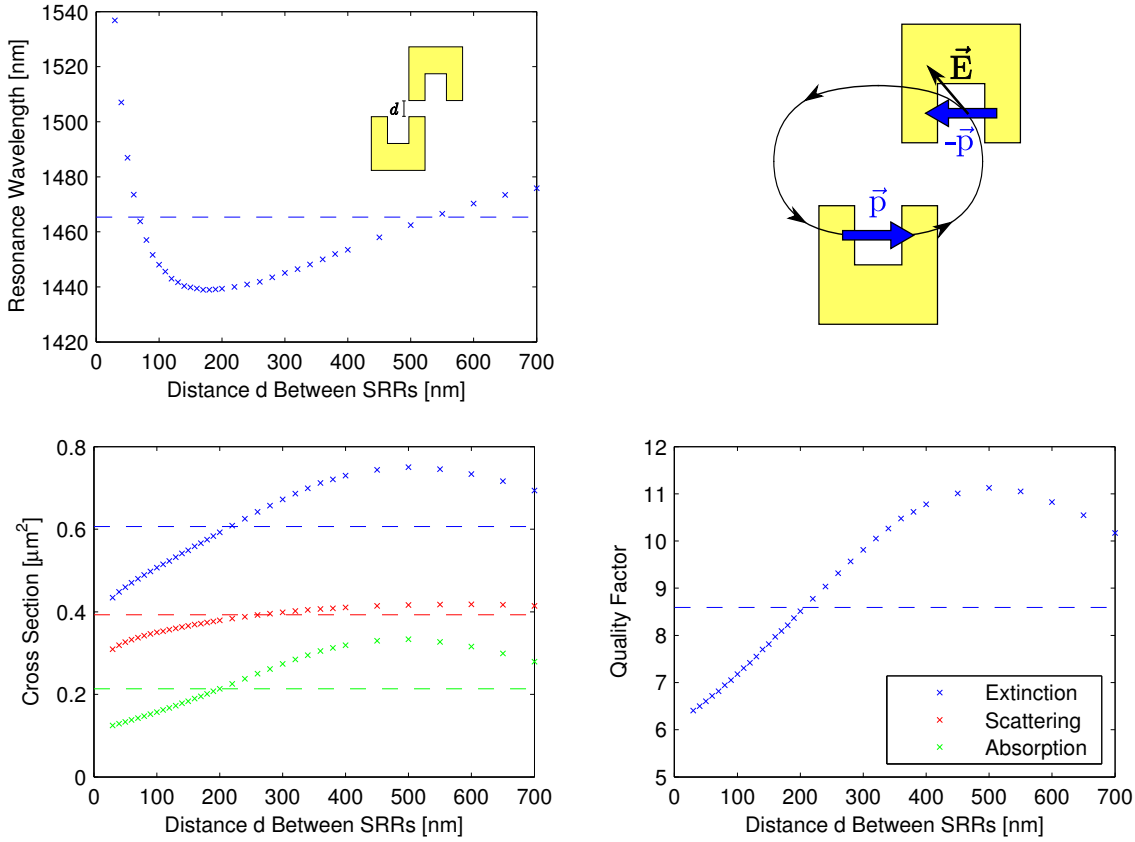


Figure 9.29: *Distance dependence of resonance characteristics for two coupled SRRs in cheek-to-cheek configuration. Please note the different abscissa in this plot compared to Figs.* 9.22 *to* 9.28 *which highlights the near-field interaction. The upper right panel illustrates the electric dipole-dipole interaction between both SRRs. As compared to the face-to-face configuration, the dipole moment* $-\vec{\mathrm{p}}$ *of the second SRR includes an angle with the electric field caused by the dipole moment* $\vec{\mathrm{p}}$ *of the first SRR. This weakens the influence of the electric dipole-dipole coupling for medium distances.*

both SRRs accumulate charges of opposite signs. This should lead to an attractive Coulomb force which increases the effective capacitance C of each resonator and leads to strong electric fields in between the opposing arms (Fig. 9.26). This translates to a larger resonance wavelength as compared to the individual reference SRR. This is indeed what we observe in a series of numerical experiments for very small distances $d < 70\,\mathrm{nm}$ (Fig. 9.29).

For medium separations, we observe a behaviour similar to the face-to-face configuration with a characteristic blue shift. However, compared to the results of Fig. 9.28 the blue shift is less pronounced. Qualitatively, this is expected because the shift in the horizontal direction leads to an angle between the electric field caused by the first resonator and the dipole moment of the second SRR (see Fig. 9.29). Thus, the electric dipole-dipole coupling is not as strong as in the face-to-face configuration. Large separations finally yield the familiar resonances (not shown).

9.4.7 Influence of Substrate Index

As a final topic we investigate the influence of the substrate index on the oscillations. To this end, we replace the substrate by vacuum, i.e., we set $n = 1.0$, and perform a series of simulations as before. The results (together with reference values obtained from an individual SRR *without* a substrate) are compiled in Fig. 9.30.

Using `MATLAB`'s curve fitting toolbox, we perform a fit of the distance dependence of the quality factor. We expect a sinusoidal oscillation around the quality factor obtained for an individual SRR, which is damped by a d^{-1} dependence. We chose this dependence over an exponential decay, because we expect the coupling to be mediated via the electromagnetic fields. For dipole radiation, the electric field decays with r^{-1}, where r would be the distance between the point dipole source and an observation point. Thus, we expect the quality factor to behave like

$$Q(d) = Q_\infty + \frac{\Delta Q}{d} \cdot \sin\left(\frac{2\pi(d - d_0)}{\lambda_{\mathrm{osc}}}\right). \tag{9.9}$$

Adapting the values of the infinite-distance limit Q_∞, the oscillation amplitude ΔQ, the oscillation length λ_{osc}, and the oscillation offset d_0 to best reproduce the data given in Fig. 9.30 for $d > 500\,\mathrm{nm}$, we find:

$$Q_\infty \approx 9.95, \quad \Delta Q \approx 686, \quad \lambda_{\mathrm{osc}} \approx 1119\,\mathrm{nm} \approx 0.92\lambda_{\mathrm{res}}, \quad d_0 \approx 0.452\,\lambda_{\mathrm{osc}}. \tag{9.10}$$

As before, λ_{res} denotes the resonance frequency of the individual SRR, in this case for $n = 1.0$, obviously. We perform an analogous fit for the side-by-side configuration *with* the substrate included. From the fitting procedure, we obtain the parameters

$$Q_\infty \approx 8.61, \quad \Delta Q \approx 323, \quad \lambda_{\mathrm{osc}} \approx 1008\,\mathrm{nm} \approx 0.69\lambda_{\mathrm{res}}, \quad d_0 \approx 0.5273\,\lambda_{\mathrm{osc}}. \tag{9.11}$$

The results of both fitting procedures can be found in Fig. 9.31.

It is apparent that the vacuum fit is of much higher quality than the fit involving the substrate. Considering the parameter values (9.10) for the vacuum case, we observe that the oscillation length λ_{osc} is in the order of the resonance wavelength λ_{res}. Small deviations are to be expected because the SRRs are not point dipoles and finite-size effects play a role. In the case of the side-by-side configuration on a substrate, however, Eq. (9.11) indicates that the oscillation length is much lower

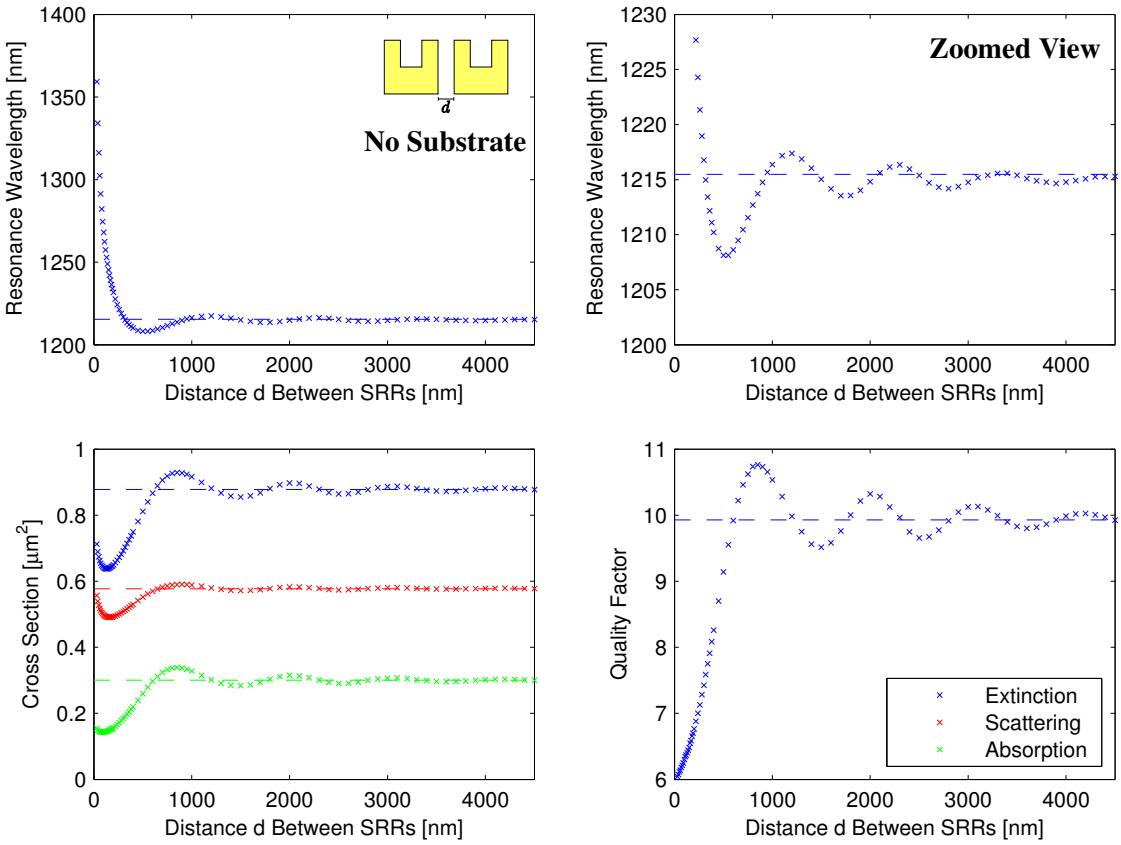


Figure 9.30: *Distance dependence of resonance characteristics for two coupled SRRs in side-by-side configuration, but* without *a substrate. See Fig.* 9.22 *for a more detailed explanation.*

than the resonance wavelength. We can merely guess that the presence of the substrate leads to an effective reduction of the oscillation length by a factor of $n' \in [1.0, n]$, potentially because of the formation of surface waves with reduced velocity as compared to the vacuum speed of light. Alternatively, the behaviour may result from beating of two waves of different wavelengths in the substrate and vacuum regions. In any case, we conclude that a simple dipole interaction picture cannot accurately describe the physics for larger separations d if interfaces are present.

9.4.8 Summary

Let us briefly summarise the results obtained in this section. If two SRRs are placed in close vicinity, we observe strong shifts of the resonance frequency. In most cases, a simple electric charge picture is sufficient to predict if we will observe a blue or a red shift. Alternatively, the interaction of two electric dipoles qualitatively describes the same behaviour. In one configuration, specifically the back-to-back configuration, we find that magnetic dipole-dipole interaction dominates the shift of the resonance because the electric dipoles are too far apart. In all configurations the quality

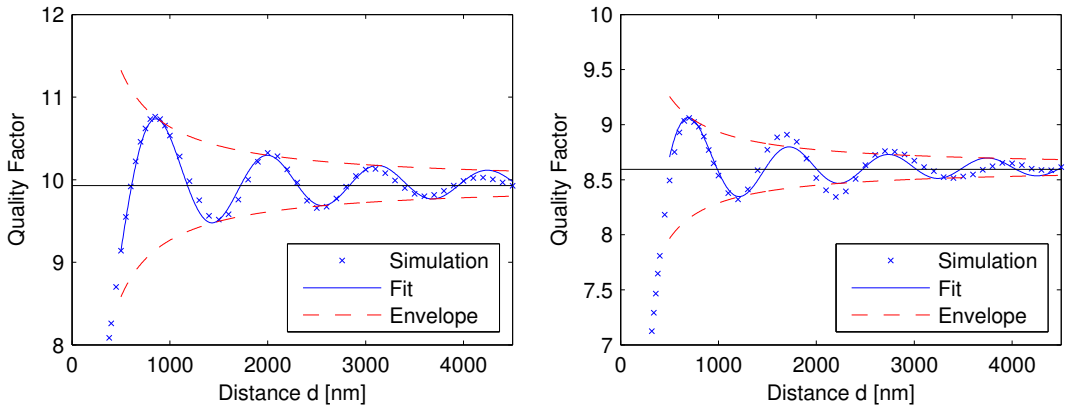


Figure 9.31: *Fits of the distance-dependent quality factors for SRRs in side-by-side configuration in vacuum (left) and on a substrate with $n = 1.5$ (right). Crosses indicate data points from DGTD simulations, the straight lines represent fits according to Eq.* (9.9)*. Red dashed lines form the envelope of the oscillations. The horizontal black line indicates the quality factors of individual SRRs in vacuum and on a substrate, respectively.*

factor decreases simultaneously with the absorption cross section for smaller distances d: SRRs act as antennas and scatter the near fields of their respective neighbours into the far field before the electromagnetic energy is dissipated.

In the far field, all configurations show a similar behaviour of oscillating resonance frequencies, quality factors, and cross sections. These effects are consistent with those observed for dimers of gold spheres in previous works [121, 122] and can be mostly attributed to dipole coupling. The presence of a substrate significantly complicates matters, but the overall effect appears to be the same.

9.5 From Individual Particles to Finite-Sized Arrays

Finally, we employ the DGTD method to construct finite-sized arrays out of individual resonators. We successively add more SRRs to the array and monitor extinction, scattering, and absorption spectra. In the limit of infinitely many SRRs, we expect a single resonance which coincides with that of a *periodic* array of corresponding lattice constant. Before we construct a full array, however, we investigate linear chains of SRRs first.

9.5.1 Horizontally Coupled Linear Chains

For a start, we construct finite linear chains of horizontally coupled SRRs. These chains consist of a single row and c columns in the x-y plane (see Fig. 9.32). Thus, we denote such chains by the symbol "$1 \times c$". The distance d measured between two adjacent SRRs is $100\,\mathrm{nm}$. Given a width of $200\,\mathrm{nm}$, this corresponds to a lattice constant of $a = 300\,\mathrm{nm}$ in a periodic array. Adding more SRRs to the scattering setup (see sections 9.2.1 and 9.4.1), we simulate chains with up to $c = 9$ columns.

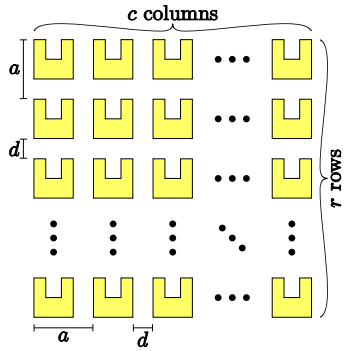


Figure 9.32: *Configuration of SRRs in linear chains and finite-sized arrays. The $r \times c$ array consists of r rows and c columns of SRRs. For horizontally coupled linear chains we have $r \equiv 1$, while for vertically coupled chains we have $c \equiv 1$. A distance $d = 100\,\mathrm{nm}$ between SRRs translates to a lattice constant $a = 300\,\mathrm{nm}$. Please note that all resonators are facing the same direction.*

Figure 9.33 presents extinction, scattering, and absorption cross section spectra for these chains. To allow for an easy comparison between chains of different lengths, all cross sections are normalised to the number of SRRs in the chain. Thus, the spectra show the average contribution of an individual SRR to the chain's total cross sections.

For $c = 1$ (individual SRR) and $c = 2$ (dimer in side-by-side configuration) we observe a single resonance of Lorentzian shape in the considered wavelength range. For $c = 3$ a pronounced kink appears in the spectrum. For larger chains, the main resonance shifts towards higher wavelengths. The kink somewhat fades away for higher frequencies, although it is still visible in the absorption cross section spectra of larger chains. Intuitively, this kink must be caused by another resonance which overlaps with the main resonance. Since the resonances are rather broad and, conversely, the quality factors are rather low, it is near impossible to extract individual resonance frequencies, quality factors, and amplitudes of individual resonances from the spectra.

Concerning the amplitudes of the cross sections, we find that all cross sections decrease with increasing c. In particular, the ratio between scattering and absorption further decreases as compared to the dimer case. In addition, the spectrum seems to broaden. All these effects are expected from our experience concerning dimers: Due to the close vicinity of the SRRs in the chain, resonators in the middle of the chain have two neighbours which act as antennas and scatter light into the far field. This enables radiative decay, leads to lower quality factors and favours scattering over absorption. Furthermore, the extinction cross section of an individual SRR is much larger than the geometrical cross section of an SRR ($C_{\mathrm{geo}} = 0.032\,\mu\mathrm{m}^2$) and the size of a "unit cell" ($a^2 = 0.09\,\mu\mathrm{m}^2$ for a periodic array with $a = 300\,\mathrm{nm}$). Thus, there ought to be some "overlap" of extinction cross sections of neighbouring resonators. Since one cannot extinguish the same light twice, this overlap effectively reduces the extinction per SRR.

All in all, we find that horizontal coupling of SRRs amplifies the effects found in resonator dimers in side-by-side configuration (section 9.4). In particular, we observe a strong red shift compared to the individual SRR, and an even stronger red shift as compared to the periodic array. This is a hint that horizontal coupling is not dominant in periodic SRR arrays.

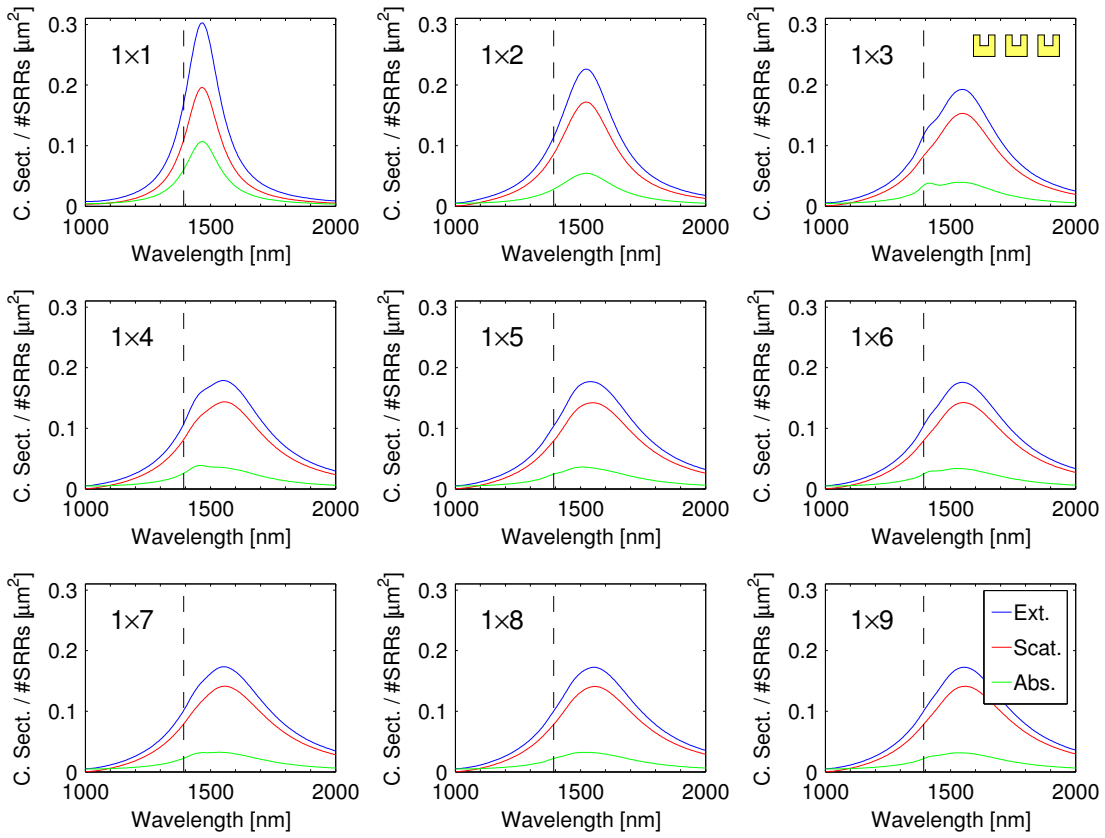


Figure 9.33: *Cross sections for finite-sized horizontally coupled linear chains with a lattice constant* $a = 300\,\mathrm{nm}$. *The plots show the extinction (blue), scattering (red), and absorption (green) cross sections* normalised *to the number of SRRs in the chain, i.e., they specify the* average *contribution which can be associated with a single SRR. The vertical dashed line indicates the resonance position of a* periodic *array of resonators with the same lattice constant* $a = 300\,\mathrm{nm}$.

9.5.2 Vertically Coupled Linear Chains

At the next stage, we investigate vertical chains of SRRs, i.e., we set the number of columns $c = 1$ and vary the number of rows r (see Fig. 9.32).[7] Spectra for various r are compiled in Fig. 9.34.

For one and two rows, i.e., an individual SRR and a dimer in on-top configuration, we obtain spectra of Lorentzian shape. As for the horizontal chain, kinks appear in the spectrum for $r = 3$ and diminish for larger chains. Both the main resonance and the kink shift towards the resonance of the SRR array, i.e., there is a blue shift with respect to the individual SRR's resonance wavelength. The balance between scattering and absorption is more lopsided than for the individual particle, but not as much as for the horizontal chain. Furthermore, the resonance remains rather sharp even for nine rows.

[7] The number of rows should not be confused with the corner radius, which is set to $0\,\mathrm{nm}$ in all these experiments.

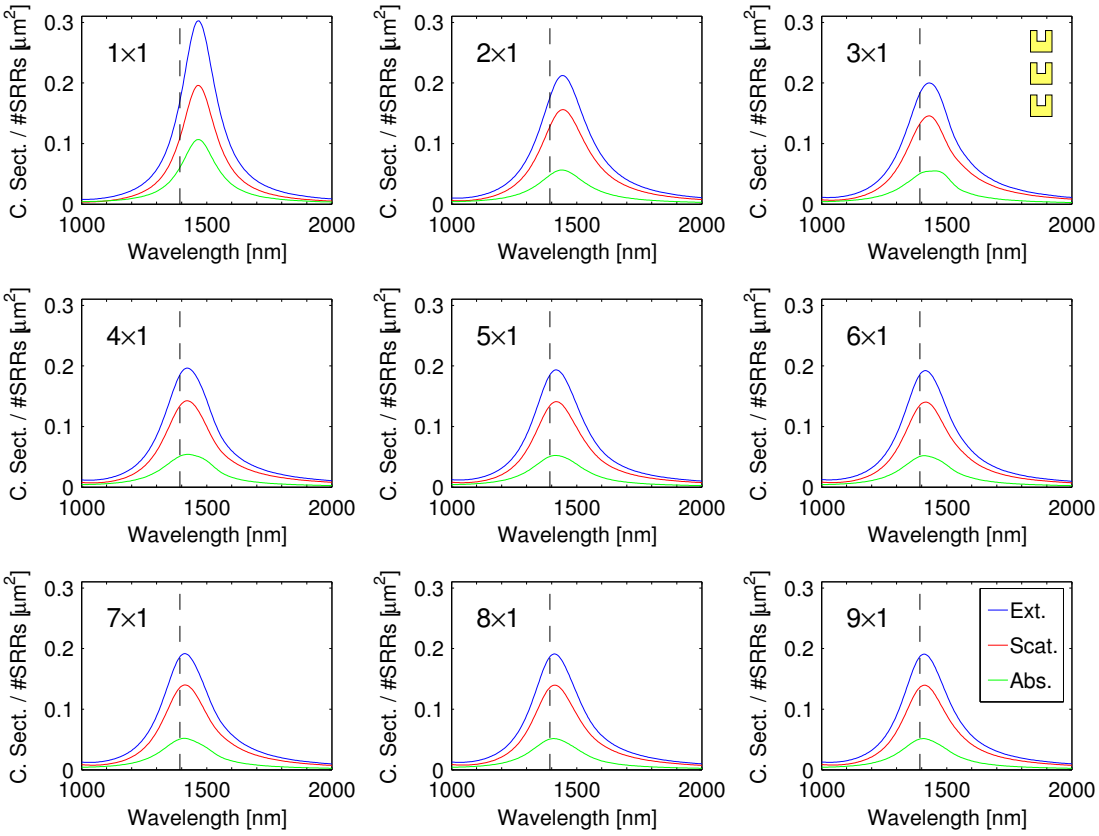


Figure 9.34: *Cross sections for finite-sized vertically coupled linear chains with a lattice constant* $a = 300\,\mathrm{nm}$*. The plots show the extinction (blue), scattering (red), and absorption (green) cross sections* normalised *to the number of SRRs in the chain, i.e., they specify the* average *contribution which can be associated with a single SRR. The vertical dashed line indicates the resonance position of a* periodic *array of resonators with the same lattice constant* $a = 300\,\mathrm{nm}$*.*

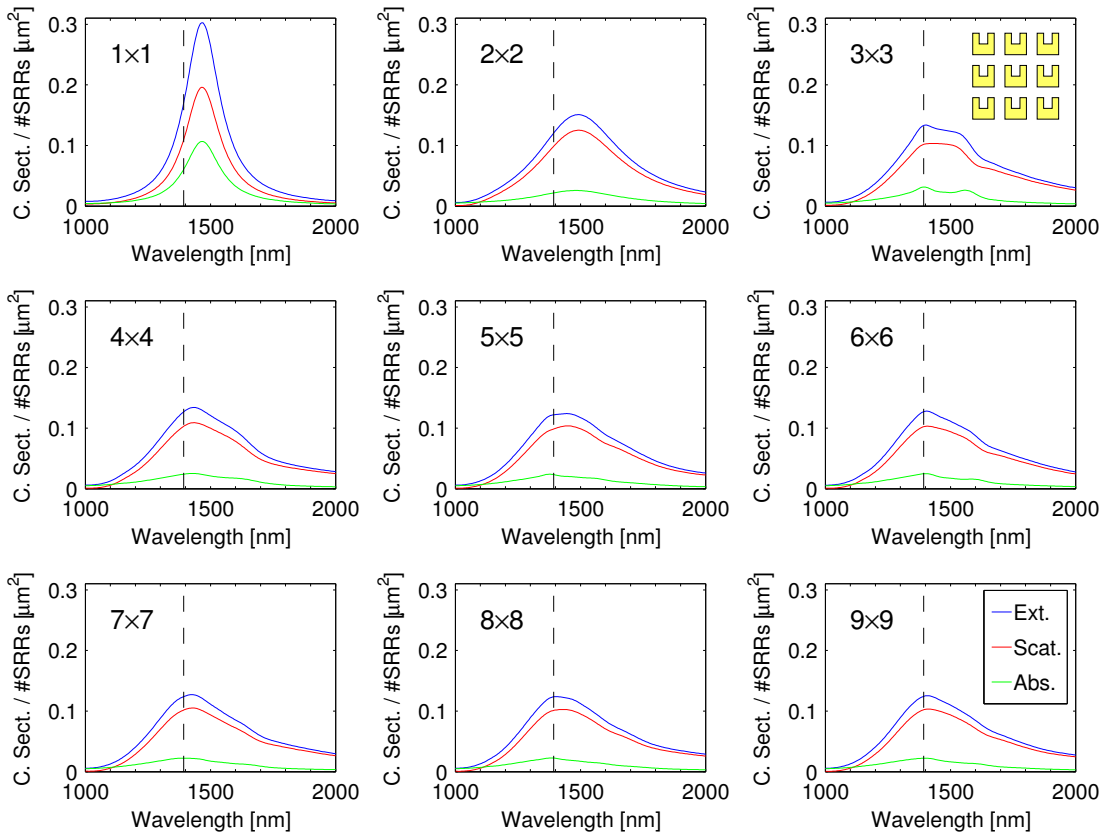


Figure 9.35: *Cross sections for finite-sized quadratic arrays with a lattice constant* $a = 300\,\mathrm{nm}$. *The plots show the extinction (blue), scattering (red), and absorption (green) cross sections* normalised *to the number of SRRs in the array, i.e., they specify the* average *contribution which can be associated with a single SRR. The vertical dashed line indicates the resonance position of a* periodic *array of resonators with the same lattice constant* $a = 300\,\mathrm{nm}$. *For arrays larger than* 2×2 *resonators the spectra strongly deviate from a Lorentzian shape. This is caused by a superposition of hybrid modes with low quality factors, which makes it difficult to identify individual resonances.*

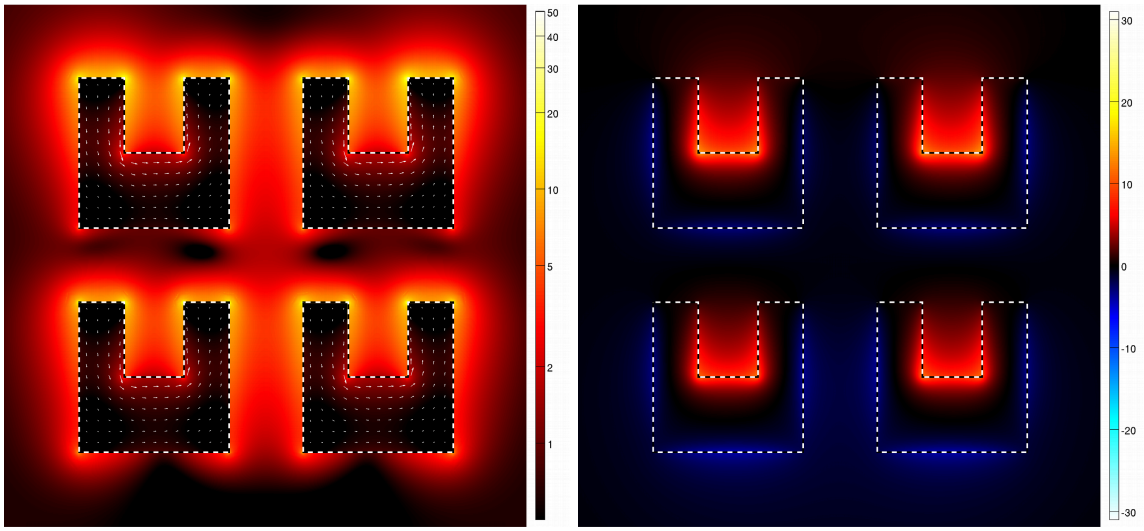


Figure 9.36: *Electric field and charge polarisation current (left panel) and out-of-plane magnetic field H_z (right panel) at resonance wavelength $\lambda_{\mathrm{res}} \approx 1490\,\mathrm{nm}$ for a 2×2 array of SRRs ($a = 300\,\mathrm{nm}$) with reference parameters (see section 9.1.1). For a more detailed explanation please refer to section 9.2.3 or Fig. 9.3.*

9.5.3 Finite-Sized Arrays

In a final step, we combine both types of chains to form quadratic $r \times r$ arrays of resonators for $r \leq 9$. To model the largest array with 40.5 resonators (effectively 81 due to symmetry) we use a mesh which consists of more than $700{,}000$ elements. Including all auxiliary fields in the metal and PML regions, this amounts to roughly $200{,}000{,}000$ degrees of freedom. Exploiting the parallel computing capabilities of DGTD and the general efficiency of our code, the longest computation takes just a bit over four days to finish on a computer equipped with two Intel® XEON® X5670 CPUs (12 cores) running at a clock speed of $2.93\,\mathrm{GHz}$. Out of the $48\,\mathrm{GB}$ of memory provided, the 9×9 array calculation requires less than $10\,\mathrm{GB}$. It is the efficiency of DGTD which enables us to investigate this highly complicated and extended scattering setup.

The results obtained are a continuation of those for the linear chains (Fig. 9.35). For arrays smaller than 3×3, the simulation yields a single, well-defined resonance in the considered frequency range. Figure 9.36 shows field distributions for a 2×2 array of SRRs. Starting with the 3×3 array, kinks and even side peaks appear in the spectra. Figure 9.37 shows field distributions for the 3×3 array at two wavelengths which correspond to maxima in the absorption cross section. The striking feature of these modes is that electromagnetic energy is concentrated in the outermost chains of SRRs. The central chain is almost dark. Whether the chains are aligned horizontally or vertically depends on the wavelength of the incident light. As expected from the results for linear chains, horizontal chains are visible for higher wavelengths. Vertical chains dominate for lower wavelengths. Figure 9.38 shows the field distribution for a 9×9 array at the resonance frequency of the periodic array. Vertical chains are still visible, but less pronounced: As compared to the dominant "surface" effects for smaller arrays, bulk effects are more important for larger arrays.

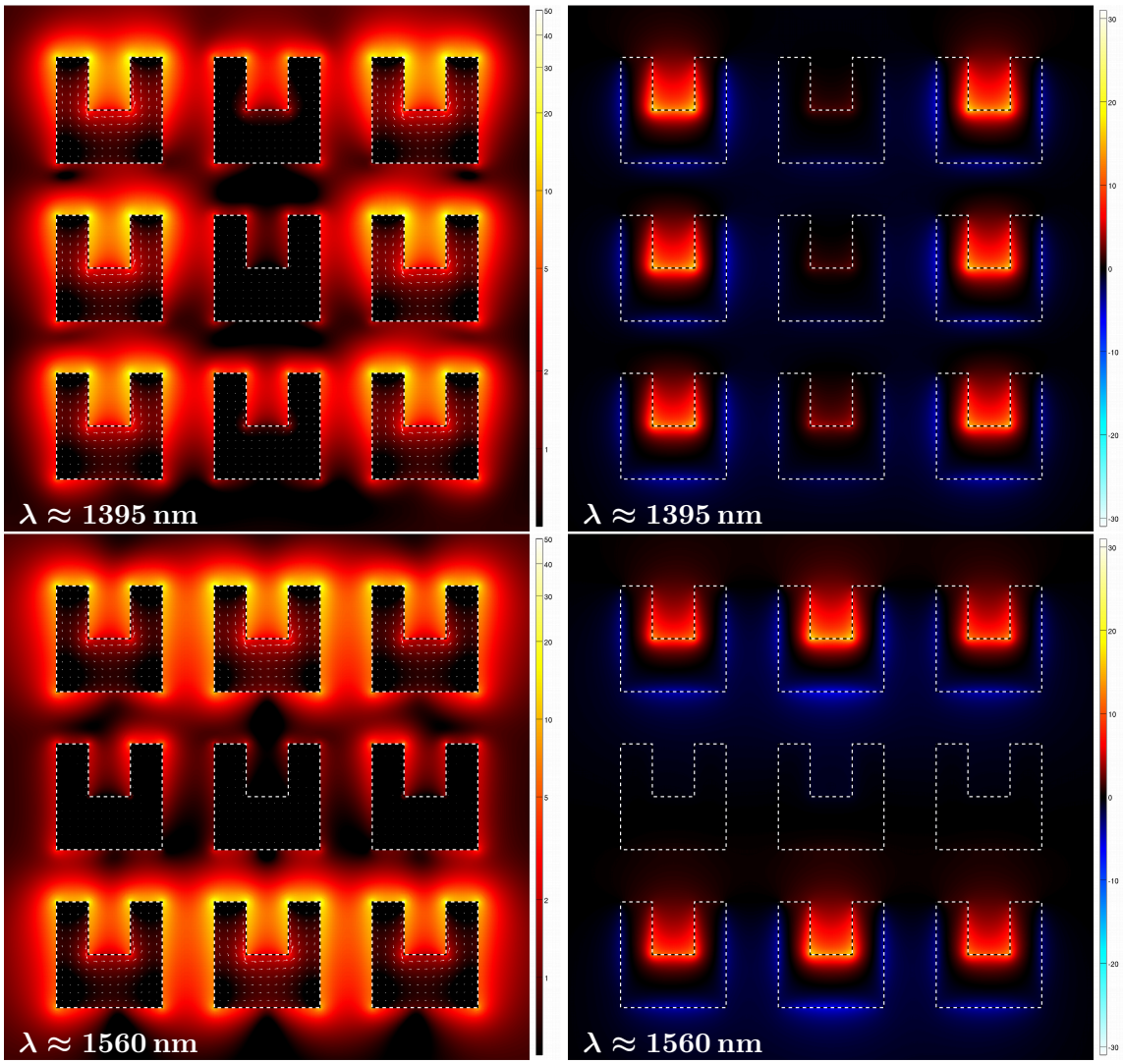


Figure 9.37: *Electric fields and charge polarisation currents (left column) and out-of-plane magnetic field component H_z (right column) at the suspected resonance wavelengths $\lambda_{\mathrm{res}} \approx 1395\,\mathrm{nm}$ (top row) and $\lambda_{\mathrm{res}} \approx 1560\,\mathrm{nm}$ (bottom row) for a 3×3 array of SRRs ($a = 300\,\mathrm{nm}$) with reference parameters (see section 9.1.1). For a more detailed explanation please refer to section 9.2.3 or Fig. 9.3.*

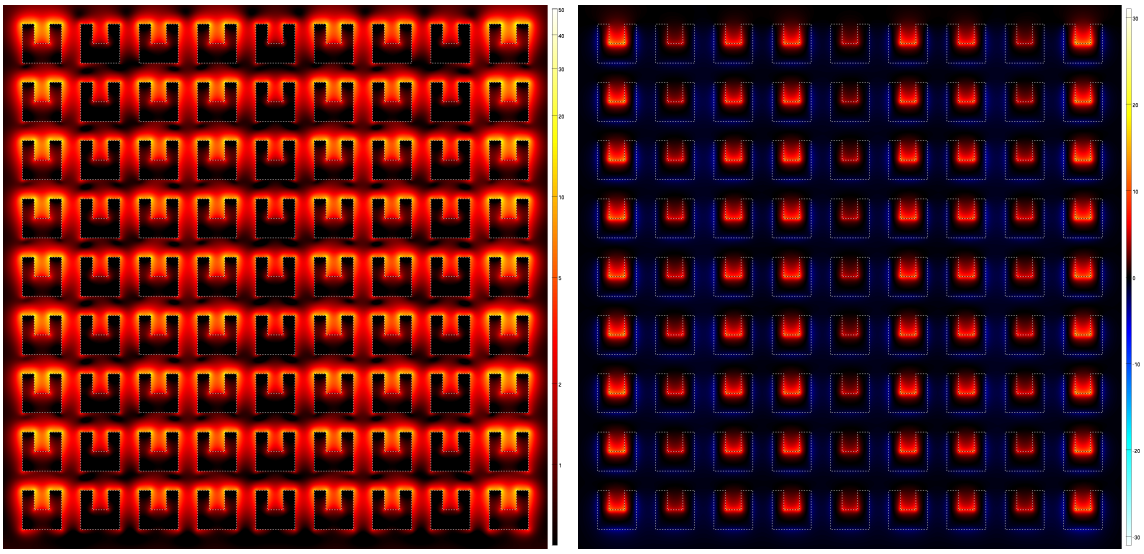


Figure 9.38: *Absolute value of the electric field (left panel) and out-of-plane magnetic field component H_z (right column) for a 9×9 array of SRRs ($a = 300\,\mathrm{nm}$) with reference parameters (see section 9.1.1). The incident wavelength corresponds to the infinitely periodic array's resonance position. For a more detailed explanation please refer to section 9.2.3 or Fig. 9.3.*

Overall, we observe the following tendencies in the spectra for larger arrays:

- The main resonance shifts towards the resonance wavelength of the infinite array.
- The spectrum retrieves smoothness.
- The spectrum broadens with respect to the individual resonator.
- Extinction is strongly dominated by scattering.

Comparing these observations with the results gathered for linear chains, it appears that an array's resonance position is mostly determined by vertical coupling. The quality factor and the ratio between scattering and absorption shows the behaviour of horizontal chains. It seems that horizontal coupling merely radiates energy from strongly coupled vertical chains to the far-field.

9.5.4 Conclusions

The last notion gives rise to educated design guidelines for periodic SRR arrays. The resonance position can be tuned by changing the parameters of the individual resonators and by the distance between SRRs in *vertical* direction. Relevant information is compiled in Table 9.1 and in Fig. 9.25. Losses and the quality factor of resonances can either be tuned by modifying the geometrical parameters or by changing the distance of SRRs in *horizontal* direction. The impact of changing the latter can be deduced from Fig. 9.22.

As a test for these guidelines, we try to design a periodic array of SRRs with a resonance at roughly $1.55\,\mu\mathrm{m}$ (telecommunication wavelength [123]), a high quality factor and a high fill factor. Following the advice above, we take the reference SRR and change the gap height to $g_h = 140\,\mathrm{nm}$

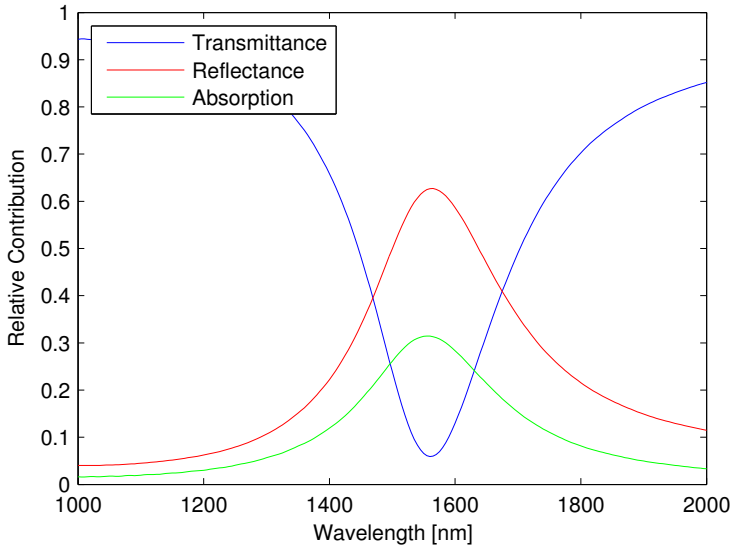


Figure 9.39: *Transmittance, reflectance, and absorption spectra for a periodic array of SRRs. The size of the unit cell is* $500\,\mathrm{nm}$ *in horizontal direction and* $300\,\mathrm{nm}$ *in vertical direction. Except for a modified gap height* $g_h = 140\,\mathrm{nm}$ *and a modified thickness* $t = 45\,\mathrm{nm}$*, the SRR shares geometrical parameters with the reference SRR of section* 9.1.1.

and its thickness to $t = 45\,\mathrm{nm}$. The unit cell has a size of $500\,\mathrm{nm} \times 300\,\mathrm{nm}$. A simulation with these parameters leads to a resonance wavelength $\lambda_{\mathrm{res}} = 1563\,\mathrm{nm}$ and a quality factor of $Q \approx 6.07$. Although this quality factor does not seem particularly high, it exceeds the quality factor of the quadratic array with $a = 390\,\mathrm{nm}$ (whose unit cell has roughly the same area) by 13%. We would like to point out that we have only performed a single simulation. In particular, there was no kind of optimisation process involved.

9.6 Summary

In this section we have presented an extensive numerical study related to SRRs. We have seen that the resonance of an individual resonator is strongly affected by geometrical parameters and the surrounding materials. The situation is more involved for periodic arrays of SRRs. In particular, the resonance wavelength and the quality factor of the resonance are strongly affected by the lattice constant and show an oscillatory behaviour for larger separations.

The origins of the observed physical effects can be traced back to the mutual coupling of SRR dimers. In particular, the distance between both split-rings plays a dominant role and controls the resonance frequency, quality factor, and extinction, scattering, and absorption cross sections. For small separations we have found reduced quality factors and absorption cross sections: Close vicinity of SRRs leads to additional coupling and radiation of near-field energy to the far-field. Resonance shifts can be explained in terms of simple charge and dipole-dipole interaction models. We have found substantial evidence for long-range coupling between SRRs, which can be attributed to dipole-

dipole coupling.

Exploiting our efficient DGTD computer code, we have taken our analysis one step further and have investigated finite-sized arrays of up to 81 SRRs. From these studies it is apparent that the resonance position of a periodic array is mainly determined by vertical coupling of linear chains. Horizontal coupling has a large influence on the absorption cross section of—and thus losses in—SRR arrays and influences the quality factor. The basic physics of the array, however, is the same as for the respective SRR dimers.

Our results indicate that a combination of knowledge obtained for single SRRs and SRR dimers provides a useful guideline for the design of periodic SRR arrays. We have successfully demonstrated the practicability of this approach with the help of a simple example.

10 Chapter 10 Summary and Outlook

We would like to conclude this thesis with the traditional summary and outlook. In section 10.1, we briefly highlight the main techniques and advances presented in this thesis. More details can be found at the ends of the respective chapters. Section 10.2 provides an overview of technical advances which may further improve the computational capabilities of the DG method. A few closing remarks constitute the end of this thesis in section 10.3.

10.1 Summary

The discontinuous Galerkin (DG) method is a numerical method which is well suited for hyperbolic partial differential equations like Maxwell's curl equations. It is essentially a variant of conventional finite element techniques and discretises spatial variables on the elements of a geometry-adapted mesh. On each element, the electromagnetic fields are expanded in terms of Lagrange polynomials of adjustable order p. Higher orders lead to a more accurate discretisation. A conceptual difference to other methods is that the electromagnetic fields are not uniquely defined at mutual interfaces of neighbouring elements. Instead, each element defines its own set of values at its boundaries.

This has two important consequences. First of all, the method has no immanent difficulties with material interfaces: Accuracy and convergence is not impaired by inherently discontinuous material parameters, fields, and corresponding derivatives. At the same time, this also leads to a high degree of locality in the scheme: Potentially expensive operations can be performed on all elements separately. The numerical flux, a concept borrowed from finite volume methods, weakly enforces field matching conditions at element interfaces. In this fashion, it transports information from one cell to its neighbours and allows for wave propagation. In the end, the DG scheme reduces to a set of matrix-vector multiplications between small matrices and vectors of expansion coefficients of the electromagnetic fields. For these operations, we provide assembly-based implementations for maximum efficiency.

The DG method is only applied to the spatial variables in Maxwell's curl equations. For the time variable two approaches have been presented. The discontinuous Galerkin time-domain (DGTD) method combines the spatial discretisation with an explicit low-storage Runge-Kutta time stepping scheme.

Alternatively, one may consider systems driven by an external source of fixed frequency and inves-

tigate time-harmonic field distributions. This approach effectively reduces to the solution of a system of linear equations. A number of strategies to solve such systems are available. Direct solvers are fast, but require lots of memory. Iterative solvers, on the other hand, are slow, but memory-efficient. Block Jacobi preconditioners in conjunction with iterative solvers find a balance between both extremes. Multigrid methods provide a better initial guess for iterative solvers. With these algorithms, we can investigate two- and three-dimensional problems with a realistic amount of computational resources.

The basic time- and frequency-domain techniques can be extended in numerous ways to handle dispersive materials, curved material interfaces, or sources of arbitrary spatial and temporal profiles. Novel implementations of perfectly matched layers and anisotropic materials have been presented and seemlessly interface with the standard algorithms.

Finally, we have investigated metallic nanoparticles called split-ring resonators (SRRs), which are often employed as building blocks of photonic metamaterials. The fundamental resonance of single resonators essentially depends on geometrical and material parameters. Periodic arrays of SRRs show that coupling heavily influences the resonance, in particular its wavelength and quality factor. For larger separations, these quantities show oscillations with the lattice constant.

Numerical experiments with SRR dimers indicate strong near-field coupling which significantly reduces the quality factor and leads to more scattering than absorption. For large separations, dimer resonances also show characteristic oscillations which can be attributed to dipole-dipole interaction between the SRRs. Adding more SRRs to the system leads to finite-sized chains and arrays. It is here where we rely most on the superior computational power of our DGTD implementation, which is able to handle several hundred thousands of elements and hundreds of millions of degrees of freedom with reasonable computational resources.

Small arrays show pronounced surface modes, while the distribution of electromagnetic energy is more homogeneous in larger arrays. Coupling between SRRs in vertical chains determines the resonance frequency of periodic arrays. Coupling in horizontal chains affects the width of the resonance and determines the ratio of scattered and absorbed light. The basic physics can be understood in terms of single resonators and SRR dimers, from which we can deduce design guidelines to tune the resonance of periodic arrays.

10.2 Outlook

From the scope of this thesis it is apparent that the DG method presents a very broad area of research. The application to nanophotonic systems is still relatively new and there is a huge number of potential systems which may benefit from DG's higher order accuracy.

Still, there are a number of technical extensions which may further increase the computational power of the DG method.

Mesh Generation

The spatial discretisation we have presented in chapter 3 is not ideal in a number of ways. Though tetrahedrons allow us to model almost arbitrary geometries, we usually do not require this freedom in the entire computational domain.

For example, a computational domain could be enclosed by an axes-aligned perfectly matched layer (chapter 7). While the actual physical domain with arbitrary scatterers will benefit from conventional tetrahedral meshes, such boundary regions could be easily tessellated into hexahedral elements [105]. Here, nodes would be positioned in terms of tensor products of one-dimensional node distributions. As derivatives would only incorporate degrees of freedom along a line, performance gains are immediately expected. Furthermore, the surface-to-volume ratio of a cuboid is lower than that of the tetrahedrons it can be divided into (compare Fig. 7.3, for example). Hence, we have less degrees of freedom associated with element boundaries. As we have to store these values twice, hexahedrons lead to a more memory-efficient scheme.

In a similar way, one could speed up simulations of layered systems such as split-ring resonators on substrates. In such systems, prism-shaped elements allow for a triangular representation of in-plane geometrical details, while providing a simple and efficient representation of the normal direction.

Unfortunately, such meshes pose significant technological challenges to mesh generation and handling, in particular if we want to mix elements of different shapes (tetrahedrons and hexahedrons, for example) in the same mesh. Appropriate data structures are needed to allow a mesh to hold elements of different types and to relate them with each other efficiently. More fundamentally, sophisticated algorithms which create optimal meshes for arbitrary geometries are required.

Spatial Basis Functions

Besides the element shape, the set of basis functions is another aspect of the DG discretisation to refine. While the nodal scheme is well tested and convenient due to the close relation between expansion coefficients and field values at the nodes, it supports fields of non-vanishing divergence (see section 5.2). Expressed in other terms, this means that the nodal basis contains superfluous degrees of freedom.

To eliminate this redundant information, one can construct a *modal*, vectorial basis which locally satisfies the divergence condition, i.e., in an element-wise fashion [76, 78, 124]. However, since locally divergence-free basis functions are not invariant under affine transformations, such an implementation dramatically increases the memory consumption. In addition, the calculation of the numerical flux also requires more computational effort in a modal representation when compared to the nodal version. Thus, it is not clear yet whether a locally divergence-free basis actually leads to performance improvements.

Local Adaptivity

Finally, one usually assumes that the fields in all element are discretised with the same local order of accuracy p. In addition to the local mesh refinement (h-refinement), DG also allows the local refinement (or coarsening) of p. Consider a structure with just a few small features which need to be resolved by a couple of small elements. Compared to the other elements, the feature region is discretised by considerably more degrees of freedom, and probably more than necessary. Hence, one could locally reduce the order of the basis functions. Following the discussion of section 4.3, this will increase the time step, as the distance between adjacent nodes is larger for lower orders. Technological difficulties are mostly restricted to the efficient computation of the numerical flux across elements with different local orders of accuracy.

Time Stepping

In contrast to FDTD, the spatial discretisation of DGTD is strictly separated from the time evolution of the fields. Though Runge-Kutta schemes show satisfactory performance, it might be worthwhile to investigate alternative time stepping techniques. The main motivation lies in the observation that in nanophotonics the time step is often limited by just a few exceptionally small or awkwardly shaped elements. If a time stepper could treat these elements separately, one could maintain a larger time step for the bulk of the elements. Hence, depending on the mesh dramatic performance boosts can be expected.

One possible strategy is to use explicit solvers with different time steps for differently sized elements [53, 125, 126]. For example, we might divide the elements into the categories small, medium, and large. Fields in small elements are evolved in steps of Δt, fields in medium-sized elements in steps of $2\Delta t$, and the largest elements in steps of $4\Delta t$. Problems arise because the numerical flux connects small and medium elements and we have to make sure that fields in both elements are known at the same time.

While such schemes are available and employed, e.g., in finite volume calculations [127], they are usually restricted to lower orders. Higher-order schemes [128] (or schemes of mixed order) appear more suitable to accompany the higher-order spatial discretisation of DGTD. Recently, higher-order Taylor approximations [129] and third-order Adams-Bashforth schemes [130] were employed to design local time stepping schemes for large electromagnetic problems.

Alternatively, one can employ hybrid implicit-explicit time stepping schemes [131]. Implicit time steppers require the solution of a system of linear equations, which is in contrast to the explicit nature of the usual DGTD update process. However, implicit schemes are unconditionally stable and allow large time steps with comparatively small numerical errors. By applying an implicit scheme to the smallest couple of elements only, one can increase the overall time step while keeping the computational effort per time step within reasonable bounds. The fields in the remaining elements are evolved in the usual explicit manner.

Massively Parallel Implementations

One of the key strengths of the DGTD method is that the algorithm essentially consists of a large number of small matrix-vector products. Over and over again, the same operation is applied to millions of independent data sets. Thus, DGTD lends itself to massively parallel implementations on processors with reduced instruction sets. Recently, scientific computing on graphics processors (GPUs) has seen much interest due to their superior computational power for simple instructions as compared to standard CPUs [63]. Here, the main challenge lies in providing a hardware-independent implementation which can handle not only the basic algorithm, but also extensions like dispersive materials and perfectly matched layers. DGTD simulations on GPUs will allow the investigation of large-scale systems with unprecedented speed and accuracy in the near future.

Material Models and Coupled System Dynamics

The material models presented in this thesis should be sufficient for most linear experiments in nanophotonics. However, nonlinear material responses [64] have attracted interest of a large number

of research groups and become more and more important.

Classical nonlinearities of $\chi^{(2)}$ and $\chi^{(3)}$ type are responsible for the optical Pockels and Kerr effects, wave mixing phenomena, etc., and are often found in substrate materials such as GaAs. The inclusion of nonlinear effects in the DGTD framework is not easy. The main problem is to find an efficient and accurate expression for the numerical flux.

In addition, nonlinear system dynamics arise when dealing with more sophisticated material models, too. For example, the simple Drude theory of metals could be extended to a hydrodynamic model, which describes the free metal electrons in terms of charge and current densities [132]. One obtains additional equations which have to be solved concurrently with Maxwell's equations. The coupling between both types of equations via currents and driving fields introduces nonlinearities and leads to the generation of higher harmonics. Other examples of coupled systems include the interaction of classical light fields with quantum mechanical systems. For example, the radiation dynamics of light emitted from quantum wells or atoms can be significantly influenced by the surrounding media [64].

While the linear regime seems to be well understood, it appears that it has not been reported yet how to incorporate nonlinear material models in the DG framework for Maxwell's equations. Once this issue is solved, it will open a whole new research area to the powerful machinery that is the DG method.

10.3 Closing Remarks

We believe that the discontinuous Galerkin method is on a good way to become one of the most efficient and versatile simulation tools for all kinds of nanophotonic systems. We are eager to see which further developments the future holds.

Appendix A

A The Finite-Difference Time-Domain Method

The finite-difference time-domain method (FDTD[1]) is probably the most widely employed simulation tool in nanophotonics and presented in detail in Taflove's excellent monograph [21]. First proposed by Kane Yee in 1966 [20], numerous sophisticated implementations are available today, including dozens of commercial and freely available simulation tools plus countless student exercises. In this section we *briefly* summarise the key concepts which have made FDTD such a popular tool, while we also mention the algorithm's limitations. This knowledge helps us appreciate the benefits of the discontinuous Galerkin method.

A.1 Finite Differences

Maxwell's curl equations (2.18) feature derivatives with respect to time and space. FDTD employs finite-difference approximations of these derivatives. Considering the Taylor series

$$F(x \pm h) = F(x) \pm \frac{1}{1!}\partial_x F(x)h + \frac{1}{2!}\partial_x^2 F(x)h^2 \pm \frac{1}{3!}\partial_x^3 F(x)h^3 + \mathcal{O}(h^4), \tag{A.1}$$

we can approximate the derivative of any *smooth* function $F(x)$ at any point x by

$$\partial_x F(x) = \frac{F\left(x + \frac{h}{2}\right) - F\left(x - \frac{h}{2}\right)}{h} + \mathcal{O}(h^2). \tag{A.2}$$

This approximation is second-order accurate, i.e., if we divide h by a factor of two, the discretisation error will reduce to a quarter of its previous value.[2] Furthermore, we approximate the derivative at x by taking values of F left and right of this position. Thus, Eq. (A.2) represents central differences.

[1] In oral presentations sometimes called FDDT, FTDT, FTTD, and—more unusual—FDT, FDDDT and FTDDT.
[2] Provided h is sufficiently small, that is.

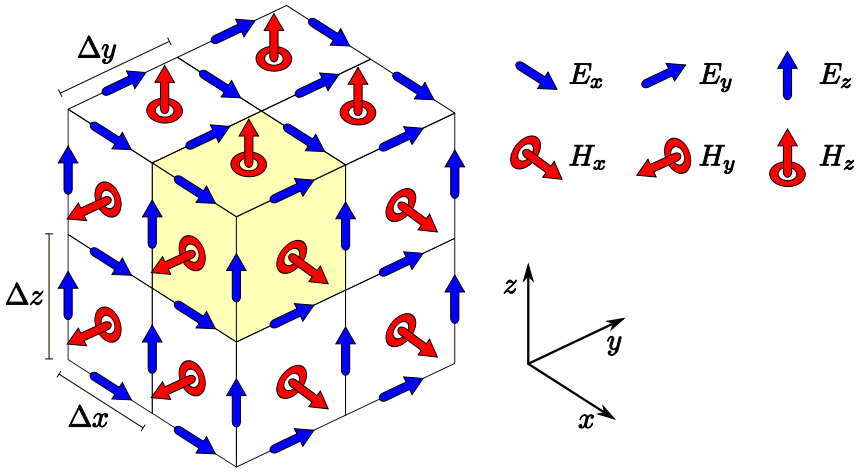


Figure A.1: *Spatial discretisation via the three-dimensional Yee grid. One cell is highlighted by a yellow shade. Blue arrows represent the components of the electric field, while red arrows (located on the centres of the faces of each cell) represent magnetic field components.*

A.2 Spatial Discretisation and the Yee Grid

We start discretising the curl equations by applying the finite-difference formula (A.2) to the spatial variables. To this end, we introduce a rectangular grid (Fig. A.1), where each cell has the same size $\Delta x \times \Delta y \times \Delta z$. In addition, we employ the shorthand notation

$$F(t)\Big|_{i,\,j,\,k} \equiv F(i\Delta x,\, j\Delta y,\, k\Delta z,\, t) \tag{A.3}$$

to discretise

$$\partial_t H_x(\vec{\mathrm{r}},\, t) = \mu^{-1}(\vec{\mathrm{r}}) \cdot \Big(\partial_z E_y(\vec{\mathrm{r}},\, t) - \partial_y E_z(\vec{\mathrm{r}},\, t)\Big). \tag{A.4}$$

Inserting finite-difference expression for the spatial derivatives ($h \to \Delta y$ and $h \to \Delta z$, respectively) yields

$$\begin{aligned} \partial_t H_x(t)\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}} = {} & \mu^{-1}\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}} \cdot \frac{1}{\Delta z}\left(E_y(t)\Big|_{i,\,j+\frac{1}{2},\,k+1} - E_z(t)\Big|_{i,\,j+\frac{1}{2},\,k}\right) \\ & - \mu^{-1}\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}} \cdot \frac{1}{\Delta y}\left(E_z(t)\Big|_{i,\,j+1,\,k+\frac{1}{2}} - E_z(t)\Big|_{i,\,j,\,k+\frac{1}{2}}\right). \end{aligned} \tag{A.5}$$

We observe that to calculate the temporal derivative of H_x we require the electric field at four different positions which form a "loop" around the magnetic field component. Repeating the procedure for the components of the electric field at these positions, we find that all electromagnetic field components must be staggered with respect to each other, and hence form the famous Yee grid depicted in Fig. A.1.

A.3 A Leapfrog Scheme for Time-Evolution

Equation (A.5) is what one might call the semi-discrete form of FDTD. In principle, this form could be combined with any time stepping algorithm such as Runge-Kutta (section 4.2) or Krylov subspace methods [47, 133]. In the case of FDTD, however, temporal derivatives are approximated using central differences as well. To this end, we discretise the time variable in terms of equidistant time steps Δt and define the convenient notation

$$F\Big|_{i,\,j,\,k}^{n} \equiv F(i\Delta x,\, j\Delta y,\, k\Delta z,\, n\Delta t). \tag{A.6}$$

Approximating the temporal derivative in Eq. (A.5) by Eq. (A.2) with $h \to \Delta t$ and solving for the most recent time step yields the update equation

$$\begin{aligned} H_x\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}}^{n+1} = \quad & H_x\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}}^{n} \\ & + \mu^{-1}\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}} \cdot \frac{\Delta t}{\Delta z}\left(E_y\Big|_{i,\,j+\frac{1}{2},\,k+1}^{n+\frac{1}{2}} - E_z\Big|_{i,\,j+\frac{1}{2},\,k}^{n+\frac{1}{2}} \right) \\ & - \mu^{-1}\Big|_{i,\,j+\frac{1}{2},\,k+\frac{1}{2}} \cdot \frac{\Delta t}{\Delta y}\left(E_z\Big|_{i,\,j+1,\,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z\Big|_{i,\,j,\,k+\frac{1}{2}}^{n+\frac{1}{2}} \right). \end{aligned} \tag{A.7}$$

Additional update equations are available for the remaining field components. Similarly to the spatial discretisation, introducing central differences staggers the electromagnetic fields in time. The magnetic field is required at integral time steps, while we need to know the electric field at *half*-integer time steps.

This allows for a very simple time stepping scheme. Starting from the magnetic field given at $t = 0$ and the electric field given at $t = \frac{1}{2}\Delta t$, we use the update equations to evolve the magnetic field to $t = \Delta t$.[3] Thereby, we can safely replace previous values of the magnetic field with updated ones, because the update equation (A.7) only features the previous value of the same field component at the same position as the updated one. Once we have finished updating the magnetic field, we employ this new data to update the electric field to $t = \frac{3}{2} \cdot \Delta t$. This process of updating both fields in turn is called a leapfrog scheme, which is repeatedly applied until the desired simulation time is reached.

A.4 Discussion

FDTD is a conceptually simple, yet very powerful algorithm. The update equations are straightforwardly derived and do not feature any linear algebra at all. Hence, time stepping is as explicit as it could possibly be and applying the update equations is very fast. Secondly, the algorithm is very memory-efficient. The time step procedure merely overwrites field values of the previous time step, and thus only one set of field values needs to be stored. Last but not least, a multitude of extensions to the basic algorithm is available [21].

[3] To maintain numerical stability, the time step Δt must obey the Courant-Friedrichs-Lewy (CFL) condition [21]. If $\Delta x = \Delta y = \Delta z$, the latter reads

$$\Delta t \leq \frac{\Delta x}{c\sqrt{3}},$$

where c is the speed of light.

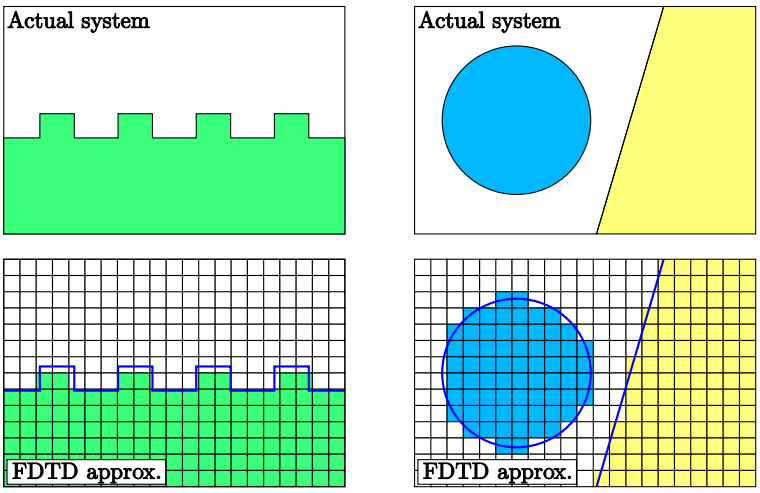


Figure A.2: *Staircase effect due to the regular Yee grid. The top panels show sketches of physical systems. The bottom panels show the respective representations of the geometries using the Yee grid, where the blue lines indicate the original geometry. As a result, aspect ratios of gratings change (bottom left), while curved and oblique interfaces show steps in the numerical representation (bottom left).*

Unfortunately, there are a number of shortcomings which—depending on the system of interest—may be of severe importance. The main issue is related to material interfaces (section 2.1.2). As soon as material interfaces are present in the system, the electromagnetic fields feature kinks and discontinuities. Consequently, the Taylor expansion (A.1) for smooth functions is no longer applicable. As a result, accuracy is lost. In many cases, it is not even clear which material parameters should be used within a given cell. The regular mesh cannot accurately model fine details, round objects, or oblique interfaces (see Fig. A.2). Instead, a staircase approximation of the geometry will be simulated. For highly sensitive systems, especially in nanoplasmonics, this often leads to strong distortions in both near fields and spectra. More recent efforts lead to remarkable progress regarding this issue, but a systematic treatment of interfaces remains challenging [22–24]. Finally, even if one uses central differences which improve on the second-order accuracy of Eq. (A.2), material interfaces intrinsically limit the order of accuracy to $(\Delta x)^2$. As a consequence, one often needs vast amounts of degrees of freedom to obtain sufficiently accurate results [38].

Acknowledgements

My thesis in its current form would not have been possible without the support and dedication of a number of people. First of all, I would like to thank *Prof. Dr. Kurt Busch*, who has presented me with the opportunity to work on an interesting numerical method in an exciting research area. His spirit, humour, and continued backing has greatly helped me to persevere. I want to express my gratitude towards *Prof. Dr. Martin Wegener*, who has kindly agreed to co-review this thesis. Furthermore, I am deeply indebted to *Prof. Dr. Thomas F. Krauss* for inviting me to St Andrews, Scotland, where I spent a brilliant six months in his group.

Many thanks must go to *Dr. Jens Niegemann*. We had countless fruitful discussions about all aspects of my work, be it implementation issues or numerical experiments. His experience was a great inspiration and his advice was always much appreciated.

Regarding the length of this thesis, proofreading was a tedious task. I am glad to have had patient, careful, and critical proofreaders in *Dr. Jens Niegemann*, *Dipl.-Phys. Christian Wolff*, *Dr. Sabine Essig*, *Dr. Patrick Mack*, *Dipl.-Phys. Thomas Zebrowski*, and *Dipl.-Phys. Benjamin Lutz*.

Furthermore, I would like to thank a number of students which have contributed in one way or the other to my work: *Dipl.-Phys. Christoper Prohm*, *M. Sc. Richard Diehl*, *Dipl.-Phys. Kai Stannigel*, *Dipl.-Phys. Michael Pieper*, and *Dipl.-Math. Elisabeth Blank*. Furthermore, I had numerous stimulating discussions with *Dipl.-Phys. Christian Wolff*.

I also benefitted from the cooperation with the group of *Prof. Dr. Martin Wegener*. In particular, I would like to thank *Dr. Nils Feth* and *Dipl.-Phys. Martin Husnik* for a collaboration both enjoyable and successful, even though I was not always as quick with my results as probably hoped for. Their experiments have presented an immediate purpose for my work. Being able to compare my simulations with their measurements was very rewarding.

While staying at St Andrews, many people have taught me, a humble theoretician (as some never got tired to mention), how to fabricate and characterise metallic nanostructures. Foremost, I have learned much from *Dr. Andrea Di Falco* and *Dr. Christopher Reardon*, who always took the patience to help me out with electron beam lithography and other experimental techniques. I had a productive time working with my fellow students *Abdul Shakoor* and *Martin Ploschner* on metamaterials. I fondly remember lunch, sports, and parties (international evening!) with *Marcel Spurny*, *Karl Welna*, *Shu Chen*, *Abdul Shakoor*, *Fengqiao Dong*, *Vikash Venkataramana*, *Isabella Rey*, *Chiara Platania*, *Chris Evi-Parker*, *Nadine Landolina*, *Daisuke Nakahara*, and my flatmate *Marco Grande*. I had a wonderful time with you guys.

A number of organisations have provided invaluable support for my research. The Karlsruhe School of Optics and Photonics (KSOP), directed by *Dr. Judith Elsner*, granted me a scholarship for the first year of my research project and provided funds for computers and travel expenses. Immaterial benefits such as management modules and seminars are also gratefully acknowledged.

The Karlsruhe House of Young Scientists (KHYS) awarded me a scholarship for my stay in Scot-

land, which greatly eased living in a foreign country. *Gaby Weick* proved to be an amiable and helpful contact.

Furthermore, I owe my gratitude to the Studienstiftung des Deutschen Volkes for granting me a scholarship for the longest time of my research project. Of all the people involved, I would like to thank *Prof. Dr. Uwe Hartmann* from Saarland University for a pleasant interview and his recommendation to accept my application, my liaison professor *Prof. Dr. Christoph Stiller* for his help and statements, and my contact person *Dr. Peter Antes* for reminding me of imminent deadlines and kind replies to my reports.

I could not close this thesis without mentioning the people at Photonics Group which have made working the last few years so worthwhile. Apart from our regular lunch at the cafeteria and the pizza seminar, I particularly enjoyed the Christmas parties, Antje and the Funky Photons, exhausting basketball matches, poker sessions, LAN parties, bring-crappy-movies nights, and—of course—barbecues.

Last but not least, I would like to thank my parents, my brother, and my girlfriend Melanie for their never-ending support through all these years and in particular the final stages of writing up this thesis.

Thank you.

Publications

Review Articles in Regular Peer-Reviewed Journals

- *Discontinuous Galerkin Methods in Nano-Photonics*; K. Busch, M. König, and J. Niegemann, Laser & Photonics Reviews, DOI: 10.1002/lpor.201000045 (in press, 2011)

Research Articles in Regular Peer-Reviewed Journals

- *Stretched-Coordinate PMLs for Maxwell's Equations in the Discontinuous Galerkin Time-Domain Method*; M. König, C. Prohm, K. Busch and J. Niegemann, Optics Express **19**, 4618–4631 (2011)
- *The Discontinuous Galerkin Time-Domain Method for Maxwells Equations with Anisotropic Materials*; M. König, K. Busch and J. Niegemann, Photonics and Nanostructures - Fundamentals and Applications **8**, 303–309 (2010)
- *Electromagnetic Interaction of Split-Ring Resonators: The Role of Separation and Relative Orientation*; N. Feth, M. König, M. Husnik, K. Stannigel, J. Niegemann, K. Busch, M. Wegener and S. Linden, Optics Express **18**, 6545–6554 (2010)
- *Discontinuous Galerkin Time-Domain Computations of Metallic Nanostructures*; K. Stannigel, M. König, J. Niegemann and K. Busch, Optics Express **17**, 14934–14947 (2009)
- *Higher-Order Time-Domain Methods for the Analysis of Nano-Photonic Systems*; J. Niegemann, M. König, K. Stannigel and K. Busch, Photonics and Nanostructures: Fundamentals and Applications **7**, 2–11 (2009)
- *Absolute Extinction Cross-Section of Individual Magnetic Split-Ring Resonators*; M. Husnik, M. W. Klein, N. Feth, M. König, J. Niegemann, K. Busch, S. Linden and M. Wegener, Nature Photonics **2**, 614–617 (2008)

Conference Proceedings

- *Simulations of Nano-Antennas with the Discontinuous Galerkin Time-Domain Method*; J. Niegemann, M. König, and K. Busch; Proceedings of SPIE **7713**, 77130Z (2010).
- *Using Curved Elements in the Discontinuous Galerkin Time-Domain Approach*; J. Niegemann, M. König, C. Prohm, R. Diehl, and K. Busch, AIP Conference Proceedings **1291**, 76–78 (2010)
- *Coupling between Split-Ring Resonators*; M. König, K. Stannigel, J. Niegemann, and K. Busch, AIP Conference Proceedings **1176**, 69–71 (2009)

- *Analysis of Metallic Nanostructures via a Discontinuous-Galerkin Time-Domain Approach*; K. Stannigel, M. König, J. Niegemann, and K. Busch, Proceedings of SPIE **7353**, 73530C (2009)
- *Efficient Modeling of Nonlinear Wave Propagation and Radiation Dynamics in Nano-Photonic Systems*; M. König, J. Niegemann, M. Pototschnig, L. Tkeshelashvili and K. Busch, Proceedings of SPIE **6775**, 67750D (2007)

Oral Presentations

- *Complementary Solvers: From DGTD to the Frequency-Domain*; 6th Workshop on Numerical Methods for Optical Nano Structures, Zurich, (2010)
- *Numerical Methods for Nanophotonics*; KSOP meets SICK AG, Waldkirch (2010)
- *Coupling Between Split-Ring Resonators*; 2nd International Workshop on Theoretical and Computational Nano-Photonics, Bad Honnef (2009)
- *Discontinuous Galerkin Time-Domain Analysis of Metallic Nanostructures*; 5th Workshop on Numerical Methods for Optical Nano Structures, Zurich, (2009)
- *The Discontinuous Galerkin Time-Domain Method for Nanophotonics*; DPG Frühjahrstagung, Dresden (2009)
- *Higher-Order Time-Domain Methods for the Analysis of Nano-Photonic Systems*; E-MRS Spring Meeting, Strasbourg (2008)
- *Time-Domain Simulations for Metallic Nano-Structures*; DPG Frühjahrstagung, Berlin (2008)

Bibliography

[1] K. Busch, G. von Freymann, S. Linden, S. Mingaleev, L. Tkeshelashvili, and M. Wegener. Periodic nanostructures for photonics. *Phys. Rep.-Rev. Sec. Phys. Lett.*, **444**(3–6): 101–202 (Jun. 2007). doi:10.1016/j.physrep.2007.02.011.

[2] J. Joannopoulos, S. Johnson, J. Winn, and R. Meade. *Photonic crystals: Molding the flow of light* (Princeton Univerity Press, Princeton, 2008), 2nd edition.

[3] K. Sakoda. *Optical properties of photonic crystals* (Springer, Berlin, 2005), 2nd edition.

[4] J.-M. Lourtioz, H. Benisty, V. Berger, J.-M. Gérard, D. Maystre, A. Tchelnokov, and D. Pagnoux. *Photonic crystals: Towards nanoscale photonic devices* (Springer, Berlin, 2008), 2nd edition.

[5] G. Dolling, M. Wegener, C. Soukoulis, and S. Linden. Negative-index metamaterial at 780 nm wavelength. *Opt. Lett.*, **32**(1): 53–55 (Jan. 2007). doi:10.1364/OL.32.000053.

[6] J. B. Pendry. Negative refraction makes a perfect lens. *Phys. Rev. Lett.*, **85**(18): 3966–3969 (Oct. 2000). doi:10.1103/PhysRevLett.85.3966.

[7] W. Cai, U. Chettiar, A. Kildishev, and V. Shalaev. Optical cloaking with metamaterials. *Nature Photon.*, **1**(4): 224–227 (Apr. 2007). doi:10.1038/nphoton.2007.28.

[8] M. Stewart, C. Anderton, L. Thompson, J. Maria, S. Gray, J. Rogers, and R. Nuzzo. Nanostructured plasmonic sensors. *Chem. Rev.*, **108**(2): 494–521 (Jan. 2008). doi:10.1021/cr068126n.

[9] A. Grigorenko, N. Roberts, M. Dickinson, and Y. Zhang. Nanometric optical tweezers based on nanostructured substrates. *Nature Photon.*, **2**(6): 365–370 (Jun. 2008). doi:10.1038/nphoton.2008.78.

[10] B. Hecht, B. Sick, U. Wild, V. Deckert, R. Zenobi, O. Martin, and D. Pohl. Scanning near-field optical microscopy with aperture probes: Fundamentals and applications. *J. Chem. Phys.*, **112**(18): 7761–7774 (2000). doi:10.1063/1.481382.

[11] A. Armani, R. Kulkarni, S. Fraser, R. Flagan, and K. Vahala. Label-free, single-molecule detection with optical microcavities. *Science*, **317**(5839): 783–787 (Aug. 2007). doi:10.1126/science.1145002.

[12] J. Hu, X. Sun, A. Argawal, and L. Kimerling. Design guidelines for optical resonator biochemical sensors. *J. Opt. Soc. Am. B*, **26**(5): 1032–1041 (May 2009). doi:10.1364/JOSAB.26.001032.

[13] Y. Sun and X. Fan. Optical ring resonators for biochemical and chemical sensing. *Anal. Bioanal. Chem.*, **399**(1): 205–211 (Jan. 2011). doi:10.1007/s00216-010-4237-z.

[14] C. Hafner. Boundary methods for optical nano structures. *phys. stat. sol. (b)*, **244**(10): 3435–3447 (Oct. 2007). doi:10.1002/pssb.200743165.

[15] K. Busch, C. Blum, A. Graham, D. Hermann, M. Köhl, P. Mack, and C. Wolff. The photonic Wannier function approach to photonic crystal simulations: Status and perspectives. *J. Mod. Opt.*, **58**(5&6): 365–383 (2010). doi:10.1080/09500340.2010.526256.

[16] C. Blum, C. Wolff, and K. Busch. Photonic-crystal time-domain simulations using Wannier functions. *Opt. Lett.*, **36**(2): 307–309 (Jan. 2011). doi:10.1364/OL.36.000307.

[17] L. Li. Formulation and comparison of two recursive matrix algorithms for modeling layered diffraction gratings. *J. Opt. Soc. Am. A*, **13**(5): 1024–1035 (May 1996). doi: 10.1364/JOSAA.13.001024.

[18] L. Li. New formulation of the Fourier modal method for crossed surface-relief gratings. *J. Opt. Soc. Am. A*, **14**(10): 2758–2767 (Oct. 1997). doi:10.1364/JOSAA.14.002758.

[19] S. Essig and K. Busch. Generation of adaptive coordinates and their use in the Fourier modal method. *Opt. Express*, **18**(22): 23258–23274 (Oct. 2010). doi:10.1364/OE.18.023258.

[20] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propagat.*, **14**(3): 302–307 (May 1966). doi: 10.1109/TAP.1966.1138693.

[21] A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method* (Artech House, Boston, 2005), 3rd edition.

[22] A. Farjadpour, D.R., A. Rodriguez, M. Ibanescu, P. Bermel, J. Joannopoulos, S. Johnson, and G. Burr. Improving accuracy by subpixel smoothing in the finite-difference time domain. *Opt. Lett.*, **31**(20): 2972–2974 (Oct. 2006). doi:10.1364/OL.31.002972.

[23] A. Deinega and I. Valuev. Subpixel smoothing for conductive and dispersive media in the finite-difference time-domain method. *Opt. Lett.*, **32**(23): 3429–3431 (Dec. 2007). doi: 10.1364/OL.32.003429.

[24] A. Oskooi, C. Kottke, and S. Johnson. Accurate finite-difference time-domain simulation of anisotropic media by subpixel smoothing. *Opt. Lett.*, **34**(18): 2778–2780 (Sep. 2009). doi: 10.1364/OL.34.002778.

[25] J. Jin. *The Finite Element Method in Electromagnetics* (John Wiley & Sons, New York, 2002), 2nd edition.

[26] P. Monk. *Finite Element Methods for Maxwell's Equations* (Oxford Science Publications, Oxford, 2003).

[27] J. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods—Algorithms, Analysis, and Applications* (Springer, Berlin, 2007).

[28] J. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations. *J. Comput. Phys.*, **181**(1): 186–221 (2002). doi: 10.1006/jcph.2002.7118.

[29] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, **141**(2): 199–224 (1998). doi: 10.1006/jcph.1998.5892.

[30] Y. Xu and C. Shu. Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. *Commun. Comput. Phys.*, **7**(1): 1–46 (Jan. 2010). doi: doi:10.4208/cicp.2009.09.023.

[31] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory (1973).

[32] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case. *Math. Comput.*, **54**(190): 545–581 (Apr. 1990). doi:10.1090/S0025-5718-1990-1010597-0.

[33] Y. Shi and C.-H. Liang. Simulations of the left-handed medium using discontinuous Galerkin method based on the hybrid domains. *Prog. Electromagn. Res.*, **63**: 171–191 (2006). doi: 10.2528/PIER06050803.

[34] J. Niegemann, M. König, K. Stannigel, and K. Busch. Higher-order time-domain methods for the analysis of nano-photonic systems. *Photonics Nanostruct. Fundam. Appl.*, **7**(1): 2–11 (2009). doi:10.1016/j.photonics.2008.08.006.

[35] A. Hille, R. Kullock, S. Grafström, and L. Eng. Improving nano-optical simulations through curved elements implemented within the discontinuous Galerkin method computational. *J. Comput. Theor. Nanosci.*, **7**: 1581–1586 (Aug. 2010). doi:10.1166/jctn.2010.1522.

[36] G. Tang, R. Panetta, and P. Yang. Application of a discontinuous Galerkin time domain method to simulation of optical properties of dielectric particles. *Appl. Opt.*, **49**(15): 2827–2840 (May 2010). doi:10.1364/AO.49.002827.

[37] R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics (Cambridge University Press, Cambridge, 2002).

[38] J. Niegemann, W. Pernice, and K. Busch. Simulation of optical resonators using DGTD and FDTD. *J. Opt. A: Pure Appl. Opt.*, **11**(11): 114015 (2009).

[39] M. König, C. Prohm, K. Busch, and J. Niegemann. Stretched-coordinate PMLs for Maxwell's equations in the discontinuous Galerkin time-domain method. *Opt. Express*, **19**(5): 4618–4631 (Feb. 2011).

[40] M. König, K. Busch, and J. Niegemann. The discontinuous Galerkin time-domain method for Maxwell's equations with anisotropic materials. *Photonics Nanostruct. Fundam. Appl.*, **8**(4): 303–309 (Sep. 2010).

[41] K. Busch, M. König, and J. Niegemann. Discontinuous Galerkin methods in nanophotonics. *Laser Photon. Rev.* (2011). doi:10.1002/lpor.201000045. (in press).

[42] J. Maxwell. A dynamical theory of the electromagnetic field. *Phil. Trans. R. Soc. Lond.*, **155**: 459–512 (Jan. 1865). doi:10.1098/rstl.1865.0008.

[43] J. Jackson. *Classical Electrodynamics* (Wiley, New York, 1999), 3rd edition.

[44] M. Fabrizio and A. Morro. *Electromagnetism of Continuous Media* (Oxford University Press, Oxford, 2003).

[45] P. Johnson and R. Christy. Optical constants of the noble metals. *Phys. Rev. B*, **6**(12): 4370–4379 (Dec. 1972). doi:10.1103/PhysRevB.6.4370.

[46] A. Vial, A.-S. Grimault, D. Macías, D. Barchiesi, and M. de la Chapelle. Improved analytical fit of gold dispersion: Application to the modeling of extinction spectra with a finite-difference time-domain method. *Phys. Rev. B*, **71**(8): 085416 (Feb. 2005). doi: 10.1103/PhysRevB.71.085416.

[47] M. König. *Time-Domain Simulations for Metallic Nano-Structures*. Master's thesis, Institut für Theoretische Festkörperphysik, Universität Karlsruhe (TH) (Nov. 2007).

[48] M. Born and E. Wolf. *Principles of Optics* (Cambridge University Press, Cambridge, 2002), 7th edition.

[49] N. Ashcroft and N. Mermin. *Festkörperphysik* (Oldenbourg Verlang, München, 2005), 2nd edition.

[50] C. Bohren and D. Huffman. *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, Inc., New York, 1983).

[51] M. Ordal, L. Long, R. Bell, S. Bell, R. Bell, R. A. Jr., and C. Ward. Optical properties of the metals Al, Co, Cu, Au, Fe, Pb, Ni, Pd, Pt, Ag, Ti, and W in the infrared and far infrared. *Appl. Opt.*, **22**(7): 1099–1119 (Apr. 1983).

[52] H. van de Hulst. *Light Scattering by Small Particles* (Dover Publications, Inc., New York, 1981).

[53] G. Cohen, X. Ferrieres, and S. Pernet. A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell's equations in time domain. *J. Comput. Phys.*, **217**(2): 340–363 (2006). doi:10.1016/j.jcp.2006.01.004.

[54] J. Niegemann. *Higher-Order Methods for Solving Maxwell's Equations in the Time-Domain*. Ph.D. thesis, Institut für Theoretische Festkörperphysik, Universitä Karlsruhe (TH) (Jan. 2009).

[55] J. Schöberl. NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, **1**(1): 41–52 (Jul. 1997). doi:10.1007/s007910050004.
URL `http://sourceforge.net/projects/netgen-mesher/`

[56] H. Si. Tetgen.
URL `http://tetgen.berlios.de`

[57] C. Geuzaine and J.-F. Remacle. Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.*, **79**(11): 1309–1331 (Sep. 2009). doi:10.1002/nme.2579.

[58] T. Warburton. An explicit construction of interpolation nodes on the simplex. *J. Eng. Math.*, **56**(3): 247–262 (Nov. 2006). doi:10.1007/s10665-006-9086-6.

[59] R. Pasquetti and F. Rapetti. Spectral element methods on unstructured meshes: Which interpolation points? *Numer. Algorithms*, **55**(2–3): 349–366 (Nov. 2010). doi:10.1007/s11075-010-9390-0.

[60] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comput.*, **6**(4): 345–390 (1991). doi:10.1007/BF01060030.

[61] Intel® 64 and IA-32 architectures software developer's manual (Apr. 2011).
URL `http://www.intel.com/products/processor/manuals/`

[62] A. Fog. The microarchitecture of Intel, AMD and VIA CPUs (Jan. 2011).
URL `http://www.agner.org/optimize/`

[63] A. Klöckner, T. Warburton, J. Bridge, and J. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *J. Comput. Phys.*, **228**(21): 7863–7882 (2009). doi:10.1016/j.jcp.2009.06.041.

[64] P. Mandel. *Theoretical problems in cavity nonlinear optics* (Cambridge University Press, Cambridge, 2005).

[65] M. Carpenter and C. Kennedy. Fourth-order 2N-storage Runge-Kutta schemes. Technical Report 109112, NASA Technical Memorandum (Jun. 1994).

[66] J. Williamson. Low-storage Runge-Kutta schemes. *J. Comput. Phys.*, **35**(1): 48–56 (Mar. 1980). doi:10.1016/0021-9991(80)90033-9.

[67] K. Stannigel, M. König, J. Niegemann, and K. Busch. Discontinuous Galerkin time-domain computations of metallic nanostructures. *Opt. Express*, **17**(17): 14934–14947 (Aug. 2009).

[68] R. Diehl, K. Busch, and J. Niegemann. Comparison of low-storage Runge-Kutta schemes for discontinuous Galerkin time-domain simulations of Maxwell's equations. *J. Comput. Theor. Nanosci.*, **7**(8): 1572–1580 (Aug. 2010). doi:10.1166/jctn.2010.1521.

[69] J. Niegemann, R. Diehl, and K. Busch. Efficient low-storage Runge-Kutta schemes with optimized stability regions. *J. Comput. Phys.* (2011). (submitted).

[70] R. Diehl. *Advanced Time Stepping Methods for Discontinuous Galerkin Time-Domain Simulators*. Master's thesis, Institut für Theoretische Festkörperphysik, Universität Karlsruhe (TH) (Oct. 2009).

[71] H. Kreiss and L. Wu. On the stability definition of difference approximations for the initial boundary value problem. *Appl. Numer. Math.*, **12**(1–3): 213–227 (1993). doi:10.1016/0168-9274(93)90119-C.

[72] J. Niegemann and K. Busch. Time-stepping and convergence characteristics of the discontinuous Galerkin time-domain approach for the Maxwell equations. In *AIP Conf. Proc.*, volume 1147, pp. 22–29 (2009). doi:10.1063/1.3183437.

[73] T. Toulorge and W. Desmet. CFL conditions for Runge-Kutta discontinuous Galerkin methods on triangular grids. *J. Comput. Phys.*, **230**(12): 4657–4678 (Jun. 2011). doi:10.1016/j.jcp.2011.02.040.

[74] R. Lehoucq and D. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, **17**(4): 789–821 (1996). doi:10.1137/S0895479895281484.
URL `http://www.caam.rice.edu/software/ARPACK/`

[75] R. Spiteri and S. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J. Numerical Analysis*, **40**(2): 469–491 (2002). doi:10.1137/S0036142901389025.

[76] D. Sármány, F. Izsák, and J. van der Vegt. Optimal penalty parameters for symmetric discontinuous Galerkin discretisations of the time-harmonic Maxwell equations. *J. Sci. Comput.*, **44**(3): 219–254 (Sep. 2010). doi:10.1007/s10915-010-9366-1.

[77] R. Lehoucq, D. Sorensen, and C. Yang. ARPACK users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods (Oct. 1997).
URL `http://www.caam.rice.edu/software/ARPACK/`

[78] B. Cockburn, F. Li, and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *J. Comput. Phys.*, **194**(2): 588–610 (2004). doi:10.1016/j.jcp.2003.09.007.

[79] Y. Saad. *Iterative Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2003), 2nd edition.

[80] T. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, **30**: 165–195 (Jun. 2004). doi:10.1145/992200.992205.

[81] O. Schenk and K. Gärtner. On fast factorization pivoting methods for sparse symmetric indefinite systems. *Elec. Trans. Numer. Anal.*, **23**: 158–179 (2006).

[82] G. Sleijpen and D. Fokkema. BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum. *Elec. Trans. Numer. Anal.*, **1**: 11–32 (Sep. 1993).

[83] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**(3): 856–869 (Jul. 1986).

[84] J. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain (Aug. 1994).

[85] K. Hiremath, J. Niegemann, and K. Busch. Analysis of light propagation in slotted resonator based systems via coupled-mode theory. *Opt. Express*, **19**(9): 8641–8655 (Apr. 2011). doi: 10.1364/OE.19.008641.

[86] K. Okamoto. *Fundamentals of Optical Waveguides* (Academic Press, San Diego, 2000).

[87] L. Novotny and B. Hecht. *Principles of Nano-Optics* (Cambridge University Press, New York, 2006).

[88] J. Kong. *Electromagnetic Wave Theory* (EMW Publishing, Cambridge, 2005).

[89] M. Aeschlimann, M. Bauer, D. Bayer, T. Brixner, F. G. de Abajo, W. Pfeiffer, M. Rohmer, C. Spindler, and F. Steeb. Adaptive subwavelength control of nano-optical fields. *Nature*, **446**: 301–304 (Mar. 2007).

[90] G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Trans. Electromagn. Compat.*, **23**(4): 377–382 (Nov. 1981). doi:10.1109/TEMC.1981.303970.

[91] M. I. Stockman, S. V. Faleev, and D. J. Bergman. Coherent control of femtosecond energy localization in nanosystems. *Phys. Rev. Lett.*, **88**(6): 067402 (Jan. 2002). doi: 10.1103/PhysRevLett.88.067402.

[92] X. Li and M. Stockman. Highly efficient spatiotemporal coherent control in nanoplasmonics on a nanometer-femtosecond scale by time reversal. *Phys. Rev. B*, **77**(19): 195109 (May 2008). doi:10.1103/PhysRevB.77.195109.

[93] H. Fahs. Discontinuous Galerkin method for time-domain electromagnetics on curvilinear domains. *Appl. Math. Sci. (Ruse)*, **4**(19): 943–958 (2010).

[94] J. Niegemann, M. König, C. Prohm, R. Diehl, and K. Busch. Using curved elements in the discontinuous Galerkin time-domain approach. In D. N. Chigrin, editor, *AIP Conf. Proc.*, volume 1291, pp. 76–78 (2010). doi:10.1063/1.3506136.

[95] L.-B. Zhang, T. Cui, and H. Liu. A set of symmetric quadrature rules on triangles and tetrahedra. *J. Comput. Math.*, **27**(1): 89–96 (2009).

[96] T. Warburton. A low storage curvilinear discontinuous Galerkin time-domain method for electromagnetics. In *URSI International Symposium on Electrogmagnetic Theory*, pp. 996–999 (2010). doi:10.1109/URSI-EMTS.2010.5637392.

[97] J. Cooley and J. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, **19**(90): 297–301 (1965).

[98] M. Frigo and S. Johnson. The design and implementation of FFTW3. *Proc. IEEE*, **93**(2): 216–231 (Feb. 2005). doi:10.1109/JPROC.2004.840301.

[99] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comp.*, **14**(6): 1368–1393 (Nov. 1993).

[100] T. Hagstrom and S. Lau. Radiation boundary conditions for Maxwell's equations: A review of accurate time-domain formulations. *J. Comp. Math.*, **25**(3): 305–336 (2007).

[101] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.*, **114**(2): 185–200 (1994). doi:10.1006/jcph.1994.1159.

[102] Z. Sacks, D. Kingsland, R. Lee, and J.-F. Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *IEEE Trans. Antennas Propagat.*, **43**(12): 1460–1463 (Dec. 1995). doi:10.1109/8.477075.

[103] W. C. Chew and W. H. Weedon. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. *Microwave Opt. Technol. Lett.*, **7**(13): 599–604 (1994). doi:10.1002/mop.4650071304.

[104] J.-P. Bérenger. *Perfectly matched layer (PML) for computational electromagnetics* (Morgan & Claypool Publishers, San Rafael, CA, 2007).

[105] T. Lu, P. Zhang, and W. Cai. Discontinuous Galerkin methods for dispersive and lossy Maxwell's equations and PML boundary conditions. *J. Comput. Phys.*, **200**(2): 549–580 (2004). doi:10.1016/j.jcp.2004.02.022.

[106] J. Li and J. Pendry. Hiding under the carpet: A new strategy for cloaking. *Phys. Rev. Lett.*, **101**(20): 203901 (Nov. 2008). doi:10.1103/PhysRevLett.101.203901.

[107] U. Leonhardt and T. Tyc. Broadband invisibility by non-Euclidean cloaking. *Science*, **323**(5910): 110–112 (Jan. 2009).

[108] S. Chun and J. Hesthaven. High-order accurate thin layer approximations for time-domain electromagnetics. Part I: General metal backed coatings. *J. Comput. Appl. Math.*, **231**(2): 598–611 (2009).

[109] J. Pendry, A. Holden, D. Robbins, and W. Stewart. Magnetism from conductors and enhanced nonlinear phenomena. *IEEE Trans. Microwave Theory Tech.*, **47**(11): 2075–2084 (Nov. 1999). doi:10.1109/22.798002.

[110] S. Linden, C. Enkrich, M. Wegener, J. Zhou, T. Koschny, and C. Soukoulis. Magnetic response of metamaterials at 100 terahertz. *Science*, **306**: 1351–1353 (Nov. 2004).

[111] M. Husnik, M. Klein, N. Feth, M. König, J. Niegemann, K. Busch, S. Linden, and M. Wegener. Absolute extinction cross-section of individual magnetic split-ring resonators. *Nature Photon.*, **2**: 614–617 (Oct. 2008).

[112] N. Feth, M. König, M. Husnik, K. Stannigel, J. Niegemann, K. Busch, M. Wegener, and S. Linden. Electromagnetic interaction of split-ring resonators: The role of separation and relative orientation. *Opt. Express*, **18**(7): 6545–6554 (Mar. 2010).

[113] N.-A. Feth. *Nonlinear Optics of Planar Metamaterial Arrays and Spectroscopy of Individual "Photonic Atoms"*. Ph.D. thesis, Institut für Angewandte Physik, Karlsruhe Institute of Technology (KIT) (Jan. 2010).

[114] C. Rockstuhl, T. Zentgraf, H. Guo, N. Liu, C. Etrich, I. Loa, K. Syassen, J. Kuhl, F. Lederer, and H. Giessen. Resonances of split-ring resonator metamaterials in the near infrared. *Appl. Phys. B*, **84**: 219–227 (2006). doi:10.1007/s00340-006-2205-2.

[115] J. Z. T., Koschny, M. Kafesaki, E. Economou, J. Pendry, and C. Soukoulis. Saturation of the magnetic response of split-ring resonators at optical frequencies. *Phys. Rev. Lett.*, **95**(22): 223902 (Nov. 2005). doi:10.1103/PhysRevLett.95.223902.

[116] M. Klein, C. Enkrich, M. Wegener, C. Soukoulis, and S. Linden. Single-slit split-ring resonators at optical frequencies: Limits of size scaling. *Opt. Lett.*, **31**(9): 1259–1261 (May 2006). doi:10.1364/OL.31.001259.

[117] G. Dolling, M. Wegener, C. Soukoulis, and S. Linden. Negative-index metamaterial at 780 nm wavelength. *Opt. Lett.*, **32**(1): 53–55 (Jan. 2007). doi:10.1364/OL.32.000053.

[118] C. Enkrich, F. Pérez-Willard, D. Gerthsen, J. Zhou, T. Koschny, C. Soukoulis, M. Wegener, and S. Linden. Focused-ion-beam nanofabrication of near-infrared magnetic metamaterials. *Adv. Mater.*, **17**(21): 2547–2549 (2005). doi:10.1002/adma.200500804.

[119] M. Decker, N. Feth, C. Soukoulis, S. Linden, and M. Wegener. Retarded long-range interaction in split-ring-resonator square arrays (2011).

[120] R. Singh, C. Rockstuhl, and W. Zhang. Strong influence of packing density in terahertz metamaterials. *Appl. Phys. Lett.*, **97**(24): 241108 (2010). doi:10.1063/1.3525169.

[121] C. Dahmen, B. Schmidt, and G. von Plessen. Radiation damping in metal nanoparticle pairs. *Nano Lett.*, **7**(2): 318–322 (2007). doi:10.1021/nl062377u.

[122] P. Olk, J. Renger, M. Wenzel, and L. Eng. Distance dependent spectral tuning of two coupled metal nanoparticles. *Nano Lett.*, **8**(4): 1174–1178 (2008). doi:10.1021/nl080044m.

[123] H. Dutton. *Understanding Optical Communications* (Prentice Hall PTR, New Jersey, 1998).

[124] M. Grote, A. Schneebeli, and D. Schötzau. Interior penalty discontinuous Galerkin method for Maxwell's equations: optimal L^2-norm error estimates. *IMA J. Numer. Anal.*, **28**(3): 440–468 (2008). doi:10.1093/imanum/drm038.

[125] S. Piperno. Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems. *ESAIM-Math. Model. Numer. Anal.-Model. Math. Anal. Numer.*, **40**(5): 815–841 (2006).

[126] E. Montseny, S. Pernet, X. Ferriéres, and G. Cohen. Dissipative terms and local time-stepping improvements in a spatial high order discontinuous Galerkin scheme for the time-domain Maxwell's equations. *J. Comput. Phys.*, **227**(14): 6795–6820 (2008). doi: 10.1016/j.jcp.2008.03.032.

[127] C. Fumeaux, D. Baumann, P. Leuchtmann, and R. Vahldieck. A generalized local time-step scheme for efficient FVTD simulations in strongly inhomogeneous meshes. *IEEE Trans. Microwave Theory Tech.*, **52**(3): 1067–1076 (Mar. 2004). doi:10.1109/TMTT.2004.823595.

[128] M. Dumbser, M. Käser, and E. Toro. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes — V. Local time stepping and p-adaptivity. *Geophys. J. Int.*, **171**(2): 695–717 (2007). doi:10.1111/j.1365-246X.2007.03427.x.

[129] S. Schomann, N. Gödel, T. Warburton, and M. Clemens. Local timestepping techniques using Taylor expansion for modeling electromagnetic wave propagation with discontinuous Galerkin-FEM. *IEEE Trans. Magn.*, **46**(8): 3504–3507 (Aug. 2010). doi: 10.1109/TMAG.2010.2043656.

[130] N. Gödel, S. Schomann, T. Warburton, and M. Clemens. GPU accelerated Adams-Bashforth multirate discontinuous Galerkin FEM simulation of high-frequency electromagnetic fields. *IEEE Trans. Magn.*, **46**(8): 2735–2738 (Aug. 2010). doi:10.1109/TMAG.2010.2043655.

[131] V. Dolean, H. Fahs, L. Fezoui, and S. Lanteri. Locally implicit discontinuous Galerkin method for time domain electromagnetics. *J. Comput. Phys.*, **229**(2): 512–526 (2010). doi: 10.1016/j.jcp.2009.09.038.

[132] J. E. Sipe, V. So, M. Fukui, and G. Stegeman. Analysis of second-harmonic generation at metal surfaces. *Phys. Rev. B*, **21**(10): 4389–4402 (May 1980).

[133] K. Busch, J. Niegemann, M. Pototschnig, and L. Tkeshelashvili. A Krylov-subspace based solver for the linear and nonlinear Maxwell equations. *Phys. Status Solidi B-Basic Solid State Phys.*, **244**(10): 3479–3496 (2007). doi:10.1002/pssb.200743290.

[134] T. Pratchett. *Interesting Times* (Transworld Publishers, London, 1996).

++?????++ Out of Cheese Error. Redo From Start. [134]