Analysis of Metallic Nanostructures by a Discontinuous Galerkin Time-Domain Maxwell Solver on Graphics Processing Units

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

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

genehmigte

DISSERTATION

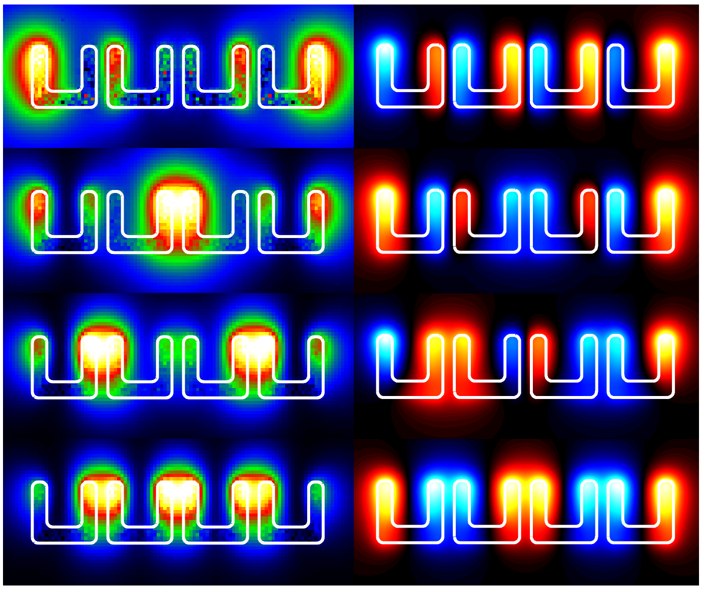
von

M. Sc. Richard Timothy Helmut Diehl
aus München

Tag der mündlichen Prüfung: 14.12.2012
Referent: Prof. Dr. K. Busch
Korreferent: Prof. Dr. M. Wegener

Fakultät für Physik
Institut für Theoretische Festkörperphysik

Analysis of Metallic Nanostructures by a Discontinuous Galerkin Time-Domain Maxwell Solver on Graphics Processing Units



PhD Thesis

by

M. Sc. Richard Timothy Helmut Diehl

14.12.2012

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

Contents

1

Chapter 1

Introduction

In this thesis, we examine the optical properties of metallic nanostructures with typical feature sizes of the order of visible light. The interaction of light with such structures can be accurately described by classical electrodynamics. Thus, for the analysis of metallic nanostructures within this thesis, we will employ Maxwell's equations [1] to model the physical behavior of the electromagnetic fields. On these length-scales, light does not respond to the atomic-scale variation induced by the individual nuclei and electrons of matter, and, hence, a rigorous treatment in terms of quantum electrodynamics is not required.

1.1 Plasmonic Nanostructures

Plasmonics refers to a relatively young field of optical science in which the properties of nano-scale metallic structures and their interaction with light is investigated. Metal structures support localized oscillations of free electrons, so called plasmons. These plasmons couple to the incident electromagnetic and can be resonantly enhanced which results in a potentially strong enhancement of the local field. Such strong local fields are employed in a variety of methods and techniques, e. g., surface enhanced Raman spectroscopy, surface enhanced infrared absorption, or nonlinear frequency conversion (cf. Refs. [2, 3] and references therein).

Besides local enhancement, the field tends to be strongly localized. In other words, the field strongly varies over short distances which renders plasmonic structures suitable devices for sensing. For example, the detection of single molecule events has been reported [4]. Scanning near-field optical microscopy (SNOM) is a technique that exploits the strong localization. This leads to a spatial resolution below the diffraction limit [5–7].

Remarkable effects occur when individual metallic structures are periodically arranged. Among these, the discovery of an effective magnetic response from non-magnetic conducting media by Pendry *et al.* [8] is outstanding. Based on their findings, negative refraction has been demonstrated for several different arrangements, e. g., split-ring resonators [9] (cf. Fig. 1.1), net-structures [10], and nanostrips [11]. Such behavior has not been observed in nature and, hence, these materials, fabricated from periodically arranged metallic building-blocks[1], are suitably coined "metamaterials" [10]. In order to obtain these exceptional properties, the extent of the metaatoms must be (significantly) smaller than the wavelength of the incident radiation.

[1] We also refer to the building-blocks by "metaatoms".

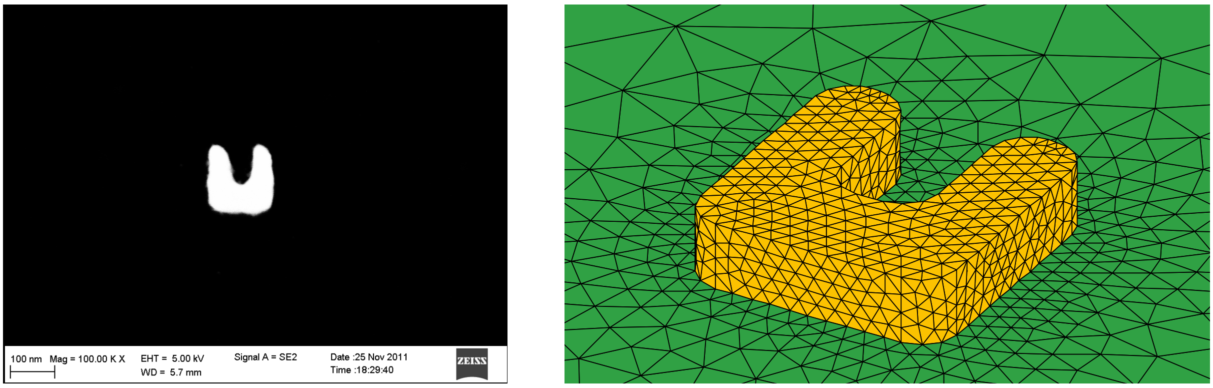


Figure 1.1: *Scanning electron microscope image (left panel, courtesy of M.Husnik and M.Wegener) and tetrahedral mesh (right panel) of an individual split-ring resonator. The side-length of the split-ring resonator is approximately 140 nm.*

The properties of a metamaterial are the result of the *collective* response of the individual metaatoms. In order to understand the underlying coupling mechanisms, the electromagnetic properties of an *individual* metaatom must be characterized. Among a large amount of characterization methods, two sophisticated methods are considered in the context of this thesis:

Recently, Husnik *et al.* [12] demonstrated the simultaneous *and* quantitative measurement of the scattering and the extinction cross section of individual nanostructures. Their novel setup combines a spatial modulation technique with a common-path interferometer. Since this newly developed technique cannot be compared to any other measurement, we supported the development by numerical investigations (Chap. 6).

The second characterization method, which we consider in this thesis, is electron energy loss spectroscopy (EELS). Due to a spatial resolution on the nm-scale and a high energy resolution (sub-eV), EELS gained considerable attention throughout the last years. In this thesis, both individual and coupled systems of split-ring resonators are treated numerically and are compared to a series of other (numerical) characterization methods (Chap. 7). Parts of this analysis along with experimental data[2] will be published soon [13].

1.2 Numerical Simulation's Contribution to Research

Albeit a remarkable progress in fabrication and measurement techniques, the analysis of nanostructures is a costly and time-consuming task. In particular, extensive parameter studies are challenging in terms of time-consumption and reproducibility. A parameter study of split-ring resonators was numerically performed by use of the discontinuous Galerkin time-domain (DGTD) method in Ref. [14]. Therein, the dependence of the scattering, the absorption, and the extinction cross section on numerous geometry parameters (e. g., height, thickness, etc.) was examined. This example demonstrates that numerical simulation methods are comparatively cheap, flexible, and reliable tools.

Besides these aspects, numerical simulations provide physical quantities which are inaccessible in the experiment. Very naturally, the time-evolution of the electromagnetic field distribution can be observed by employing time-domain simulations [15]. Due to the high frequencies in the optical regime, such

[2] The experimental work is accomplished by F. von Cube and S. Linden.

an analysis cannot be easily obtained by experimental techniques. For a particular system, a *single* time-domain simulation suffices to obtain a spectrally broad response when employing a short pulse for excitation.

A variety of numerical methods exist which handle Maxwell's equations in both the time- and the frequency-domain. Undoubtedly, the most famous one is the finite-different (FD) method and, in particular, the finite-difference time-domain (FDTD) method. In its basic form, the FD method lays down an structured grid in space and approximates spatial derivatives by differences across neighboring grid points [16]. Being conceptually simple and intuitive, the method suffers from a low order of accuracy when dealing with material interfaces [17]. Bound to a necessarily structured grid, the FD method leads to unsatisfactory results when dealing with round structures. In particular, the grid may induce spurious field enhancement when simulating metallic objects.

Another well-established numerical method is the finite-element method (FEM) [18, 19]. In contrast to FD, unstructured grids are supported, i. e., the geometry of the physical system is modeled by elements of different sizes and shapes. An unstructured mesh of a split-ring resonator consisting of tetrahedra is depicted in the right panel of Fig. 1.1. Obviously, the geometry can be accurately modeled by an unstructured tetrahedral mesh.

In the FEM, the numerical solution of the physical problem is approximated by high-order basis functions. Both the size of the grid elements and the order of approximation can be employed to control the accuracy of the solution which is known as *hp*-adaptivity. The discretization according to the FEM method results in a (potentially large) implicit semi-discrete set of equations. For time-domain simulations, the solution of the associated systems of linear equations can yield to an inefficient numerical scheme.

Thus, our method of choice is the discontinuous Galerkin (DG) method which combines the major advantages of both the finite-difference and the finite-element method: The representation of the geometry by an unstructured grid, high-order basis functions, and an efficient applicability in the time-domain. In contrast to the FEM, *element-local* basis functions are employed. This leads to an implicit method which *can* be efficiently inverted. Furthermore, this element-local ansatz is ideally suited for execution on massively parallel devices.

Originally, the DG method was proposed in 1973 in the context of neutron-transport [20]. A major step towards its application in the field of classical electrodynamics was done in 2002 when Hesthaven and Warburton published a nodal scheme and proved convergence of the numerical flux [21]. Since then, the DGTD method has attracted much attention and has been applied to a variety of analyses, e. g., for optical spectroscopy [12, 22, 23] and electron energy loss spectroscopy [13, 24, 25]. These analyses techniques are, however, computationally demanding which emphasizes the need for accelerated approaches, e. g., graphics processing units.

1.3 Graphics Processing Units Accelerated Research

During the last decade, graphics processing units (GPUs) have attracted a lot of attention in the scientific community. These massively parallel processing units offer steadily increasing computational raw power which — today — is an order of magnitude larger than that of CPUs [26]. Furthermore, they are also more efficient in terms of cost and power consumption. Besides this bare power, it is the evolving ecosystem around GPUs which attracts scientists to employ these devices for their research. Suitable programming languages and development kits were released which simplify the application of GPUs for general purposes.

Today, GPUs are applied in those fields of research which handle complex systems by numerical simulation tools. In physics, for example, GPUs are employed for Maxwell's equations [27], seismic wave propagation [28], and incompressible flow dynamics [29], just to mention a few. In comparison to well-established CPU codes, GPU implementations have been reported to yield a significant reduction of the simulation time which may open the door for the treatment of systems governing high complexity or demanding increased accuracy.

Based on these encouraging prospects, we address an implementation of the DGTD method for Maxwell's equations on GPUs. A detailed description of our venture is presented in this thesis. In essence, we obtained a powerful tool which enables us to perform computationally intense physical investigations of, e. g., plasmonic nanostructures.

1.4 Outline of the Thesis

A brief review of classical electrodynamics is given in Chap. 2 where the necessary physical equations governing the description of plasmonic systems are introduced. Subsequently, in Chap. 3, the set of partial differential equations is discretized in space by the nodal DG method, and the low-storage Runge-Kutta method is applied to integrate the set of ordinary differential equations in time. Extensions for the practical use of the DGTD method are presented in Chap. 4. With the theory at hand, the DGTD method for Maxwell's equations is analyzed, implemented, and tested in Chap. 5. In the following chapters, we apply the GPU implementation to plasmonic systems. First, we examine the cross sections of gold nano-antennae and compare the results to the corresponding experiment in Chaps. 6. An individual split-ring resonator and coupled systems of several split-ring resonators are analyzed by electron energy loss spectroscopy in Chap. 7. We conclude the thesis with a brief summary and an outlook in Chap. 8.

Chapter 2 Classical Electrodynamics

The physical systems discussed in this thesis can be accurately described by classical electrodynamics. In particular, quantum mechanical effects do not need to be considered. In this chapter, a brief review of classical electrodynamics is presented focusing on topics which are relevant for this thesis. For an elaborate discussion of classical electrodynamics, the reader is referred to standard textbooks, e. g., Refs. [30, 31].

2.1 Macroscopic Maxwell's Equations

The theory of classical electrodynamics is based on James Clerk Maxwell's publication in 1865 [1] and describes the propagation of electromagnetic waves and their interaction with matter. In vector notation and the SI unit system, the macroscopic Maxwell's equations in differential form read [30, 32]

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

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

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

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

where the electric field $\vec{\mathrm{E}}$, the magnetic field $\vec{\mathrm{H}}$, the electric displacement $\vec{\mathrm{D}}$, and the magnetic induction $\vec{\mathrm{B}}$ are three-dimensional vectors comprising x-, y-, and z-components. The free electric charge density ρ and the free electric current density $\vec{\mathrm{j}}$ denote sources due to free charge in the system, e. g., charge that is not bound to the cores of atoms. These quantities depend on position $\vec{\mathrm{r}} = (x, y, z)^T$ and time t.

The macroscopic Maxwell's equations[1] are accurate for wavelengths large compared to atomic scales as the fields and the sources are considered averaged quantities of a microscopic theory. The systems discussed in this thesis comprise typical length-scales of the order of 100 nm which justifies the application of the macroscopic approach.

[1] For brevity, we refer to Eq. (2.1) by simply "Maxwell's equations" in the remainder of the thesis.

2.1.1 Constitutive Relations

Maxwell's equations comprise twelve unknown field components in eight independent equations. Thus, Eq. (2.1) cannot be solved until $\vec{\mathrm{D}}$ and $\vec{\mathrm{H}}$ are known in terms of $\vec{\mathrm{E}}$ and $\vec{\mathrm{B}}$ [30]. The missing relations are expressed by the constitutive relations which model the interaction between the fields and matter by

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

where $\vec{\mathrm{P}}$ and $\vec{\mathrm{M}}$ denote the macroscopic polarization and magnetization, respectively. Upon introduction of the vacuum permittivity ϵ_0 and the vacuum permeability μ_0, an essential arbitrariness of dimensions is introduced [31] — a fact that we exploit in Sec. 2.1.3 to introduce a dimensionless formulation. Implicitly, ϵ_0 and μ_0 must fulfill

$$c = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \ , \tag{2.3}$$

where c denotes the vacuum speed of light. The polarization $\vec{\mathrm{P}}$ and the magnetization $\vec{\mathrm{M}}$ may depend on $\vec{\mathrm{E}}$, $\vec{\mathrm{B}}$, $\vec{\mathrm{r}}$, and t in a complex way, expressed by $[\cdot]$, e. g., they can depend on past history, be nonlinear, etc. Consequently, media are conveniently classified by the dependencies of $\vec{\mathrm{P}}$ and $\vec{\mathrm{M}}$.

In the regime of weak fields, the material response of most media other than ferroelectrics or ferromagnets can be accurately modeled by the linear relation [30]

$$\begin{aligned} D_\alpha\big(\vec{\mathrm{r}},t\big) &= \sum_\beta \int\limits_{-\infty}^{\infty} \mathrm{d}^3 r' \int\limits_{-\infty}^{\infty} \mathrm{d}t' \ \epsilon_{\alpha\beta}\Big(\vec{\mathrm{r}}',t'\Big)\ E_\beta\Big(\vec{\mathrm{r}}-\vec{\mathrm{r}}',t-t'\Big) \ , \\ H_\alpha\big(\vec{\mathrm{r}},t\big) &= \sum_\beta \int\limits_{-\infty}^{\infty} \mathrm{d}^3 r' \int\limits_{-\infty}^{\infty} \mathrm{d}t' \ \mu'_{\alpha\beta}\Big(\vec{\mathrm{r}}',t'\Big)\ B_\beta\Big(\vec{\mathrm{r}}-\vec{\mathrm{r}}',t-t'\Big) \ , \end{aligned} \tag{2.4}$$

which supports non-local, dispersive, and anisotropic behavior. For many simple metals and dielectrics, the material response is local and isotropic in space and most conveniently described in the frequency-domain by

$$\begin{aligned} \vec{\mathrm{D}}\big(\vec{\mathrm{r}},\omega\big) &= \epsilon_0\epsilon_r(\omega)\cdot\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega\big) \ , \\ \vec{\mathrm{B}}\big(\vec{\mathrm{r}},\omega\big) &= \mu_0\mu_r(\omega)\cdot\vec{\mathrm{H}}\big(\vec{\mathrm{r}},\omega\big) \ , \end{aligned} \tag{2.5}$$

where ϵ_r and μ_r are the *relative* permittivity and permeability, respectively. For dispersionless media, the constitutive relations in the time-domain — obtained by the inverse Fourier transform of Eq. (2.5) — read

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

Employing Eq. (2.6), Maxwell's equations for local, isotropic, and dispersionless media simplify to

$$\vec{\nabla} \cdot \Big(\epsilon_r(\vec{\mathrm{r}}) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \Big) = \frac{\rho(\vec{\mathrm{r}}, t)}{\epsilon_0} \,, \tag{2.7a}$$

$$\vec{\nabla} \cdot \Big(\mu_r(\vec{\mathrm{r}}) \cdot \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \Big) = 0 \,, \tag{2.7b}$$

$$\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) = \quad \epsilon_0 \epsilon_r(\vec{\mathrm{r}}) \cdot \partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) + \vec{\mathrm{j}}(\vec{\mathrm{r}}, t) \ , \tag{2.7c}$$

$$\vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) = -\mu_0 \mu_r(\vec{\mathrm{r}}) \cdot \partial_t \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \ . \tag{2.7d}$$

2.1.2 Time-Evolution of the Electromagnetic Fields

The evolution of the electromagnetic fields in time is governed by the curl equations Eqs. (2.1c) and (2.1d). The divergence conditions Eqs. (2.1a) and (2.1b) are satisfied for all time, if the divergence conditions were initially fulfilled and the fields are evolved in time by the curl equations [21]. Hence, Eqs. (2.1a) and (2.1b) can be regarded as initial conditions for the fields and need not be considered for time-evolution.

To show that, we require the conservation of charge

$$\partial_t \rho(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{j}}(\vec{\mathrm{r}}, t) = 0 \,, \tag{2.8}$$

which is obtained[2] by application of the divergence to Eq. (2.1c) and subsequent insertion of Eq. (2.1a).

Applying the divergence to the curl equations and using Eq. (2.8) yields

$$\begin{aligned} \partial_t \Big(\vec{\nabla} \cdot \vec{\mathrm{D}}(\vec{\mathrm{r}}, t) - \rho(\vec{\mathrm{r}}, t) \Big) &= 0 \,, \\ \partial_t \Big(\vec{\nabla} \cdot \vec{\mathrm{B}}(\vec{\mathrm{r}}, t) \Big) &= 0 \,, \end{aligned} \tag{2.9}$$

which implies that if the divergence conditions are satisfied at some (initial) time t_0, they will not change upon time-evolution by the curl equations as long as charge remains preserved.

2.1.3 Normalization to Dimensionless Units

In the previous sections, we expressed Maxwell's equations in the SI unit system. For numerical simulations, a dimensionless unit system is more convenient and, thus, we scale all lengths λ with respect to a reference length λ_0 — in SI units — and express time by the (normalized) distance that light propagates in free space. In this system, the vacuum speed of light becomes 1:

$$\lambda = \frac{\lambda_{\mathrm{SI}}}{\lambda_0} \,, \qquad t = \frac{1}{\lambda_0} \cdot c_{\mathrm{SI}} t_{\mathrm{SI}} \,, \qquad c = \frac{\lambda}{t} = 1 \,. \tag{2.10}$$

In essence, the interval $[0, \lambda_0]_{\mathrm{SI}}$ is mapped to $[0, 1]$, and light in vacuum propagates 1 "dimensionless unit" of length in 1 "dimensionless unit of time".

[2] The divergence of a curl of a vector is identical to zero, i. e., $\vec{\nabla} \cdot \Big(\vec{\nabla} \times \vec{\mathrm{A}}(\vec{\mathrm{r}}) \Big) \equiv 0$.

Quantity	SI Unit	Dimensionless Units
Length	$\lambda_{\mathrm{SI}} = \lambda_0 \cdot \lambda$	$\lambda = \frac{1}{\lambda_0} \cdot \lambda_{\mathrm{SI}}$
Time	$t_{\mathrm{SI}} = \frac{\lambda_0}{c_{\mathrm{SI}}} \cdot t$	$t = \frac{c_{\mathrm{SI}}}{\lambda_0} \cdot t_{\mathrm{SI}}$
Frequency	$\omega_{\mathrm{SI}} = \frac{c_{\mathrm{SI}}}{\lambda_0} \cdot \omega$	$\omega = \frac{\lambda_0}{c_{\mathrm{SI}}} \cdot \omega_{\mathrm{SI}}$
Electric Field	$\vec{\mathrm{E}}_{\mathrm{SI}} = E_0 \cdot \vec{\mathrm{E}}$	$\vec{\mathrm{E}} = \frac{1}{E_0} \cdot \vec{\mathrm{E}}_{\mathrm{SI}}$
Magnetic Field	$\vec{\mathrm{H}}_{\mathrm{SI}} = E_0 \sqrt{\frac{\epsilon_0}{\mu_0}} \cdot \vec{\mathrm{H}}$	$\vec{\mathrm{H}} = \frac{1}{E_0} \sqrt{\frac{\mu_0}{\epsilon_0}} \cdot \vec{\mathrm{H}}_{\mathrm{SI}}$

Table 2.1: *Conversion rules between SI system and dimensionless units. Throughout the thesis, we set $E_0 = 1\frac{V}{m}$ and $\lambda_0 = 1\,\mu$m.*

In a next step, we eliminate ϵ_0 and μ_0 from the curl equations by rescaling of the fields. For linear media, we can additionally normalize the fields by an arbitrary field strength E_0 — in SI units. Thus, the fields in dimensionless units are obtained by

$$\vec{\mathrm{E}} = \frac{\vec{\mathrm{E}}_{\mathrm{SI}}}{E_0}, \qquad \vec{\mathrm{H}} = \sqrt{\frac{\mu_0}{\epsilon_0}} \cdot \frac{\vec{\mathrm{H}}_{\mathrm{SI}}}{E_0}, \qquad \vec{\mathrm{j}} = \sqrt{\frac{\mu_0}{\epsilon_0}} \cdot \frac{\lambda_0 \vec{\mathrm{j}}_{\mathrm{SI}}}{E_0}. \tag{2.11}$$

Employing Eqs. (2.10) and (2.11), the curl equations for linear, local, isotropic, dispersionless media are transformed to

$$\partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) = \quad \epsilon^{-1}(\vec{\mathrm{r}}) \cdot \left(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) - \vec{\mathrm{j}}(\vec{\mathrm{r}}, t) \right), \tag{2.12a}$$

$$\partial_t \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) = -\mu^{-1}(\vec{\mathrm{r}}) \cdot \vec{\nabla} \times \vec{\mathrm{E}}(\vec{\mathrm{r}}, t). \tag{2.12b}$$

This set of equations comprises only dimensionless quantities. Note that the index r of the relative permittivity and permeability has been dropped for convenience, and we will merely employ quantities in the dimensionless unit system in the remainder of the thesis unless stated otherwise. The transformation rules are summarized in Tab. 2.1. For all simulations of this thesis, we set $\lambda_0 = 1\,\mu$m and $E_0 = 1\frac{V}{m}$.

2.2 Dispersive Materials in the Time-Domain

Plasmonic structures are most commonly fabricated from metals such as silver, gold, or aluminum. Within the visible and near-infrared regime, the real and imaginary part of the permittivity of these metals can vary by orders of magnitude, i. e., these materials exhibit considerable dispersion. Dispersion enters Maxwell's equations via the constitutive relations and can be expressed in both the frequency- and the time-domain. In the frequency-domain, dispersion is conveniently modeled by a frequency-dependent permittivity and permeability (cf. Eq. (2.5)). In contrast, the time-domain formulation contains a convolution integral (cf. Eq. (2.4)) which is notoriously difficult to handle in numeric methods. A feasible technique to include linear, dispersive materials in numerical time-domain methods is based on auxiliary differential equations [15].

2.2.1 Auxiliary Differential Equations

In order to efficiently include dispersive media in a time-domain numerical method, we seek for a simple expression of the time-derivative of the constitutive relation

$$\vec{\mathrm{D}}\big(\vec{\mathrm{r}},t\big) = \int\limits_{-\infty}^{\infty} \mathrm{d}t'\ \epsilon\Big(t'\Big) \cdot \vec{\mathrm{E}}\Big(\vec{\mathrm{r}},t-t'\Big)\ , \tag{2.13}$$

which enters Maxwell's equations via Eq. (2.1c). The time-derivative of Eq. (2.13) transformed into frequency-domain reads

$$-\,\mathrm{i}\omega\vec{\mathrm{D}}\big(\vec{\mathrm{r}},\omega\big) = -\mathrm{i}\omega\left(\epsilon(\omega)\cdot\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega\big)\right)\ , \tag{2.14}$$

where the convolution becomes a simple product and the time-derivative transforms as

$$\partial_t \longleftrightarrow -\mathrm{i}\omega. \tag{2.15}$$

Despite the transformed time-derivative, Eq. (2.14) corresponds to Eq. (2.5) in dimensionless units. The permittivity $\epsilon(\omega)$ can be decomposed into a constant background permittivity ϵ_∞ and a frequency-dependent susceptibility $\chi(\omega)$ by

$$\epsilon(\omega) = \epsilon_\infty + \chi(\omega)\ . \tag{2.16}$$

Upon insertion of Eq. (2.16) into Eq. (2.14), we obtain

$$-\mathrm{i}\omega\vec{\mathrm{D}}\big(\vec{\mathrm{r}},\omega\big) = -\mathrm{i}\omega\epsilon_\infty\cdot\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega\big) + \vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},\omega\big)\ , \tag{2.17a}$$

$$\vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},\omega\big) = -\mathrm{i}\omega\chi(\omega)\cdot\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega\big)\ , \tag{2.17b}$$

where $\vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},\omega\big)$ denotes a frequency-dependent auxiliary polarization current density. Thus, the polarization in the frequency-domain reads

$$\vec{\mathrm{P}}\big(\vec{\mathrm{r}},\omega\big) = \chi(\omega)\cdot\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega\big) = \frac{\vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},\omega\big)}{-\mathrm{i}\omega}\,. \tag{2.18}$$

An inverse Fourier transform of Eqs. (2.17a) and (2.17b) yields the desired time-domain expression for dispersive, linear media. While Eq. (2.17b) depends on the particular material model in charge and must be rearranged before transformation in order to avoid a convolution integral in the time-domain, Eq. (2.17a) is easily transformed:

$$\partial_t\vec{\mathrm{D}}\big(\vec{\mathrm{r}},t\big) = \epsilon_\infty\partial_t\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big) + \vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},t\big)\ . \tag{2.19}$$

Upon insertion of Eq. (2.19) into the Eq. (2.1c), the modified curl equation including the auxiliary polarization current density in dimensionless units reads

$$\partial_t\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big) = \epsilon_\infty^{-1}\cdot\left(\vec{\nabla}\times\vec{\mathrm{H}}\big(\vec{\mathrm{r}},t\big) - \vec{\mathrm{j}}\big(\vec{\mathrm{r}},t\big) - \vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},t\big)\right)\ . \tag{2.20}$$

For suitable susceptibilities $\chi(\omega)$, i. e., rational functions with respect to iω, an inverse Fourier transform of Eq. (2.17b) leads to (several) auxiliary differential equations (ADEs) that are to be solved in parallel with Maxwell's equations [15].

2.2.2 The Drude Model

Around 1900, Paul Drude described the material response of metals by means of the kinetic theory of gases [33, 34]. In the Drude model, the metal is considered to consist of heavy immobile ion cores and bound and free electrons. Bound electrons are located in the vicinity of the ion cores and stated immobile as well while the free electrons behave like an ideal kinetic gas, i. e., in absence of external forces the momentum of the free electrons remains unchanged. In particular, the free electrons interact neither with the ion cores (free electron approximation) nor with each other (independent electron approximation) [35].

Consequently, the velocity of a single electron is modeled by

$$\partial_t \vec{\mathrm{v}}(t) + \gamma_D \vec{\mathrm{v}}(t) = \frac{-e}{m_e} \vec{\mathrm{E}}(t) \ , \tag{2.21}$$

where e denotes the (positive-valued) elementary charge, m_e represents the electron mass, and γ_D corresponds to a phenomenological damping term, also referred to by collision frequency. Solving Eq. (2.21) for the velocity in the frequency-domain yields

$$\vec{\mathrm{v}}(\omega) = \frac{1}{\mathrm{i}\omega - \gamma_D} \frac{e}{m_e} \vec{\mathrm{E}}(\omega) \ . \tag{2.22}$$

The harmonically driven electron is associated with a macroscopic (polarization) current density, i. e.,

$$\vec{\mathrm{j}}_D(\omega) = -en\vec{\mathrm{v}}(\omega) \ , \tag{2.23}$$

where n refers to the free electron density. Upon insertion of Eq. (2.23) into Eq. (2.18), we obtain the susceptibility

$$\chi_D\left(\omega\right) = -\frac{\omega_D^2}{\omega\left(\omega + \mathrm{i}\gamma_D\right)} \tag{2.24}$$

with the plasma frequency

$$\omega_D = \sqrt{\frac{ne^2}{m_e}} \, . \tag{2.25}$$

The permittivity according to $\epsilon(\omega) = \epsilon_\infty + \chi_D(\omega)$ is visualized in the left panel of Fig. 2.1. For frequencies larger than the plasma frequency, the metal behaves similarly to a dielectric, i. e., $\Re\{\epsilon\}$ slowly approaches the positive-valued ϵ_∞ accompanied by a practically vanishing $\Im\{\epsilon\}$. Below ω_D, Drude metals exhibit a strong dispersive response with a negative $\Re\{\epsilon\}$ and significant, positive $\Im\{\epsilon\}$.

The Drude model accurately describes metals in the near- and far-infrared regime where the photon energy is insufficient to invoke interband transitions. Gold in the near-infrared, for example, features a plasma frequency of $\omega_{D,\mathrm{Au}} \approx 14 \cdot 10^{15}\,\mathrm{Hz}$. In comparison, light of $\lambda = 1500\,\mathrm{nm}$ corresponds to a frequency of $\omega \approx 1.3 \cdot 10^{15}\,\mathrm{Hz}$, i. e., the light frequency is well below the plasma frequency. In practice, the free parameters ϵ_∞, ω_D, and γ_D are obtained by the fitting Drude model permittivity to experimental data as described in Sec. 2.2.4.

In order to employ the Drude model in time-domain simulations, the frequency-dependent susceptibility $\chi_D(\omega)$ must be expressed in the ADE formalism (cf. Sec. 2.2.1). Therefore, Eq. (2.24) is inserted into Eq. (2.17b), rearranged, and transformed into the time-domain [15], leading to a single ADE:

$$\partial_t \vec{\mathrm{j}}_D(t) = \omega_D^2 \vec{\mathrm{E}}(t) - \gamma_D \vec{\mathrm{j}}_D(t) \tag{2.26}$$

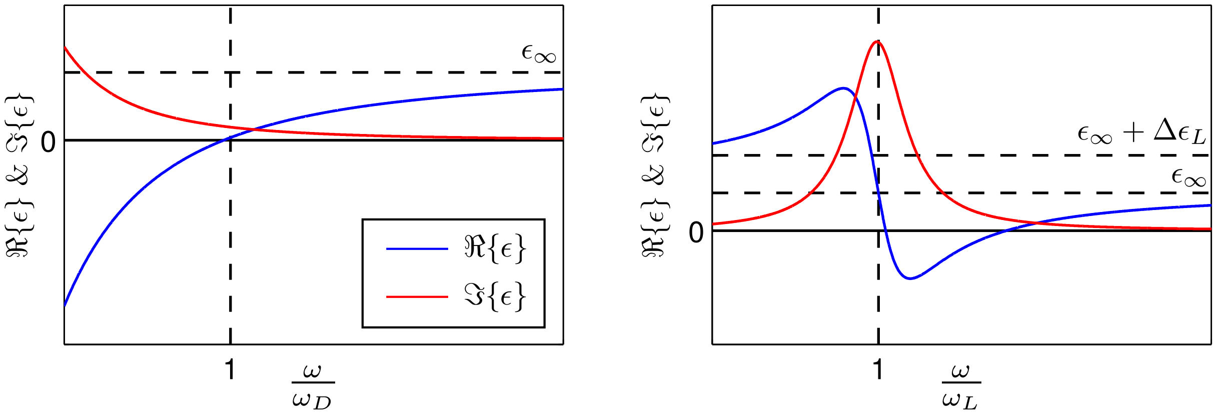


Figure 2.1: *Electric permittivity $\epsilon(\omega)$ for the Drude (left panel) and the Lorentz (right panel) model. Real and imaginary parts of the permittivity are indicated in blue and red, respectively.*

2.2.3 The Lorentz Model

By considering merely free electrons, the Drude model fails to describe interband transitions in the high energy range of the visible spectrum. Hendrik Antoon Lorentz extended the Drude model by taking bound electrons into account which are modeled as harmonic oscillators. In an equivalent mechanical model, an electron is bound to an ion by an elastic spring which results in a restoring force [32]. Consequently, polarizable matter is understood as a collection of identical, independent, isotropic harmonic oscillators.

The equation of motion for an electron of charge $-e$ and mass m_e is given by

$$\partial_t^2 \vec{\mathrm{x}}(t) + \gamma_L \partial_t \vec{\mathrm{x}}(t) + \omega_L^2 \vec{\mathrm{x}}(t) = \frac{-e}{m_e} \vec{\mathrm{E}}(t) \ , \tag{2.27}$$

where $\vec{\mathrm{x}}$ represents the displacement from the equilibrium position, ω_L denotes the eigenfrequency of the oscillator and γ_L corresponds to a phenomenological damping term. Solving for $\vec{\mathrm{r}}$ in Fourier space yields

$$\vec{\mathrm{x}}(\omega) = \frac{1}{\omega_L^2 - \mathrm{i}\gamma_L \omega - \omega^2} \cdot \frac{-e}{m_e} \vec{\mathrm{E}}(\omega) \ . \tag{2.28}$$

The displacement is associated with a dipole moment by $\vec{\mathrm{p}}(\omega) = -e\vec{\mathrm{x}}(\omega)$. As the individual oscillators are considered non-interacting, the macroscopic polarization is obtained by $\vec{\mathrm{P}}(\omega) = n\vec{\mathrm{p}}(\omega)$, where n denotes the density of microscopic oscillators. Employing Eq. (2.18), the susceptibility of the Lorentz oscillator model reads

$$\chi_L(\omega) = \frac{\Delta\epsilon_L \cdot \omega_L^2}{\omega_L^2 - \mathrm{i}\gamma_L \omega - \omega^2} \tag{2.29}$$

with the oscillator strength

$$\Delta\epsilon_L = \frac{\omega_D^2}{\omega_L^2} \ , \qquad \left(\text{with } \omega_D^2 = \frac{ne^2}{m_e} \right) \ . \tag{2.30}$$

Note that the density of microscopic oscillators and the free electron density of the Drude model differ and — despite the formal equity of ω_D and the Drude plasma frequency — $\Delta\epsilon_L$ is a free parameter of the Lorentz model. In practice, the free parameters ω_L, γ_L, and $\Delta\epsilon_L$ are obtained by a fit procedure, described in Sec. 2.2.4.

The permittivity according to $\epsilon(\omega) = \epsilon_\infty + \chi_L(\omega)$ is depicted in the right panel of Fig. 2.1. The imaginary part exhibits a significant increase around ω_L which models absorption due to interband transitions. In the same range, the real part is strongly dispersive.

The ADE formulation of the Lorentz model is obtained by insertion of the susceptibility Eq. (2.29) into Eq. (2.17b), rearrangement, and subsequent transformation into the time-domain. Two ADEs are required to model the time-domain polarization current density [15]:

$$\partial_t \vec{\mathrm{j}}_L(t) = \Delta\epsilon_L \omega_L^2 \vec{\mathrm{E}}(t) + \vec{\mathrm{q}}_L(t) \ , \tag{2.31a}$$

$$\partial_t \vec{\mathrm{q}}_L(t) = -\omega_L^2 \vec{\mathrm{j}}_L(t) - \gamma_L \Delta\epsilon_L \omega_L^2 \vec{\mathrm{E}}(t) - \gamma_L \vec{\mathrm{q}}_L(t) \ . \tag{2.31b}$$

2.2.4 Calculation of Model Parameters from Experimental Data

Both the Drude and the Drude-Lorentz model accurately describe the electric permittivity within a limited spectral range. Previously published material parameters were often designed to model large spectral regions leading to a suboptimal representation of the permittivity within narrow ranges of the spectrum, e. g., in the visible and near-infrared regime. For practical purposes, the model must be accurate within the range of illumination and observation. Thus, we determine the free parameters of a material model by fitting the permittivity to experimental values within the spectral range of interest.

Assume a metal of which the permittivity is to be modeled by one Drude and one Lorentz pole, i. e.,

$$\begin{aligned} \epsilon_{\mathrm{fit}}\big(\omega;\Pi\big) &= \epsilon'_{\mathrm{fit}}(\omega) + \mathrm{i}\epsilon''_{\mathrm{fit}}(\omega) \\ &= \epsilon_\infty - \frac{\omega_D^2}{\omega\left(\omega + \mathrm{i}\gamma_D\right)} + \frac{\Delta\epsilon_L \cdot \omega_L^2}{\omega_L^2 - \mathrm{i}\gamma_L\omega - \omega^2} \end{aligned} \tag{2.32}$$

with the free parameters $\Pi \ni \{\epsilon_\infty, \omega_D, \gamma_D, \Delta\epsilon_L, \omega_L, \gamma_L\}$. For a given set of N experimentally determined values of the complex electric permittivity

$$\epsilon_{\mathrm{exp}}(\omega_i) = \epsilon'_{\mathrm{exp}}(\omega_i) + \mathrm{i}\epsilon''_{\mathrm{exp}}(\omega_i) \, , \; i \in [1, N] \, , \tag{2.33}$$

the relative deviation of the real and imaginary part of the experimental values and the model permittivity are defined by

$$\delta\epsilon'_i = \frac{\epsilon'_{\mathrm{fit}}(\omega_i) - \epsilon'_{\mathrm{exp}}(\omega_i)}{\epsilon'_{\mathrm{exp}}(\omega_i)} \qquad \text{and} \qquad \delta\epsilon''_i = \frac{\epsilon''_{\mathrm{fit}}(\omega_i) - \epsilon''_{\mathrm{exp}}(\omega_i)}{\epsilon''_{\mathrm{exp}}(\omega_i)} \, , \tag{2.34}$$

respectively. The set of free parameters Π is determined by

$$\min_{\Pi} \left(\sum_{i=1}^{N} \left(\delta\epsilon'_i\right)^2 + \sum_{i=1}^{N} \left(\delta\epsilon''_i\right)^2 \right) \ . \tag{2.35}$$

Here, the sum of the squared relative deviation of both the real and the imaginary part is minimized which implies that the real and the imaginary parts are optimized together. This means that each of the $2N$ data points of $\{\epsilon'_{\mathrm{exp}}(\omega_i)\, , \epsilon''_{\mathrm{exp}}(\omega_i)\}$, $i \in [1, N]$ contributes to the minimization functional with equal weight. In order to ensure reasonable physical values, the free parameters are bound to limits, e. g., $\{\epsilon_\infty, \Delta\epsilon_L\} \in [1, \infty[$ and $\{\omega_D, \gamma_D, \omega_L, \gamma_L\} \in [0, \infty[$.

The nonlinear minimization algorithm employs the subspace trust-region method and is based on the interior-reflective Newton method [36, 37]. In general, nonlinear minimization algorithms cannot guarantee to arrive at a global minimum but return local minima in the parameter space Π. The result of the minimization strongly depends on the initial condition which must be provided to the algorithm.

Previously established model parameters often serve as suitable initial conditions. For gold, silver, and copper, Ref. [38] provides both initial conditions for the Drude model and experimental values at optical frequencies and in the near-infrared. Vial *et al.* [39] extended the Drude model parameters by an additional Lorentz pole. Experimental values for a large number of elements and compound materials are compiled in, e. g., Ref. [40].

2.3 Scattering of Electromagnetic Radiation

Photonic and plasmonic structures can be characterized by the amount of energy they absorb, i. e., convert into non-radiative energy or heat, and scatter, i. e., reradiate into all directions. The total amount of energy that is removed from an incident electromagnetic field is referred to by extinction. Sophisticated experimental setups allow to obtain the absorption, scattering, and extinction cross section of individual particles and nanostructures [12, 23, 41]. This section briefly summarizes relevant aspects of scattering of electromagnetic radiation for the purposes of this thesis. For an extensive introduction, we refer the reader to Refs. [32, 42].

The flux of energy carried by an electromagnetic field is determined by the Poynting vector [32]

$$\vec{\mathrm{S}}\big(\vec{\mathrm{r}}, t\big) = \vec{\mathrm{E}}\big(\vec{\mathrm{r}}, t\big) \times \vec{\mathrm{H}}\big(\vec{\mathrm{r}}, t\big) \ . \tag{2.36}$$

For light in the visible and near-infrared regime, the Poynting vector Eq. (2.36) oscillates at frequencies above 100 Thz. Most detectors are not able to follow these rapid oscillations [32]. The physical quantity which is accessible in the experiment is the time-averaged Poynting vector

$$\vec{\mathrm{S}}\big(\vec{\mathrm{r}}, \omega\big) = \frac{1}{2}\Re\left\{\vec{\mathrm{E}}\big(\vec{\mathrm{r}}, \omega\big) \times \vec{\mathrm{H}}^*\big(\vec{\mathrm{r}}, \omega\big)\right\} \ , \tag{2.37}$$

which determines the energy transported by an electromagnetic wave per unit area and time within the period $T = 2\pi\omega^{-1}$ [15]. Note that the time-averaged Poynting vector depends on frequency and is not the Fourier transform of Eq. (2.36).

The net rate $W(\omega)$ at which electromagnetic energy crosses the closed surface ∂V of a volume V is given by [32]

$$W(\omega) = -\int\limits_{\partial V} \vec{\mathrm{S}}\big(\vec{\mathrm{r}}, \omega\big) \cdot \hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)\, \mathrm{d}^2 r \,, \tag{2.38}$$

where $\hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)$ is the outwardly directed normal vector on the surface ∂V and the minus sign ensures that energy flowing into the volume is accounted for by a positive value.

Let us consider a scatterer embedded in a homogeneous, lossless, linear medium. Due to the linearity of Maxwell's equations, we can decompose the total field, labeled "tot", into the incident field "inc" and the scattered field "scat" by

$$\begin{aligned} \vec{\mathrm{E}}_{\mathrm{tot}}\big(\vec{\mathrm{r}}, \omega\big) &= \vec{\mathrm{E}}_{\mathrm{inc}}\big(\vec{\mathrm{r}}, \omega\big) + \vec{\mathrm{E}}_{\mathrm{scat}}\big(\vec{\mathrm{r}}, \omega\big) \ , \\ \vec{\mathrm{H}}_{\mathrm{tot}}\big(\vec{\mathrm{r}}, \omega\big) &= \vec{\mathrm{H}}_{\mathrm{inc}}\big(\vec{\mathrm{r}}, \omega\big) + \vec{\mathrm{H}}_{\mathrm{scat}}\big(\vec{\mathrm{r}}, \omega\big) \ . \end{aligned} \tag{2.39}$$

In such a medium, the rate at which energy is absorbed within V reads

$$W_{\text{abs}}(\omega) = -\int\limits_{\partial V} \vec{\mathrm{S}}_{\text{tot}}\big(\vec{\mathrm{r}},\omega\big) \cdot \hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)\,\mathrm{d}^2 r\,, \tag{2.40}$$

where $\vec{\mathrm{S}}_{\text{tot}}\big(\vec{\mathrm{r}},\omega\big)$ is the time-averaged Poynting vector of the total fields $\vec{\mathrm{E}}_{\text{tot}}\big(\vec{\mathrm{r}},\omega\big)$ and $\vec{\mathrm{H}}_{\text{tot}}\big(\vec{\mathrm{r}},\omega\big)$. Similarly we calculate the rate of energy that is scattered from within the volume through ∂V by

$$W_{\text{scat}}(\omega) = +\int\limits_{\partial V} \vec{\mathrm{S}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big) \cdot \hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)\,\mathrm{d}^2 r\,. \tag{2.41}$$

Here, the time-averaged Poynting vector of the scattered fields $\vec{\mathrm{S}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big)$ is calculated from $\vec{\mathrm{E}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big)$ and $\vec{\mathrm{H}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big)$. Note that we inverted the minus sign in order to account for energy that is transported out of the volume V by a positive value. In absence of scatterers within V, the scattered fields vanish, i. e., $\vec{\mathrm{E}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big) = \vec{\mathrm{H}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big) \equiv 0$, which implies that both $W_{\text{abs}}(\omega)$ and $W_{\text{scat}}(\omega)$ equal zero, too.

Both $W_{\text{abs}}(\omega)$ and $W_{\text{scat}}(\omega)$ directly depend on the strength of the incident field. In order to eliminate this dependence, both rates are normalized by the modulus of the incident energy rate which yields the absorption and the scattering cross section, respectively:

$$C_{\text{abs}}(\omega) = -\frac{1}{|\vec{\mathrm{S}}_{\text{inc}}(\omega)\,|} \cdot \int\limits_{\partial V} \vec{\mathrm{S}}_{\text{tot}}\big(\vec{\mathrm{r}},\omega\big) \cdot \hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)\,\mathrm{d}^2 r\,, \tag{2.42}$$

$$C_{\text{scat}}(\omega) = \frac{1}{|\vec{\mathrm{S}}_{\text{inc}}(\omega)\,|} \cdot \int\limits_{\partial V} \vec{\mathrm{S}}_{\text{scat}}\big(\vec{\mathrm{r}},\omega\big) \cdot \hat{\mathrm{n}}\big(\vec{\mathrm{r}}\big)\,\mathrm{d}^2 r\,. \tag{2.43}$$

The extinction cross section is expressed as the sum of the absorption and scattering cross section [32]:

$$C_{\text{ext}}(\omega) = C_{\text{abs}}(\omega) + C_{\text{scat}}(\omega)\ . \tag{2.44}$$

In terms of geometrical optics, the extinction cross section corresponds to the size of a shadow cast by a perfectly opaque object. In the regime of wave optics, however, the “shadow” may differ considerably from the geometric cross section [32].

In many cases, the incident field is a homogeneous plane wave of which the modulus of the Poynting vector in a lossless medium with real-valued permittivity ϵ and permeability μ reads [32]

$$\big|S(\omega)\big| = \frac{1}{2}\sqrt{\frac{\epsilon}{\mu}}\,\Big|\vec{\mathrm{E}}_0(\omega)\Big|^2\ . \tag{2.45}$$

3 Chapter 3

The Discontinuous Galerkin Time-Domain Method

With the exception of some usually highly symmetric cases, analytic solutions to Maxwell's equations are elusive. In contrast, numerical techniques provide universal access to an approximate solution of a physical system in both the time- and frequency-domain. Roughly, we can categorize numerical methods into general-purpose and specialized methods. While the former make few assumptions about the physical system at hand, the latter utilizes specific properties of the system which renders these methods very efficient in their field but unsuitable in other cases.

In this chapter, the nodal discontinuous Galerkin time-domain (DGTD) method for Maxwell's equations is presented which can be considered a general-purpose method. The nodal discontinuous Galerkin (DG) discretization supports a flexible representation of the physical system's geometry by an unstructured mesh which can approximate almost arbitrarily shaped geometries. In principle, the DG approach inherently supports curvilinear mesh elements, i. e., elements of which the faces may be curved [21]. However, highest computational efficiency is achieved for planar-faced elements. Unstructured meshes help to avoid artificial field enhancement due to sharp edges which was observed for methods employing an orthogonal grid, e. g., the popular finite-difference (FD) methods.

Another noteworthy advantage of the DG method is its arbitrary order basis functions which provide excellent control over the discretization error compared to, e. g., the finite-difference time-domain (FDTD) method which is bound to second order accuracy when dealing with material interfaces [17].

The DG method provides discretized matrix operators for the spatial derivatives of a set of PDEs and leaves the temporal derivatives untouched. This is known as the method of lines approach. Consequently, DG discretized Maxwell's equations can be solved in both the time- and frequency-domain by a multitude of different approaches [15].

Most naturally, the propagation of electromagnetic waves and their interaction with matter is investigated in the time-domain. However, the response of a physical system is more conveniently described in terms of frequency-resolved spectra. Employing short pulses in the time-domain gives access to the response of the physical system in a broad frequency range by a single simulation.

From a practical point of view, the simulation time, i. e., the time a simulation requires to be executed, is important, too. The DGTD method provides a highly parallel numerical scheme, especially under the assumption of planar-faced elements [21], for which we will present an implementation on Graphics Processing Units in Chap. 5. Recently, a similarly efficient approach was presented for curvilinear elements, too [43].

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

Table 3.1: *Notation of vector types employed throughout the thesis. The notation is in agreement with Ref. [15].*

3.1 Spatial Discretization via the Nodal Discontinuous Galerkin Method

In the spirit of the method of lines approach, this section comprises the discretization of the spatial derivatives according to the nodal discontinuous Galerkin method. The discussion is mainly guided by Refs. [14, 15, 21, 44] and concentrates on providing a conclusive derivation of the numerical scheme while omitting proofs in the interest of brevity. For details on proofs, the reader is referred to Refs. [21, 45]. In order to provide a compact description, we employ the notation of Ref. [15], which is summarized in Tab. 3.1.

3.1.1 Maxwell's Curl Equations in Conservation Form

The first step is to formulate the physical problem in terms of the common basis for the DG discretization, i. e., we formulate Maxwell's equations in conservation form. In order to focus on the method itself, we consider linear, isotropic, dispersionless media resulting in simple constitutive relations (cf. Eq. (2.6)) and neglect current density sources, i. e., $\vec{\mathrm{j}}\big(\vec{\mathrm{r}}, t\big) \equiv 0$. Given suitable initial conditions, the time-evolution of Maxwell's equations is governed by the curl equations (cf. Sec. 2.1.2).

Casting Eq. (2.12) in conservation form, Maxwell's curl equations in dimensionless units read [21]

$$\mathcal{Q}\big(\vec{\mathrm{r}}\big) \cdot \partial_t \mathbf{q}\big(\vec{\mathrm{r}}, t\big) + \vec{\nabla} \cdot \vec{\mathrm{F}}\big(\mathbf{q}\big) = 0\,, \tag{3.1}$$

where material matrix $\mathcal{Q}\big(\vec{\mathrm{r}}\big)$ and the state vector $\mathbf{q}\big(\vec{\mathrm{r}}, t\big)$ are defined by

$$\mathcal{Q}\big(\vec{\mathrm{r}}\big) = \begin{pmatrix} \epsilon\big(\vec{\mathrm{r}}\big) & 0 \\ 0 & \mu\big(\vec{\mathrm{r}}\big) \end{pmatrix} \quad \text{and} \quad \mathbf{q}\big(\vec{\mathrm{r}}, t\big) = \begin{pmatrix} \vec{\mathrm{E}}\big(\vec{\mathrm{r}}, t\big) \\ \vec{\mathrm{H}}\big(\vec{\mathrm{r}}, t\big) \end{pmatrix}\,, \tag{3.2}$$

respectively, and the flux is given by

$$\vec{\mathrm{F}} = \Big(\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z\Big)^T\,, \quad \mathbf{F}_i\big(\mathbf{q}\big) = \begin{pmatrix} -\hat{\mathrm{e}}_i \times \vec{\mathrm{H}}\big(\vec{\mathrm{r}}, t\big) \\ \hat{\mathrm{e}}_i \times \vec{\mathrm{E}}\big(\vec{\mathrm{r}}, t\big) \end{pmatrix}\,. \tag{3.3}$$

Here, the material matrix comprises the position-dependent material properties, i. e., the (relative) permittivity $\epsilon(\vec{\mathrm{r}})$ and permeability $\mu(\vec{\mathrm{r}})$. The state vector is a super-vector of the electric and magnetic field, i. e., it contains two physical vectors of three components each (cf. Tab. 3.1). The flux vector is a physical vector of three components of which each is itself a super-vector depending on the components of $\mathbf{q}$. The Cartesian unit vector into the i^{th} direction is denoted by $\hat{\mathrm{e}}_i$, $i \in \{x, y, z\}$.

At first glance, the notation might seem confusing but serves to support a compact notation later on. Note, that the product of the material matrix and the state vector is to be understood as

$$\mathcal{Q}(\vec{\mathrm{r}}) \cdot \mathbf{q}(\vec{\mathrm{r}}, t) = \begin{pmatrix} \epsilon(\vec{\mathrm{r}}) \cdot \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \\ \mu(\vec{\mathrm{r}}) \cdot \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \end{pmatrix} ,$$

and the divergence of the flux vector is given by

$$\vec{\nabla} \cdot \vec{\mathrm{F}} = \partial_x \begin{pmatrix} -\hat{\mathrm{e}}_x \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \\ \hat{\mathrm{e}}_x \times \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \end{pmatrix} + \partial_y \begin{pmatrix} -\hat{\mathrm{e}}_y \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \\ \hat{\mathrm{e}}_y \times \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \end{pmatrix} + \partial_z \begin{pmatrix} -\hat{\mathrm{e}}_z \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \\ \hat{\mathrm{e}}_z \times \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) \end{pmatrix} ,$$

i. e., both operations yield physical state vectors of six components. The remainder of this section addresses the discretization of Eq. (3.1) by the DG method following Refs. [15, 21, 45].

3.1.2 Tesselation of the Computational Domain

The DG method inherits properties from both the finite-volume and the finite-element method. As such, the volume in which we evaluate Maxwell's equations, referred to by computational domain Ω, is finite in extent. In order to allow for a flexible representation of the physical system's geometry, the computational domain Ω is tessellated into K non-overlapping body-conforming simplices $D^{\triangle}$, where $\triangle \in \{1, \ldots, K\}$ refers to the element index.

According to Ref. [21], the simplices, also referred to by (physical) elements, may by curvilinear, i. e., their faces are not necessarily planar. However, the assumption of planar faces will simplify the implementation considerably and lead to an efficient numerical scheme.

The shape of the elements is not uniquely defined. Amongst others, tetrahedra, hexahedra, prisms, or pyramids can be employed for three-dimensional systems. A mesh may even contain combinations of different element types which complicates matters in practice considerably. By far the most common choice is to employ merely tetrahedra [45]. As an example, a sphere modeled by tetrahedra is depicted in Fig. 3.1.

Operations of the physical elements $D^{\triangle}$ can be expressed in terms of those defined on a reference element D^{ref}, which is a fundamental strength of the DG method. Therefore, we introduce a mapping which uniquely relates the two simplices [21]

$$\Psi : D^{\mathrm{ref}} \to D^{\triangle} . \tag{3.4}$$

For linear elements, i. e., elements with planar faces, this relation is expressed by an affine transform of coordinates:

$$\vec{\mathrm{r}}^{\triangle} = \vec{\mathrm{r}}_0^{\triangle} + \mathcal{J}^{\triangle} \vec{\mathrm{s}}, \tag{3.5a}$$

$$\mathcal{J}^{\triangle} = \frac{\partial \vec{\mathrm{r}}}{\partial \vec{\mathrm{s}}} . \tag{3.5b}$$

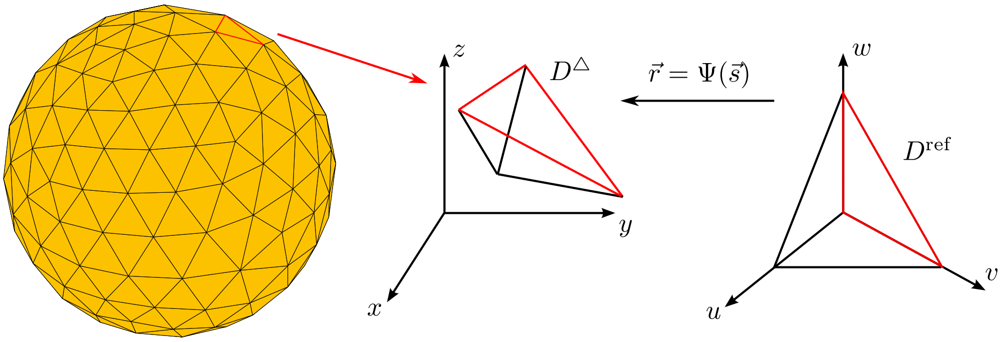


Figure 3.1: *Tesselation of the computational domain and mapping to the reference element. The left panel shows a sphere modeled by tetrahedra. The geometry is approximated by elements of different sizes and shapes. The right panel depicts a physical element and its mapping to the reference element.*

Here, $\mathcal{J}^{\triangle}$ is the element-local Jacobian matrix, and $\vec{\mathrm{r}} = (x, y, z)^T \in D^{\triangle} \subset R^3$ and $\vec{\mathrm{s}} = (u, v, w)^T \in D^{\text{ref}} \subset R^3$ are local coordinate vectors of a physical element $D^{\triangle}$ and the reference element D^{ref}, respectively. The values of $\vec{\mathrm{r}}_0^{\triangle}$ and $\mathcal{J}^{\triangle}$ depend on the explicit vertices of the reference element and the physical elements.

Later in this chapter, we want to express the operators of physical elements in terms of those defined on the reference element. Therefore, we require the inverse Jacobian matrix

$$\left(\mathcal{J}^{\triangle}\right)^{-1} = \frac{\partial \vec{\mathrm{s}}}{\partial \vec{\mathrm{r}}} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} = \begin{pmatrix} \mathcal{J}^{\triangle}_{xu} & \mathcal{J}^{\triangle}_{yu} & \mathcal{J}^{\triangle}_{zu} \\ \mathcal{J}^{\triangle}_{xv} & \mathcal{J}^{\triangle}_{yv} & \mathcal{J}^{\triangle}_{zv} \\ \mathcal{J}^{\triangle}_{xw} & \mathcal{J}^{\triangle}_{yw} & \mathcal{J}^{\triangle}_{zw} \end{pmatrix}, \tag{3.6}$$

which contains merely constant values for planar-faced elements.

3.1.3 Element-Local Approximate Solution

Our ultimate goal is to find the exact solution $\mathbf{q}$ of the problem formulated in Eq. (3.1). The solution obtained by a numerical method $\mathbf{q}_N$ generally differs from the — in general unknown — exact solution while convergence ensures that the error is bounded.

On a particular element, the numerical solution $\mathbf{q}_N^{\triangle}$ cannot satisfy Eq. (3.1) exactly which results in a residuum:

$$\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) = \text{res}\,. \tag{3.7}$$

In some sense, it must be defined how the numerical solution shall satisfy Eq. (3.1) exactly. The DG discretization inherits this property from FEM. Let $L_i(\vec{\mathrm{r}})$ represent a scalar basis function of a finite, linear function space and demand that the residuum is orthogonal to this function space. On element $D^{\triangle}$, this condition is expressed by a vanishing scalar product of the residuum and the test functions:

$$\int\limits_{V_{\triangle}} \left(\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right)\right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r = \int\limits_{V_{\triangle}} \text{res} \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r \overset{!}{=} 0\,. \tag{3.8}$$

The local solution $\mathbf{q}_N^{\triangle}$ must satisfy Eq. (3.8) for all test functions $L_i(\vec{\mathrm{r}})$ while the global solution $\mathbf{q}_N$ must satisfy Eq. (3.8) for all elements. Note that in contrast to FEM, where orthogonality to the test functions is demanded globally over the entire computational domain Ω, Eq. (3.8) demands orthogonality locally for each element. As a consequence, the elements are completely decoupled from each other and we have to establish a mechanism to enforce the boundary conditions of Maxwell's equations.

3.1.4 Restoring a Physical Solution via the Numerical Flux

Due to the entirely local statement of Eq. (3.8), discontinuities across element-boundaries are explicitly allowed. In general, the local solutions of two adjacent elements exhibit a jump at their common interface irrespective of the physical behavior they ought to describe.

Maxwell's equations demand continuity of the electromagnetic fields within homogeneous regions and exhibit well-defined conditions at material interfaces [30]. In particular, the tangential components of $\vec{\mathrm{E}}$ and $\vec{\mathrm{H}}$ must be continuous across material interfaces while the normal components jump. These conditions must be explicitly enforced to obtain a physically reasonable, global solution.

Integrating Eq. (3.8) by parts yields

$$\int\limits_{V_\triangle} \left(\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t) \cdot L_i(\vec{\mathrm{r}}) - \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) \cdot \vec{\nabla} L_i(\vec{\mathrm{r}}) \right) \mathrm{d}^3 r = - \int\limits_{\partial V_\triangle} \left(\hat{\mathrm{n}} \cdot \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) \right) \cdot L_i(\vec{\mathrm{r}}) \, \mathrm{d}^2 r \,, \tag{3.9}$$

where $\hat{\mathrm{n}}$ denotes the outwardly oriented normal (unit-)vector on the element's boundary $\partial V_\triangle$. Intentionally, the integration by parts yields a surface term on the right-hand side which can be "employed" to transfer information between elements through their common face. In this spirit, we modify the surface term in order to account for physical conditions by introduction of the numerical flux $\vec{\mathrm{F}}^*$ which ought to enforce a physically correct solution. Obviously, the choice of the numerical flux is crucial for correctness and stability of the numerical scheme, and we will address the issue in more detail later in this section.

Undoing the integration by parts yields the strong variational form of Maxwell's equations [45]:

$$\int\limits_{V_\triangle} \left(\mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) \right) \cdot L_i(\vec{\mathrm{r}}) \, \mathrm{d}^3 r = \int\limits_{\partial V_\triangle} \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) - \vec{\mathrm{F}}^*\left(\mathbf{q}_N^{\triangle}\right) \right) \cdot L_i(\vec{\mathrm{r}}) \, \mathrm{d}^2 r \,. \tag{3.10}$$

Note that the left-hand side remains an element-local expression while the right-hand side mediates coupling of adjacent elements through their common face by the numerical flux.

The Numerical Flux for Maxwell's Equations

As the numerical flux is supposed to enforce physical conditions across element boundaries, it is specific to the physical problem at hand. Surprisingly, the numerical flux is not unique. In 2002, Hesthaven and Warburton proved convergence of a numerical upwind flux for the nodal DG method which enforces the boundary conditions of Maxwell's equations in a weak form[1] [21]. From their paper and Ref. [45],

[1] This implies continuity of the electromagnetic fields within homogeneous regions.

we obtain

$$\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) - \vec{\mathrm{F}}^*\left(\mathbf{q}_N^{\triangle}\right) \right) = \begin{pmatrix} \bar{Z}^{-1} \left(\alpha \left[\Delta\vec{\mathrm{E}} - \hat{\mathrm{n}} \left(\hat{\mathrm{n}} \cdot \Delta\vec{\mathrm{E}} \right) \right] + Z^+ \hat{\mathrm{n}} \times \Delta\vec{\mathrm{H}} \right) \\ \bar{Y}^{-1} \left(\alpha \left[\Delta\vec{\mathrm{H}} - \hat{\mathrm{n}} \left(\hat{\mathrm{n}} \cdot \Delta\vec{\mathrm{H}} \right) \right] - Y^+ \hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}} \right) \end{pmatrix} . \tag{3.11}$$

The upwind flux Eq. (3.11) comprises the impedance Z and admittance Y of the local "-" and neighboring "+" element

$$Z^\pm = \sqrt{\frac{\mu^\pm}{\epsilon^\pm}} , \quad Y^\pm = \left(Z^\pm\right)^{-1} \tag{3.12}$$

and their sums

$$\bar{Z} = Z^+ + Z^- , \quad \bar{Y} = Y^+ + Y^- . \tag{3.13}$$

In Eq. (3.11), $\hat{\mathrm{n}}$ denotes the normal unit-vector at the boundary of the local element in outward direction and the field differences across common interfaces are defined as

$$\Delta\vec{\mathrm{E}} = \vec{\mathrm{E}}^+ - \vec{\mathrm{E}}^- , \quad \Delta\vec{\mathrm{H}} = \vec{\mathrm{H}}^+ - \vec{\mathrm{H}}^- . \tag{3.14}$$

Eq. (3.11) contains the so-called upwind parameter α. Hesthaven and Warburton showed [21] that a stable, convergent scheme can be achieved for $\alpha \in [0, 1]$. For nodal methods, an upwind flux ($\alpha \neq 0$) is preferable over an energy conserving central flux ($\alpha = 0$) since unphysical spurious oscillations are damped. Optimal convergence is obtained for $\alpha = 1$ [21, 45]. Hence, for the simulations performed in the context of this thesis, we employ $\alpha = 1$. The derivation of the numerical flux involves the solution of a Riemann problem, and is discussed in detail in, e. g., Refs. [14, 21, 45].

3.1.5 Boundary Conditions

As a consequence of the truncation of space, we are obliged to define the behavior at the boundary of the computational domain by means of boundary conditions. These boundary conditions ought to mimic the behavior of the physical system under consideration. For example, assume a scattering experiment in which an object located in (an infinite) free space is illuminated by some source. Scattered light leaves the vicinity of the object and the experiment without further interaction. This behavior is modeled by absorbing radiation impinging onto the boundary of the computational domain by employing absorbing boundary conditions. In the context of the thesis, we employ reflecting and absorbing boundary conditions (ABCs) albeit there exist several more, e. g., periodic boundary conditions [17].

Within the DG method, information between elements is exchanged by the numerical flux. Thus, it is not surprising that boundary conditions are enforced by the numerical flux. In case of Maxwell's equations, the field differences mediate in between adjacent elements (cf. Eq. (3.11)). Boundary elements, i. e., elements at the boundary of the computational domain, are missing a neighbor to evaluate the field difference and, thus, virtual elements are provided to evaluate the field differences Eq. (3.14). These virtual elements are assigned the material properties of the local elements, i. e.,

$$Z^+ \equiv Z^- \quad \text{and} \quad Y^+ \equiv Y^- , \tag{3.15}$$

and the field differences are defined according to Tab. 3.2.

Boundary Condition (BC)	$\Delta\vec{\mathrm{E}}$	$\Delta\vec{\mathrm{H}}$
Perfect Electric Conductor (PEC)	$-2\vec{\mathrm{E}}^-$	0
Perfect Magnetic Conductor (PMC)	0	$-2\vec{\mathrm{H}}^-$
Silver-Müller (first order absorbing)	$-2\vec{\mathrm{E}}^-$	$-2\vec{\mathrm{H}}^-$

Table 3.2: *Field differences for selected boundary conditions [44].*

Perfect electric conductor (PEC) and perfect magnetic conductor (PMC) boundary conditions are reflecting boundary conditions which are typically employed to model cavities (cf. Sec. 5.3.1) or to enforce symmetry (cf. Sec. 7.2.4).

Absorbing boundary conditions, e. g., the Silver-Müller boundary conditions, mimic open space by absorbing incident radiation. Unfortunately, the Silver-Müller boundary conditions provide unacceptable performance for oblique incidence, and more sophisticated approaches must be employed to overcome this disadvantage. In practice, so-called perfectly matched layers (PMLs) are required, which are addressed in Sec. 4.3.

3.1.6 Expansion of the Electromagnetic Fields

In Sec. 3.1.3, we defined in which sense the approximate solution $\mathbf{q}_N^{\triangle}$ must fulfill the set of PDEs Eq. (3.1). In this section, we explicitly define the approximate solution in terms of the previously defined test functions $L_i(\vec{\mathrm{r}})$.

We assume that the exact solution $\mathbf{q}^{\triangle}$ is well approximated on element $D^{\triangle}$ by [15, 21]

$$\mathbf{q}_N^{\triangle}(\vec{\mathrm{r}},t) = \sum_{i=1}^{N_p} \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}_i,t)\, L_i(\vec{\mathrm{r}}) = \sum_{i=1}^{N_p} \tilde{\mathbf{q}}_i^{\triangle}(t)\, L_i(\vec{\mathrm{r}}) \ , \tag{3.16}$$

where N_p nodal points $\vec{\mathrm{r}}_i$ are utilized, and $\tilde{\mathbf{q}}_i^{\triangle}(t) = \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}_i,t)$ serves as short-hand notation for the expansion coefficients (cf. Tab. 3.1). Employing the same function space for both the test functions and the expansion of the unknowns is commonly known as the Galerkin choice.

Due to the nodal expansion, the space- and time-dependence are separated, i. e., the time-evolution is fully captured by the expansion coefficients $\tilde{\mathbf{q}}_i^{\triangle}(t)$ at the nodal points $\vec{\mathrm{r}}_i$ while the space-dependence is governed by the multivariate Lagrange interpolation polynomials $L_i(\vec{\mathrm{r}})$.

Due to the defining property of the interpolation polynomials

$$L_i\left(\vec{\mathrm{r}}_j\right) = \delta_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases} \quad i,j \in \{1,\ldots,N_p\}\,, \tag{3.17}$$

the expansion coefficients directly represent unknown field values at the nodal points $\vec{\mathrm{r}}_i$. Exploiting Eq. (3.17), this property is easily shown:

$$\mathbf{q}_N^{\triangle}\left(\vec{\mathrm{r}}_j,t\right) = \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}_i,t)\, L_i\left(\vec{\mathrm{r}}_j\right) = \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}_i,t)\,\delta_{ij} = \tilde{\mathbf{q}}_j^{\triangle}(t) \ . \tag{3.18}$$

For convenience, we applied Einstein notation, i. e., summation of repeating indices. This convention is employed throughout the thesis.

p	N_p	N_{fp}	$N_f N_{fp}$	b	i
2	10	6	24	10	0
3	20	10	40	20	0
4	35	15	60	34	1
5	56	21	84	52	4
6	84	28	112	74	10

Table 3.3: *Characteristic quantities of the DG method: the number of nodes per element N_p, the number of nodes on a single face N_{fp}, the product $N_f N_{fp}$, the number of nodes on the element's boundary b, and the number of nodes inside the element i are listed for the practically most relevant polynomial orders $p = \{2, \ldots, 6\}$, where $N_f = 4$ is the number of faces per element for three-dimensional systems.*

The Lagrange polynomials of order p are given by [15]

$$L_i(\vec{r}) = \sum_{k,l,m=0}^{k+l+m\leq p} a_{k,l,m}^{(i)} \cdot x^k y^l z^m \,, \tag{3.19}$$

where the coefficients $a_{k,l,m}^{(i)}$ are determined by the position of the nodes $\vec{r}_i$ and Eq. (3.17). The choice of the nodal points crucially influences the accuracy of the approximation Eq. (3.16) with which $\mathbf{q}_N^{\triangle}(\vec{r}, t)$ is interpolated in between the nodes. Interestingly, there exists a set of optimal node positions for one-dimensional systems, given by the Legendre-Gauss-Lobatto quadrature nodes [45]. For higher dimensions, the empirical Warp&Blend method is employed to create a suitable set of nodal points [46].

In order to set up the numerical scheme, we must calculate the Lagrange polynomials explicitly. For a set of predetermined nodal points $\vec{r}_i$, the Lagrange polynomials can be determined from Eqs. (3.17) and (3.19). Unfortunately, the resulting linear systems of equations are ill conditioned and suffer low precision, especially for high orders [21]. Consequently, the Lagrange polynomials are constructed using an intermediate basis of Koornwinder-Dubiner polynomials [21, 45, 47].

According to Eq. (3.19), the number of basis functions N_p is given in dependence of the polynomial order p by

$$N_p = \frac{1}{6} \cdot (p+1) \cdot (p+2) \cdot (p+3) \tag{3.20}$$

for three-dimensional systems. Note that N_p also represents the number of interpolation nodes $\vec{r}_i$ per element and the number of expansion coefficients for each component of the state vector $\tilde{\mathbf{q}}^{\triangle}$. Thus, the polynomial order controls both accuracy and convergence rate of the approximation. Obviously, increasing p yields improved accuracy and convergence at the cost of additional computational effort. A compilation of characteristic quantities for the DG method resulting from the node distribution according to the Warp&Blend method is listed in Tab. 3.3.

3.1.7 The Semi-Discrete Form

The last step of the discretization procedure is the insertion of the expansion of the unknowns Eq. (3.16) into the strong variational form of Maxwell's equations Eq. (3.10).

Therefore, we must evaluate the integrals

$$I_{1,i}^{\triangle} = \int\limits_{V_\triangle} \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r\,, \tag{3.21a}$$

$$I_{2,i}^{\triangle} = \int\limits_{V_\triangle} \vec{\nabla} \cdot \vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r\,, \tag{3.21b}$$

$$I_{3,i}^{\triangle} = \int\limits_{\partial V_\triangle} \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}}\left(\mathbf{q}_N^{\triangle}\right) - \vec{\mathrm{F}}^{*}\left(\mathbf{q}_N^{\triangle}\right)\right) \cdot L_i(\vec{\mathrm{r}})\, \mathrm{d}^2 r\,, \tag{3.21c}$$

which leads to the so-called semi-discrete matrix formulation. In the following, we examine each of the above integrals.

The Mass Matrix

First, we evaluate Eq. (3.21a) by considering the E_x-component of the state vector $\mathbf{q}_N^{\triangle}(\vec{\mathrm{r}}, t)$. Similar expressions for other state vector components follow straightforwardly. Upon insertion of Eq. (3.16) into Eq. (3.21a), we obtain

$$\begin{aligned} I_{1,i}^{\triangle, E_x} &= \int\limits_{V_\triangle} \epsilon(\vec{\mathrm{r}}) \cdot \partial_t \tilde{\mathrm{E}}_{x,j}^{\triangle}(t) \cdot L_j(\vec{\mathrm{r}}) L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r \\ &= \partial_t \tilde{\mathrm{E}}_{x,j}^{\triangle}(t) \int\limits_{V_\triangle} \epsilon(\vec{\mathrm{r}}) \cdot L_j(\vec{\mathrm{r}}) L_i(\vec{\mathrm{r}})\, \mathrm{d}^3 r \end{aligned} \tag{3.22}$$

due to the separation of the space- and time-dependence of the nodal expansion. Demanding constant material parameters within each element, the integrand reduces to $L_i(\vec{\mathrm{r}})L_j(\vec{\mathrm{r}})$. This assumption will allow for a very efficient numerical scheme and does not impose severe restrictions [14].

Thus, replacing the position-dependent permittivity $\epsilon(\vec{\mathrm{r}})$ by the element-constant $\epsilon^{\triangle}$ yields

$$I_{1,i}^{\triangle, E_x} = \epsilon^{\triangle} \cdot \partial_t \left(\left(\mathcal{M}^{\triangle}\right)_{ij} \tilde{\mathrm{E}}_{x,j}^{\triangle}(t) \right)\,, \tag{3.23}$$

where the element-local mass matrix $\mathcal{M}^{\triangle}$ is defined as

$$\left(\mathcal{M}^{\triangle}\right)_{ij} = \int\limits_{V_\triangle} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\, \mathrm{d}^3 r\,. \tag{3.24}$$

Note that the mass matrix $\mathcal{M}^{\triangle}$ is defined on element $D^{\triangle}$, and we have not yet assumed a particular relation to the reference element D^{ref}.

The efficiency of the DG method is partially based on the creation of template operators which are defined on the reference element D^{ref} [21]. Acknowledging Eq. (3.5), the mass matrix of the reference element is defined as

$$\left(\mathcal{M}^{\triangle}\right)_{ij} = \int\limits_{V_{\mathrm{ref}}} L_i(\vec{\mathrm{s}}) L_j(\vec{\mathrm{s}}) \cdot \det\left(\mathcal{J}^{\triangle}(\vec{\mathrm{s}})\right) \mathrm{d}^3 s\,, \tag{3.25}$$

where the Lagrange polynomials $L_i(\vec{\mathrm{s}})$ are defined by the nodal points of the reference element. For planar-faced elements, the Jacobian matrix merely contains constant-valued entries [21]. Consequently, the determinant of the Jacobian matrix

$$J^{\triangle} = \det\left(\mathcal{J}^{\triangle}\right) \tag{3.26}$$

is constant-valued, too. Eq. (3.25) simplifies to

$$\mathcal{M}^{\triangle} = J^{\triangle}\mathcal{M}\,, \tag{3.27}$$

where $\mathcal{M}$ is the mass matrix of the reference element being defined as

$$\mathcal{M} = \int\limits_{V_{\mathrm{ref}}} L_i(\vec{\mathrm{s}})L_j(\vec{\mathrm{s}})\,\mathrm{d}^3 s\,. \tag{3.28}$$

Employing, Eq. (3.27), we can construct the mass matrices of each element by a simple scaling operation which results in an efficient numerical scheme later on.

The Stiffness and the Differentiation Matrices

In a similar fashion, we treat the second integral Eq. (3.21b). Employing the definition of the flux Eq. (3.3) and the expansion of the unknowns Eq. (3.16), the integral for the E_x-component can be expressed by

$$\begin{aligned} I_{2,i}^{\triangle,E_x} &= \int\limits_{V_\triangle} \left(\partial_z \vec{\mathrm{H}}_{y,N}^{\triangle}(\vec{\mathrm{r}},t) - \partial_y \vec{\mathrm{H}}_{z,N}^{\triangle}(\vec{\mathrm{r}},t)\right)\cdot L_i(\vec{\mathrm{r}})\,\mathrm{d}^3 r \\ &= \int\limits_{V_\triangle} \tilde{\mathrm{H}}_{y,j}^{\triangle}(t)\cdot L_i(\vec{\mathrm{r}})\left(\partial_z L_j(\vec{\mathrm{r}})\right)\mathrm{d}^3 r - \int\limits_{V_\triangle} \tilde{\mathrm{H}}_{z,j}^{\triangle}(t)\cdot L_i(\vec{\mathrm{r}})\left(\partial_y L_j(\vec{\mathrm{r}})\right)\mathrm{d}^3 r \\ &= \left(\mathcal{S}_z^{\triangle}\right)_{ij}\cdot \tilde{\mathrm{H}}_{y,j}^{\triangle}(t) - \left(\mathcal{S}_z^{\triangle}\right)_{ij}\cdot \tilde{\mathrm{H}}_{y,j}^{\triangle}(t)\ , \end{aligned} \tag{3.29}$$

where the element-local stiffness matrices $\mathcal{S}_k^{\triangle}$, $k \in \{x,y,z\}$ of $D^{\triangle}$ are defined as

$$\left(\mathcal{S}_k^{\triangle}\right)_{ij} = \int\limits_{V_\triangle} L_i(\vec{\mathrm{r}})\left(\partial_k L_j(\vec{\mathrm{r}})\right)\mathrm{d}^3 r\,. \tag{3.30}$$

Again, the separation of space- and time-dependence within the expansion is essential. As a consequence, the spatial derivatives merely act on the Lagrange polynomials. For a compact notation in later sections, we introduce the vector of stiffness matrices

$$\vec{\mathcal{S}}^{\triangle} = \left(\mathcal{S}_x^{\triangle}, \mathcal{S}_y^{\triangle}, \mathcal{S}_z^{\triangle}\right)^T\ . \tag{3.31}$$

Similar to the mass matrix, we strive to find a definition of the stiffness matrix on the reference element. The derivative of any polynomial of order p is a polynomial of order $p-1$. Thus, there exists a unique representation of the derivative in terms of Lagrange polynomials with the coefficients given by $\partial_k L_j$ at the nodal points [14]. Employing this, the stiffness matrices can be expressed by

$$\begin{aligned}\left(\mathcal{S}_k^{\triangle}\right)_{ij} &= \int\limits_{V_\triangle} L_i(\vec{\mathrm{r}})\,\left(\partial_k L_j(\vec{\mathrm{r}}_m)\right) L_m(\vec{\mathrm{r}})\,\mathrm{d}^3 r \\ &= \left(\mathcal{D}_k^{\triangle}\right)_{mj} \cdot \int\limits_{V_\triangle} L_i(\vec{\mathrm{r}}) L_m(\vec{\mathrm{r}})\,\mathrm{d}^3 r \\ &= \left(\mathcal{M}^{\triangle}\mathcal{D}_k^{\triangle}\right)_{ij}\,,\end{aligned} \tag{3.32}$$

where we defined the element-local differentiation matrices $\mathcal{D}_k^{\triangle}$ as

$$\left(\mathcal{D}_k^{\triangle}\right)_{mj} = \partial_k L_j(\vec{\mathrm{r}}_m)\ . \tag{3.33}$$

For reasons that will become clear by the end of this section, we reformulate Eq. (3.32) to

$$\mathcal{D}_k^{\triangle} = \left(\mathcal{M}^{\triangle}\right)^{-1} \mathcal{S}_k^{\triangle} \tag{3.34}$$

and introduce the vector of differentiation matrices

$$\vec{\mathcal{D}}^{\triangle} = \left(\mathcal{D}_x^{\triangle}, \mathcal{D}_y^{\triangle}, \mathcal{D}_z^{\triangle}\right)^T\ . \tag{3.35}$$

The differentiation matrices of the reference element D^{ref} are defined in analogy to Eq. (3.33):

$$\left(\mathcal{D}_d\right)_{mj} = \partial_d L_j(\vec{\mathrm{s}}_m)\ , \quad d \in \{u, v, w\}\,. \tag{3.36}$$

Acknowledging Eq. (3.6), we can construct the local-element differentiation matrices from the differentiation matrices of the reference element:

$$\mathcal{D}_k^{\triangle} = \sum_{d=1}^{3} \left(\mathcal{J}^{\triangle}\right)^{-1}_{kd} \cdot \mathcal{D}_d\,. \tag{3.37}$$

Instead of storing the element-local differentiation matrices for each element, they can be efficiently constructed from the template matrices $\mathcal{D}_d$. Therefore, the inverse Jacobian matrix for each element is required. However, for three-dimensional systems, this amounts to nine values per element only.

The Face-Mass and the Lift Matrices

The last integral Eq. (3.21c) involves the numerical flux Eq. (3.11). Conceptually, there is no difference compared to the derivations of the mass, the stiffness, and the differentiation matrices. In the interest of brevity, we omit the lengthy derivation and state the result, as usual, for the E_x-component:

$$I_{3,i}^{\triangle,E_x} = \sum_f \left[\mathcal{F}_f^{\triangle}\left[\frac{\alpha\left(\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}}(\hat{\mathrm{n}}\cdot\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle})\right) + Z^{+}\hat{\mathrm{n}}\times\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}}{\bar{Z}}\right]_x\right]_i \tag{3.38}$$

Here, $\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle}$ and $\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}$ correspond to physical vectors of three components. Each component comprises a vector of expansion coefficients which contains the field difference across face f (cf. Tab. 3.1).

For each face of element $D^{\triangle}$, we introduced the element-local face-mass matrix $\mathcal{F}_f^{\triangle}$ which is defined as

$$\left(\mathcal{F}_f^{\triangle}\right)_{ij} = \int\limits_{f^{\triangle}} L_i(\vec{\mathrm{r}}) L_j(\vec{\mathrm{r}})\,\mathrm{d}^2 r\,, \quad \vec{\mathrm{r}} \in f^{\triangle}\,. \tag{3.39}$$

Note that the integration is performed over a single face $f^{\triangle}$. Due to Eq. (3.17), a Lagrange polynomial $L_j\left(\vec{\mathrm{r}}\right)$ identically vanishes on face f, if the associated node $\vec{\mathrm{r}}_j$ does not lie on this face [14]. Consequently, many entries of the face-mass matrix are zero.

The element-local face-mass matrices are easily expressed in terms of template operators acknowledging the analogy to the mass matrix. Thus, the face-mass matrix on face f of element $D^{\triangle}$ can be constructed by

$$\mathcal{F}_f^{\triangle} = J_f^{\triangle} \cdot \mathcal{F}_f\,, \tag{3.40}$$

where the face-mass matrix of the reference element D^{ref} is defined as

$$\mathcal{F}_f = \int\limits_{f^{D^{\mathrm{ref}}}} L_i(\vec{\mathrm{s}}) L_j(\vec{\mathrm{s}})\,\mathrm{d}^2 s\,. \tag{3.41}$$

As it turns out, it is advantageous to define the so-called lift matrix of face f on the reference element:

$$\mathcal{L}_f \equiv \mathcal{M}^{-1} \cdot \mathcal{F}_f\,. \tag{3.42}$$

Thus, the element-local lift matrix is given by

$$\left(\mathcal{M}^{\triangle}\right)^{-1} \cdot \mathcal{F}_f^{\triangle} \equiv \mathcal{L}_f^{\triangle} = \frac{J_f^{\triangle}}{J^{\triangle}} \mathcal{L}_f\,. \tag{3.43}$$

Semi-Discrete Form of Maxwell's Equations

Finally, we can express the strong variational form in terms of the previously derived matrix operators. The semi-discrete form of Maxwell's equations for local, isotropic, linear, dispersionless media with element-wise constant parameters reads

$$\begin{aligned}
\partial_t \tilde{\vec{\mathrm{E}}}^{\triangle} &= \frac{1}{\epsilon^{\triangle}} \left(\mathcal{M}^{\triangle}\right)^{-1} \left(\quad \vec{\mathcal{S}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle} + \bar{Z}^{-1} \mathcal{F}_f^{\triangle} \left[\alpha \left(\Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}}\left(\hat{\mathrm{n}} \cdot \Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle}\right)\right) + Z^{+} \hat{\mathrm{n}} \times \Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle} \right] \right), \\
\partial_t \tilde{\vec{\mathrm{H}}}^{\triangle} &= \frac{1}{\mu^{\triangle}} \left(\mathcal{M}^{\triangle}\right)^{-1} \left(-\vec{\mathcal{S}}^{\triangle} \times \tilde{\vec{\mathrm{E}}}^{\triangle} + \bar{Y}^{-1} \mathcal{F}_f^{\triangle} \left[\alpha \left(\Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle} - \hat{\mathrm{n}}\left(\hat{\mathrm{n}} \cdot \Delta\tilde{\vec{\mathrm{H}}}_f^{\triangle}\right)\right) - Y^{+} \hat{\mathrm{n}} \times \Delta\tilde{\vec{\mathrm{E}}}_f^{\triangle} \right] \right).
\end{aligned} \tag{3.44}$$

In this particular form, merely element-local matrix operators are employed. Eq. (3.44) holds for both linear and curvilinear elements since we have not yet assumed a particular mapping between the element $D^{\triangle}$ and the reference element D^{ref}.

Note that the mass and the face-mass matrix act on physical vectors of expansion coefficient vectors (cf. Tab. 3.1). In this sense, the face-mass matrix acts on a vector of expansion coefficients $\tilde{\vec{\mathrm{a}}}$ by

$$\mathcal{F}_f^{\triangle} \cdot \tilde{\vec{\mathrm{a}}} \equiv \left(\mathcal{F}_f^{\triangle} \cdot \tilde{\mathrm{a}}_x,\, \mathcal{F}_f^{\triangle} \cdot \tilde{\mathrm{a}}_y,\, \mathcal{F}_f^{\triangle} \cdot \tilde{\mathrm{a}}_z\right)^T\,. \tag{3.45}$$

The semi-discrete form describes the time-evolution of an element $D^{\triangle}$. Since all elements are treated equivalently, it also represents the time-evolution of the global electromagnetic fields.

Semi-Discrete Form of Maxwell's Equations on the Reference Element

The semi-discrete form Eq. (3.44) employs element-local operators. In order to utilize this form in a numerical scheme, we would have to store the matrix operators of all elements simultaneously which would result in an immense memory consumption and an inefficient numerical method.

Assuming a particular mapping $\Psi : D^{\mathrm{ref}} \to D^{\triangle}$, we can cast Eq. (3.44) into a computationally more efficient form. Under the assumption of planar-faced elements, the Jacobi matrix (cf. Sec. 3.1.2) and its determinant are constants. This results in simple linear relations between element-local matrix operators and those defined on the reference element (cf. Eqs. (3.27),(3.37), and (3.40)).

Upon insertion of the previously derived template operators, we obtain the semi-discrete form of Maxwell's equations for planar-faced elements

$$\begin{aligned} \partial_t \tilde{\vec{\mathrm{E}}}^{\triangle} &= \frac{1}{\epsilon^{\triangle}} \left(\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle} + \mathcal{L}_f \cdot \tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle} \right) , \\ \partial_t \tilde{\vec{\mathrm{H}}}^{\triangle} &= \frac{1}{\mu^{\triangle}} \left(-\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{E}}}^{\triangle} + \mathcal{L}_f \cdot \tilde{\vec{\mathrm{F}}}_f^{\mathrm{H},\triangle} \right) . \end{aligned} \tag{3.46}$$

Here, we have employed Eqs. (3.34) and (3.43) to absorb the inverse mass matrix. Note that the element-local differentiation matrices can be efficiently calculated using Eq. (3.37).

The flux vectors $\tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle}$ and $\tilde{\vec{\mathrm{F}}}_f^{\mathrm{H},\triangle}$ are defined as

$$\begin{aligned} \tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle} &= \bar{Z}^{-1} \frac{J_f^{\triangle}}{J^{\triangle}} \left[\alpha \left(\Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle} - \hat{\mathrm{n}} \left(\hat{\mathrm{n}} \cdot \Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle} \right) \right) + Z^{+} \hat{\mathrm{n}} \times \Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle} \right] , \\ \tilde{\vec{\mathrm{F}}}_f^{\mathrm{H},\triangle} &= \bar{Y}^{-1} \frac{J_f^{\triangle}}{J^{\triangle}} \left[\alpha \left(\Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle} - \hat{\mathrm{n}} \left(\hat{\mathrm{n}} \cdot \Delta \tilde{\vec{\mathrm{H}}}_f^{\triangle} \right) \right) - Y^{+} \hat{\mathrm{n}} \times \Delta \tilde{\vec{\mathrm{E}}}_f^{\triangle} \right] . \end{aligned} \tag{3.47}$$

Note that element-local constants, such as $J_f^{\triangle}$, $J^{\triangle}$, and the material parameters are absorbed in the flux vectors.

Due to the application of template matrix operators, Eq. (3.46) merely requires the differentiation and lift matrices of the reference element and the mapping constants of each element. Element-local operators can be constructed on-the-fly at moderate computational effort.

Finally, note that the template matrix operators are relatively small in size, i. e., each differentiation matrix contains $N_p \times N_p$ entries and the lift matrix of face f reduces to $N_p \times N_{fp}$ entries (cf. Tab. 3.3). Thus, in order to implement the DG approach efficiently, one must be capable to handle a large number of small matrix-vector-products.

3.1.8 Convergence and Error of the Spatial Discretization

A rigorous proof of convergence is a necessary condition for every numerical method. For the DG method, convergence can be proven by use of the Lax-Richtmyer equivalence theorem [48]. However, this yields suboptimal error estimates [45].

Sharp error bounds for the DG method applied to Maxwell's equations including the upwind flux Eq. (3.11) were determined in Ref. [21]. In this case, the error of the numerical solution $\mathbf{q}_N$ is bounded by [15]

$$\left\| \mathbf{q}^{\mathrm{ref}} - \mathbf{q}_N \right\| = \mathcal{O}\left(h^{p+1}\right) , \tag{3.48}$$

where $\mathbf{q}^{\mathrm{ref}}$ denotes the — in general unknown — exact solution, h represents the size of the largest element of the mesh, and p is the polynomial order.

The convergence is algebraic in h and exponential in p [15] which is known as hp-convergence. Hence, for a constant order of interpolation p, the error can be reduced by decreasing the size of the elements in the mesh. Alternatively, the polynomial order p can be increased to enhance the accuracy of the solution. In general, both h and p can be different for each element which is known as hp-adaptivity. In principle, p does not need to be constant during a time-domain simulation (adaptive p-refinement) [45]. For the simulations performed in the context of this theses, p is equal for all elements of the mesh and is not changed during the simulation.

3.2 Time Integration

The discretization according to the DG method results in the semi-discrete form of Maxwell's equations (cf. Eq. (3.46)). The missing part towards a time-domain solver is the integration in time.

In order to simplify the notation, we rewrite Eq. (3.46) as [15]

$$\partial_t \tilde{\mathbf{p}}(t) = \mathcal{H} \cdot \tilde{\mathbf{p}}(t) + \tilde{\mathbf{p}}^{\mathrm{source}}(t) , \tag{3.49}$$

where $\mathcal{H}$ denotes the system operator and $\tilde{\mathbf{p}}^{\mathrm{source}}(t)$ corresponds to a source term. The system operator acts on the state vector of expansion coefficients and represents the right-hand side of Eq. (3.46). Eq. (3.49) is a set of coupled ordinary differential equations (ODEs) of first order in time, which are in a more general form described by

$$\partial_t \tilde{\mathbf{p}}(t) = \tilde{\mathbf{f}}\left(\tilde{\mathbf{p}}(t) , t\right) . \tag{3.50}$$

A set of first order ODEs can be dealt with by a multitude of different methods. An overview of the general structure of these methods is depicted in Fig. 3.2. The choice of a particular method should account for the properties of the set of ODEs, especially those of the system operator $\mathcal{H}$.

In case of the DG discretization, the temporal order of accuracy should be sufficiently high to accompany the spatial order of discretization p. In addition, we usually face systems with a large amount of unknowns, typically of the order of 10^7 and above. Thus, the time-integration method ought to consume little memory.

Based on these requirements, we opt for an explicit low-storage Runge-Kutta (LSRK) method of the Williamson formulation [50]. Currently established LSRK methods provide a temporal accuracy of up to fourth order. Employing such a method, the numerical error is usually dominated by the spatial discretization. The LSRK method demands to store a total of (only) $2N$ coefficients simultaneously, where N refers to the number of unknowns in $\tilde{\mathbf{p}}$.

3.2.1 The Low-Storage Runge Kutta Method

In order to integrate Eq. (3.50) in time, we divide the time-axis into a number of — not necessarily equidistant — time-steps t_n. For the $2N$-LSRK method, we prepare two equivalent state vectors of expansion coefficients $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$, each containing N entries.

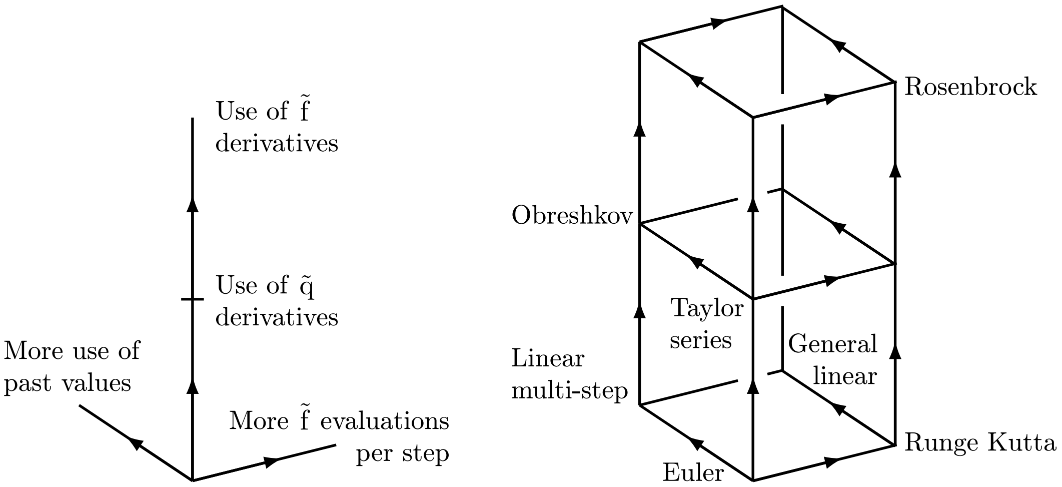

Use of $\tilde{\mathrm{f}}$ derivatives
Use of $\tilde{\mathrm{q}}$ derivatives
More use of past values
More $\tilde{\mathrm{f}}$ evaluations per step
Rosenbrock
Obreshkov
Taylor series
General linear
Linear multi-step
Euler
Runge Kutta

Figure 3.2: *Classification of general method types for the time-integration of ODEs. The figure is adapted from Ref. [49].*

The set of ODEs is evolved from t_n to t_{n+1} by [15]

$$\tilde{\mathbf{q}}_0 = \tilde{\mathbf{q}}(t_n)\,, \tag{3.51a}$$

$$\tilde{\mathbf{p}}_i = A_i \cdot \tilde{\mathbf{p}}_{i-1} + \Delta t \cdot \tilde{\mathbf{f}}\big(\tilde{\mathbf{q}}_{i-1}, t_n + c_i \Delta t\big)\,, \tag{3.51b}$$

$$\tilde{\mathbf{q}}_i = \tilde{\mathbf{q}}_{i-1} + B_i \cdot \tilde{\mathbf{p}}_i\,, \tag{3.51c}$$

$$\tilde{\mathbf{q}}\big(t_{n+1}\big) = \tilde{\mathbf{q}}_s\,. \tag{3.51d}$$

Eqs. (3.51b) and (3.51c) define the so-called stages $i = \{1, \ldots, s\}$, where s is the number of stages. For a given number of stages s and temporal order of accuracy p_t, the parameters A_i, B_i, and c_i define a particular scheme of the LSRK method. Interestingly, the scheme is not unique for a given set (s, p_t). The time-step $\Delta t = t_{n+1} - t_n$ denotes the time-interval integrated by a single evaluation of Eq. (3.51). Per se, the time-step is a free parameter which is, however, bound to conditions. We will address the time-step in more detail in the following section. Note that the amount of storage is independent of the number of stages s and the order of accuracy p_t.

A five-stage, fourth order scheme of Carpenter and Kennedy [51] is most commonly employed for DGTD simulations of Maxwell's equations [21, 22, 52]. In previous work, we developed LSRK schemes of third and fourth order which can yield increased performance compared to Carpenter and Kennedy's scheme [53, 54]. In several cases, these schemes allow for a 40% − 50% increased time-step compared to the scheme of Ref. [51].

3.2.2 Stability

Since we chose an *explicit* LSRK method, the time-integration by Eq. (3.51) is conditionally stable. This conditional stability imposes conditions on the time-step Δt, i. e., we cannot choose Δt arbitrarily large. The relation between the unknowns of two consecutive time-steps is given by the amplification factor

$$R(z) = 1 + \gamma_1 z + \gamma_2 z^2 + \cdots + \gamma_s z^s\,, \quad z \in \mathbb{C}\,. \tag{3.52}$$

Here, the parameters γ_i, $i = \{1, \ldots, s\}$ can be directly related to parameters of the LSRK scheme A_i, B_i, and c_i [49, 53, 54]. Employing the amplification factor, a necessary condition of stability is

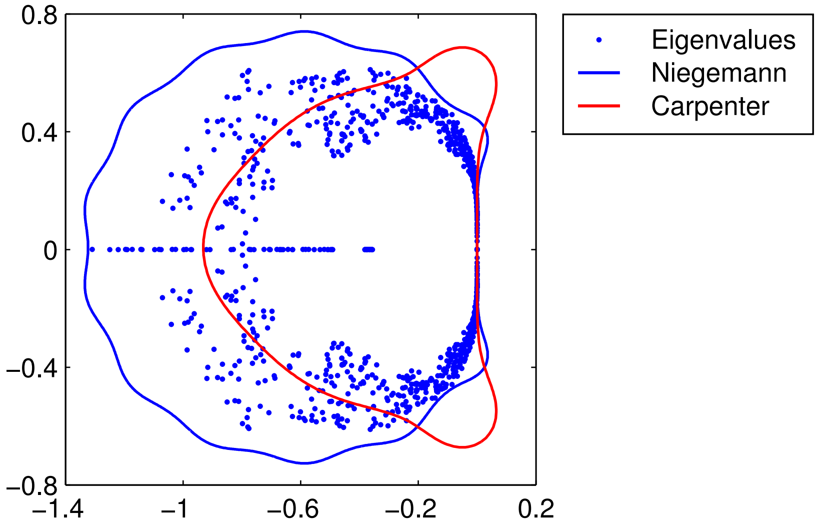


Figure 3.3: *Eigenvalue spectrum of an empty cavity (blue dots) and normalized stability contours of Niegemann's 14-stage (blue) and Carpenter and Kennedy's five-stage (red) LSRK scheme. Both schemes are of fourth order in time.*

expressed by [49, 55]

$$\forall i : \Big| R\big(\Delta t \cdot \lambda_i\big) \Big| \leq 1 \,, \tag{3.53}$$

where λ_i are the eigenvalues of the system operator $\mathcal{H}$ in the absence of sources (cf. Eq. (3.49)). Note that Eq. (3.53) does not guarantee stability. However, it served as a good approximation of a sufficient condition [45, 56].

Violation of the stability condition Eq. (3.53) results in an unphysical exponential growth of the unknowns. Based on Eq. (3.53), we can formulate an explicit condition for the maximum stable time-step $\Delta t_{\max}$:

$$\forall i : \Big| R\big(\Delta t_{\max} \cdot \lambda_i\big) \Big| = 1 \,. \tag{3.54}$$

Consequently, the maximum time-step depends on both the physical system, represented by λ_i, and the integration scheme, i. e., the set of parameters A_i, B_i, and c_i (cf. Eq. (3.52)). The region of stability $\{z : \big|R(z)\big| \leq 1\}$ of schemes employing different numbers of stages s can be quantitatively compared by the so-called normalized region of absolute stability [53, 54]

$$\left\{ \frac{z}{s} : \big|R(z)\big| \leq 1 \right\} \,. \tag{3.55}$$

The normalized stability region of Carpenter and Kennedy's five-stage, fourth order scheme, as well as our 14-stage, fourth order scheme, are visualized[2] in Fig. 3.3. In order to fulfill the stability condition Eq. (3.53) for a given normalized time-step $\Delta t_s = \frac{\Delta t}{s}$, *all* eigenvalues must reside within the normalized stability contour. In case of Fig. 3.3, the normalized time-step per stage Δt_s is chosen such that the stability condition is exactly fulfilled for the 14-stage scheme. In contrast, employing the same Δt_s for the five-stage scheme violates stability, i. e., the normalized time-step is too large to fulfill Eq. (3.53).

[2] In fact, the normalized stability contour $\{z/s : |R(z)\,| = 1\}$ is depicted. The region of normalized stability governs the closed area within the normalized stability contour.

As a result, for this particular eigenvalue spectrum, the 14-stage scheme allows for a larger maximum normalized time-step than the five-stage scheme [53, 54]. An empirical guideline to chose the time-step in practice is provided in Ref. [15].

In order to achieve a large, stable, normalized time-step, two main approaches can be derived from Eq. (3.55):

- Manipulate the eigenvalue spectrum such that it fits into the stability contour.
- Form the stability contour according to the eigenvalue spectrum.

Since the eigenvalue spectrum directly depends on the physical problem at hand, there is only limited freedom to manipulate its shape. Furthermore, the exact spectrum is in general not available.

In case of the simulation of Maxwell's equations, the eigenvalue spectrum can be directly manipulated by the upwind parameter α [45, 54, 57]. By variation in the range $\alpha = [0, 1]$, significantly increased maximum stable time-steps of the order of $40\% - 50\%$ compared to the conventionally employed $\alpha = 1$ were observed [54, 57].

The second point, adapting the stability contour according to the eigenvalue spectrum, offers more flexibility. In order to do this, the eigenvalue spectrum is required but in general unknown. Based on generic shapes of the eigenvalue spectrum, Niegemann *et al.* [53, 54] developed LSRK schemes which are optimized for the DGTD method.

In their approach, the degrees of freedom emerging from an increased number of stages s are employed to optimize those parameters γ_i which are not bound to accuracy conditions. In a subsequent step, a nonlinear minimization yields the LSRK parameters A_i, B_i, and c_i. Such optimized schemes show up to $40\% - 50\%$ enhanced performance compared to the most frequently used scheme of Carpenter and Kennedy for an upwind parameter $\alpha = 1$ [53, 54].

Chapter 4

4 Extensions for Practical Use

In the previous chapter, Maxwell's equations were discretized according to the nodal discontinuous Galerkin method. In order to focus on the approach itself, we excluded sources and restricted the constitutive relations to linear dispersionless media. Only a limited number of simple systems can be treated due to these constraints which must be remedied for realistic physical setups. In particular, we extend the previously presented method to include metals which are described by the Drude and the Drude-Lorentz model in Sec. 4.1. In addition, the total-field/scattered-field and scattered-field techniques are introduced to include practically arbitrary electromagnetic sources in Sec. 4.2. Finally, perfectly matched layers (PMLs), a sophisticated approach to limit the computational domain by an absorbing medium, are addressed in Sec. 4.3.

4.1 The Discretized Drude-Lorentz Model

Essential properties of metals originate from their dispersive response described by a frequency-dependent electric permittivity $\epsilon(\omega)$. A general approach to arrive at a computationally feasible time-domain formulation for dispersive media by means of auxiliary differential equations was introduced and applied for both the Drude and the Drude-Lorentz model (cf. Sec. 2.2). In this section, the ADEs of the Drude-Lorentz model are discretized according to the DG formalism[1]. We consider a Drude-Lorentz metal with one Drude and one Lorentz pole of which the electric permittivity is given by

$$\epsilon(\omega) = \epsilon_\infty - \frac{\omega_D^2}{\omega\left(\omega + \mathrm{i}\gamma_D\right)} + \frac{\Delta\epsilon_L \cdot \omega_L^2}{\omega_L^2 - \mathrm{i}\gamma_L\omega - \omega^2} \,. \tag{4.1}$$

For convenience, we collect the set of PDEs emerging from the ADE formalism of permittivity Eq. (4.1), which were previously derived in Sec. 2.2:

$$\partial_t \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) = \epsilon_\infty^{-1} \cdot \left(\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) - \vec{\mathrm{j}}_D(\vec{\mathrm{r}}, t) - \vec{\mathrm{j}}_L(\vec{\mathrm{r}}, t)\right) \,, \tag{4.2a}$$

$$\partial_t \vec{\mathrm{j}}_D(\vec{\mathrm{r}}, t) = \omega_D^2 \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) - \gamma_D \vec{\mathrm{j}}_D(\vec{\mathrm{r}}, t) \,, \tag{4.2b}$$

$$\partial_t \vec{\mathrm{j}}_L(\vec{\mathrm{r}}, t) = \Delta\epsilon_L \omega_L^2 \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) + \vec{\mathrm{q}}_L(\vec{\mathrm{r}}, t) \,, \tag{4.2c}$$

$$\partial_t \vec{\mathrm{q}}_L(\vec{\mathrm{r}}, t) = -\omega_L^2 \vec{\mathrm{j}}_L(\vec{\mathrm{r}}, t) - \gamma_L \Delta\epsilon_L \omega_L^2 \vec{\mathrm{E}}(\vec{\mathrm{r}}, t) - \gamma_L \vec{\mathrm{q}}_L(\vec{\mathrm{r}}, t) \,. \tag{4.2d}$$

[1] The formulation for the Drude model is implicitly contained in the Drude-Lorentz model and obtained by neglecting the terms emerging from the Lorentz pole.

We arrive at the discretization of Eq. (4.2) in a straightforward manner following the same basic steps as presented in Sec. 3.1. The state vector $\mathbf{q}$ is extended to hold the additional DOFs of the ADEs. Hence, each auxiliary field is expanded in terms of Lagrange polynomials in full analogy to the electromagnetic fields (cf. Eq. (3.16)). The ADEs of the Drude and the Drude-Lorentz model do not include spatial derivatives. Consequently, merely the mass matrix appears in the discretization of ADEs, and the flux is not affected at all. For brevity, we omit a general form and immediately include the simplifications arising from employing linear elements and homogeneous material properties on individual elements which yields

$$\partial_t \tilde{\vec{\mathrm{E}}}^{\triangle} = \frac{1}{\epsilon_\infty^{\triangle}} \cdot \left(\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle} - \tilde{\vec{\mathrm{j}}}_D^{\triangle} - \tilde{\vec{\mathrm{j}}}_L^{\triangle} + \mathcal{L}_f \cdot \tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle} \right) , \tag{4.3a}$$

$$\partial_t \tilde{\vec{\mathrm{j}}}_D^{\triangle} = \left(\omega_D^{\triangle} \right)^2 \cdot \tilde{\vec{\mathrm{E}}}^{\triangle} - \gamma_D^{\triangle} \cdot \tilde{\vec{\mathrm{j}}}_D^{\triangle} , \tag{4.3b}$$

$$\partial_t \tilde{\vec{\mathrm{j}}}_L^{\triangle} = \Delta\epsilon_L^{\triangle} \left(\omega_L^{\triangle} \right)^2 \tilde{\vec{\mathrm{E}}}^{\triangle} + \tilde{\vec{\mathrm{q}}}_L^{\triangle} , \tag{4.3c}$$

$$\partial_t \tilde{\vec{\mathrm{q}}}_L^{\triangle} = - \left(\omega_L^{\triangle} \right)^2 \tilde{\vec{\mathrm{j}}}_L^{\triangle} - \gamma_L^{\triangle} \Delta\epsilon_L^{\triangle} \left(\omega_L^{\triangle} \right)^2 \tilde{\vec{\mathrm{E}}}^{\triangle} - \gamma_L^{\triangle} \tilde{\vec{\mathrm{q}}}_L^{\triangle} . \tag{4.3d}$$

In practice, the core algorithm of Sec. 3.1 does not need to be modified except for the additional current density terms in Eq. (4.3a). The additional DOFs must be stored for metallic elements only which generally results in a memory efficient approach [15]. The ADEs of the Drude and Drude-Lorentz model induce little computational effort, as expensive matrix-vector-products do not occur.

4.2 Sources

Initial conditions in form of initial values of the DOFs can be considered as the most simplistic source for a time-domain method. No cost arises from this type of source as, in any case, the DOFs must be initialized at the start of a simulation. Unfortunately, initial conditions provide only limited flexibility and are — for our purposes — employed for simple (test) systems only, e. g., for the preparation of a particular mode in the empty cubic cavity of Sec. 5.3.1.

For realistic physical setups, sources of various spatial and temporal dependencies are required, e. g., plane waves, waveguide modes, or electron sources. From the sources of Maxwell's equations, i. e., ρ and $\vec{\mathrm{j}}$ (cf. Eq. (2.1)), merely the current density on the right-hand side of Eq. (2.12a) remains applicable due to the reduction to the curl equations. Another ansatz is given by the coupling of adjacent elements.

In this section, the total-field/scattered-field and the scattered-field source are introduced which employ both principle approaches and allow to include almost arbitrary electromagnetic fields. Both sources rely on the linearity of Maxwell's equations which allows us to split the total field, i. e., the physical field, labeled "tot", into the incident field "inc" and the scattered field "scat" by

$$\begin{aligned} \vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}}) &= \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}) + \vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}}) \ , \\ \vec{\mathrm{H}}_{\mathrm{tot}}(\vec{\mathrm{r}}) &= \vec{\mathrm{H}}_{\mathrm{inc}}(\vec{\mathrm{r}}) + \vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}}) \ . \end{aligned} \tag{4.4}$$

Note that Eq. (4.4) holds in both the time- and the frequency-domain. Here, the incident fields are considered to be known for all space and time while the scattered and, hence, the total fields are unknowns. As a consequence, the scattered fields can immediately be retrieved in terms of the total fields, and vice versa.

4.2.1 Total-Field/Scattered-Field Source

The total-field/scattered-field (TF/SF) source is a well known technique in the field of finite element methods. It is based on the modification of the coupling of (adjacent) elements in a mesh and commonly applied to inject, e. g., plane waves [17].

Following Ref. [15], the computational domain is divided into the total-field region and the scattered-field region. In the total-field region, we solve for the total field and equivalently in the scattered-field region for the scattered field. Due to the linearity of Maxwell's equations *and* assuming that the incident field implicitly fulfills Maxwell's equations, we can solve

$$\begin{aligned} \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_{\mathrm{tot}}(\vec{\mathrm{r}}, t) \;+ \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_{\mathrm{tot}}) \;&= 0\,, \\ \mathcal{Q}(\vec{\mathrm{r}}) \cdot \partial_t \mathbf{q}_{\mathrm{scat}}(\vec{\mathrm{r}}, t) + \vec{\nabla} \cdot \vec{\mathrm{F}}(\mathbf{q}_{\mathrm{scat}}) &= 0\,, \end{aligned} \tag{4.5}$$

in the total- and scattered-field region, respectively. As a fundamental principle of the DG formalism, elements only couple to adjacent neighbors via the numerical flux. In particular, for Maxwell's equations, information is exchanged between elements by the field differences (cf. Eq. (3.11)). Within the total-field and the scattered-field region, the field differences can be evaluated as usual (cf. Eq. (3.14)).

At the interface of both regions, i. e., on the TF/SF contour, elements of both domains share a common face and the field values of mutual neighbors cannot be used to determine the field difference directly. In order to evaluate the field difference of an element in the TF region, for example, the total field of the neighbor is required. However, the neighbor is located in the scattered-field region and merely provides the scattered field. Employing Eq. (4.4), this issue is readily resolved, and the field differences can be calculated by

$$\begin{aligned} \Delta\vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}}, t) &= \vec{\mathrm{E}}^{+}_{\mathrm{scat}}(\vec{\mathrm{r}}, t) - \vec{\mathrm{E}}^{-}_{\mathrm{tot}}(\vec{\mathrm{r}}, t) \;+ \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)\;, \\ \Delta\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}}, t) &= \vec{\mathrm{E}}^{+}_{\mathrm{tot}}(\vec{\mathrm{r}}, t) \;- \vec{\mathrm{E}}^{-}_{\mathrm{scat}}(\vec{\mathrm{r}}, t) - \vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)\;. \end{aligned} \tag{4.6}$$

In essence, $\vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)$ is either added or subtracted to restore the total field from the scattered field, and vice versa. A similar expression holds for the magnetic field.

The algorithm of Sec. 3.1 needs only slight adaption to employ the TF/SF source. The calculation of the field difference for elements at the TF/SF contour must be performed by Eq. (4.6) instead of Eq. (3.14) which renders the TF/SF technique an ideal extension to the DGTD method [15]. Field distributions for a number of relevant sources, i. e., analytic expressions for $\vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)$ and $\vec{\mathrm{H}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)$, are given in many textbooks, e. g., Refs. [7, 15, 30, 32].

4.2.2 Scattered-Field Source

The scattered-field (SF) source technique relies on the decomposition of the total fields $\vec{\mathrm{E}}_{\mathrm{tot}}(\vec{\mathrm{r}}, \omega)$ and $\vec{\mathrm{H}}_{\mathrm{tot}}(\vec{\mathrm{r}}, \omega)$ into known incident fields and unknown scattered fields according to Eq. (4.4). In contrast to the TF/SF source, the computational domain is not decomposed into a total-field and a scattered-field region [17], but the entire domain is regarded a scattered-field domain. In particular, only the scattered fields $\vec{\mathrm{E}}_{\mathrm{scat}}(\vec{\mathrm{r}}, t)$ and $\vec{\mathrm{H}}_{\mathrm{scat}}(\vec{\mathrm{r}}, t)$ are evolved in time, and the incident fields $\vec{\mathrm{E}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)$ and $\vec{\mathrm{H}}_{\mathrm{inc}}(\vec{\mathrm{r}}, t)$ do not propagate within the grid, as is the case for the TF/SF source. As a consequence, the total field is not available from the grid and must be calculated a posteriori to the simulation, if required. The formulation of the scattered-field source is developed in the following, focusing on the electric field. Similar expressions for the magnetic field follow analogously in a straightforward manner.

Assuming no external sources, the curl equation of the total electric field for linear dispersive materials modeled by Eq. (2.16) in the frequency-domain reads

$$-\,\mathrm{i}\omega\left(\epsilon_\infty + \chi(\omega)\right)\cdot \vec{\mathrm{E}}_{\mathrm{tot}}\big(\vec{\mathrm{r}},\omega\big) = \vec{\nabla}\times\vec{\mathrm{H}}_{\mathrm{tot}}\big(\vec{\mathrm{r}},\omega\big)\ . \tag{4.7}$$

Similarly, the curl equation of the incident field in vacuum is given by

$$-\,\mathrm{i}\omega\vec{\mathrm{E}}_{\mathrm{inc}}\big(\vec{\mathrm{r}},\omega\big) = \vec{\nabla}\times\vec{\mathrm{H}}_{\mathrm{inc}}\big(\vec{\mathrm{r}},\omega\big)\ . \tag{4.8}$$

Due to the linearity of Maxwell's equations, Eq. (4.8) can be subtracted from Eq. (4.7) [17]. Substituting Eq. (4.4), we obtain

$$-\,\mathrm{i}\omega\epsilon_\infty\vec{\mathrm{E}}_{\mathrm{scat}}\big(\vec{\mathrm{r}},\omega\big) = \vec{\nabla}\times\vec{\mathrm{H}}_{\mathrm{scat}}\big(\vec{\mathrm{r}},\omega\big) + \mathrm{i}\omega\chi(\omega)\cdot\vec{\mathrm{E}}_{\mathrm{tot}}\big(\vec{\mathrm{r}},\omega\big) + \mathrm{i}\omega\left(\epsilon_\infty - 1\right)\cdot\vec{\mathrm{E}}_{\mathrm{inc}}\big(\vec{\mathrm{r}},\omega\big)\ . \tag{4.9}$$

An inverse Fourier transform yields the desired time-domain formulation of the scattered-field source. Employing the ADE formalism, the term involving the susceptibility is absorbed in a polarization current density:

$$\partial_t\vec{\mathrm{E}}_{\mathrm{scat}}\big(\vec{\mathrm{r}},t\big) = \epsilon_\infty^{-1}\left(\vec{\nabla}\times\vec{\mathrm{H}}_{\mathrm{scat}}\big(\vec{\mathrm{r}},t\big) + \vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},t\big)\right) + \left(\epsilon_\infty^{-1} - 1\right)\partial_t\vec{\mathrm{E}}_{\mathrm{inc}}\big(\vec{\mathrm{r}},t\big)\ . \tag{4.10}$$

Here, $\vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},t\big)$ is obtained by an inverse Fourier transform of $\mathrm{i}\omega\chi(\omega)\cdot\vec{\mathrm{E}}_{\mathrm{tot}}\big(\vec{\mathrm{r}},\omega\big)$. In order to avoid a convolution integral in the time-domain, $\chi(\omega)$ must be a rational function with respect to $\mathrm{i}\omega$ (cf. Sec. 2.2.1) and the expression must be suitably rearranged before transformation. Note that $\vec{\mathrm{j}}_p\big(\vec{\mathrm{r}},t\big)$ represents the polarization current density of the *total* field. This implies, that additional ADEs besides those introduced by the dispersive material itself are *not* required.

Due to the term $(\epsilon_\infty^{-1}-1)\partial_t\vec{\mathrm{E}}_{\mathrm{inc}}\big(\vec{\mathrm{r}},t\big)$, Eq. (4.10) must be evaluated for all elements in a mesh modeling $\epsilon_\infty \neq 1$. In particular, the time-derivative of the incident field must be calculated even for non-dispersive materials which can — but need not necessarily — render the SF source time-expensive.

Scattered-Field Source for the Discretized Drude-Lorentz Model

Due to the analogy to the ADE formalism (cf. Eqs. (4.10) and (2.20)), we can immediately formulate the SF source for Drude and Drude-Lorentz metals. The time-domain formulation of the polarization current density is identical to Eqs. (4.2b)-(4.2d), acknowledging the fact that the driving electric field is given by the total field $\vec{\mathrm{E}}_{\mathrm{tot}}\big(\vec{\mathrm{r}},t\big)$. Including the incident field term of Eqs. (4.10), we obtain

$$\partial_t\tilde{\vec{\mathrm{E}}}^{\triangle} = \frac{1}{\epsilon_\infty^{\triangle}}\cdot\left(\vec{\mathcal{D}}^{\triangle}\times\tilde{\vec{\mathrm{H}}}^{\triangle} - \tilde{\vec{\mathrm{j}}}_D^{\triangle} - \tilde{\vec{\mathrm{j}}}_L^{\triangle} + \mathcal{L}_f\cdot\tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle}\right) + \left(\frac{1}{\epsilon_\infty^{\triangle}} - 1\right)\cdot\partial_t\tilde{\vec{\mathrm{E}}}_{\mathrm{inc}}^{\triangle}\,, \tag{4.11a}$$

$$\partial_t\tilde{\vec{\mathrm{j}}}_D^{\triangle} = \left(\omega_D^{\triangle}\right)^2\cdot\left(\tilde{\vec{\mathrm{E}}}^{\triangle} + \tilde{\vec{\mathrm{E}}}_{\mathrm{inc}}^{\triangle}\right) - \gamma_D^{\triangle}\cdot\tilde{\vec{\mathrm{j}}}_D^{\triangle}\,, \tag{4.11b}$$

$$\partial_t\tilde{\vec{\mathrm{j}}}_L^{\triangle} = \Delta\epsilon_L^{\triangle}\left(\omega_L^{\triangle}\right)^2\left(\tilde{\vec{\mathrm{E}}}^{\triangle} + \tilde{\vec{\mathrm{E}}}_{\mathrm{inc}}^{\triangle}\right) + \tilde{\vec{\mathrm{q}}}_L^{\triangle}\,, \tag{4.11c}$$

$$\partial_t\tilde{\vec{\mathrm{q}}}_L^{\triangle} = -\left(\omega_L^{\triangle}\right)^2\tilde{\vec{\mathrm{j}}}_L^{\triangle} - \gamma_L^{\triangle}\Delta\epsilon_L^{\triangle}\left(\omega_L^{\triangle}\right)^2\left(\tilde{\vec{\mathrm{E}}}^{\triangle} + \tilde{\vec{\mathrm{E}}}_{\mathrm{inc}}^{\triangle}\right) - \gamma_L^{\triangle}\tilde{\vec{\mathrm{q}}}_L^{\triangle}\,, \tag{4.11d}$$

where we dropped the label of the scattered field for convenience.

For $\epsilon_\infty^{\triangle} = 1$, the last term in Eq. (4.11a) vanishes which holds for most of the published material models, e. g., Ref. [38]. The previously developed algorithm requires minor modification to include the SF source. In particular, only the terms involving the incident field have to be added which contain arithmetically light vector operations.

4.3 Perfectly Matched Layers

Many setups of interest comprise nanostructures (on substrate) in free space. Light impinging onto theses structures is scattered into all directions and leaves the vicinity of the experimental setup. When treated in a numerical method, the setup of the numerical simulation must ensure a similar behavior, i. e., light radiated into free space may not return or interact with the structure again.

As a volume method, the DG method does not inherently support infinitely extended systems due to the locality of the element's basis functions and the truncation of the computational domain. Consequently, free space must be "created" by absorbing electromagnetic radiation of arbitrary wavelength, polarization, and direction of propagation.

Waves impinging onto the boundary of the computational domain act according to boundary conditions (cf. Sec. 3.1.5). Thus, a straightforward approach is to apply absorbing boundary conditions (ABCs). The class of analytic ABCs (AABCs) provides physical boundary conditions which support outgoing radiation and suppresses reflection, e. g., Silver-Müller boundary conditions [19]. Unfortunately, AABCs provide insufficient absorption under oblique incidence [17].

A conceptually different approach was introduced by Bérenger [58]. Instead of applying (only) boundary conditions, an artificial — not necessarily physical — medium is introduced in between the physical region of interest and the boundary of the computational domain. In other words, the physical region is enclosed by a shell of which the material is specifically designed. The artificial medium attenuates propagating waves and prohibits reflection at the interface by matching the impedance of the physical region at the shared interface (cf. Fig. 4.1). Hence, this ansatz is also referred to by perfectly matched layers (PMLs). In the following, two different approaches, uniaxial PMLs (UPMLs) and stretched coordinate PMLs (SC-PMLs), are addressed.

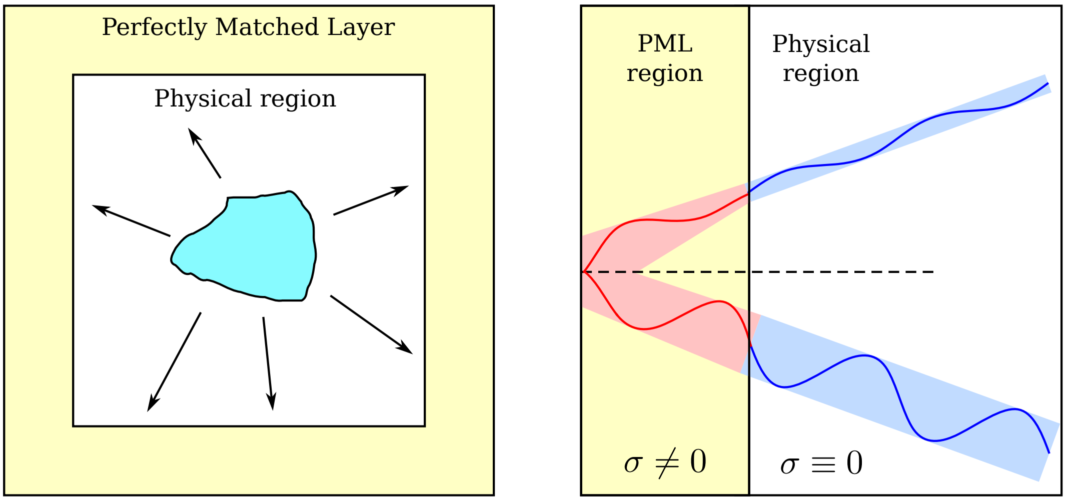


Figure 4.1: *Principle of perfectly matched layers. An artificial medium (yellow) enclosing the physical region of interest absorbs propagating radiation and prohibits reflection at the shared interface due to impedance matching. Attenuation within the PML region is visualized in the right panel (reproduced from Ref. [15]). Note that no reflected waves occur at the interface.*

4.3.1 Uniaxial Perfectly Matched Layers

For a PML, two aspects must be achieved: Impedance matching and absorption within the PML layer. In the context of UPMLs, absorption is taken into account by the dispersive, anisotropic material tensors [15, 22]

$$\underline{\epsilon}' \equiv \underline{\lambda}\epsilon\,, \quad \underline{\mu}' \equiv \underline{\lambda}\mu\,, \quad \underline{\lambda} \equiv \begin{pmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_x} \end{pmatrix} \tag{4.12}$$

with the complex, dispersive parameters

$$s_k(\omega) \equiv 1 - \frac{\sigma_k}{\mathrm{i}\omega}\,, k \in \{x, y, z\}\,. \tag{4.13}$$

Here, ϵ and μ represent the material parameters of the adjacent physical region which are emulated in the PML to achieve impedance matching. In Eq. (4.13), the real-valued σ_k controls the damping along direction k.

As thoroughly examined in Ref. [15], the ADE technique can be applied to handle the dispersive property of the material tensors $\underline{\epsilon}'(\omega)$ and $\underline{\mu}'(\omega)$, following the principle steps of Sec. 2.2.1. Assuming dispersionless material properties ϵ and μ, this results in one ADE for each electromagnetic field component, i. e., for the E_x-component, the set of PDEs reads

$$\begin{aligned} \partial_t E_x(\vec{\mathrm{r}}, t) &= \epsilon^{-1} \left[\vec{\nabla} \times \vec{\mathrm{H}}(\vec{\mathrm{r}}, t) \right]_x + \left(\sigma_x - \sigma_y - \sigma_z \right) \cdot E_x(\vec{\mathrm{r}}, t) - \epsilon^{-1} p_x(\vec{\mathrm{r}}, t)\ , \\ \partial_t p_x(\vec{\mathrm{r}}, t) &= \left(\sigma_x^2 - \sigma_x\sigma_y - \sigma_x\sigma_z + \sigma_y\sigma_z \right) \cdot \epsilon E_x(\vec{\mathrm{r}}, t) - \sigma_x p_x(\vec{\mathrm{r}}, t)\ , \end{aligned} \tag{4.14}$$

where p_x denotes an auxiliary field for the E_x-component, and $[\cdot]_x$ returns the x-component of the vector in square brackets. Note that outside of the PML region, $\sigma_k = 0$ and Eq. (4.14) reduces to the set of equations for the material parameters ϵ and μ, i. e., Eq. (4.14) must be evaluated inside the PML region only.

The discretization of Eq. (4.14) by the DG method (cf. Sec. 3.1) follows straightforwardly which yields [15]:

$$\begin{aligned} \partial_t \tilde{\mathrm{E}}_x^{\triangle}(t) &= \frac{1}{\epsilon^{\triangle}} \left[\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle}(t) \right]_x + \left(\sigma_x^{\triangle} - \sigma_y^{\triangle} - \sigma_z^{\triangle} \right) \cdot \tilde{\mathrm{E}}_x^{\triangle}(t) - \frac{1}{\epsilon^{\triangle}} \tilde{\mathrm{p}}_x^{\triangle}(t) + \mathcal{L}_f \cdot \tilde{\vec{\mathrm{F}}}_{x,f}^{\mathrm{E},\triangle}\ , \\ \partial_t \tilde{\mathrm{p}}_x^{\triangle}(t) &= \left((\sigma_x^{\triangle})^2 - \sigma_x^{\triangle}\sigma_y^{\triangle} - \sigma_x^{\triangle}\sigma_z^{\triangle} + \sigma_y^{\triangle}\sigma_z^{\triangle} \right) \cdot \epsilon^{\triangle} \tilde{\mathrm{E}}_x^{\triangle}(t) - \sigma_x^{\triangle} \tilde{\mathrm{p}}_x^{\triangle}(t)\ . \end{aligned} \tag{4.15}$$

As before, the discretized formulation assumes constant material parameters $\sigma_k^{\triangle}$ within an element $\triangle$, and the additional DOFs of the ADE are to be absorbed in the state vector $\tilde{\mathbf{q}}$. The ADE in Eq. (4.14) does not contain derivatives with respect to space and, hence, there are no additional matrix-vector-products contained in Eq. (4.15) which renders the UPML formulation computationally cheap.

As the ADE formalism is applied to the material tensors $\underline{\epsilon}'(\omega)$ and $\underline{\mu}'(\omega)$ and not to $\underline{\lambda}(\omega)$, the properties of ϵ and μ are explicitly taken into account during the derivation of the ADEs. Consequently, the resulting ADEs directly depend on the material of the physical region which is to be matched.

Note that the equations above are valid for dispersionless media only. An UPML formulation and subsequent discretization according to the DG method for dispersive media is provided in, e. g., Ref. [15]. A numerical study of the free parameters σ_k in the context of the DG method is provided in Ref. [22].

4.3.2 Stretched Coordinate Perfectly Matched Layers

A disadvantage of UPMLs is their direct dependence on the material to match, i. e., the ADE formalism must be derived and subsequently be discretized for each $\epsilon(\omega)$ and $\mu(\omega)$. In addition, absorption of evanescent waves does not work optimal [17, 59]. The SC-PML approach, originally proposed in Ref. [60] and recently applied to the DG method [61], overcomes these problems.

In SC-PMLs, absorption is achieved via a transform of space which maps a real-valued position $\vec{\mathrm{r}}$ to a complex-valued vector [58]. Consequently, propagating fields — commonly described by $\exp(\mathrm{i}\vec{\mathrm{k}}\cdot\vec{\mathrm{r}})$ — are attenuated due to the complex component of the transformed position vector. Within the DG framework, the transformation is introduced by a modification of the spatial derivatives [61]:

$$\frac{\partial}{\partial x} \to \frac{1}{s_x(\omega)}\frac{\partial}{\partial x}\,, \quad \frac{\partial}{\partial y} \to \frac{1}{s_y(\omega)}\frac{\partial}{\partial y}\,, \quad \frac{\partial}{\partial z} \to \frac{1}{s_z(\omega)}\frac{\partial}{\partial z}\,. \tag{4.16}$$

Here s_k, $k \in \{x, y, z\}$ is a complex, dispersive stretching factor which is typically defined as [17]

$$s_k(\omega) = \kappa_k - \frac{\sigma_k}{\mathrm{i}\omega - \alpha_k}\,. \tag{4.17}$$

It can be shown that the stretching factors do not lead to reflections [17, 59]. The free parameters κ_k, σ_k, and α_k are to be optimized for performance. Ref. [61] provides a numerical study of the parameters for a three-dimensional reference system.

The ADE approach can be employed to formulate the modified set of Maxwell's equations in the time-domain, as described in detail in Refs. [15, 61]. For the E_x-component, the ADE formalism yields

$$\begin{aligned}
\partial_t E_x(\vec{\mathrm{r}},t) &= \epsilon^{-1}\Big(\partial_y H_z(\vec{\mathrm{r}},t) - \partial_z H_y(\vec{\mathrm{r}},t) - G^E_{xy}(\vec{\mathrm{r}},t) - G^E_{xz}(\vec{\mathrm{r}},t)\Big)\ , \\
\partial_t G^E_{xy}(\vec{\mathrm{r}},t) &= \sigma_y \partial_y H_z(\vec{\mathrm{r}},t) - \Big(\alpha_y + \sigma_y\Big)\cdot G^E_{xy}(\vec{\mathrm{r}},t)\ , \\
\partial_t G^E_{xz}(\vec{\mathrm{r}},t) &= -\sigma_z \partial_z H_y(\vec{\mathrm{r}},t) - (\alpha_z + \sigma_z)\cdot G^E_{xz}(\vec{\mathrm{r}},t)\ ,
\end{aligned} \tag{4.18}$$

where $\kappa_k = 1$ is assumed without loss of generality [15]. Similar expressions hold for all other electromagnetic field components.

For each electromagnetic field component, two ADEs are required which both involve spatial derivatives. These derivatives result in computationally expensive matrix-vector-products and, in addition, induce a modification of the numerical flux. As the SC-PMLs are not employed in the context of this thesis, we omit the DG discretized form and refer the reader to Refs. [14, 15, 61].

As alluded to above, SC-PMLs implicitly fulfill impedance matching irrespective of the material properties ϵ and μ. In comparison to UPMLs, SC-PMLs require more ADEs — at least for dispersionless materials — which is accompanied by an increased memory consumption and significantly increased computational effort. However, due to the enhanced technique, SC-PMLs show an attenuation increased by an order of magnitude compared to UPMLs and can be applied to dispersive and even nonlinear materials without adaption.

The physical systems investigated in this thesis are located on substrates and embedded in air. Both these media are conveniently modeled by dispersionless dielectrics and, hence, UPMLs are employed in conjunction with Silver-Müller boundary conditions throughout the thesis.

5 Chapter 5

DGTD on Graphics Processing Units

During the last decade, graphics processing units (GPUs) became popular accelerators for scientific applications. Maxwell's equations by the DGTD method [27], seismic wave propagation by the spectral-element method [28], and the investigation of incompressible flow dynamics [29] are only a few examples from the field of computational physics on GPUs. GPUs offer tremendous computational power for single precision floating point arithmetics and memory bandwidth at low cost. Both were steadily improved over the last decade and grew at a higher rate compared to Intel's CPUs as depicted in Fig. 5.1. Today, the computational power for single precision arithmetic and the memory bandwidth offered by GPUs is an order of magnitude larger compared to CPUs, which renders GPUs ideal devices for computationally intense tasks. The vendors of GPU hardware recently began to address the developer's needs for high performance double precision routines, bridging a serious drawback of GPUs for accurate scientific calculations.

As a matter of fact, the GPU ecosystem is at a stage of rapid growth and vital evolution. This means that, in contrast to the well established environment of CPU programming, hardware architectures, programming models, libraries, and application programming interfaces (APIs) such as `OpenCL` or `CUDA` are subject to "sudden" modification. As a consequence, code designed to fit the requirements of a specific hardware architecture is quickly outdated and might need refactoring to exploit the capabilities of new hardware. Several approaches to adapt existing code bases to the GPU hardware requirements have been developed, e. g., `OpenACC` [62], but a standard has not yet been established. Libraries similar to those known from CPU programming, e. g., `BLAS` or `LAPACK`, have been implemented for the hardware of specific vendors, binding the developer to a vendor specific API. For example, `cuBLAS` [63] and `CULA` [64] are math libraries optimized for NVidia devices based on the `CUDA` programming language.

"`OpenCL` (Open Computing Language) is an open royalty-free standard for general purpose parallel programming across CPUs, GPUs and other processors, giving software developers portable and efficient access to the power of these heterogeneous processing platforms." [65]. As a cross platform standard, `OpenCL` can be employed for any given hardware as long as the vendors support the standard by providing drivers. Often, these drivers lag behind the latest version of the `OpenCL` standard. In contrast, GPU vendors release evolved hardware side-by-side with updated drivers and adapted programming languages on a regular basis, providing full support for their own development package. The evolution of the `OpenCL` standard is not bound to follow these modifications, resulting — if at all — in delayed adaption with respect to vendors on the one hand, but long term backward compatibility on the other. In order to remain independent of a particular vendor, we choose `OpenCL` [65].

Developing new code or porting existing code to GPU hardware is time-consuming and often not

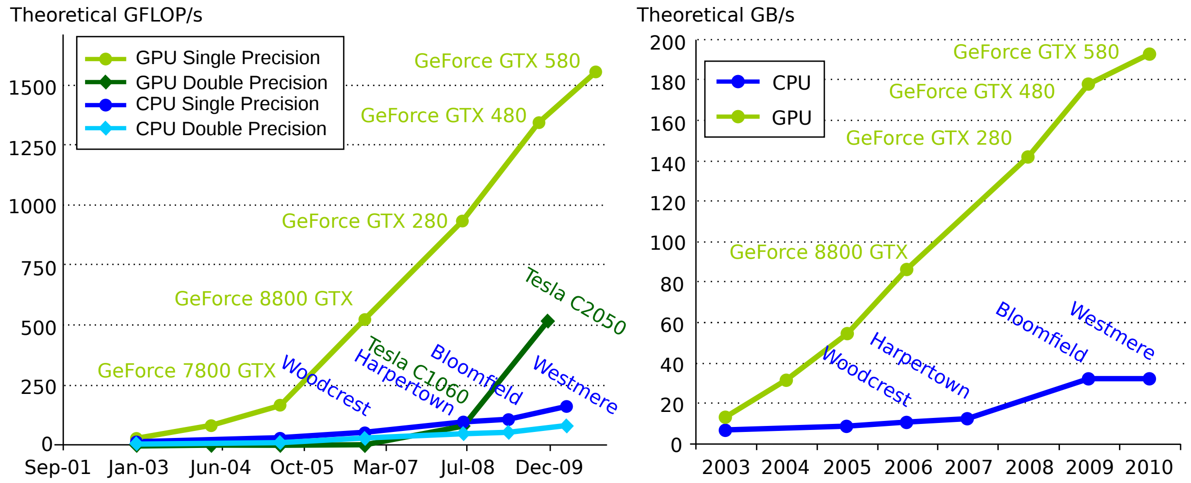


Figure 5.1: *Evolution of computational power (left panel) and memory bandwidth (right panel) of NVidia graphics processing units and Intel CPUs over the last decade. The graphs are reproduced from [26].*

feasible in a straightforward way, as the programming languages employed to write GPU compliant code do not conform with standard languages such as `C/C++` or `Fortran`. For example, `OpenCL` uses a subset of `ISO C99`, and NVidia's `CUDA` programming language supports `CUDA C` and `CUDA Fortran`, languages similar to `C` and `Fortran` with GPU specific extensions, respectively. As a consequence, large parts of legacy code must often be reimplemented as long as suitable libraries are not available.

The computational effort of the DGTD method is dominated by the evaluation of many rather small matrix-vector-products (cf. Sec. 3.1.7). Unfortunately, math libraries for application on GPU devices are usually designed for large matrix-vector-products and perform poorly for small matrices. Consequently, these fundamental math operations must be implemented by hand, as demonstrated in the first DGTD implementation of Maxwell's equations employing the `CUDA` programming language by Klöckner *et al.* [27].

5.1 GPU Basics

As alluded to in the introduction of this chapter, fundamental math operations must be implemented by hand. This requires insight into the concepts and work flow of GPUs. Therefore, we must account for the basic idea of the GPU and its design paradigms described in Sec. 5.1.1. As we implement the DGTD method for NVidia hardware, the compute unified device architecture (`CUDA`), NVidia's platform for general purpose GPU (GPGPU) programming, is presented in Sec. 5.1.2, which comprises the fundamental concepts and work flow of Fermi based devices. From the principles of optimal performance in Sec. 5.1.3 and the concepts of kernel design in Sec. 5.1.4, we deduce performance relevant guidelines for the implementation of the DGTD method which are summarized in Sec. 5.1.5.

5.1.1 The Graphics Processing Unit Design Paradigm

By design, a graphics processing unit is a device for highly parallel, arithmetically intense computations. From this "paradigm" the hardware design of GPUs is derived. In comparison to CPUs, much more

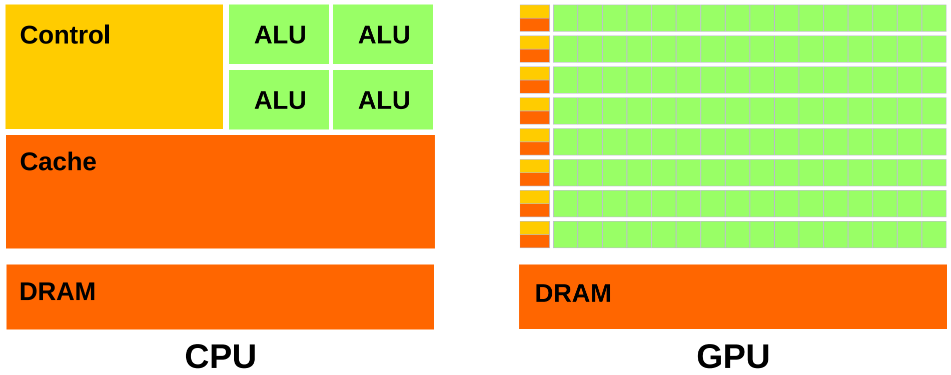


Figure 5.2: *Schematic design of CPU (left panel) and GPU (right panel) hardware. In contrast to the CPU, the GPU devotes much more transistors to ALUs and much less to cache and cache management (control). The graphs are reproduced from [26].*

transistors are spent on arithmetic logic units (ALUs) and much less on caching or flow control as schematically visualized in Fig. 5.2. Little cache size for a large amount of ALUs seems surprising at first glance, but it is at the heart of GPU's design to hide load latency, i. e., the time required to transfer data to and from memory, by a large number of arithmetic operations rather than big cache, as is the case in CPU design. The concept of latency hiding by means of hardware multithreading is described in detail in Sec. 5.1.2.

The GPU is designed for data parallel operation, i. e., an operation is executed on a large set of independent data at a time. Consequently, only those algorithms which provide sufficient data parallelism *and* high arithmetic intensity can fully utilize the GPU's power. An analysis of the DGTD method with respect to, amongst others, data parallelism and arithmetic intensity is conducted in Sec. 5.2.1.

5.1.2 NVidia's Compute Unified Device Architecture

The compute unified device architecture (`CUDA`) is NVidia's platform for general purpose applications on GPUs. It defines hardware mechanisms and serves as an interface for APIs such as the `CUDA` programming languages `CUDA C` and `OpenCL`. In the following, the key features of `CUDA` GPUs are summarized and a detailed description of the mechanisms realizing the GPU design paradigm is presented. `CUDA` and the `OpenCL` architecture are closely related in their entities although, in principle, they are completely independent of each other. As we use the `OpenCL` API to implement the DGTD method on `CUDA` based hardware, both NVidia and `OpenCL` terminology is introduced side-by-side and emphasized by special formatting: *CUDA term* and *`OpenCL term`*.

A Scalable Programming Model by Hierarchical Hardware

While hardware is evolving fast in comparison to large (legacy) code bases, it is generally undesirable to adapt a given code basis upon change of the executing hardware. `CUDA` accounts for this need by a scalable programming model and hierarchically organized hardware as follows.

A data parallel program is executed by mapping mutually independent data to *threads* (*`work-items`*), i. e., each datum is treated by a single, addressable thread. A program will typically issue hundreds if not thousands of threads at the same time; not all are executed at the same time, though. Threads are organized in groups, referred to as *thread blocks* (TB) or *`work groups`*. thread blocks are themselves

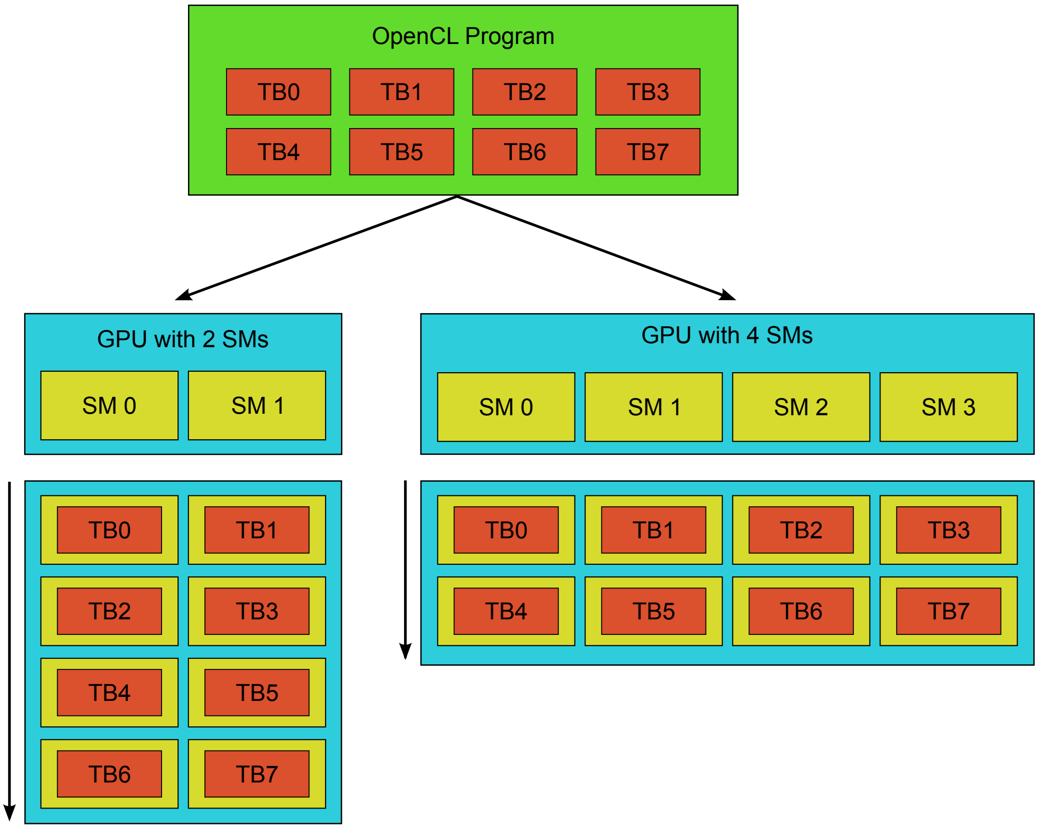


Figure 5.3: *Distribution of work in the scalable programming model. The thread blocks (TB) of the `OpenCL` program are distributed among the available streaming multiprocessors (SM) by the hardware. The horizontal distribution of the TBs indicates parallelization among the SMs, while the vertical distribution shows their execution in time. Several TBs can simultaneously reside and be processed on a single SM which, for clarity's sake, is not depicted.*

organized in a *grid of thread blocks* (*`NDRange`*). Consequently, a three level hierarchy comprising threads, thread blocks, and a grid[1] is established.

These purely conceptual entities of the `CUDA` (`OpenCL`) programming model are mapped to `CUDA` hardware entities to implement the scalable programming model. GPU hardware compliant to `CUDA` is structured in *`CUDA` cores* (*`processing elements`*), i. e., a unit performing floating point and integer operations, and *streaming multiprocessors* (SM), also referred to as *`compute units`*. In the schematic GPU of Fig. 5.2, one green square represents a single `CUDA` core that executes a single thread at a time. One "row" of `CUDA` cores, cache, and control units makes up a single streaming multiprocessor, and the entire GPU comprises 8 SMs. In the *Fermi* architecture [66], a single SM consists of 32 `CUDA` cores. The number of SMs is device specific. However, a maximum of 16 SMs per chip was implemented for Fermi based GPUs. NVidia's latest GPU architecture, the *Kepler GK110* maintains the principle structure of multiprocessors and cores. Kepler's [67] new streaming multiprocessor SMX contains 192 single precision `CUDA` cores, 64 double precision units and 32 special function units (SFU).

[1] In `OpenCL` terms: Work-items, work-groups, and NDRange.

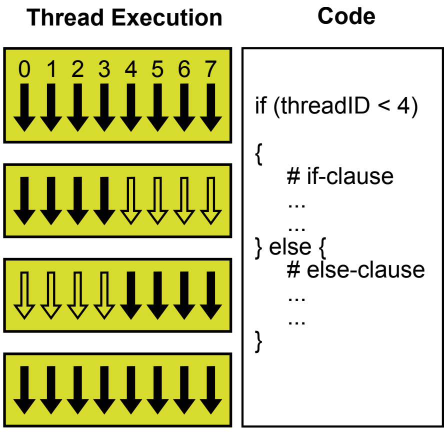


Figure 5.4: *Branching and serialization of a warp. An `if`-condition splits the execution path of the threads (arrows) in a warp*[3] *(yellow box). While the first half of the warp executes the `if`-clause (solid arrows), the second half idles (hollow arrows), and vice versa in the `else`-clause. The execution paths merge after the clauses and the threads execute synchronously again.*

Scalability of a program is achieved by distribution of thread blocks among the multiprocessors of a GPU during runtime. As sketched in Fig. 5.3, the 8 thread blocks of a grid, defined by the programmer, are scheduled automatically to the multiprocessors available on the GPU. The more SMs, the "wider" the TBs are parallelized and the faster the program is executed. The hierarchy of thread blocks mapped to multiprocessors gives vendors a possibility to enhance their devices, e. g., increase the number of SMs or improve the productivity of an SM, while code remains productive without adaption.

Single-Instruction Multiple-Thread (SIMT) Architecture and Hardware Multithreading

The challenges of implementing high-performance code for GPUs emerge from the mechanisms of the hardware, i. e., how the device distributes work among its processing units, burdens the programmer with conditions and limitations. In this section, details about the work flow of a GPU are described and performance relevant guidelines emanating thereof are deduced.

As alluded to above, a streaming multiprocessor is designed to handle hundreds and thousands of threads at the same time. In order to manage this large amount of threads, the single-instruction multiple-thread (SIMT) architecture is employed. SIMT allows the programmer to write thread-parallel code. This means that a kernel, i. e., a set of instructions similar to a function in `C/C++`, defines the operations of all threads of all thread blocks in a grid of thread blocks.

In the SIMT architecture, the SM "creates, manages, schedules, and executes threads in groups of 32 parallel threads called *warps*." [26] The number of threads in a warp corresponds to the number of `CUDA` cores in one SM of the Fermi architecture.[2] The threads of a warp are launched in parallel at the same time. However, they need not follow the same execution path, i. e., the threads of a warp may branch, e. g., by condition, and execute independently.

A warp executes a single instruction at a time. In essence, threads on a common execution path are

[2] The warp size is maintained for devices of the Kepler architecture although the number of `CUDA` cores increased to 192 per SM [67].

[3] The size of the warp has been reduced to 8 for simplification of the figure.

executed simultaneously while divergent threads of the warp pause, as sketched in Fig. 5.4. Divergent threads lead to a serialization of instructions, i. e., parts of the warp execute while other parts of the warp idle. Obviously, this is a performance relevant issue as a lot of time can be wasted upon branching of the threads in a warp. Hence, maximum efficiency is achieved when all threads in a warp follow the same execution path and do not branch.

When a kernel is launched, CUDA hardware partitions each thread block on a SM into warps and schedules the warps for execution by the *warp scheduler*. If the number of threads in the thread block is not an integer multiple of the warp size, not all threads of the last warp are utilized for computation, although they are reserved by the hardware since instructions are always issued per warp. Consequently, threads are wasted and performance diminished. However, if the number of threads in a thread block is much larger than the warp size, the performance loss becomes negligible.

Another consequence of the three-level hierarchy is the potential mismatch of the number of thread blocks and SMs. As alluded to in Sec. 5.1.2, the thread blocks of a given kernel are distributed among the SMs of a device. If the number of thread blocks is not an integer multiple of the number of SMs, the last heat of thread blocks does not fill all available SMs, and computational power is wasted. Again, this effect becomes negligible when the number of thread blocks is much larger than the number of SMs.

Hardware Multithreading

"The number of clock cycles it takes for a warp to be ready to execute its next instruction is called *latency*" [26]. Most of the time, latency occurs when an instruction's input operands are not yet available, e. g., because data is to be transferred. By design, latency on GPUs is supposed to be hidden by computation rather than big cache or smart control, as alluded to in Sec. 5.1.1. In this section, hardware multithreading, the key mechanism to achieve latency hiding on GPUs, is introduced and the consequences for the programmer aiming at high-performance code are addressed.

Full utilization of the hardware is achieved, when all cores are busy all the time, i. e., when the warp scheduler issues instructions of some warp at every clock cycle. This implies that the latency of all warps must be hidden completely. In order to implement this concept, the *execution context*, i. e., program counters, registers, etc., of each warp processed by a SM is maintained on-chip during the entire lifetime of the warp. As alluded to above, thread blocks are distributed among the SMs of a GPU and partitioned into warps by the warp scheduler, i. e., at a given time, several warps reside simultaneously on a SM. As the execution context is kept on-chip, switching from one execution context to another has no cost, i. e., no time is spent on changing the warp to be processed on a SM. Consequently, latency can be hidden by switching to a warp ready to execute while the other resident warps get prepared.

Therefore, latency can be efficiently hidden, if enough warps reside on a SM simultaneously and the arithmetic intensity of the kernel in process is sufficiently large. The first condition, referred to as *occupancy*, can be influenced by the programmer as described in detail in Sec. 5.1.4. The latter is bound to the algorithm to be executed, and the programmer has only limited design opportunity to interfere.

5.1.3 Optimal Performance

Based on the mechanisms of CUDA hardware presented in the previous sections, the most important issues related to optimal performance are addressed in this section. For every computing device, optimal performance is achieved when the bus *and* the computing units work at their peak throughput. The bus transfers data from DRAM to the compute units and vice versa (cf. Fig. 5.2). When the bus bandwidth, i. e., the number of bytes transferred per unit time, is insufficient to saturate the compute units, the latter idle and the performance of the kernel is bandwidth limited. In contrast, when the instruction throughput, i. e., the number of instructions executed per unit time, of the compute units is too small to saturate the bus, bandwidth is wasted and the kernel is bound by instruction. Thus, optimal performance is achieved when the bus and the compute units are balanced and simultaneously execute at their device specific peak values. The ratio of instruction throughput and bus bandwidth is referred to as *instruction to byte (I/B) ratio*. Each device has a specific optimum I/B ratio, e. g., the NVidia® GeForce® GTX580, the device on which all computations of this thesis were performed, executes one single precision floating point instruction (e. g., add, multiply, or fused multiply-add (FMA)) each clock cycle [66]. Employing all 512 cores at 1544 MHz results in $790 \cdot 10^9$ instructions per second. Dividing this by the maximum bandwidth of 192.4 GByte/s [68] yields the theoretical, hardware specific, optimum I/B ratio of approximately 4.1 single precision floating point instructions per byte transferred over the bus.

In order to familiarize ourselves with the I/B ratio, some matrix-vector operations required in the DGTD method are examined. The matrix-vector-product is the computationally most intense part of the DGTD method. For example, the I/B ratio of an $m \times n$ matrix being multiplied to a vector of n components whose result is written back to memory (DRAM) is given by[4]

$$\mathrm{I/B}_{[m\times n]\cdot n} = \frac{m \cdot n}{(m+2)\cdot n \cdot 4} \xrightarrow{m,n\gg 1} \frac{1}{4}\,. \tag{5.1}$$

For each of the $m \cdot n$ entries of the matrix, a single FMA instruction performing $y \leftarrow y + a \cdot b$ is required to compute the result which corresponds to 2 floating point operations (Flops) adding the product of a and b to y. However, $m \cdot n$ components of the matrix and n elements of the vector must be loaded and n elements of the result stored. For reasonably large m, n, the I/B ratio approaches $\frac{1}{4}$, which is far below the hardware optimum of the GPU.

The I/B ratio of the addition of two vectors comprising n components is even lower, as, per vector component, one add instruction is issued for 2 floats loaded and 1 stored, i. e.,

$$\mathrm{I/B}_{n\text{-vector add}} = \frac{n}{3n \cdot 4} = \frac{1}{12}\,. \tag{5.2}$$

As intuitively applied in the examples above, the I/B ratio is determined by load and store instructions to DRAM (*global memory*) and the instructions performed by each thread of the kernel. Consequently, the programmer must encounter the I/B ratio at kernel level. This is the reason why math libraries for matrix-vector calculus on GPUs often perform poorly as the I/B ratios of hardware and math operation strongly differ and cannot be compensated for. For our implementation of the DGTD method we are free to decide how to segment the algorithm, i. e., what kernels execute which part of the method. The segmentation of the DGTD method based on an analysis of the I/B ratio is described in Sec. 5.2.

Since the I/B ratio is the fundamental access to high performance, basically all optimization strategies and techniques approach either memory throughput or instruction throughput. As illustrated by the examples above, code is limited by bandwidth in most cases. Hence, memory management and traffic minimization dominate optimization guidelines thoroughly discussed in Refs. [26, 69]. Only little room for improvement is left if a kernel is bound by instruction. However, in principle, accuracy can be traded for speed by employing low-accuracy directives.

[4] We consider single precision arithmetics with 4 bytes per float.

Symbol	Description	Value
$T_{\mathrm{TB}}^{\max}$	Maximum number of threads per thread block	1024
$B_{\mathrm{SM}}^{\max}$	Maximum number of resident thread blocks per multiprocessor	8
$W_{\mathrm{SM}}^{\max}$	Maximum number of resident warps per multiprocessor	48
$T_{\mathrm{SM}}^{\max}$	Maximum number of resident threads per multiprocessor	1536

Table 5.1: *Kernel design relevant hardware limits of Fermi based GPUs of compute capability 2.0.*

5.1.4 Kernel Design and Occupancy

In order to arrive at a high-performance code, more than the I/B ratio must be considered when designing a kernel. Assume a kernel with ideal I/B ratio. At some point data must be loaded and stored which will unavoidably cause latency to the warp in progress. Full utilization of the device can only be achieved when the latency is fully hidden by hardware multithreading (cf. Sec. 5.1.2). Consequently, several warps on a SM are required.

The ratio of the number of resident warps per multiprocessor W_{SM} to the maximum number of resident warps per multiprocessor supported by the hardware $W_{\mathrm{SM}}^{\max}$ (cf. Tab. 5.1) is referred to as occupancy, and must be sufficiently large to effectively apply hardware multithreading. Often, an occupancy of 0.5 is sufficient [69], however, for kernels with low I/B ratio the occupancy should be as large as possible. Thus, it is at the heart of kernel design to implement both a suitable I/B ratio *and* a sufficiently high occupancy.

Full occupancy is achieved when W_{SM} reaches the device specific maximum $W_{\mathrm{SM}}^{\max}$ which is specified by the *compute capability* (CC) for NVidia GPUs. Fermi and Kepler based GPUs are referred to by CC2.x and CC3.x, respectively.

The number of resident warps on a single multiprocessor W_{SM} is determined by

$$W_{\mathrm{SM}} = W_{\mathrm{TB}} \cdot B_{\mathrm{SM}}\,, \tag{5.3}$$

$$W_{\mathrm{TB}} = \left\lceil \frac{T_{\mathrm{TB}}}{W_{\mathrm{S}}} \right\rceil_1\,, \tag{5.4}$$

where W_{TB} is the number of warps per thread block, B_{SM} is the number of thread blocks per SM, T_{TB} is the number of threads per thread block, W_{S} is the number of threads in a warp, referred to as warp size, and $\lceil x \rceil_y$ is a function returning x rounded up to the nearest multiple of y. Note that these quantities are bound to device specific hardware limits listed in Tab. 5.1, and it is the programmer's task to simultaneously balance them to achieve high occupancy.

Within hardware limits, T_{TB}, and consequently W_{TB}, can be defined by the programmer according to the kernel's needs, however, these quantities affect the performance, too, and must therefore be chosen with great care. For Fermi based GPUs, the maximum number of warps per SM, i. e., full occupancy, can only be achieved when $192 \leq T_{\mathrm{TB}} \leq 512$ as, per SM, 8 thread blocks of 192 threads and 3 thread blocks of 512 threads result in 1536 threads, i. e., 48 warps which is equivalent to the hardware limit $W_{\mathrm{SM}}^{\max}$.[5]

[5] In fact, $T_{\mathrm{TB}} = 189$ with $B_{\mathrm{SM}} = 8$ yields $W_{\mathrm{SM}} = 48$, too. However, the last warp comprises a single active thread and 31 idle threads due to the SIMT architecture (cf. Sec. 5.1.2). Therefore, when calculating occupancy, T_{TB} is assumed to be an integer multiple of W_{S}.

In contrast to T_{TB}, the number of resident thread blocks per multiprocessor B_{SM} cannot be freely defined by the programmer but depends on the resources consumed by a thread block of a given kernel as well as hardware- and CC-specific resources [26] and limits (cf. Tab. 5.1). In particular, B_{SM} is directly limited by $B_{\mathrm{SM}}^{\mathrm{max}}$ and indirectly by $T_{\mathrm{SM}}^{\mathrm{max}}$ through T_{TB}. For example, assuming $T_{\mathrm{TB}} = T_{\mathrm{TB}}^{\mathrm{max}}$ for Fermi based GPUs, only a single TB per SM is possible due to $T_{\mathrm{SM}}^{\mathrm{max}}$ (since threads must be invoked at thread block granularity).

In addition, B_{SM} is also limited by memory resource requirements. Each SM on a GPU comprises a CC-dependent amount of registers and shared memory, i. e., on-chip fast local memory in contrast to slow global memory (DRAM). For a given kernel and T_{TB}, the number of registers and shared memory required by a thread block is evaluated by the `OpenCL` compiler. The memory resources are allocated at thread block granularity, i. e., per thread block, the number of registers R_{TB} and shared memory S_{TB} allocated for a given kernel on Fermi based GPUs is given by [26]

$$R_{\mathrm{TB}} = \lceil R_k \cdot W_{\mathrm{S}} \rceil_{G_T} \cdot W_{\mathrm{TB}} \,, \tag{5.5}$$

$$S_{\mathrm{TB}} = \lceil S_k \rceil_{G_S} \,, \tag{5.6}$$

where R_k is the number of registers required by a single thread of the kernel, G_T is the thread allocation granularity (64 for Fermi), W_{TB} is given by Eq. (5.4), S_k is the amount of shared memory in bytes required by a thread block of the kernel, and G_S is the shared memory allocation granularity (128 for Fermi). The number of registers required by a kernel depends on the instructions issued in the kernel, i. e., the kernel code, and is therefore bound to the algorithm. In contrast, shared memory is fully controlled by the programmer.

The number of thread blocks per SM due to memory restrictions $B_{\mathrm{SM}}^{\text{mem-bound}}$ is then given by

$$B_{\mathrm{SM}}^{\text{mem-bound}} = \min\left(\left\lfloor \frac{R_{\mathrm{TB}}^{\mathrm{HW}}}{R_{\mathrm{TB}}} \right\rfloor_1 , \left\lfloor \frac{S_{\mathrm{TB}}^{\mathrm{HW}}}{S_{\mathrm{TB}}} \right\rfloor_1 \right) , \tag{5.7}$$

where $R_{\mathrm{TB}}^{\mathrm{HW}}$ and $S_{\mathrm{TB}}^{\mathrm{HW}}$ are the CC-specific hardware limits of registers and shared memory per SM, respectively [26], and $\lfloor x \rfloor_y$ is a function returning x rounded down to the nearest multiple of y. In essence, the available local memory resources of a SM are distributed among fully equipped thread blocks resulting in a limited number of thread blocks per SM. When a kernel requires lots of registers, for example, a high amount of the available registers is consumed by a single thread block resulting in few thread blocks per SM and low occupancy.

5.1.5 Summary of Performance Related Issues

This section summarizes the most important performance related issues including those introduced in the previous sections. A detailed list of guidelines for kernel design, memory and instruction throughput optimization is available in Ref. [69].

Instruction to byte ratio The I/B ratio (cf. Sec. 5.1.3) is the fundamental guideline to optimal performance. Suitable and ineligible sections of an algorithm with respect to the GPU design paradigm can be identified a priori to the implementation which helps to partition the algorithm into kernels.

Occupancy High occupancy (cf. Sec. 5.1.4) is mandatory for efficient hardware multithreading (cf. Sec. 5.1.2). Occupancy is determined by the resource requirements of the kernel, the thread block, and hardware specific limits and resources.

Levels of parallelism `CUDA` hardware and the `OpenCL` programming model exploit a three level hierarchy (thread, thread block, grid mapped to core, multiprocessor, GPU) to parallelize a task into subtasks and to efficiently distribute the work among the available multiprocessors (cf. Sec. 5.1.2). Consequently, the algorithm's potential for parallelization must be thoroughly examined to appropriately utilize the hierarchy levels.

Warp serialization Avoid branching of the threads' execution paths which causes serialization of the instructions (cf. Sec. 5.1.2).

Minimize host to device memory traffic Host memory, i. e., the CPU's DRAM, and device memory, i. e., the GPU's DRAM, are connected via a low bandwidth PCIe bus causing very large latency. Traffic minimization between host and device is of high priority.

Minimize global memory traffic Although the bandwidth between global memory, i. e., the program model's term for the GPU's DRAM, and the cores is several hundreds of GByte/s, it is a performance limiter in many cases. The minimization of global memory traffic and the optimization of the transfer efficiency is of high priority.

5.2 Implementation of the DGTD Method on GPUs

5.2.1 Analysis of the DGTD Method

The first step towards an efficient implementation of the DGTD method on GPUs is an analysis with respect to performance relevant issues determined in the previous sections. We identify levels of parallelism, i. e., sections of the DGTD method that can be executed simultaneously and, hence, be distributed among the compute units of the GPU followed by an investigation of the I/B ratio of the method's algorithm.

Levels of Parallelism

The time-integration via the LSRK method is the "outermost" algorithm of the DGTD method. It solely involves scale and add operations on the state vector of expansion coefficients (cf. Eq. (3.51)) and can therefore be parallelized on component level. This means that each component of $\tilde{\mathbf{q}}$ can be treated by an individual thread.

The kernel of the time-integration $\tilde{\mathbf{f}}(\tilde{\mathbf{q}}, t)$, i. e., the right-hand-side of Eq. (3.46) comprises computationally intense matrix-vector-products and simple vector operations. The set of equations is spanned by three levels of parallelism. The first level originates from the tessellation of the computational domain into elements and the subsequent formulation of the physical problem on a single element. This element based level of parallelism is very broad, i. e., in general, the number of elements is large compared to GPU related quantities such as the size of the thread block or the number of SMs on a device. The vector nature of Maxwell's curl equations yields a second level of parallelism, i. e., each component of the two vector equations in Eq. (3.46) can be evaluated simultaneously. The third level rises from the expansion of the fields into Lagrange polynomials, i. e., the level of expansion coefficients.

Besides these structural levels, the particular form Eq. (3.46) provides parallelism of arithmetic parts. The matrix-vector-products involving the element-local differentiation and the flux lifting, i. e., the application of the lift matrix to the flux vectors, could be evaluated in parallel [27]. However, there is no advantage in a simultaneous execution of several kernels as the levels of parallelism provided by the method suffice to saturate the compute units of a GPU completely.

Instruction to Byte Ratio

As the fundamental approach to high performance, we analyze the I/B ratio (cf. Sec. 5.1.3) of the individual parts of the DGTD method and identify suitable and ineligible sections for implementation on the GPU. The time-integration via the LSRK method comprises little arithmetic instructions per DOF (cf. Eq. (3.51)), resulting in a very low I/B ratio.

Per se, the matrix-vector-products of Eq. (3.44) suffer a low I/B ratio, as dense matrices must be loaded for each individual element. For linear elements and the simplified form of Eq. (3.46), the differentiation matrices of an element $\mathcal{D}_i^{\triangle}$, $i \in \{x, y, z\}$ are related to the reference element's differentiation matrices $\mathcal{D}_j$, $j \in \{u, v, w\}$ by a simple linear transformation (cf. Eq. (3.37)). A similar approach for curvilinear elements has recently been proposed [43]. It is possible to tune the I/B ratio when $\mathcal{D}_j$ is kept in memory or cache and is reused for the calculation of the differentiation matrices of several elements.

Assume a differentiation kernel whose thread block evaluates $\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{H}}}^{\triangle}$ and $-\vec{\mathcal{D}}^{\triangle} \times \tilde{\vec{\mathrm{E}}}^{\triangle}$ for κ elements employing Eq. (3.37). The differentiation matrices $\mathcal{D}_j$, each of N_p^2 entries, are loaded once. The inverse Jacobi matrix $(\mathcal{J}^{\triangle})^{-1}$ contains 9 entries for each element, and $6N_p$ expansion coefficients of the fields are loaded and stored. For each element, each $\mathcal{D}_i^{\triangle}$, $i \in \{x, y, z\}$ is constructed by 2 FMA instructions and 1 multiplication and for the 6 field components, a matrix-vector-product consumes N_p^2 FMA instructions, which yields an I/B ratio of

$$\mathrm{I/B_{diff}} = \frac{\kappa \cdot \left(6N_p^2 + 3N_p^2\right)}{\left[\kappa \cdot \left(6N_p \cdot 2 + 9\right) + 3N_p^2\right] \cdot 4}. \tag{5.8}$$

Due to the fact that instead of a full $N_p \times N_p$ matrix only the inverse Jacobi matrix must be loaded for each element, $\mathrm{I/B_{diff}}$ steadily increases with increasing κ (cf. Tab. 3.3 for typical values of N_p). Hence, a tunable I/B ratio kernel can be implemented for the element-local differentiation. Note that $\mathrm{I/B_{diff}}$ increases with the polynomial order p. Employing this technique in Sec. 5.2.3, I/B ratios beyond the hardware optimum are achieved, and the full computational power of the GPU becomes accessible.

A similar situation arises for the lift operation. In general, the lift of a single element suffers a low I/B ratio (cf. Eq. (3.44)). Due to the application of linear elements, resulting in an arithmetically cheap relation between the face-mass matrix of an element $\mathcal{F}_f^{\triangle}$ and the reference element $\mathcal{F}_f$ (cf. Eq. (3.40)), the I/B ratio of the lift operation can be tuned as well.

Assume a lift kernel whose thread block performs κ matrix-vector-products of the lift matrix $\mathcal{L}_f$ with the flux vector $\tilde{\mathrm{F}}_{x,f}^{E,\triangle}$ of one field component, here E_x. The resulting I/B ratio is given by

$$\mathrm{I/B_{Lift}} = \frac{\kappa \cdot \left(N_f N_{fp} \cdot N_p\right)}{\left[\kappa \cdot \left(N_f N_{fp} + N_p\right) + N_f N_{fp} \cdot N_p\right] \cdot 4}, \tag{5.9}$$

where for each element $N_f N_{fp}$ entries of the flux vector are loaded and N_p lifted values are stored. The $N_f N_{fp} \cdot N_p$ entries of the lift matrix are loaded once. Each (lift-)matrix entry is processed by one FMA instruction. As a consequence, $\mathrm{I/B_{Lift}}$ steadily increases with increasing κ, resulting in a tunable and again p-dependent I/B ratio.

The calculation of the flux vectors (cf. Eq. (3.47)), also referred to as gather stage, is not an element-local operation [27] and involves only simple vector operations that are specific for each element. Consequently, the I/B ratio is fixed and low.

The ADEs for the Drude and Drude-Lorentz model as well as the uniaxial PMLs result in arithmetically little effort, as the ADEs themselves involve simple vector operations (cf. Eqs. (4.3b), (4.3c), (4.3d), (4.15)) and are coupled to the curl equations by additional current terms only (cf. Eq. (2.20)). Within the ADEs, data cannot be reused and, hence, they govern a low I/B ratio.

5.2.2 Segmentation of the DGTD Algorithm into Kernels

The next step towards an implementation is the segmentation of the entire DGTD algorithm into kernels. This design process is a balancing act between performance on the one hand and maintainability, extensibility, and modularity on the other. One approach is to hide parts of low I/B ratio in a common kernel with parts of tunable I/B ratio. This results in improved performance at the cost of decreased modularity which, in turn, hinders maintainability and extensibility of the code. Based on the analysis of the previous section, our DGTD implementation for GPUs comprises the following kernels. Our segmentation of the core kernels, i. e., local-element differentiation, lift, and flux is in analogy with Ref. [27].

LSRK Kernels

Depending on the application of interest, the DG method can be used with other time-integration methods, e. g., implicit or exponential integrators, or even in the frequency-domain [15]. Therefore, modularity of the code is favored over potential speedup for the low I/B ratio LSRK algorithm. The LSRK time-integration method requires 2 kernels, one for each step of the LSRK algorithm, i. e., Eq. (3.51b) and Eq. (3.51c), respectively.

Flux

The calculation of the flux vectors (cf. Eq. (3.47)) is a struggling task on the GPU. Loading the DOFs on adjacent faces for two different elements in a 3D system unavoidably results in scattered access patterns which deteriorates the per se low I/B ratio.

A fused lift and flux kernel seems an appropriate approach at first glance but cannot be implemented in an efficient way for several reasons [27]. Algorithms comprising vector operations or small matrix-vector-products often apply the one-thread-per-output principle, i. e., a single thread is employed per output value to be computed. Per field component and element, the number of output values of the lift operation N_p and the length of the flux vector of all faces $N_f N_{fp}$ are in general not equal (cf. Tab. 3.3). Consequently, the one-thread-per-output principle cannot be applied without wasting threads during either the lift or the gather operation. Assume that the thread block is designed to fit the lift operation, then, an artificial index space for the flux calculation must be calculated by computationally very expensive modulo and integer division instructions. At last, kernels diverse in their instructions consume more registers than simple kernels and, hence, potentially suffer from low occupancy.

We implemented both approaches, i. e., one fused lift-flux kernel and two individual kernels, and found that two individual kernels execute more efficiently than one fused kernel.

Lift and Local-Element Differentiation Kernel

The calculation of the lift operation, i. e., the application of the lift matrix on the flux vectors, and the local-element differentiation, i. e., the evaluation of the curl operator, are the computationally most intense sections of the DGTD method. Both govern a tunable I/B ratio (cf. Sec. 5.2.1) and their performant implementation is the basis for an efficient DGTD method.

Albeit both can be fused in a common kernel following the one-thread-per-output principle, each operation is implemented in an individual kernel to provide modularity and maintainability.

ADEs of the Uniaxial PMLs, the Drude, and the Drude-Lorentz Model

ADEs of uniaxial PMLs, the Drude and the Drude-Lorentz model result in additional PDEs to be solved simultaneously with the curl equations. The ADEs are coupled to Maxwell's curl equations by current density terms (cf. Eq. (2.20)) which could be easily absorbed in the tunable lift and local-element differentiation kernels. However, this would result in a large amount of kernels as *each combination* of ADEs with one of the tunable kernels would have to be implemented and executed individually.

In practically relevant numerical setups, a low percentage of elements are metallic, and the time consumed by kernels evaluating the Drude and Drude-Lorentz operations in a present CPU implementation is short compared to the lift and differentiation kernels. Consequently, the Drude and Drude-Lorentz algorithms (cf. Eq. (4.3)) are implemented by individual kernels and executed for metallic elements only.

In contrast, the number of PML elements is often large enough to be performance relevant. For this reason, the differentiation kernel is diversified for PMLs, i. e., there exist two kernels implementing the element-local differentiation with and without the PML's ADEs.

TF/SF and SF Sources

Due to the large number of spatial profiles and temporal dependences, sources must be implemented in a maintainable and modular fashion. The TF/SF source enters the algorithm via the manipulation of the field differences (cf. Sec. 4.2.1). In order to avoid explicit storage of the field differences in global memory, the TF/SF source is integrated into the calculation of the flux vectors. This means, that the flux kernel is diversified for the TF/SF source, i. e., there exist two kernels, one performing only the gather stage and one that evaluates the flux vectors with the field differences modified by the source.

The SF source enters the algorithm via the manipulation of the metallic elements' auxiliary fields (cf. Sec. 4.2.2). For reasons of modularity, two individual kernels implement SF Drude and SF Drude-Lorentz sources.

5.2.3 Implementation

This section comprises detailed information about the implementation of the core kernels of the DGTD algorithm. In particular, the design of the element-local differentiation and the lift operation are presented in conjunction with the memory layout. In contrast to Ref. [27], we do not employ textures as they do not support double-precision arithmetics according to the `OpenCL` specification [65]. Our kernels are templated to be utilized in both single and double precision without further modification. We intend to design the kernels and the memory layout such that the computationally intense differentiation and lift kernels achieve highest throughputs. Note that the implementation is explicitly designed for 3D problems of variable polynomial order p, i. e., there are 6 field components for electromagnetic fields, a vector in space has 3 components, and an element has $N_f = 4$ faces.

Notation Reference

Symbol	Description
$[a]$	Size of index space of variable a
$\lvert a, b\rangle$	Set of integers from the half-open interval $[a, b[$
$\{a\}$	Predetermined expression for a during initialization
t_x, t_y, t_z	Local thread indices within a thread block
T_x, T_y, T_z	Number of threads within a thread block
g_x, g_y, g_z	Global thread indices within a grid
G_x, G_y, G_z	Number of threads within a grid
b_x, b_y, b_z	thread block indices within a grid
B_x, B_y, B_z	Number of thread blocks within a grid
K	Total number of elements (in a mesh)
κ	Number of elements processed by a single thread block
k	Global element index
$\mathrm{A} := a$	Preprocessor macro for a constant value a
Q, P	DOF variable containing data of $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}$
F	Flux vector
$\mathcal{L}$	Lift matrix of the reference element

Table 5.2: *Notation within algorithms of the implementation.*

Symbol	Storage region
a	Variable in register
$a^{\mathtt{S}}$	Variable in shared memory
$a^{\mathtt{G}}$	Variable in global memory
$a^{\mathtt{C}}$	Variable in constant memory

Table 5.3: *Typography of the GPU memory regions.*

Thread Indexing

Each thread and each thread block within a grid are attributed several indices for identification. The `OpenCL` programming language [65] provides two sets of indices for each thread, spanning an index-space in up to three dimensions:

- Local thread indices $t_x \in \lvert 0, T_x\rangle$, $t_y \in \lvert 0, T_y\rangle$, $t_z \in \lvert 0, T_z\rangle$ within each thread block.
- Global thread indices $g_x \in \lvert 0, G_x\rangle$, $g_y \in \lvert 0, G_y\rangle$, $g_z \in \lvert 0, G_z\rangle$ within the grid.

Here T_i and G_i, $i \in \{x, y, z\}$ refer to the number of threads in dimension i within a thread block and within the grid, respectively (cf. Tab. 5.2). A thread block is identified by the thread block indices $b_x \in \lvert 0, B_x\rangle, b_y \in \lvert 0, B_y\rangle, b_z \in \lvert 0, B_z\rangle$ within the grid, where B_i, $i \in \{x, y, z\}$ refers to the number of thread blocks within dimension i of the grid. Global and local thread indices are related by

$$g_i = b_i \cdot T_i + t_i,\ i \in \{x, y, z\}\,. \tag{5.10}$$

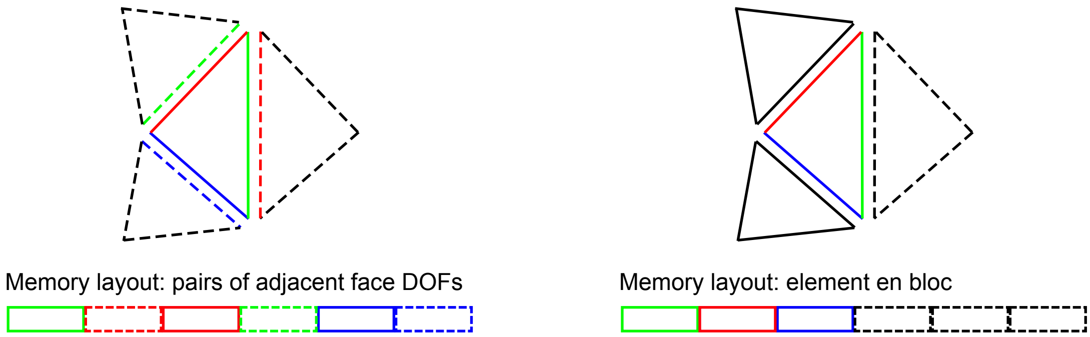

Memory layout: pairs of adjacent face DOFs
Memory layout: element en bloc

Figure 5.5: *Memory layout dilemma. A memory layout of contiguously aligned pairs of adjacent faces (left panel) leads to a distribution of the center element's DOFs (solid lined boxes) in memory. In contrast, the DOFs of the center element are stored en bloc in the layout of the right panel.*

On `CUDA` devices, a thread is attributed a unique *thread ID* t within a thread block which is related to the local thread indices by

$$\begin{aligned} t &= t_x && \text{for 1D thread blocks}\,, \\ t &= t_x + t_y T_x && \text{for 2D thread blocks}\,, \\ t &= t_x + t_y T_x + t_z T_x T_y && \text{for 3D thread blocks}\,. \end{aligned} \tag{5.11}$$

The thread ID is employed to partition the threads of the thread block into warps [26], i. e., batches of W_S contiguous threads identified by t make up a warp. Memory transfer efficiency is strongly related to the pattern by which the threads of a warp — ordered by the thread ID — access global memory data. As a consequence, the "thread layout", i. e., the partitioning of threads into the dimensions of the thread block is crucial for the performance of the kernel.

DOF Memory Layout

The layout of the variables in memory and the implementation of the kernels mutually depend on each other. This renders the definition of the memory layout a very difficult task. A change in memory layout induces time-consuming refactorings of a potentially large number of kernels. In principle, the memory layout can be defined freely by the programmer. However, it must match the kernels' load patterns in order to yield efficient memory accesses.

The DOFs of the DGTD method cannot be arranged such that all previously defined kernels can access the data in a contiguous way. The gather stage, for example, requires data of adjacent faces to be stored contiguously, while the differentiation and lift require a compact representation of the DOFs of each element as visualized in the left and right panels of Fig. 5.5, respectively. The memory layout that contiguously arranges the DOFs of adjacent faces results in a scattered access pattern for the element-local matrix-vector-products in the differentiation and lift kernel, and vice versa. For polynomial orders $p \leq 3$, all DOFs of an element are located on the element's faces (cf. Tab. 3.3) and an optimized memory layout can be constructed. However, for the sake of generality of our approach, we define a memory layout of "best practice" that does not depend on p.

We decided to optimize the layout with respect to the differentiation and lift kernels' access patterns. The arrangement of all DOFs is described in detail in Fig. 5.6. The DOFs of the electromagnetic and

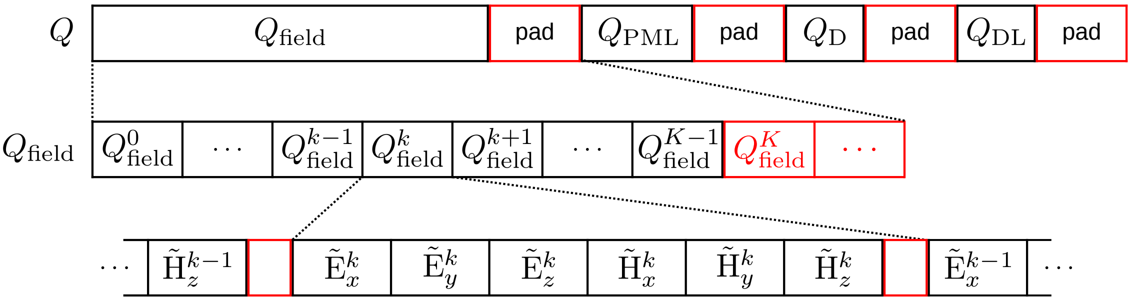


Figure 5.6: *Structure of DOF containers Q, P for the GPU implementation of the DGTD method.*

auxiliary fields are stored in a single structure, referred to by Q in analogy to the state vector $\tilde{\mathbf{q}}$. As Q contains all DOFs that need to be integrated in time, only a single kernel for each step of Eq. (3.51) is launched to process all DOFs. The LSRK method requires a second identically structured DOF container, referred to by P in analogy to $\tilde{\mathbf{p}}$ (cf. Eq. (3.51)).

Q and P are structured by a four level hierarchy: Field type, element, field component, and node.

Field type level segments DOFs into the electromagnetic field Q_{field} and the problem-specific auxiliary fields Q_{PML}, Q_{D}, and Q_{DL} resulting from the ADE formulation of the PMLs, the Drude, and the Drude-Lorentz model, respectively.

Element level segments the DOFs of a particular field type at element granularity, e. g., Q^k_X, $k \in |0, K_X\rangle$ contains the DOFs of the field type X of element k. Note that the number of elements of the field type K_X is problem-dependent and depends on the field type X. The segmentation of the electromagnetic field DOFs Q_{field} on element level is depicted in Fig. 5.6.

Field component level segments the DOFs within an element according to the components of a given field type, i. e., $Q^k_{\text{field}} \ni \{\tilde{\mathrm{E}}^k_x, \tilde{\mathrm{E}}^k_y, \tilde{\mathrm{E}}^k_z, \tilde{\mathrm{H}}^k_x, \tilde{\mathrm{H}}^k_y, \tilde{\mathrm{H}}^k_z\}$, $Q^k_{\text{PML}} \ni \{\tilde{\mathrm{p}}^k_{E_x}, \tilde{\mathrm{p}}^k_{E_y}, \tilde{\mathrm{p}}^k_{E_z}, \tilde{\mathrm{p}}^k_{H_x}, \tilde{\mathrm{p}}^k_{H_y}, \tilde{\mathrm{p}}^k_{H_z}\}$, $Q^k_{\text{D}} \ni \{\tilde{\mathrm{j}}^k_{D,x}, \tilde{\mathrm{j}}^k_{D,y}, \tilde{\mathrm{j}}^k_{D,z}\}$, and $Q^k_{\text{DL}} \ni \{\tilde{\mathrm{j}}^k_{L,x}, \tilde{\mathrm{j}}^k_{L,y}, \tilde{\mathrm{j}}^k_{L,z}, \tilde{\mathrm{j}}^k_{P,x}, \tilde{\mathrm{j}}^k_{P,y}, \tilde{\mathrm{j}}^k_{P,z}\}$.

Node level structures the order within a field component of a particular element and field type, e. g., $Q^k_{\mathrm{E_x}}[i]$, $i \in |0, N_p\rangle$ accesses the E_x-component of element k at the i^{th} node. Alternatively, the individual DOFs are accessed by $Q^k_{\text{field}}[\upsilon \cdot N_p + i]$, $i \in |0, N_p\rangle$, $\upsilon \in |0, 6\rangle$.

The structure of Q also contains padding of two different kinds, marked by red boxes in Fig. 5.6. At element level, each DOF block of a particular field type Q^k_X, $k \in |0, K_X\rangle$ is elongated by several elements of the same field type Q^k_X, $k \in |K_X, K_X + 10\rangle$ in order to allow for a simplified treatment of kernels that execute more than one element per thread block, i. e., $\kappa > 1$.

If the last thread block of a kernel with $\kappa > 1$ is processed and the total number of elements K processed by the kernel is not an integer multiple of κ, out-of-bound memory accesses may occur in non-padded data-structures. The problem of how to handle kernels with $\kappa > 1$ safely can in principle be solved by two approaches:

- Check boundedness of memory accesses during runtime within the kernel.
- Provide sufficient padding elements to allow safe out-of-bound accesses.

We decided for the latter solution as the problem is solved during initialization of the simulation and not during runtime as additional conditional instructions would be necessary causing avoidable overhead. The waste of memory is negligible as currently 10 element entities are padded after each field type.

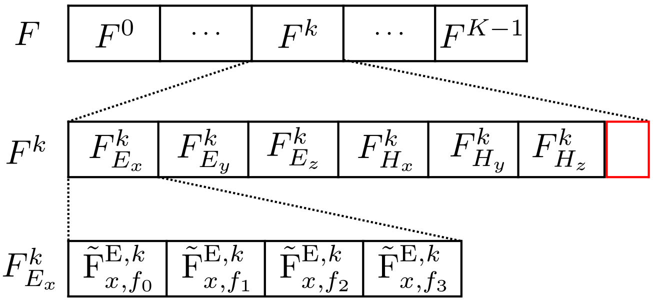

F
F^0
$\cdots$
F^k
$\cdots$
F^{K-1}
F^k
$F^k_{E_x}$
$F^k_{E_y}$
$F^k_{E_z}$
$F^k_{H_x}$
$F^k_{H_y}$
$F^k_{H_z}$
$F^k_{E_x}$
$\tilde{\mathrm{F}}^{\mathrm{E},k}_{x,f_0}$
$\tilde{\mathrm{F}}^{\mathrm{E},k}_{x,f_1}$
$\tilde{\mathrm{F}}^{\mathrm{E},k}_{x,f_2}$
$\tilde{\mathrm{F}}^{\mathrm{E},k}_{x,f_3}$

Figure 5.7: *Memory layout of the flux vector F. At element level, the DOF block is padded to the next 128 Byte boundary.*

p	$R(6N_p)$ [%]	$R(N_f N_{fp})$ [%]
2	93.75	90.00
3	93.75	93.75
4	93.75	93.75
5	95.45	98.44
6	98.44	100

Table 5.4: *Ratio of used to allocated memory due to field component level padding determined by Eq. (5.12). The values are based on single precision and 3D settings, i.e., $N_f = 4$.*

This is much less than the total number of elements, i. e., in realistic systems, padded memory accounts for less then 1‰ of total allocated memory.

Another different kind of padding is applied at field component level. Memory transfers yield highest efficiencies when data is transferred in large blocks and is aligned to 128 Byte boundaries in memory space. Thus, each DOF block of a particular element (for all field types) is elongated to the next 128 Byte boundary by padding of additional floats. The number of floats in a DOF block at field component level is calculated by $\lceil x \rceil_{32}$, where x is the number of used floats of the block, e. g., $x = 6N_p$ in case of the electromagnetic fields, and 32 corresponds to the number of floats within 128 Bytes. Consequently, the ratio of used to the allocated memory due to the padding at field component level is given by

$$R(x) = \frac{x}{\lceil x \rceil_{32}} . \tag{5.12}$$

An overview of ratios for commonly used polynomial orders is listed in Tab. 5.4. Note that field component level and element level padding are different. While the former adds floats up to the next 128 Byte boundary, the latter pads full entities of the element level.

The flux vectors $\vec{\tilde{\mathrm{F}}}^{\mathrm{E},\triangle}_f$ and $\vec{\tilde{\mathrm{F}}}^{\mathrm{H},\triangle}_f$, $f \in |0, N_f\rangle$ are explicitly stored in a variable referred to by F in the GPU's global memory. It is structured similarly to the DOF container Q, as depicted in Fig. 5.7. The flux values of the N_f faces of a particular field component are contiguously aligned, i. e., $F^k_{E_x}[f \cdot N_{fp} + i]$, $f \in |0, N_f\rangle$, $i \in |0, N_{fp}\rangle$ accesses the flux of the E_x-component of element k on face f at node i. Resulting memory usage ratios are listed in Tab. 5.4. Our approach of padding yields comparable efficiency to the one proposed in Ref. [27]. Finally, it should be noted that we apply padding to 128 Byte boundaries to data, too, e. g., the inverse Jacobi matrix $(\mathcal{J}^{\triangle})^{-1}$.

Differentiation Kernel

The differentiation kernel evaluates the curl operations of Eq. (3.46) and scales the intermediate results by $\frac{\Delta t}{\epsilon^{\triangle}}$ and $\frac{\Delta t}{\mu^{\triangle}}$, respectively, i. e., the kernel performs

$$\begin{aligned}
P^{\triangle}_{\tilde{\mathrm{E}}_x} &\leftarrow P^{\triangle}_{\tilde{\mathrm{E}}_x} + \frac{\Delta t}{\epsilon^{\triangle}}\left(\mathcal{D}^{\triangle}_{y}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_z} - \mathcal{D}^{\triangle}_{z}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_y}\right), & P^{\triangle}_{\tilde{\mathrm{H}}_x} &\leftarrow P^{\triangle}_{\tilde{\mathrm{H}}_x} - \frac{\Delta t}{\mu^{\triangle}}\left(\mathcal{D}^{\triangle}_{y}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_z} - \mathcal{D}^{\triangle}_{z}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_y}\right),\\
P^{\triangle}_{\tilde{\mathrm{E}}_y} &\leftarrow P^{\triangle}_{\tilde{\mathrm{E}}_y} + \frac{\Delta t}{\epsilon^{\triangle}}\left(\mathcal{D}^{\triangle}_{z}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_x} - \mathcal{D}^{\triangle}_{x}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_z}\right), & P^{\triangle}_{\tilde{\mathrm{H}}_y} &\leftarrow P^{\triangle}_{\tilde{\mathrm{H}}_y} - \frac{\Delta t}{\mu^{\triangle}}\left(\mathcal{D}^{\triangle}_{z}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_x} - \mathcal{D}^{\triangle}_{x}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_z}\right),\\
P^{\triangle}_{\tilde{\mathrm{E}}_z} &\leftarrow P^{\triangle}_{\tilde{\mathrm{E}}_z} + \frac{\Delta t}{\epsilon^{\triangle}}\left(\mathcal{D}^{\triangle}_{x}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_y} - \mathcal{D}^{\triangle}_{y}\cdot Q^{\triangle}_{\tilde{\mathrm{H}}_x}\right), & P^{\triangle}_{\tilde{\mathrm{H}}_z} &\leftarrow P^{\triangle}_{\tilde{\mathrm{H}}_z} - \frac{\Delta t}{\mu^{\triangle}}\left(\mathcal{D}^{\triangle}_{x}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_y} - \mathcal{D}^{\triangle}_{y}\cdot Q^{\triangle}_{\tilde{\mathrm{E}}_x}\right),
\end{aligned} \tag{5.13}$$

where $\mathcal{D}^{\triangle}_{i}$, $i \in \{x, y, z\}$ is calculated by Eq. (3.37).

The algorithm of the differentiation kernel is presented in Alg. 1. The thread block is given by $\kappa \times N_p$, i. e., a two-dimensional index-space is spanned to evaluate the kernel. The thread index in the first dimension t_x reflects the number of elements processed by a single thread block, referred to as elementblock. The second dimension represents the number of DOFs per element and field component which is motivated by the one-thread-per-output principle for the calculation of the matrix-vector-products in Eq. (5.13), i. e., the index-space spanned by t_y is equal to the number of rows in $\mathcal{D}^{\triangle}_{i}$, $i \in \{x, y, z\}$. Consequently, we launch a grid of $\lceil \frac{K}{\kappa} \rceil_{\kappa}$ thread blocks which are indexed by b_x (cf. Tab. 5.2). During initialization, κ is determined by evaluation of the minimum execution time of the kernel in the range of $\kappa \in \{1, \ldots, 10\}$. As K is in general not an integer multiple of κ, Q and P are padded in order to prevent out-of-bound accesses by the last thread block.

A single thread processes all field components at a given node in order to achieve maximum reuse of the element-local differentiation matrix elements calculated during the execution of the kernel. In contrast to the implementation in Ref. [27], we store the field DOFs in shared memory. More precisely, the expansion coefficients of the electromagnetic fields of all elements in the elementblock are kept in shared memory, hence, $[u^{\mathrm{S}}] = N_p \times \kappa \times 6$ is allocated by each thread block. Macros are used to represent constant integer values and to facilitate readability of the pseudo-code.

For the calculation of the I/B ratio (cf. Eq. (5.8)) we assumed that the input vector to a matrix-vector-product is loaded only once. We implement this assumption by loading the expansion coefficients of the electromagnetic field of κ elements into shared memory. Hence, when requested, the DOFs are loaded from fast memory.

Tunability of the I/B ratio is achieved by an appropriate ordering of $\mathcal{D}_j$, $j \in \{u, v, w\}$ in memory and specifically designed global memory accesses. Let us consider the first warp of the thread block and assume $T_x \equiv \kappa = 3$. At line 12 of Alg. 1, κ threads with contiguous thread IDs t access the same address in global memory due to Eq. (5.11), and $\lceil \frac{W_{\mathrm{S}}}{\kappa} \rceil_1 = 11$ contiguously aligned floats are requested by the warp. `CUDA` devices of CC2.0 always issue 128 Byte transfers, i. e., 32 floats are loaded although only 11 are requested by the first warp. These 32 floats are first loaded into the so-called *L1 cache* and remain there until another global memory request is issued whose data is not yet available in L1 cache. Note that, by design, no global data other than the differentiation matrix elements of the reference element are requested within the `for` loop, i. e., the cache is not evicted by other global memory accesses. As a consequence, the following warps access data already available in L1 cache. This is exactly the point at which differentiation matrix elements are reused for multiple elements, and the I/B ratio becomes tunable.

Algorithm 1: Local-element differentiation with tunable I/B ratio and field in shared memory.

```
Thread block: $\kappa \times N_p$ , i. e., $t_x \in |0, \kappa\rangle$, $t_y \in |0, N_p\rangle$
Grid        : $\lceil \frac{K}{\kappa} \rceil_\kappa$, i. e., $b_x \in |0, \lceil \frac{K}{\kappa} \rceil_\kappa\rangle$
Data        : Differentiation matrix of the reference element $[D^{\mathtt{G}}] = N_p \times 3 \times N_p$ (Fig. 5.8)
Shared mem  : $[u^{\mathtt{S}}] = N_p \times \kappa \times 6$ for field expansion coefficients of $\kappa$ elements
Macros      : $\mathrm{E_x} := 0$, $\mathrm{E_y} := 1$, $\mathrm{E_z} := 2$, $\mathrm{H_x} := 3$, $\mathrm{H_y} := 4$, $\mathrm{H_z} := 5$
Macros      : $\mathrm{u} := 0$, $\mathrm{v} := 1$, $\mathrm{w} := 2$, $\mathrm{x} := 0$, $\mathrm{y} := 1$, $\mathrm{z} := 2$

// Prepare element index
$k \longleftarrow b_x \cdot \kappa + t_x$
// Load fields into shared memory and synchronize threads
if $t_x < 3$ then
    for $l \longleftarrow 0$ to $\kappa$ do
        $u^{\mathtt{S}}[t_y][l][0 + t_x] \longleftarrow Q_{\text{field}}^{\mathtt{G}, b_x \cdot \kappa + l}[(0 + t_x) \cdot N_p + t_y]$
        $u^{\mathtt{S}}[t_y][l][3 + t_x] \longleftarrow Q_{\text{field}}^{\mathtt{G}, b_x \cdot \kappa + l}[(3 + t_x) \cdot N_p + t_y]$
    end
end
=== Barrier and Memory Fence ===
// Initialize registers for matrix-vector-product
$u_\upsilon \longleftarrow 0$ for $\upsilon \in \{\mathrm{E_x}, \mathrm{E_y}, \mathrm{E_z}, \mathrm{H_x}, \mathrm{H_y}, \mathrm{H_z}\}$
// Load inverse Jacobi matrix entries
$J_{i,j} \longleftarrow J_{i,j}^{\mathtt{G},k}$ for $i \in \{\mathrm{x}, \mathrm{y}, \mathrm{z}\}, j \in \{\mathrm{u}, \mathrm{v}, \mathrm{w}\}$
// Loop over (major) columns of $D^{\mathtt{G}}$
for $c \in |0, N_p\rangle$ do
    // Load reference element entries, $\kappa$ threads access a single address
    $D^j \longleftarrow D^{\mathtt{G}}[c][j][t_y]$ for $j \in \{\mathrm{u}, \mathrm{v}, \mathrm{w}\}$
    // Assemble element-local differentiation matrices $D_i$
    $D_i \longleftarrow J_{i,\mathrm{u}} \cdot D^{\mathrm{u}} + J_{i,\mathrm{v}} \cdot D^{\mathrm{v}} + J_{i,\mathrm{w}} \cdot D^{\mathrm{w}}$ for $i \in \{\mathrm{x}, \mathrm{y}, \mathrm{z}\}$
    // Perform matrix-vector-product for column $c$
    $u_{\mathrm{E_x}} \longleftarrow u_{\mathrm{E_x}} + D_{\mathrm{y}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_z}] - D_{\mathrm{z}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_y}]$
    $u_{\mathrm{E_y}} \longleftarrow u_{\mathrm{E_y}} + D_{\mathrm{z}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_x}] - D_{\mathrm{x}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_z}]$
    $u_{\mathrm{E_z}} \longleftarrow u_{\mathrm{E_z}} + D_{\mathrm{x}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_y}] - D_{\mathrm{y}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{H_x}]$
    $u_{\mathrm{H_x}} \longleftarrow u_{\mathrm{H_x}} + D_{\mathrm{y}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_z}] - D_{\mathrm{z}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_y}]$
    $u_{\mathrm{H_y}} \longleftarrow u_{\mathrm{H_y}} + D_{\mathrm{z}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_x}] - D_{\mathrm{x}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_z}]$
    $u_{\mathrm{H_z}} \longleftarrow u_{\mathrm{H_z}} + D_{\mathrm{x}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_y}] - D_{\mathrm{y}} \cdot u^{\mathtt{S}}[c][t_x][\mathrm{E_x}]$
end
// Store result of matrix-vector-product onto second DOF vector
$P_{\text{field}}^{\mathtt{G},k}[\upsilon N_p + t_y] \longleftarrow P_{\text{field}}^{\mathtt{G},k}[\upsilon N_p + t_y] + \Delta t^{\mathtt{C}} \cdot \{\epsilon^{-1}\}^{\mathtt{G},k} \cdot u_\upsilon$ for $\upsilon \in \{\mathrm{E_x}, \mathrm{E_y}, \mathrm{E_z}\}$
$P_{\text{field}}^{\mathtt{G},k}[\upsilon N_p + t_y] \longleftarrow P_{\text{field}}^{\mathtt{G},k}[\upsilon N_p + t_y] + \Delta t^{\mathtt{C}} \cdot \{\mu^{-1}\}^{\mathtt{G},k} \cdot u_\upsilon$ for $\upsilon \in \{\mathrm{H_x}, \mathrm{H_y}, \mathrm{H_z}\}$
```

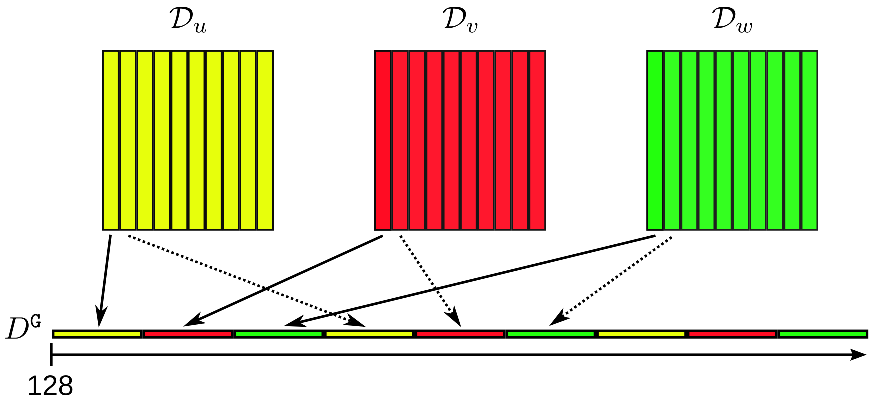


Figure 5.8: *Storage of the reference element differentiation matrices $\mathcal{D}_j$, $j \in \{u, v, w\}$. Columns of equal index are contiguously aligned as visualized by the color code. The entire memory block of $D^{\mathtt{G}}$ is aligned to a 128 Byte boundary. A major column of $D^{\mathtt{G}}$ refers to aligned columns of $\mathcal{D}_j$, $j \in \{u, v, w\}$ of equal column index.*

Per element, the differentiation kernel performs $\Big((3 \cdot 5 + 2 \cdot 2 \cdot 6) \cdot N_p + 12\Big) \cdot N_p$ Flops, i. e., for each differentiation matrix element, $3 \cdot 5$ Flops are required to construct the element-local differentiation matrices, and the matrix-vector-products consume $2 \cdot 2 \cdot 6$ Flops. The operations on lines 21 and 22 cost $2 \cdot 6 \cdot N_p$ Flops. In contrast, the diversified differentiation kernel which also performs the ADEs of the PMLs performs $\big(39Np + 66\big) \cdot N_p$ Flops. The Flop-count is used to determine the practically achieved computational power for the test cases in Sec. 5.3.

Lift Kernel

The lift kernel evaluates the matrix-vector-product of the lift matrix $\mathcal{L}_f$ of face f and the flux vectors $\tilde{\vec{\mathrm{F}}}_f^{\mathrm{E},\triangle}$ and $\tilde{\vec{\mathrm{F}}}_f^{\mathrm{H},\triangle}$ for a single element (cf. Eq. (3.46)). The sum over the faces f is absorbed into an extended matrix-vector-product by casting the lift matrix and flux vectors in the form depicted in Fig. 5.9 and Fig. 5.7, respectively. The element-dependent factors $\bar{Z}^{-1}\frac{J_f^\triangle}{J^\triangle}$ and $\bar{Y}^{-1}\frac{J_f^\triangle}{J^\triangle}$ are absorbed in the field differences of the flux vectors and evaluated during the flux kernel. In essence, the lift kernel calculates

$$\begin{aligned} P_{\tilde{\mathrm{u}}}^{\triangle} &\longleftarrow P_{\tilde{\mathrm{u}}}^{\triangle} + \frac{\Delta t}{\epsilon^{\triangle}} \mathcal{L} F_{\mathrm{u}}^{\triangle} \quad \text{for } \mathrm{u} \in \Big\{E_x, E_y, E_z\Big\} \ , \\ P_{\tilde{\mathrm{u}}}^{\triangle} &\longleftarrow P_{\tilde{\mathrm{u}}}^{\triangle} + \frac{\Delta t}{\mu^{\triangle}} \mathcal{L} F_{\mathrm{u}}^{\triangle} \quad \text{for } \mathrm{u} \in \Big\{H_x, H_y, H_z\Big\} \ . \end{aligned} \tag{5.14}$$

The basic layout of the lift kernel's algorithm (cf. Alg. 2) is similar to the differentiation kernel, i. e., the tunable I/B ratio is implemented by maintaining lift matrix segments in cache while the flux vectors are stored in shared memory. The thread block size is given by $\kappa \times N_p$, i. e., t_x and t_y index the elements and nodes processed within the thread block, respectively. As a consequence, the grid spans $\lceil\frac{K}{\kappa}\rceil_\kappa$ thread blocks, and b_x identifies the corresponding elementblock. The lift matrix is stored in global memory such that contiguous indices i access contiguously aligned memory locations by $\mathcal{L}^{\mathtt{G}}[c][i]$, $c \in |N_f N_{fp}\rangle$, $i \in |0, N_p\rangle$, where c refers to the columns of $\mathcal{L}$. The input vector to the matrix-vector-products, the flux vectors of all elements of the elementblock, are loaded into shared memory.

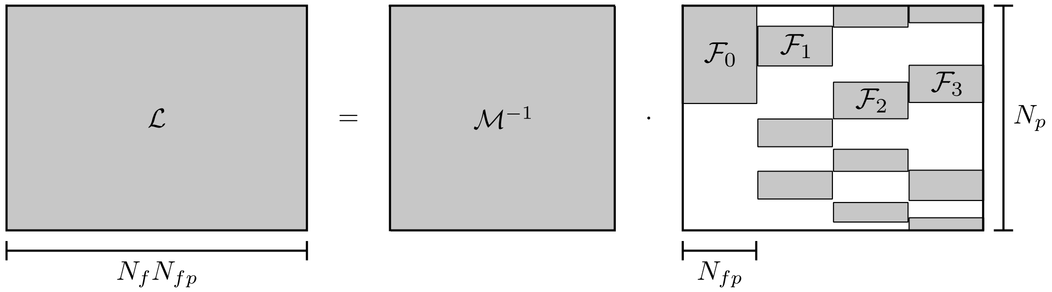


Figure 5.9: *Construction of the reference element's lift matrix $\mathcal{L}$ which results from the product of the inverse mass matrix $\mathcal{M}^{-1}$ and a constructed matrix consisting of the face-mass matrices $\mathcal{F}_f$, $f \in |0, N_f\rangle$.*

Algorithm 2: Lift kernel with tunable I/B ratio and flux vector in shared memory.

```
Thread block: $\kappa \times N_p$, i. e., $t_x \in |0, \kappa\rangle$, $t_y \in |0, N_p\rangle$
Grid        : $\lceil \frac{K}{\kappa} \rceil_\kappa$, i. e., $b_x \in |0, \lceil \frac{K}{\kappa} \rceil_\kappa\rangle$
Data        : Lift matrix of the reference element $[\mathcal{L}^{\mathtt{G}}] = N_f N_{fp} \times N_p$
Shared mem  : $[F^{\mathtt{S}}] = \kappa \times 6 \times N_f N_{fp}$ for flux vectors of $\kappa$ elements
Macros      : $\mathrm{E_x} := 0$, $\mathrm{E_y} := 1$, $\mathrm{E_z} := 2$, $\mathrm{H_x} := 3$, $\mathrm{H_y} := 4$, $\mathrm{H_z} := 5$

// Prepare element index
$k \longleftarrow b_x \cdot \kappa + t_x$

// Load flux vectors into shared memory
// Cast of dimensionality:  $[F^{\mathtt{S}}] = \kappa \times 6 N_f N_{fp}$
for $l \longleftarrow 0$ to $\kappa$ do
    for $m \longleftarrow t_x + t_y \kappa$ to $6 N_f N_{fp}$ increment by $\kappa N_p$ do
        $F^{\mathtt{S}}[l][m] \longleftarrow F^{\mathtt{G}, b_x \cdot \kappa + l}[m]$
    end
end
=== Barrier and Memory Fence ===
// Initialize registers for matrix-vector-product
$u_\upsilon \longleftarrow 0$ for $\upsilon \in \{\mathrm{E_x}, \mathrm{E_y}, \mathrm{E_z}, \mathrm{H_x}, \mathrm{H_y}, \mathrm{H_z}\}$

// Loop over nodes on faces
for $c \in |0, N_f N_{fp}\rangle$ do
    // Load lift matrix entry; $\kappa$ contiguous threads access same entry
    $\mathcal{L} \longleftarrow \mathcal{L}^{\mathtt{G}}[c][t_y]$

    // Perform matrix-vector-product for column $c$
    $u_\upsilon \longleftarrow u_\upsilon + \mathcal{L} \cdot F^{\mathtt{S}}[t_x][\upsilon][c]$ for $\upsilon \in \{\mathrm{E_x}, \mathrm{E_y}, \mathrm{E_z}, \mathrm{H_x}, \mathrm{H_y}, \mathrm{H_z}\}$
end

// Add result of matrix-vector-product onto second DOF vector
$P^{\mathtt{G},k}_{\text{field}}[\upsilon N_p + t_y] \longleftarrow P^{\mathtt{G},k}_{\text{field}}[\upsilon N_p + t_y] + \Delta t^{\mathtt{C}} \cdot \{\epsilon^{-1}\}^{\mathtt{G},k} \cdot u_\upsilon$ for $\upsilon \in \{\mathrm{E_x}, \mathrm{E_y}, \mathrm{E_z}\}$
$P^{\mathtt{G},k}_{\text{field}}[\upsilon N_p + t_y] \longleftarrow P^{\mathtt{G},k}_{\text{field}}[\upsilon N_p + t_y] + \Delta t^{\mathtt{C}} \cdot \{\mu^{-1}\}^{\mathtt{G},k} \cdot u_\upsilon$ for $\upsilon \in \{\mathrm{H_x}, \mathrm{H_y}, \mathrm{H_z}\}$
```

Similar to the differentiation kernel, $T_x \equiv \kappa$ threads access the same element of $\mathcal{L}^{\mathtt{G}}$ on line 10 of Alg. 2. During the evaluation of the matrix-vector-product, no data from global memory is fetched. The lift matrix remains in cache, leading to an effective reuse. Shared memory allows for a very efficient access on line 11, too, where $T_y \equiv N_p$ threads access the same address. Multiple threads can access one or more addresses in shared memory simultaneously without causing latency in a mechanism referred to as (multi)-broadcast, available for CC≥2.0 [26]. Finally, the result of the matrix-vector-product is scaled and added to P.

Per element, the lift kernel consumes $\Big((N_f N_{fp} + 1) \cdot 2 \cdot 6\Big) \cdot N_p$ Flops, i. e., for each row of the matrix-vector-product $2N_f N_{fp}$ Flops are required, and lines 13 and 14 demand $2 \cdot 6 \cdot N_p$ Flops.

5.2.4 Host-Device Interaction, Heterogeneous Computing, and Probing

As one of the most important issues to be regarded for a high-performance implementation, the minimization of host–device traffic must be addressed (cf. Sec. 5.1.5). The DGTD method on GPUs is embedded in a well-established CPU-framework which is employed to setup all entities required during time-integration, e. g., the reference element's differentiation matrices $\mathcal{D}_j$, $j \in \{u, v, w\}$, the lift matrix $\mathcal{L}$, boundary conditions, mapping of volume-to-surface indices, and so forth. After initialization of all variables on the host, the entities required during time-integration are transferred to the device and time-integration commences.

The evaluation of the kernels is controlled by the host program, i. e., the kernels required for the physical problem at hand are launched in sequential order on the host. A single kernel is executed on the device at a time. The time-integration of the DOFs on the device proceeds without further interaction of host and GPU.

After each LSRK step, i. e., after integration of Δt time, the electromagnetic fields are copied from the device to the host for probing purposes. Neither the host program nor the GPU process further instructions during the time of the transfer, which is typically of the order of the execution time of a single stage for the 14-stage scheme (cf. Sec. 3.2.1). Consequently, a large number of stages helps to reduce the cost of the transfer while leading to a large Δt.

Additional book keeping could help to minimize the host–device traffic as currently all electromagnetic field components of all elements are transferred regardless of the configuration of the probes. Transferring only data that is actually processed by a probe may reduce the amount of data traffic. Furthermore, data might not be required at every time-step and could be fetched only when needed. However, for time-integration methods with a high number of stages, less than 10% of the execution time of a single stage can be saved.

The evaluation of the probes is performed entirely by an existing CPU-framework. The framework was extended to evaluate the probes asynchronously, i. e., the host program launches the kernels of the next time-step and the GPU recommences time-integration while the probes are being processed. This allows to hide probe evaluation completely behind the time-integration. Probe evaluation and host program synchronize at the transfer of the DOFs after each time-step, i. e., the GPU idles as long as the probe evaluation is not finished, and vice versa.

5.3 Validation and Performance Benchmark

In this section, the implementation of the DGTD method on GPUs is checked for both validity and performance by simulation of several test systems. Problems whose analytic solutions are known explicitly are used to investigate convergence and correctness of the simulated results.

Our primary motivation for the realization of a DGTD Maxwell Solver on GPUs was the potential speedup compared to conventional CPU computer systems. Here, we quantify the speed up by comparing the GPU to a highly optimized CPU implementation, which has been developed over several years. The CPU implementation is powered by hand-written matrix-vector-products that outperform Intel's `MKL BLAS` routines [14]. We limit our analysis to the practically most relevant polynomial orders $p = \{2, 3, 4, 5\}$. Higher orders are rarely employed as the accuracy of the method can also be improved by h-refinement, i. e., a refinement of the mesh, or curvilinear elements [43].

In the following test cases, we employ a NVidia® GeForce® GTX580. Our CPU test system is equipped with an Intel® Core™ i7 CPU 970 @ 3.20 GHz.

5.3.1 Empty Metallic Cavity

The first test system is an empty cubic cavity with metallic boundaries. It is employed to verify correct functionality of the DGTD-GPU implementation's core components: The LSRK time-integrator and the DG Maxwell operator (right-hand side of Eq. (3.46)) including the differentiation, lift, and flux kernel.

The physical system is a cube of edge length $L = 2$ (in dimensionless units) "filled" with vacuum, i. e., $\epsilon = 1$. We apply perfect electric conductor (PEC) boundary conditions, i. e., the tangential component of the electric field must vanish at the boundary. The boundary conditions are enforced via the manipulation of field differences according to Tab. 3.2. The analytic solution [70] provides initial field values and serves as reference solution for the calculation of the numerical error.

The fields are initialized by a (222) eigenmode. In order to suppress the error emerging from time-integration, we chose the time-step $\Delta t = 0.1 \cdot \Delta t_{\max}$ (cf. Sec. 3.2.2). Thus, any deviation from the analytic solution arises from the spatial discretization. The system is simulated with polynomial orders $p = 2, \ldots, 5$ for $T = 15T_0$, where $T_0 = \frac{2}{\sqrt{3}}$ refers to the oscillation period of the initial (222) eigenmode.

Full control of the mesh element size is guaranteed by explicit construction of the mesh. The mesh consists of identical sub-cubes of edge-length $h = \frac{L}{k}$, where k is the number of elements per cavity edge. Each sub-cube comprises 5 tetrahedra, resulting in a total number of $5 \cdot k^3$ elements to model the cavity.

We calculate the maximum absolute error with respect to the analytic solution over time and all grid points for the H_z-component[6] by

$$\Gamma = \max_{t,\vec{\mathrm{r}}} \left(\left| H_z^{\mathrm{num}}(\vec{\mathrm{r}}, t) - H_z^{\mathrm{ref}}(\vec{\mathrm{r}}, t) \right| \right) . \tag{5.15}$$

Fig. 5.10 depicts the maximum error Γ as a function of h for several polynomial orders p. The solid lines are regression curves and indicate the fitted convergence rate, listed in the inset. We observe that the error decays in accordance with Eq. (3.48) until it saturates around 10^{-5}. When saturated, the error is limited by the floating point precision of the device. Based on this analysis, we assume that our implementation of the core features mentioned above works correctly.

As a next step, we analyze the performance of the GPU and CPU implementation. As for all performance analyses of this thesis, we compare GPU and CPU calculations performed in single precision. We are interested in the time both implementations require to process, more precisely integrate, in time. Hence, we relate their times spent on the time-integration, referred to as computational time, which

[6] Every other component would yield the same conclusions.

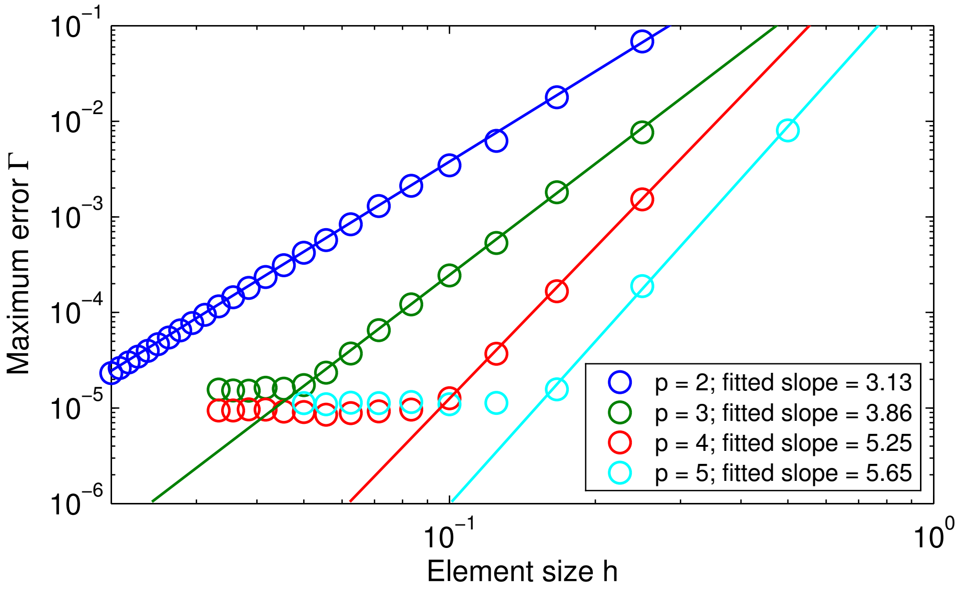


Figure 5.10: *Maximum error Γ of the H_z-component as a function of h for the polynomial orders $p = \{2, 3, 4, 5\}$. The convergence rate obtained from a linear fit is listed in the inset and visualized by solid lines. For small h, the error saturates at the single precision floating point accuracy.*

includes the evaluation of the DG operator, but excludes time consumed to prepare the simulation and evaluate probes.

Fig. 5.11 depicts the speedup of the GPU over the CPU implementation, i. e., the ratio of their computational times in dependence of the number of tetrahedra for orders $p = \{2, 3, 4, 5\}$. We observe a speedup of approximately 25-55, i. e., the GPU implementation requires 25-55 times less time than the CPU implementation on a single core. The speedup depends on the polynomial order p and saturates for large K. Increasing arithmetic intensity per DOF due to increasing polynomial order p is not necessarily connected to a growing speedup. In fact, for this particular system, the highest speedup occurs for $p = 2$. The saturation of the speedup is directly related to the saturation of the GPU's cores. Meshes consisting of only hundreds of elements do not provide sufficient work to utilize all cores (not shown). However, realistic physical systems usually require enough elements to saturate the device completely. In general, the speedup should not be overestimated as many device and implementation specific factors influence it. Nevertheless, the presented speedup shows the order of acceleration provided by the GPU implementation.

The kernel execution ratio of the employed kernels, i. e., the execution time of a particular kernel normalized to the execution time of all kernels, is summarized[7] for $p = \{2, \ldots, 5\}$ in the left panel of Fig. 5.12. The kernels of the LSRK method consume roughly 10% of the total kernel execution time. For increasing order p, the common share of the computationally intense differentiation and lift kernels rises from 50% to 70% due to the increasing number of operations per DOF which, in turn, reduces the ratio of the flux kernel from 40% to 20%. The kernel execution ratio of the flux kernel is, however, remarkably large in comparison to the number of arithmetic operations evaluated by it.

[7] We observe that the kernel execution ratio of each kernel is practically independent of K if the GPU is saturated. Hence, we average the kernel execution ratios of simulations with varying K in the range investigated in Fig. 5.11.

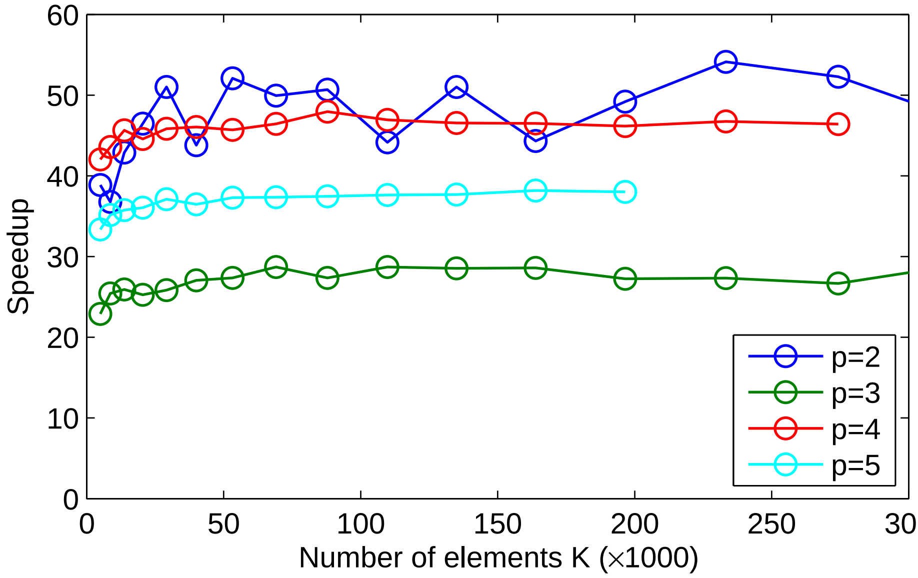


Figure 5.11: *Speedup of the GPU over the CPU implementation for the cubic metallic cavity in dependence of the number of tetrahedra K for spatial orders $p = \{2, 3, 4, 5\}$.*

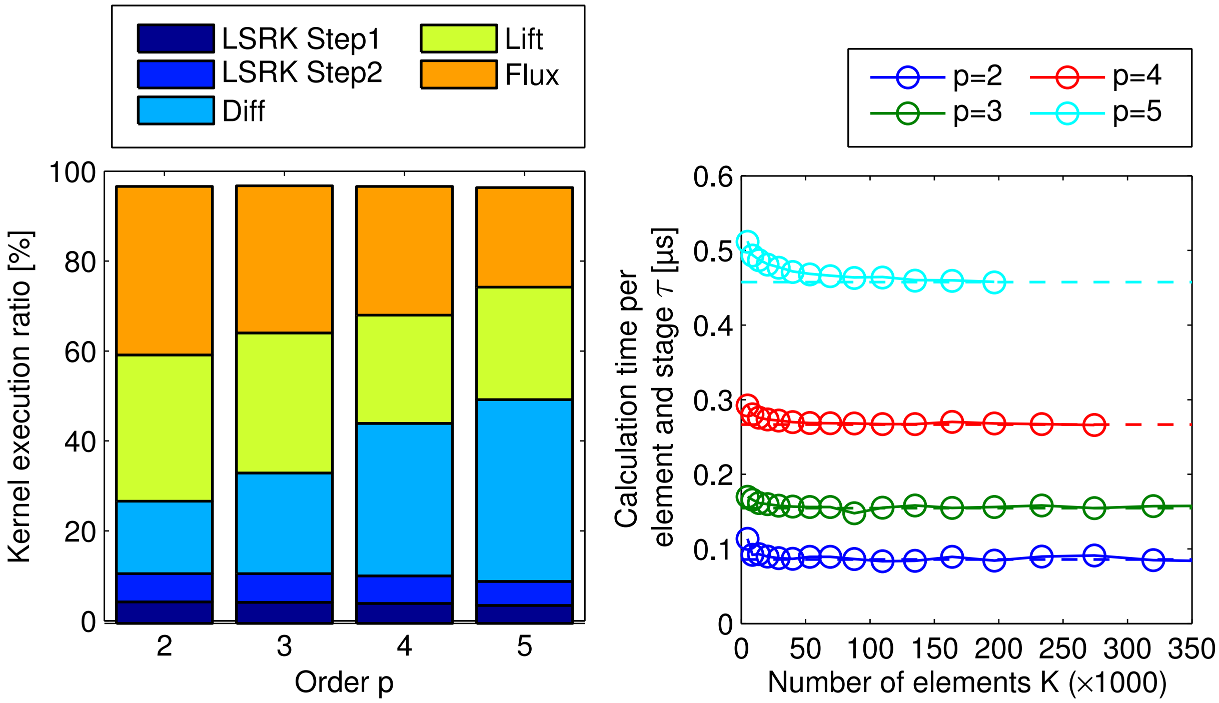


Figure 5.12: *Kernel execution ratio (left panel) and calculation time per element and stage τ (right panel) for spatial orders $p = \{2, 3, 4, 5\}$. The dashed lines mark the asymptotes for large K and serve as guides to the eye for further tests.*

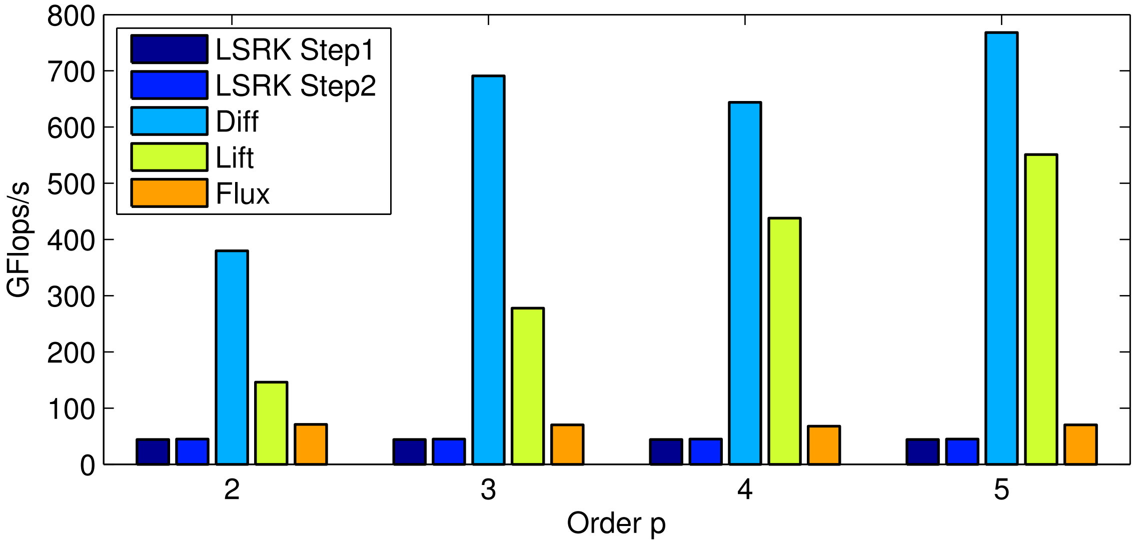


Figure 5.13: *Computational power in GFlops/s of time-consuming kernels for polynomial orders* $p = \{2, 3, 4, 5\}$.

The calculation time per element and stage τ is a measure for the absolute time required to evaluate one stage of the LSRK method for one element including the time-integration and the DG operator kernels. The right panel of Fig. 5.12 depicts the calculation time per element and stage τ as a function of the number of elements K in dependence of p. We observe an asymptotic behavior even for small element numbers, i. e., the GPU implementation scales linearly with respect to the number of elements in a mesh. Clearly, the dependence on the polynomial order is associated with the increasing number of instructions per stage. These quantities can be used directly to estimate the calculation time of simulations in advance.

At last, we analyze the number of single precision floating point operations per second (Flops/s) for each kernel in order to compare the practically achieved computational power to the available theoretical one. We calculate the computational power by counting the required Flops in each kernel, multiply by the total number of threads in the grid, and divide by the kernel execution time. Note that only floating point operations are counted.

Fig. 5.13 summarizes the computational power in GFlops/s for polynomial orders $p = \{2, 3, 4, 5\}$. The differentiation and lift kernel's computational power grows with p due to the increased I/B ratio within the kernels. For $p = 5$, a maximum of almost 800 GFlops/s is achieved, which corresponds to roughly 50% of the theoretical optimum value of approximately 1500 GFlops/s for the NVidia® GeForce® GTX580. Note that the theoretical optimum value assumes FMA operations only. The remaining kernels are bandwidth limited and, hence, suffer low computational power of the order of 50 GFlops/s. Based on the kernel execution ratio of Fig. 5.12, another factor of 2 in speedup can — in theory — still be gained as all kernels execute at best at half of the theoretical computational power.

Our implementation shows slightly improved behavior over the one presented in Ref. [27]. Therein, roughly up to one third of the theoretical computational power was utilized for both the differentiation and the lift kernel on a NVidia® GeForce® GTX280. The bandwidth limited kernels of the time-integration and the gather stage show (similar to our implementation) practically p-independent values of approximately 30 and 100-150 GFlops/s, respectively.

5.3.2 Optical Spectroscopy of a 100 nm Silver Sphere

As another example, we examine the scattering cross section of a silver sphere of 100 nm diameter in vacuum, i. e., a typical setup of a realistic physical system including external field sources and probes. With this test, the total-field/scattered-field source, the ADEs for the Drude model, and the evaluation of the flux through the TF/SF contour are verified. These setup entities are required for the detailed investigation of extinction, scattering, and absorption cross-section spectra of several different nanoantennae in Chap. 6.

The electromagnetic fields of a sphere whose dispersive material is described by a frequency-dependent electric permittivity $\epsilon(\omega)$ can be expressed analytically in terms of spherical harmonics which is commonly known as the Mie theory. Thereof, the cross sections can be directly evaluated from the scattering coefficients a_n and b_n, i. e., the expansion coefficients of the scattered electromagnetic field in terms of the vector spherical harmonics [32].

Since the setup of the system is not at the focus of interest, the reader is referred to the detailed presentation of the setup in Sec. 6.2 on page 77. The permittivity of silver is described by a Drude model (cf. Sec. 2.2.2) and corresponding material parameters $\epsilon_\infty = 1$, $\omega_D = 13.9 \cdot 10^{15}\,\mathrm{Hz}$, $\gamma_D = 3.23 \cdot 10^{13}\,\mathrm{Hz}$ are obtained by fitting the experimental values of Johnson and Christy [38] by the procedure described in Sec. 2.2.4. The system is terminated by Silver-Müller boundary conditions and, in contrast to the setup of Sec. 6.2, PMLs of 125 nm width. We excite the system by a plane wave Gaussian pulse by means of a total-field/scattered-field source (cf. Sec. 4.2.1) and simulate it with polynomial orders $p = \{2, 3, 4, 5\}$ for $T = 133\,\mathrm{fs}$.

Fig. 5.14 depicts the scattering cross section[8] calculated from DGTD simulation and Mie theory. We observe excellent agreement of numeric and analytic cross sections. Even for the relatively low order $p = 3$, no significant difference to the analytic cross section is discernible in both the overview and the zoomed view of the sharp resonances. Hence, we conclude that our implementation of the Drude ADE, PMLs, and TF/SF source works correctly.

Similar to the case of the empty cubic cavity, we analyze the performance by comparison of the calculation times of both GPU and CPU implementation for a series of h-refined meshes and polynomial orders $p = \{2, 3, 4, 5\}$. In contrast to the setup described above, the simulated time is chosen such that exactly 20 time-steps are processed which results in 280 evaluations of the DG operator using the 14-stage LSRK scheme of Sec. 3.2.1. Fig. 5.15 depicts the speedup in dependence of K and p. Similar to the empty cavity, the speedup is of the order of 25-55 and practically constant in K.

In order to determine the influence of the Drude and TF/SF kernels, we compare the calculation time per element and stage τ to the empty metallic cavity case. In the right panel of Fig. 5.16, we observe that $\tau\big(K\big)$ shows asymptotic behavior, i. e., the GPU implementation scales linearly with K. The absolute values of τ are practically unchanged in comparison to the empty metallic cavity case, indicated by the dashed lines.

These observations are consistently reproduced by the kernel execution ratio, depicted in the left panel of Fig. 5.16. The share of the Drude and TF/SF source kernels amounts to less than 3% and 1% of the total kernel execution time, respectively, while similar ratios compared to the empty metallic cavity case are observed for the core kernels (cf. Fig. 5.12). Note that the evaluation of the PMLs is embedded in a diversified local-element differentiation kernel (cf. Sec. 5.2.2) and, thus, cannot be quantified independently.

[8] In practice, we determine extinction, scattering and absorption cross section simultaneously. For this system, the extinction is dominated by scattering and absorption is very low. In order to keep the figure well-arranged, we merely depict the scattering cross section. However, the conclusions derived in the following are equivalently obtained from the analysis of extinction and absorption.

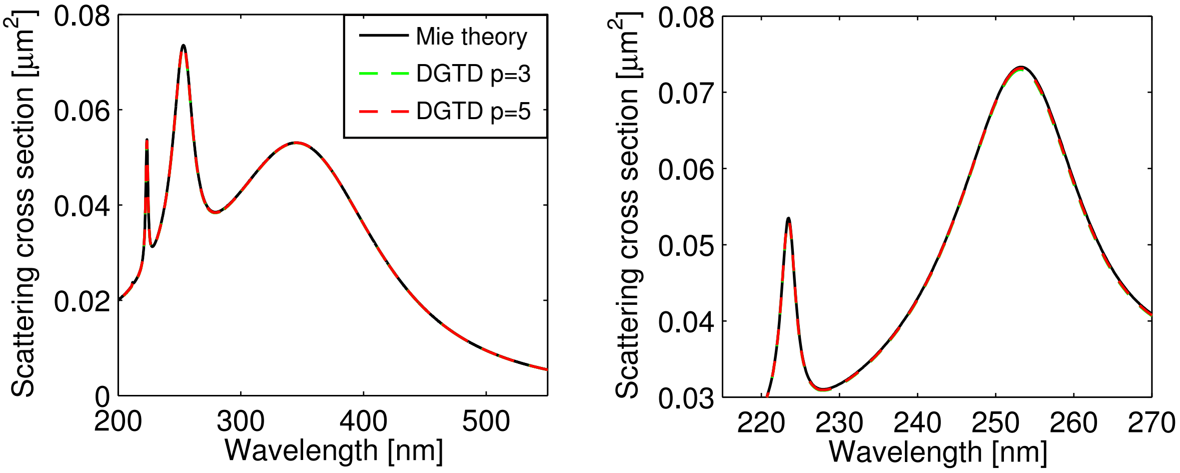


Figure 5.14: *Scattering cross section of a silver sphere of radius 50 nm calculated via analytic Mie theory and DGTD for polynomial orders $p = \{3, 5\}$. The left panel shows an overview, while the right panel depicts a zoomed view of sharp resonances.*

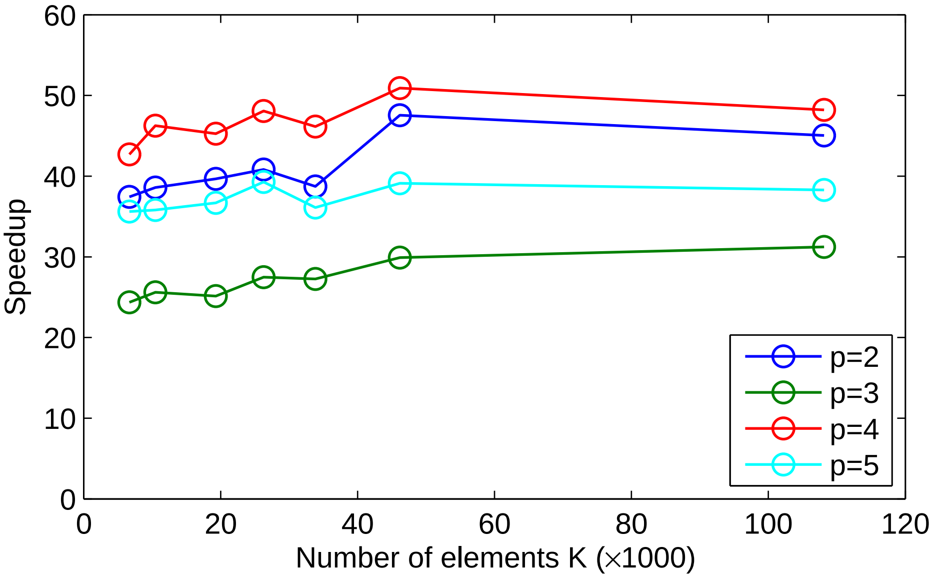


Figure 5.15: *Speedup of the GPU over the CPU implementation for a silver sphere in dependence of K for polynomial orders $p = \{2, 3, 4, 5\}$.*

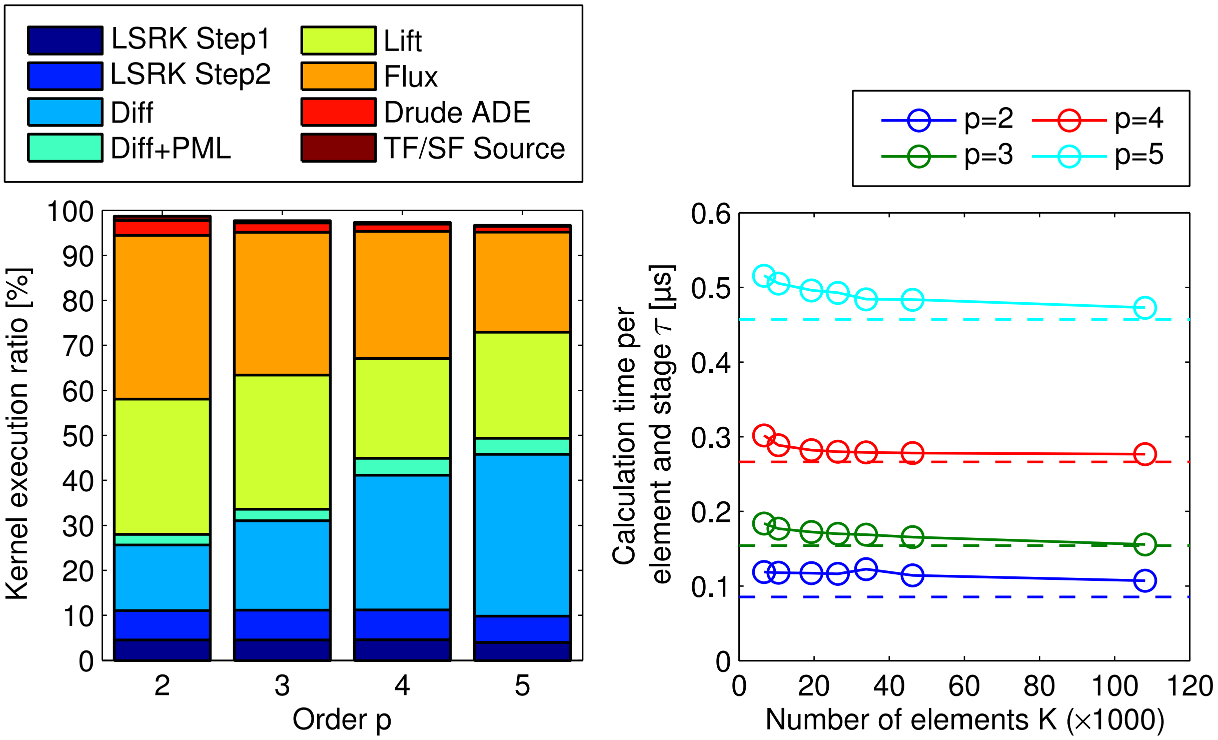


Figure 5.16: *Kernel execution ratio (left panel) and calculation time per element and stage τ (right panel) for polynomial orders $p = \{2, 3, 4, 5\}$. Dashed guides refer to the asymptotic values of the empty metallic cavity test.*

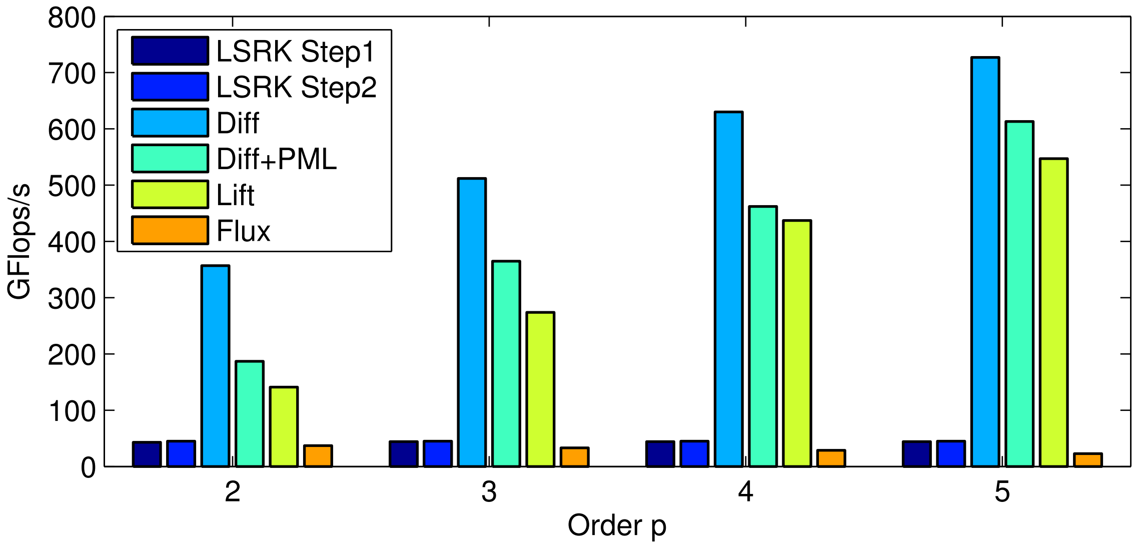


Figure 5.17: *Computational power in GFlops/s of time-consuming kernels for polynomial orders $p = \{2, 3, 4, 5\}$.*

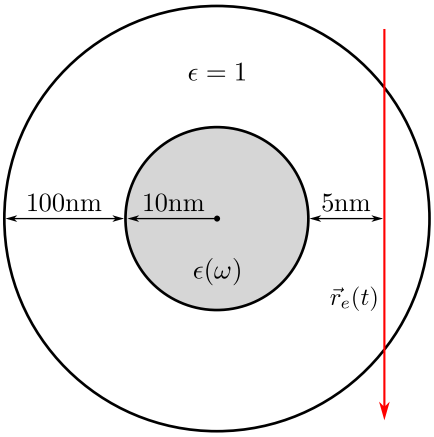


Figure 5.18: *Simulation setup of the EELS analysis of an aluminum sphere (gray) whose material properties are described by $\epsilon(\omega)$. The electron trajectory $\vec{\mathrm{r}}_e(t)$ is indicated by a straight line (red) passing the sphere at a minimum distance of 5 nm. The computational domain is limited by a sphere of radius 110 nm and filled with vacuum, i.e., $\epsilon = 1$.*

The computational power of time-consuming kernels is depicted in Fig. 5.17. In general, a behavior similar to the empty cavity case is observed, i. e., the differentiation and lift kernels show high GFlops/s values that increase with p, and the bandwidth limited kernels perform at approximately 50 GFlops/s. Again, maximum values beyond 700 GFlops/s are achieved. For high orders, additional cost of PMLs in the diversified local-element differentiation kernel decreases the computational power only slightly, whereas moderate performance loss is observed for low orders, e. g., $p = 2$.

For GPU simulations, probing is “outsourced” to the host and the time spent on probe evaluation contributes to neither the simulation nor the calculation time. Therefore, for practically realistic systems including probes, the speedup is expected to be increased over the empty metallic cavity case.

5.3.3 Electron Energy Loss Spectroscopy of a 20 nm Aluminum Sphere

As a last test system, we examine the electron energy loss spectrum of an aluminum sphere of diameter 20 nm embedded in vacuum in order to verify the implementation of a fast electron source (cf. Sec. 7.1.2) and the EELS analysis (cf. Sec. 7.1.1) used throughout Chap. 7 for a detailed analysis of split-ring resonators.

An electron of kinetic energy $E = 200\,\mathrm{keV}$, corresponding to $v = 0.6953c$, where c is the vacuum speed of light, passes the aluminum sphere by a minimum distance of 5 nm as sketched in Fig. 5.18. The electron's initial position is chosen far away from the sphere, such that the electromagnetic fields induced by the electron are negligible at the start of the simulation. The permittivity of aluminum is described by a Drude model (cf. Sec. 2.2.2) and corresponding material parameters $\epsilon_\infty = 1$, $\hbar\omega_D = 15\,\mathrm{eV}$, and $\hbar\gamma_D = 0.07\omega_D$ are taken from Ref. [71]. We simulate the system for $T = 16.6\,\mathrm{fs}$.

The electron energy loss spectra calculated by an analytical expression in Ref. [72] and the DGTD method are depicted in Fig. 5.19. The analytic spectrum was calculated employing multipoles up to 15th order. Despite minor deviations in the absolute value, the resonance position is met precisely. The overall agreement is very good even for the relatively low order $p = 3$. From these results, we assume that our implementation of the fast electron source and the subsequent analysis work correctly.

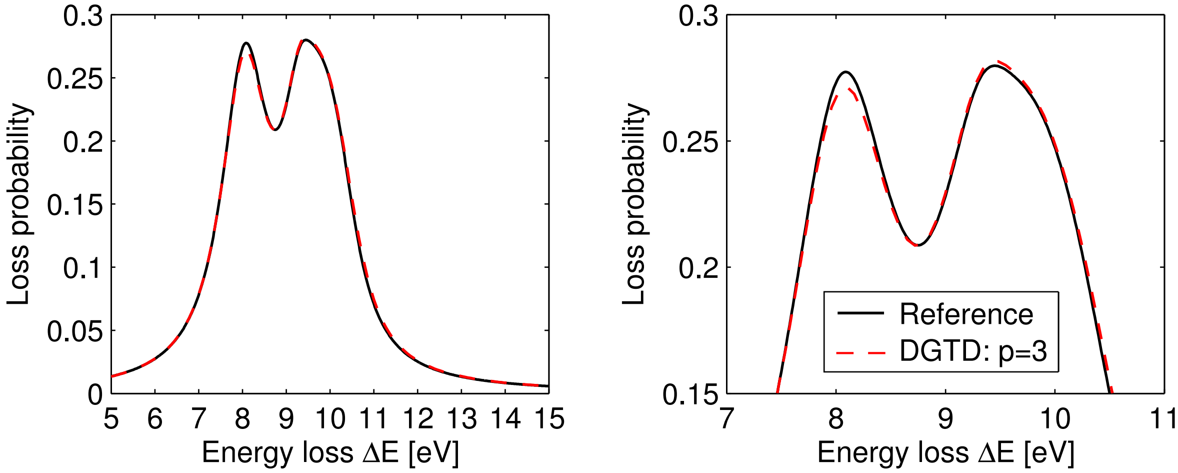


Figure 5.19: *Overview (left panel) and close up view (right panel) of electron energy loss spectra obtained via analytic theory (black solid) from Ref. [72] and DGTD simulation of order $p = 3$ (red dashed).*

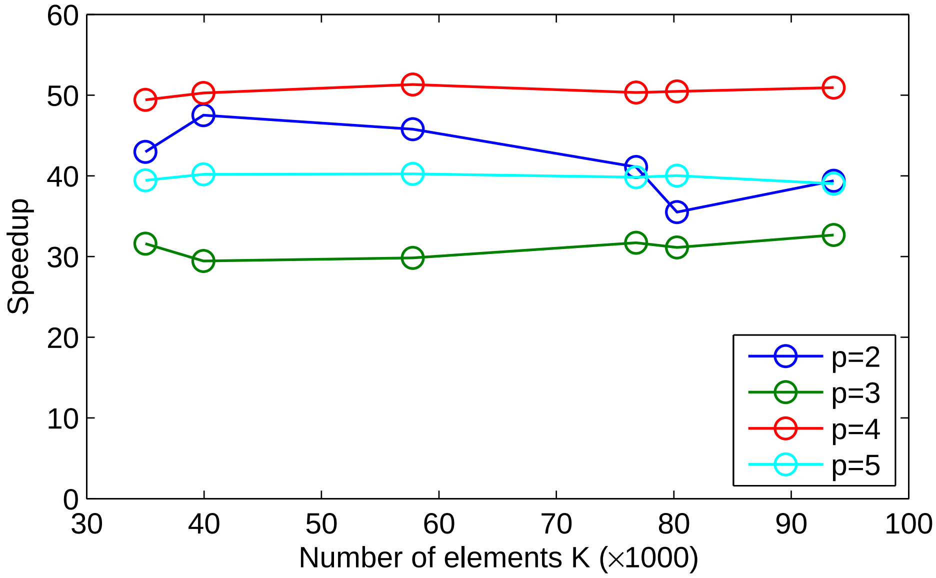


Figure 5.20: *Speedup of the GPU over the CPU implementation for the EELS analysis of a 20 nm aluminum sphere.*

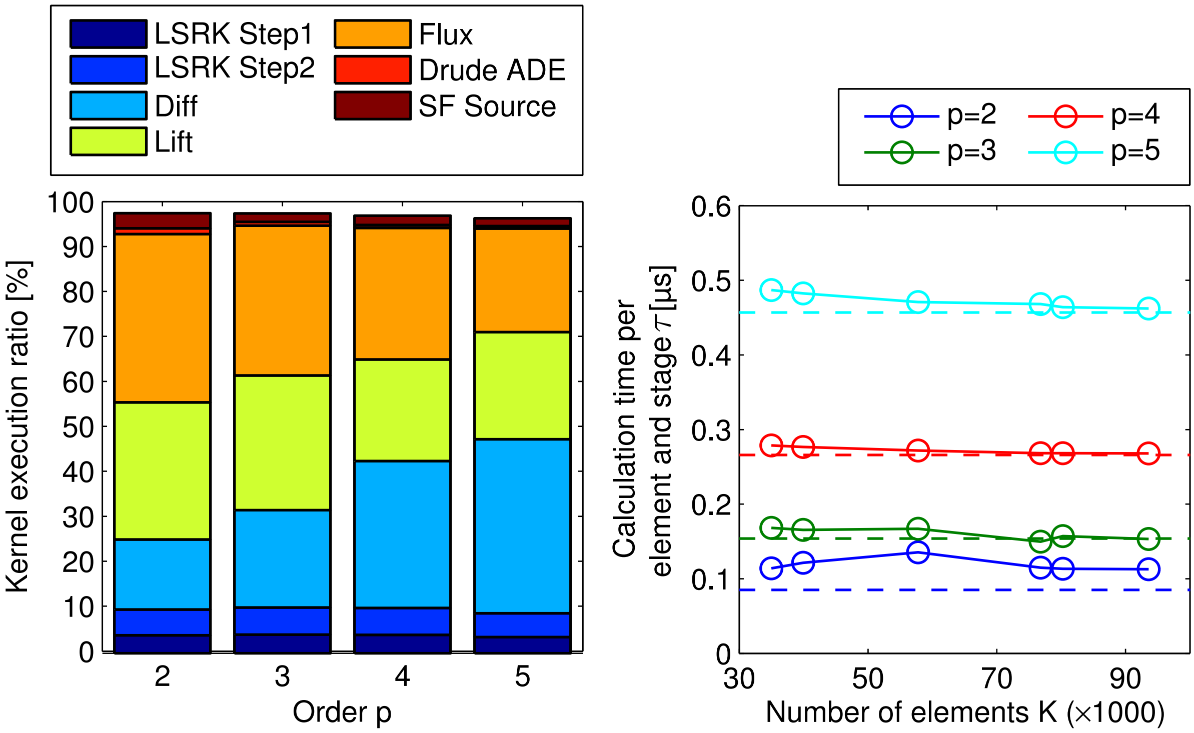


Figure 5.21: *Kernel execution ratio (left panel) and calculation time per element and stage τ (right panel) for spatial orders $p = \{2, 3, 4, 5\}$. Dashed guides refer to the asymptotic values of the empty metallic cavity test.*

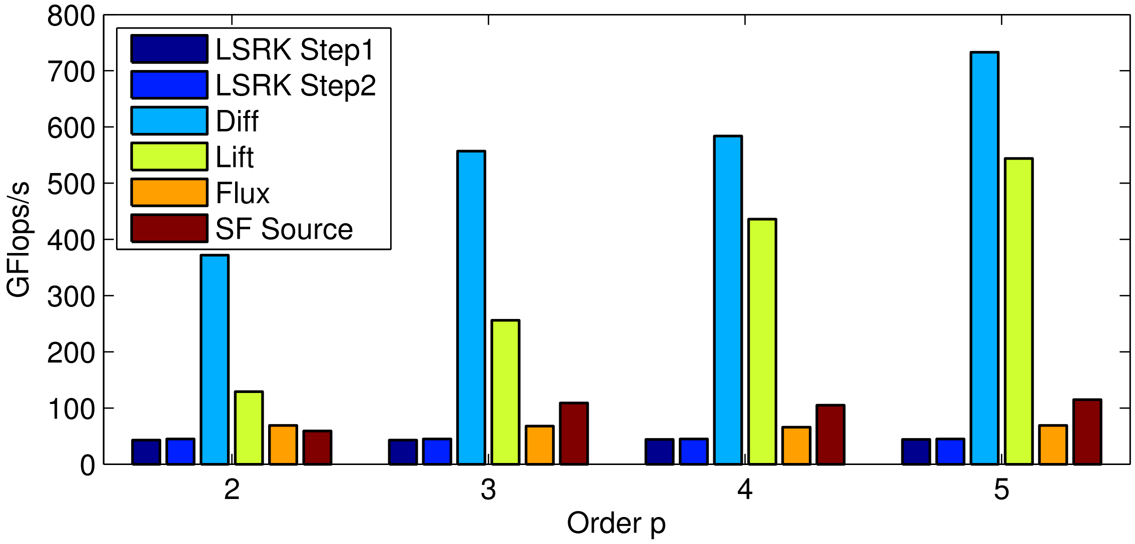


Figure 5.22: *Computational power in GFlops/s of time-consuming kernels for polynomial orders $p = \{2, 3, 4, 5\}$.*

For the performance analysis, the system is simulated for a series of h-refined meshes and polynomial orders $p = \{2, \dots, 5\}$ for 20 time-steps, i. e., 280 evaluations of the DG operator are performed.

In Fig. 5.20, the speedup is depicted in dependence of K. In agreement with the previous test cases, the speedup is approximately 30-50 and constant in K. The analysis of the kernel execution ratio, depicted in the left panel of the Fig. 5.21, shows similar behavior, too. The SF source kernel consumes more time than the Drude kernel. For this system, their total share amounts to less than 4%.

The computational power of time-consuming kernels is summarized in Fig. 5.22. In contrast to other bandwidth limited kernels, the SF source kernel shows moderate increase with order p in the range of 60-120 GFlops/s. The behavior of the remaining kernels is similar to the previous test cases with maximum computational power of 700-800 GFlops/s achieved by the differentiation kernel.

As in most cases, the mesh of this test system is dominated by non-metallic elements and, hence, τ must behave similarly to previously presented test cases, which is clearly observed in the right panel of Fig. 5.21.

5.4 Summary

Based on the fundamental hardware design paradigm of graphics processing units and NVidia's platform for general purpose GPU applications `CUDA`, performance relevant issues and the responsibilities of the programmer to achieve a high-performance DGTD implementation were presented. The instruction to byte ratio was introduced as the basic guideline to optimal performance. It was demonstrated that math libraries implementing standard matrix-vector operations cannot yield a sufficiently high I/B ratio and are, hence, not suitable for a high-performance implementation of the DGTD method.

The algorithm of the DGTD method was analyzed with respect to the I/B ratio, and suitable and ineligible sections for implementation on the GPU were identified. For linear elements used throughout the thesis, the computationally intense matrix-vector-products of the element-local differentiation and the lift of the flux vectors were identified to yield a tunable I/B ratio upon reuse of the reference element operators. Respective kernels of tunable I/B ratio were implemented and can take advantage of the GPU's computational power. Based on a thorough discussion of the partitioning of the DGTD algorithm into kernels, taking performance, maintainability, and modularity of the code into account, kernels for the LSRK time integration, the flux calculation, the ADEs of PMLs, the Drude, and the Drude-Lorentz model, as well as total-field/scattered-field and scattered-field sources of various space- and time-dependencies were implemented.

Several test cases were employed to verify correct functionality of the implementation. Three of these test cases, for which analytic reference solutions of the electromagnetic fields are available, were presented. Many more tests were performed by direct comparison of the well-established CPU and newly developed GPU implementation in order to guarantee correctness and stability. These tests also demonstrate that the reasoning which lead to partitioning of the code into kernels was justified.

The implementation of the core components of the method, i. e., the local-element differentiation and the lift operation were thoroughly described by algorithms with emphasis on the concept of the tunable I/B ratio. Furthermore, the memory layout of the implementation was described since the minimization of memory traffic is one of the most important aspects to high performance.

Employing the test cases mentioned above, the performance with respect to the well-established CPU implementation was analyzed. For polynomial orders $p = \{2, \dots, 5\}$ and practically independent of the number of elements K, the GPU code provides a speedup of 30-55 over the CPU implementation[9].

The calculation time per element and stage τ was shown to be a p-dependent constant with respect to K for the GPU implementation, i. e., the calculation time scales linearly with K. The differentiation and lift kernels yield computational powers of up to almost 800 and 600 GFlops/s, which corresponds to roughly 50% of the theoretical value. This tremendous reduction of simulation time allows us to approach computationally intense problems such as the EELS analysis of coupled split-ring resonator systems, presented in Chap. 7.

[9] We compare the calculation times of simulations performed in single precision on a NVidia® GeForce® GTX580 (GPU system) and an Intel® Core™ i7 CPU 970 @ 3.20 GHz (CPU system). The CPU system employs a single core.

6

Chapter 6

Optical Characterization of Metallic Nano-Antennae

Metallic nano-antennae convert electromagnetic radiation into localized energy and heat. In the case of conversion into localized energy, incident light is converted into localized surface plasmons and plasmon polaritons. These plasmons can be resonantly enhanced resulting in a local field enhancement, i. e., an amplification of the local field with respect to the incident field. This locally confined, large field is utilized in a variety of methods and techniques, e. g., surface enhanced Raman scattering (SERS), surface enhanced infrared absorption (SEIRA), near-field imaging, nonlinear optical frequency conversion and many more, see e. g. [2, 3] and references therein. Below the plasma frequency, the field of a plasmon is strongly confined to the surface and depends on its environment, hence, metallic nanoparticles and -antennae can serve as sensing devices, too.

Depending on the application of interest, a nano-antenna is designed to yield a defined ratio of scattering to absorption. In photothermal cancer therapy [73], for example, ideally all incident light is converted to heat, i. e., a large absorption cross section is desired. In order to tune the properties of the nano-antenna, characterization methods must be established that quantitatively measure the extinction, the scattering, and the absorption cross section of *individual* nanostructures. Recently, Husnik *et al.* [12] presented an experimental setup which determines these cross sections simultaneously and quantitatively for the first time. Our contribution to their analysis of a series of nano-antennae by numerical simulations employing the DGTD method is presented in this chapter.[1]

In order to verify the experimental setup and the associated analysis, the extinction, the scattering, and the absorption cross section of a series of nano-antennae are calculated by the DGTD method on GPUs. As in the experiment, we investigate the transition of a straight gold dipole nano-antenna, referred to as nanorod, into a split-ring resonator by bending wires of roughly equal length into a “U”-like shape. As a consequence, the nanorod and the split-ring resonators show similar resonance wavelengths.

For an ideal electric dipole, the scattering cross section scales quadratically with the dipole moment [32]. As the response of the antennae considered here is dominated by the dipole term of a multipole expansion and the dipole moment directly depends on the separation of charges, we expect the dipole antenna to show a larger scattering cross section than the split-ring resonator. Since the nano-antennae are fabricated from “wires” of nearly the same length, their ohmic resistance and, hence, their absorption cross section is supposed to be approximately equal. As a result, in transition from

[1] The general structure of this chapter is guided by Ref. [12].

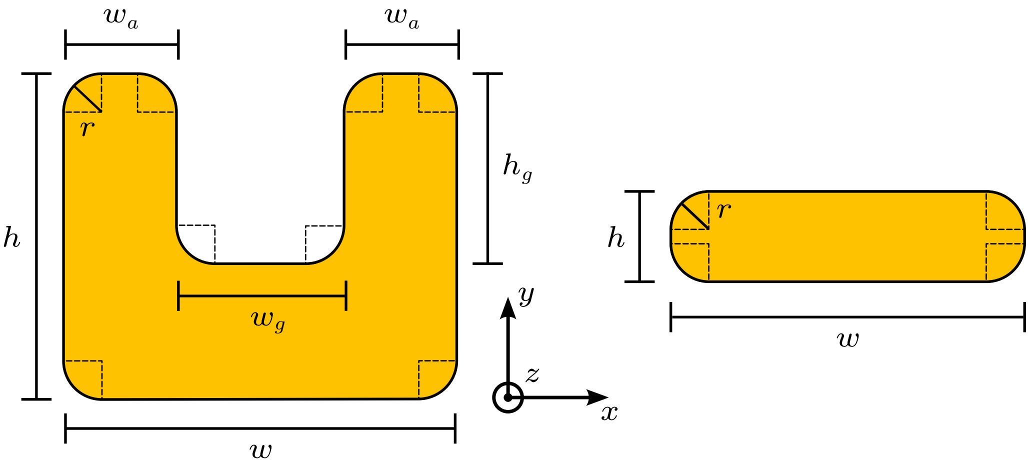


Figure 6.1: *Parametrization of the split-ring resonator (left panel) and the nanorod (right panel). The split-ring resonator is defined by its width w, height h, thickness t (not shown), gap width w_g, gap height h_g, edge radius r and mirror symmetry. Both arms are of equal width w_a. The nanorod is defined by its width w, height h, thickness t (not shown) and edge radius r.*

a nanorod to a split-ring resonator, the ratio of scattering and absorption cross section is expected to decrease. We investigate these aspects in the remainder of this chapter, which is structured as follows:

The nano-antennae, nanorods and split-ring resonators, are parametrized according to a geometric model presented in Sec. 6.1. In Sec. 6.2, the setup of the simulation including tetrahedral meshes, material models, excitation, and probing is described. Experimentally and numerically obtained extinction, scattering and absorption cross sections of the nano-antennae are compared and discussed in Sec. 6.3. The dependence of the scattering to absorption ratio of a straight dipole antenna on the ohmic resistance is analyzed in Sec. 6.4. Finally, errors and the influence of several material, geometry, and simulation parameters are addressed in Sec. 6.5.

6.1 Parametrization of Split-Ring Resonators and Nanorods

The numerical treatment of physical systems by means of the DGTD method demands a geometric model of the system in the form of a mesh. As alluded to in Sec. 3.1.2, we employ tetrahedra as building blocks for the mesh. The meshes themselves are created by NETGEN [74], an open source meshing tool. We define the geometry using constructive solid geometry, i. e., cuboids, cylinders, spheres etc. are used in conjunction with boolean operations to describe the geometry of the system. As a consequence, the split-ring resonator and the nanorod must be parametrized in terms of these objects.

The definition of the geometric model and the associated parameters are depicted in Fig. 6.1. The split-ring resonator is defined by its width w, height h, thickness t, gap width w_g, gap height h_g, and edge radius r together with the property of mirror symmetry, i. e., the SRR is located such that the structure is symmetric with respect to $x = 0$. Consequently, the arms are equal in width. The thickness t is the system's extent in z-direction and is not to be confused with the height h, i. e., the extent in y-direction. Cylinders, parametrized by the edge radius r and indicated by the dashed guides in Fig. 6.1, are used to model edges which account for the fact that fabricated nanostructures do not exhibit sharp edges — at least in the lateral dimensions.

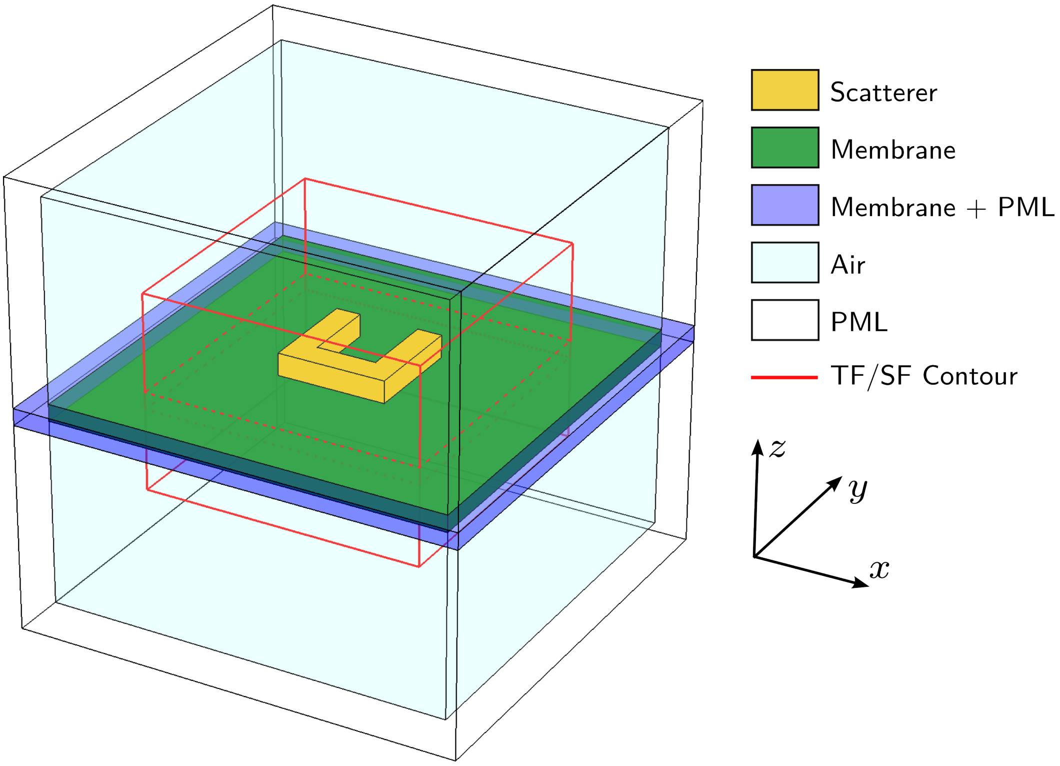


Figure 6.2: *Sketch of the simulation setup. The scatterer (gold), either split-ring resonator or nanorod, is located on a silicon nitride membrane (green). Both, scatterer and membrane are embedded in air (cyan). The computational domain is terminated by PMLs in all directions (white). The blue region represents colocated membrane and PML property, and the red box indicates the contour of the total-field/scattered-field source.*

The nanorod is modeled by a rectangular cuboid of width w, height h, and thickness t with round edges, as before, defined by the edge radius r.

6.2 Simulation Setup

An overview of the system setup is depicted in Fig. 6.2. The scatterer, either a split-ring resonator or a nanorod, is located on a silicon nitride membrane of 30 nm thickness whose material is modeled by a dielectric of constant permittivity $\epsilon_{\text{mem}} = 4$. In analogy to the experimental setup, the scatterer and the membrane are embedded in air described by $\epsilon = 1$. It is important to include the "airbox" as the electromagnetic fields of the source propagate therein. However, due to the limited amount of computer memory and finite operation speed, an infinitely extended system cannot be simulated. In order to mimic open space, the computational domain is terminated by PMLs in all directions which, in conjunction with Silver-Müller boundary conditions on the outmost surface, absorb outgoing radiation. Note that the blue region in Fig. 6.2 models both membrane and PML properties.

The scatterers themselves are defined in terms of the model presented in Sec. 6.1. The model parameters of the four systems of interest, classified as three split-ring resonators and one nanorod, were deduced from SEM images of the respective experimental structures and are summarized in Tab. 6.1.

	$w\,[nm]$	$h\,[nm]$	$t\,[nm]$	$w_g\,[nm]$	$h_g\,[nm]$	$r\,[nm]$
Split-ring resonator	139	146	35	52	88	19
Intermediate SRR #1	182	124	35	92	68	18
Intermediate SRR #2	227	104	35	145	48	17
Nanorod	292	55	35	-	-	21
Nanorod 35 nm Au	283	44	35	-	-	21
Nanorod 25 nm Au, 10 nm Cr	283	49	35	-	-	21

Table 6.1: *Geometry parameters of the scatterers investigated in this chapter based on the model defined in Sec. 6.1. The parameters of split-ring resonators and nanorods considered in Sec. 6.3 and Sec. 6.4 are summarized in the upper and lower section, respectively. Thereof constructed tetrahedral meshes are depicted in Fig. 6.3 and Fig. 6.4.*

	Gold	**Chromium**
ϵ_∞	1.00	13.6
ω_D	13.8×10^{15} Hz	7.81×10^{15} Hz
γ_D	1.08×10^{14} Hz	6.75×10^{14} Hz
$\Delta\epsilon_L$	0	17.77
ω_L		1.93×10^{15} Hz
γ_L		1.91×10^{15} Hz

Table 6.2: *Material parameters for gold and chromium in the near-infrared. The values for gold were originally published by Johnson and Christy [38]. The Drude-Lorentz parameters for chromium were obtained by a fit in the range $800 - 2000\,nm$ with experimental data from Ref. [75]. The associated permittivities are depicted in Fig. 6.5. Note that $\gamma_{D,Au}^{sim} = 1.75 \cdot \gamma_D = 1.88 \times 10^{14}$ Hz is used for simulations in practice.*

Tetrahedral meshes modeling the scatterers on the silicon nitride membrane, each consisting of approximately 37k elements, are depicted in Fig. 6.3.

In the experiment, the scatterers were fabricated from gold, hence, the material model of the simulation must account for the dispersive properties of the metal. In the (near-)infrared regime, where the fundamental resonance of the structures is expected, the electromagnetic response of gold is well described by a Drude model (cf. Sec. 2.2.2). The free parameters of the Drude model are deduced from Johnson and Christy's famous paper [38] and are listed in Tab. 6.2.

As is known from previous studies [9], the fabrication process leads to internal grain boundaries and rough surfaces of the gold structures. This leads to additional scattering of electrons and electronic surface states which, in turn, lead to significantly higher damping when compared to the measurements of Johnson and Christy. For this reason, we increase the damping used in the simulations by a factor 1.75, i. e., $\gamma_{D,\mathrm{Au}}^{\mathrm{sim}} = 1.75 \cdot \gamma_{D,\mathrm{Au}}$ is employed for the calculations of this chapter. The manipulation of γ_D influences both the real and imaginary part of the permittivity as depicted in Fig. 6.5.

The system is illuminated by a Gaussian pulse with plane wave spatial profile polarized in x-direction, i. e., parallel to the base of the split-ring resonator and to the long axis of the nanorod. Technically,

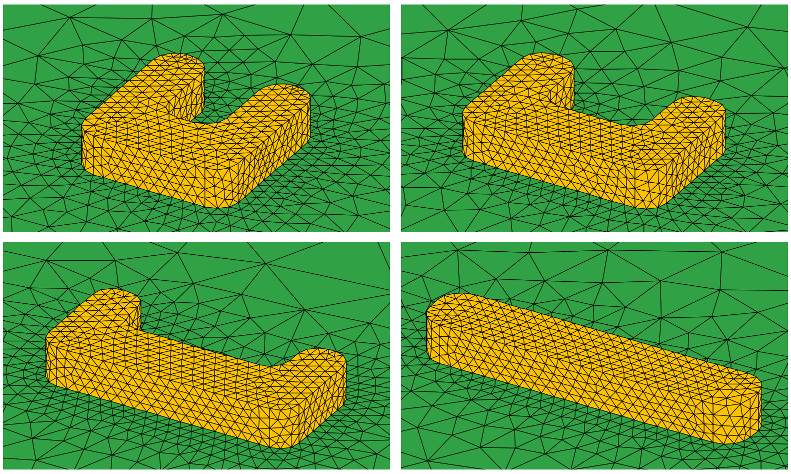


Figure 6.3: *Section of tetrahedral meshes modeling scatterers (gold) in transition from the split-ring resonator (top left) to the nanorod (bottom right) with two intermediate split-ring resonators on a silicon nitride membrane (green). The parameters according to the model in Fig. 6.1 are listed in Tab. 6.1.*

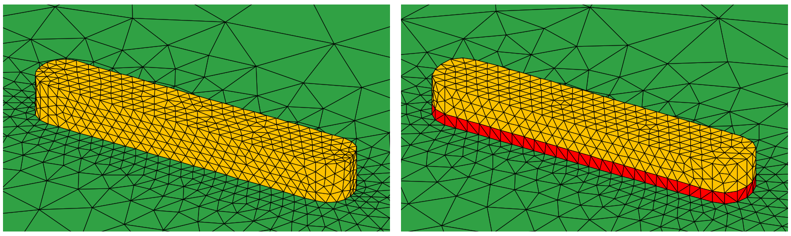


Figure 6.4: *Section of tetrahedral meshes modeling nanorods of gold (left panel) and layered chromium (red) and gold (right panel) on a silicon nitride membrane (green). The parameters according to the model in Fig. 6.1 are listed in Tab. 6.1.*

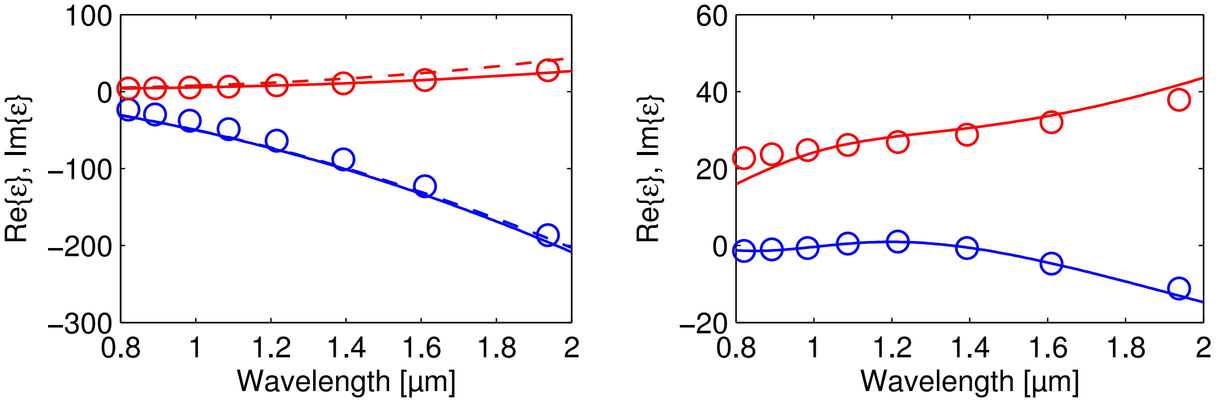


Figure 6.5: *Real (blue) and imaginary (red) part of the permittivity of gold (left panel) and chromium (right panel). Circles represent experimental data of Johnson and Christy for gold [38] and chromium [75]. The permittivities are modeled by a Drude and a Drude-Lorentz model (solid lines), respectively. Associated material parameters are listed in Tab. 6.2. In the left panel, the dashed lines result from adjusting the damping to* $\gamma^{sim}_{D,Au} = 1.75 \cdot \gamma_{D,Au}$.

the source is implemented by a total-field/scattered-field source (cf. Sec. 4.2.1). Therefore, the TF/SF contour, indicated by a red box in Fig. 6.2, divides the computational domain into the total-field region, including the scatterer and parts of the membrane on the inside, and the scattered-field region on the outside.

Employing our GPU implementation of the DGTD method, the simulations are performed in fourth order spatial discretization, i. e., $p = 4$, which yields sufficiently converged results for all systems considered in this chapter [14]. The simulation is run for 166 fs which ensures at least 250 optical cycles for waves of $\lambda = 2000$ nm, the longest wavelength (slowest oscillation) under consideration.

The scattering and absorption cross sections are calculated from the flux through a closed surface around the scatterer (cf. Sec. 2.3). Therefore, the total field on the inside and the scattered field on the outside of the TF/SF contour are processed according to Eq. (2.42) and Eq. (2.43), respectively. Finally, the extinction cross section is computed by Eq. (2.44). Note that this setup has been checked for correctness in Sec. 5.3.2.

6.3 Extinction, Scattering, and Absorption Cross Section

In this section, we compare extinction, scattering, and absorption cross sections obtained by theory and experiment for the four systems defined in Sec. 6.2. Husnik *et al.* [12] developed an experimental setup which, for the first time, determines both the extinction and the scattering cross section of individual nanostructures simultaneously and quantitatively. Consequently, the absorption cross section is calculated from extinction and absorption by Eq. (2.44). The experimental setup is based on a spatial modulation technique combined with a common-path interferometer. Their analysis employs the optical theorem and the ideal-electric-dipole approximation which limits the approach to structure sizes small compared to the wavelength of the incident light. For details about the experimental setup and the analysis, the reader is referred to Ref. [12].

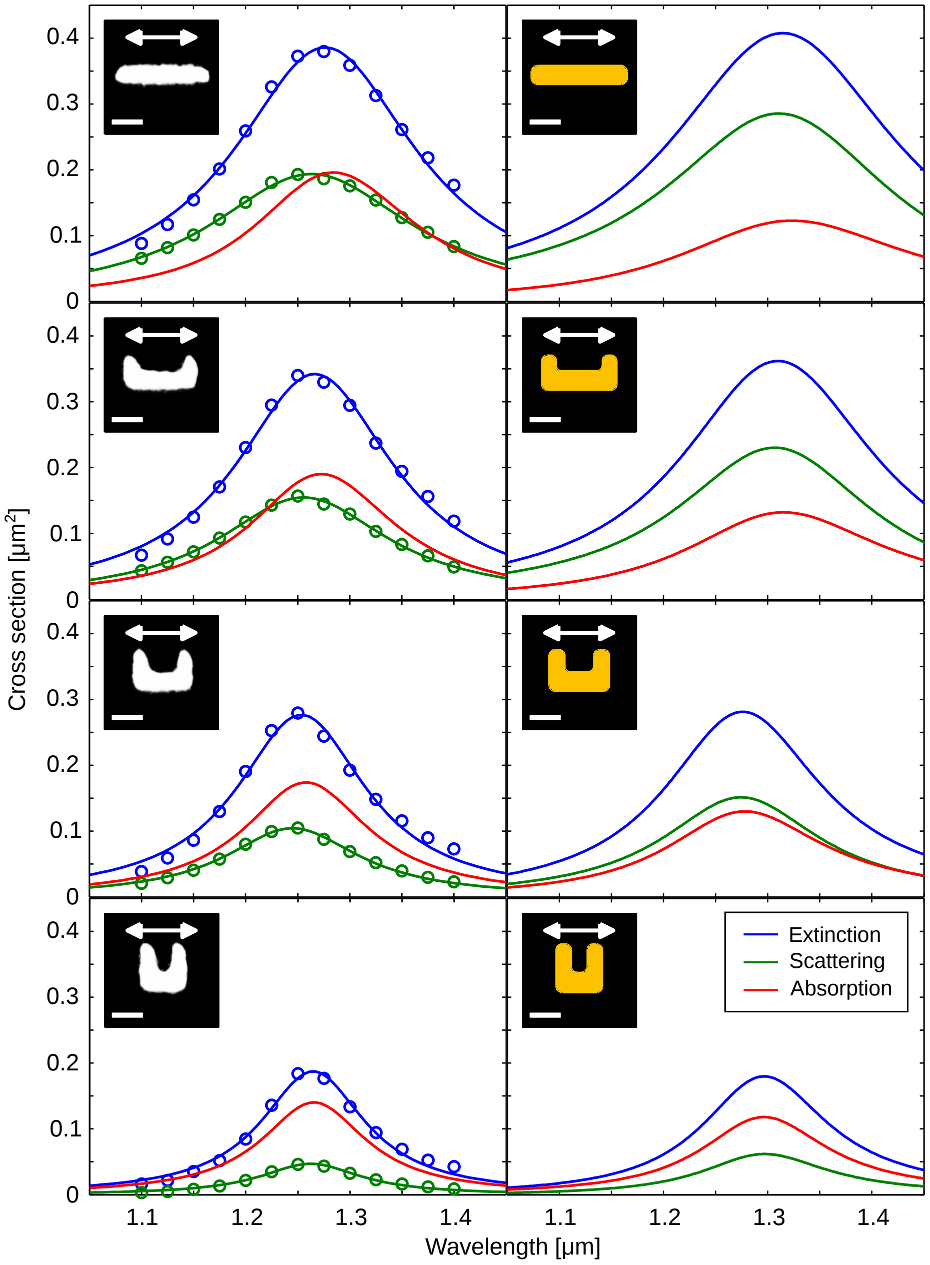


Figure 6.6: *Extinction (blue), scattering (green) and absorption (red) cross-section spectra of scatterers in transition from a nanorod to a split-ring resonator (top to bottom) in absolute units. The left column shows experimental values (circles) and Lorentzian fits (solid lines) as guides to the eye (data by courtesy of M.Husnik and M. Wegener). The simulated spectra are located in the right column. The scale bar of the insets is 100 nm, and the doublearrow indicates the polarization of the incident electric field.*

The extinction, scattering and absorption cross-section spectra of both the experiment and the simulation of the four systems are summarized Fig. 6.6. In the experiment, only extinction and scattering cross section data is accessible as marked by circles. The absorption cross section is calculated employing Eq. (2.44).

As intended, all four systems exhibit approximately the same resonance wavelength for both experiment and simulation. However, a slight, consistent red-shift is observed in the numerical results compared to the experiment.

The absolute values of the extinction cross section agree for all structures while the scattering and absorption cross sections slightly differ, especially for the dipole-like nanorod. In transition from the nanorod to the split-ring resonator, the absorption cross section decreases slightly in the experiment and is practically constant in the simulation due to the similar ohmic resistances of all structures. For both cases, the scattering cross section drops substantially in accordance with the anticipated smaller dipole moment of the SRR compared to the nanorod. In a consistent manner throughout all structures, the measured absorption is slightly larger than the calculated one while the simulated scattering cross section is larger than the measured one.

All tendencies discussed in the introduction of this chapter could be confirmed by overall good agreement between experiment and simulation. Nevertheless, deviations of experiment and simulation as well as the influence of material and geometry parameters are addressed in Sec. 6.5.

6.4 Tuning of the Scattering to Absorption Cross Section Ratio

Despite the shape of the nano-antenna, the ratio of scattering and absorption cross section also depends on the ohmic loss of the structure. Focusing on the nanorod, it is this dependence that we want to investigate in this section by manipulation of the nano-antenna's ohmic resistance. Experimentally, the ohmic resistance of the nanorod is manipulated by introduction of chromium, which is known to possess a much higher damping than gold. Consequently, the scattering to absorption cross section ratio is expected to decrease.

In both the experiment and the simulation the nanorod is structured as a layered system of 10 nm thickness of chromium underneath 25 nm of gold. A nanorod of 35 nm of gold is used for comparison. The remaining parameters were deduced from SEM images of the experimental structures according to the model in Sec. 6.1. Tab. 6.1 summarizes the geometry parameters, and, visualizing the material distribution, Fig. 6.4 depicts tetrahedral meshes of both nanorods.

The Drude model of gold, used in the previous section, is reemployed. Chromium is described by a Drude-Lorentz model, fitted in the region 800 nm - 2000 nm to experimental data from Ref. [75] employing the method of Sec. 2.2.4. The dispersion relation of both metals is depicted in Fig. 6.5, and the related material model parameters are listed in Tab. 6.2.

The cross section spectra of extinction, scattering and absorption for both experiment and simulation are depicted in Fig. 6.7. The resonance positions of experiment and simulation match well. The chromium layer causes a strong red-shift of the resonance wavelength due to the almost positive real part of the permittivity $\Re\left\{\epsilon_{\mathrm{Cr}}\right\}$ (cf. Fig. 6.5). As expected, the ratio of scattering to absorption decreases substantially due to the increase of the ohmic resistance. Interestingly, it is the scattering cross section that drops significantly while the absorption remains practically constant in absolute value. As in the previous section, the simulation shows slightly larger scattering and smaller absorption compared to the experiment. The anticipated trends, however, are confirmed.

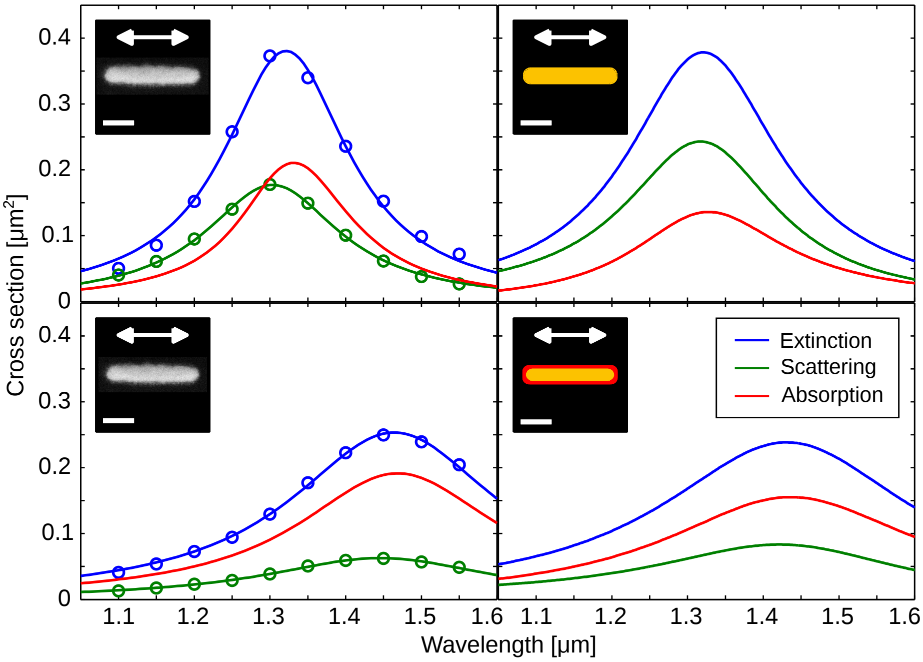


Figure 6.7: *Extinction (blue), scattering (green), and absorption (red) cross-section spectra of nanorods of 35 nm gold (top row) and layered 25 nm gold on 10 nm chromium (bottom row) in absolute units. The left column shows experimental values (circles) and Lorentzian fits (solid lines) as guides to the eye (by courtesy of M.Husnik and M.Wegener). Simulated spectra are located in the right column. The scale bar is 100 nm, and the doublearrow indicates the polarization of the incident electric field.*

6.5 Errors and Parameter Tendencies

At last, sources of errors and the influence of the parameters involved in the previously presented simulations are addressed. We distinguish between systematic errors caused by the employed methods, e. g., the spatial discretization by the DG method, and problem-specific errors induced by, e. g., geometry and material parameters.

The 14-stage 4^{th}-order LSRK time-integration method ensures that the error is dominated by the spatial discretization which, for polynomial order $p = 4$ (cf. Sec. 6.2), yields sufficiently converged results, as reported in Ref. [14]. In addition, we check for correct functionality of the TF/SF source for each of the presented results. Therefore, the metallic elements of a system are assigned $\epsilon = 1$ which resembles the source's reference behavior. Ideally, the observed extinction, scattering, and absorption cross sections are identical to zero, as, in essence, the source's reference system is simulated. Due to the limited precision of numeric simulation, non-zero cross sections several orders of magnitude below the reported absolute values of the previous sections are observed which, in turn, indicates correct functionality of the source.

Deviations between simulation and experiment are consequently caused by the description of the system itself. The abstract representation of the system's geometry by a tetrahedral mesh introduces deviations. For example, the roughness of an experimental sample is not modeled as the surfaces of the scatterers are described by simple objects such as cylinders, spheres, and planes. The lateral extent of the scatterer is well determined by SEM images of nm-precision. In contrast, the thickness of a structure is estimated rather than measured as fabrication rates are used to determine the sample's thickness.

An extensive parameter study on gold split-ring resonators, including the influence of the thickness, is presented in Ref. [14]. Therein, in the range of $20 - 50\,\text{nm}$, increasing thickness causes a blue-shift of the resonance wavelength, a decrease of absorption, and an increase of scattering while extinction remains practically constant. This behavior was consistently reproduced for both, the SRR and the nanorod. However, increased thickness would compensate only for the red-shift between theory and experiment and worsen scattering, absorption, and extinction values.

Another source of uncertainty arises from the material parameters of gold. Due to the surface roughness of the experimental sample, which is accounted for by an artificially increased damping parameter $\gamma_{D,\text{Au}}^{\text{sim}}$, a realistic material model is difficult to state. By Eq. (2.24), increasing γ_D results in a practically constant $\Re\{\epsilon(\omega)\}$ and strongly increasing $\Im\{\epsilon(\omega)\}$ employing the material parameters of Tab. 6.2. For both, the nanorod and the SRR, this results in a decrease of scattering and an increase of absorption while leaving the resonance wavelength practically unchanged. Based on these tendencies, a combination of increased thickness and γ_D approaches the experimental values in all aspects, i. e., resonance wavelength, scattering, absorption, and extinction cross sections. However, it should be emphasized that we did not "tune" the geometry and material parameters to artificially match experiment and theory.

At last, the uncertainties of the chromium model are addressed. The experimental values of ϵ_{Cr} in the near-infrared of Johnson and Christy [75] and Palik [40] differ considerably. The influence of the thickness of the chromium layer while retaining the total thickness of the nanorod was investigated using the values of Johnson and Christy. Increasing the ratio of chromium results in a red-shift of the resonance wavelength due to the "quasi dielectric" behavior of chromium, i. e., $\Re\{\epsilon_{\text{Cr}}(\lambda)\} \gg \Re\{\epsilon_{\text{Au}}(\lambda)\}$. Furthermore, both the scattering and the absorption cross section decrease slightly.

7

Chapter 7

Electron Energy Loss Spectroscopy of Split-Ring Resonators

Electron energy loss spectroscopy is a characterization method for nanostructures which gained considerable attention during the last decade [2, 76]. The method is based on the analysis of the energy distribution of initially monoenergetic electrons after their interaction with a specimen. A variety of physical systems such as bulk, films, nanoparticles and nanostructures have been investigated, see Refs. [71, 76–79] and references therein. The energy loss suffered by an electron ranges from sub-eV to keV giving access to a large number of physical phenomena, e. g., inter and intra band transitions, plasmon excitations, inner shell ionizations and Čerenkov radiation [80].

In this thesis we are interested in the properties of plasmonic structures with typical energy loss below 10 eV probed by "fast" electrons with kinetic energy larger than 100 keV, often referred to as valence EELS. In practice, these analyses are conducted using scanning transmission electron microscopes (STEM), where the sample is scanned by the electron beam and the electrons' energy distribution after interaction with the specimen is analyzed. A large number of electrons contributes to the zero loss peak (ZLP), i. e., the initial energy of the electrons is distributed around a mean value. Scattered electrons whose interaction energy is smaller than the width of the ZLP cannot be detected as is the case for, e. g., phonon interaction. The ZLP consequently defines the minimum detectable interaction energy for EELS and is continuously minimized to increase the sensitivity of the method reaching 0.3 eV in the best STEM-EELS setups [81]. In contrast to the experiment, the excitation employed in the numerical simulation is perfectly monoenergetic and, hence, the energetic resolution of the EELS analysis is not limited by the ZLP.

Inelastically scattered electrons contain information about the structure of the specimen "encoded" in their energy loss suffered from the interaction with the specimen, i. e., each electron loses an amount of kinetic energy specific to the interaction process. In the experiment, the energy loss is analyzed using a variety of energy filters. For technical details about the setup of the experiment the reader is referred to Refs. [76, 80]. In order to study fundamental interaction processes using EELS, multiple scattering processes must be avoided. Consequently, the specimen is limited in thickness by the mean free path which is about $50 - 150$ nm for 100 keV electrons [80]. Furthermore, extremely thin membranes are employed to minimize the influence of the supporting structure.

An outstanding characteristic of STEM-EELS, rendering it a suitable method for the investigation of nanostructures, is the sub-nm spatial [78] *and* sub-eV energetic resolution with reported sensitivity below 0.2 eV [25, 71, 76, 79]. In addition, due to the point-like excitation, optically bright and dark modes can be efficiently excited [82, 83].

Several nanostructures of different shapes consisting of various materials have been investigated, e. g., nanotriangles of silver [78] and gold [84], silver nanorods [85] as well as split-ring resonators of silver [86] and gold [25] with resonances in the near-infrared to visible regime. Surprisingly, only few structured systems composed of more than one metaatom have been considered, e. g., dimers of silver spheres, both, experimentally [82] and theoretically [24], and formations of gold [79]. In the context of this thesis, individual and coupled systems of gold split-ring resonators were investigated numerically and compared to experimental results [13].

The fundamentals of the EELS analysis are presented in Sec. 7.1 including the calculation of the loss probability — the central quantity in an EELS experiment representing the loss distribution suffered by an electron — from a time-domain simulation and the simulation of a relativistic electron in the context of the DGTD method. The single split-ring resonator located on a 30 nm silicon nitride membrane is analyzed in great detail using EELS in Sec. 7.2. After a comparison of the EELS analysis to a variety of other spectroscopic methods, the remainder of the thesis is dedicated to the analysis of systems of split-ring resonators composed of two and four metaatoms in Sec. 7.3.

7.1 Fundamentals of the Electron Energy Loss Spectroscopy Analysis

In this section, the theoretical fundamentals required for the EELS analysis using the DGTD method are presented. As alluded to in the introduction of this chapter, EELS is based on the analysis of the energy distribution of initially monoenergetic electrons after their interaction with a specimen. On a theoretical level, only a single electron is considered for the analysis. The theoretical analog to the energy distribution is the loss probability P which describes the probability for an electron to lose a specific amount of energy ΔE with respect to its initial (kinetic) energy. The loss probability is subsequently associated with an interaction process, e. g., the excitation of a (surface) plasmon mode. The calculation of the loss probability by a time-domain numerical simulation is described in Sec. 7.1.1.

In contrast to the experiment, a single relativistic electron is required as a source of radiation to perform the analysis presented in Sec. 7.1.1. In order to include the relativistic electron in a DGTD simulation, its space- and time-dependent electric field must be known. Hence, details about the excitation in the context of the DGTD method are addressed in Sec. 7.1.2.

7.1.1 Calculation of the Loss Probability in the Time-Domain

We follow the classical dielectric formalism [71] to describe the energy loss suffered by a single fast electron moving at constant velocity $\vec{\mathrm{v}}$ along the straight line trajectory $\vec{\mathrm{r}}_{\mathrm{e}}(t) = \vec{\mathrm{r}}_0 + \vec{\mathrm{v}}t$. This implies that the fields induced by the electron do not influence the momentum of the electron which is a reasonable assumption for fast electrons and is known as the no-recoil approximation (NRA) [71]. The energy loss is related to the force exerted by the induced electric field acting back on the electron by

$$\Delta E = e \int\limits_{-\infty}^{\infty} \mathrm{d}t \, \vec{\mathrm{v}} \cdot \vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}}_{\mathrm{e}}(t)\,, t\big)\,, \tag{7.1}$$

where $\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}}, t\big)$ is the electric field of the induced plasmonic oscillation and e is the elementary charge. Alternatively, the energy loss ΔE is expressed by

$$\Delta E = \int\limits_{0}^{\infty} \mathrm{d}\omega \, \hbar\omega P(\omega)\,, \tag{7.2}$$

where the loss probability $P(\omega)$ accounts for the probability of losing the energy $\hbar\omega$ during interaction. Replacing $\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}},t\big)$ by its Fourier transform in Eq. (7.1) and exploiting $\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}},\omega\big) = -\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}},\omega\big)^*$, the loss probability expressed in terms of the induced field reads

$$P(\omega) = \frac{e}{\pi\hbar\omega}\int\limits_{-\infty}^{\infty} \mathrm{d}t\,\Re\left[e^{-\mathrm{i}\omega t}\vec{\mathrm{v}}\cdot\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}}_{\mathrm{e}}(t)\,,\omega\big)\right], \tag{7.3}$$

where $\Re[.]$ denotes the real part. In order to examine the loss probability, a Fourier transform of $\vec{\mathrm{E}}^{\mathrm{ind}}\big(\vec{\mathrm{r}},t\big)$ must be calculated for every point along the trajectory $\vec{\mathrm{r}}_{\mathrm{e}}(t)\,, t \in \big[-\infty,\infty\big]$. Eq. (7.3) is the central equation of the EELS analysis as it provides a straightforward relation between the loss probability and the induced electric field which is directly accessible everywhere in the computational domain when exclusively employing SF sources (cf. Sec. 4.2.2). When employing total-field/scattered-field sources as is the case in Ref. [24], the induced field is directly accessible in the scattered-field region only.

7.1.2 Excitation of a Relativistic Electron in DGTD

The electromagnetic field of a fast electron can be regarded as an evanescent source of radiation “traveling” through space. The associated dispersion relation lies outside the light cone to which propagating optical fields, e. g., plane waves in homogeneous dielectric media, are limited. As a consequence, surface plasmons can be directly excited by fast electrons [71].

The relativistic electron is modeled by a scattered-field source, introduced in Sec. 4.2.2, which exploits several advantages over the originally proposed total-field/scattered-field technique used in Ref. [24]. First, the mesh does not need to be adapted in order to raster-scan a nanostructure. Second, using a total-field/scattered-field source introduces a conceptual error when scanning the nanostructure. The field of the electron does not induce electromagnetic fields in the scattered-field region which is the fundamental property of the total-field/scattered-field source. This becomes an issue when the scattered-field region intersects with the metal structure where a field is supposed to be excited on the entire structure and not exclusively in the total-field region.

The formalism of the scattered-field source requires the space- and time-dependent electric and magnetic field $\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big)$ and $\vec{\mathrm{H}}\big(\vec{\mathrm{r}},t\big)$, respectively. An electron modeled by a point charge moving at constant velocity $\vec{\mathrm{v}}$ along the trajectory $\vec{\mathrm{r}}_{\mathrm{e}}(t) = \vec{\mathrm{r}}_0 + \vec{\mathrm{v}}t$ is described by the charge density $\rho\big(\vec{\mathrm{r}},t\big) = e\delta\big(\vec{\mathrm{r}}-\vec{\mathrm{r}}_{\mathrm{e}}(t)\big)$ and the resulting current density $\vec{\mathrm{j}}\big(\vec{\mathrm{r}},t\big) = \vec{\mathrm{v}}\rho\big(\vec{\mathrm{r}},t\big)$ [24]. The electric and magnetic fields are determined from the Liénard-Wiechert potentials [30] and read

$$\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big) = \frac{\gamma\vec{\mathrm{d}}\big(\vec{\mathrm{r}},t\big)}{\left(\left|\vec{\mathrm{d}}\big(\vec{\mathrm{r}},t\big)\right|^2 + (\gamma c)^{-2}\left(\vec{\mathrm{d}}\big(\vec{\mathrm{r}},t\big)\cdot\vec{\mathrm{v}}\right)^2\right)}\,, \quad \vec{\mathrm{H}}\big(\vec{\mathrm{r}},t\big) = \frac{\vec{\mathrm{v}}}{c}\times\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big)\,, \tag{7.4}$$

where $\vec{\mathrm{d}}\big(\vec{\mathrm{r}},t\big) = \vec{\mathrm{r}}_{\mathrm{e}}(t) - \vec{\mathrm{r}}$ and $\gamma = \left(1-\big(v/c\big)^2\right)^{-1/2}$ denotes the Lorentz factor.

7.2 The Individual Split-Ring Resonator

In this section, we perform an electron energy loss spectroscopy analysis of a single gold split-ring resonator and compare the results to a variety of other characterization methods. As alluded to in the introduction of this chapter, single split-ring resonators fabricated from silver [86] and gold [25] have been analyzed in great detail including parameter studies on size-dependence. The fundamental characteristics of the system and the analysis of the individual split-ring resonator are, however, the basis of the investigation of coupled split-ring resonator systems in Sec. 7.3 where two and four individual split-ring resonators form the physical system. Therefore, we employ this section to familiarize the reader with the various setups and analysis methods applied in the remainder of the thesis, organized as follows:

In Sec. 7.2.1, the setup of the EELS analysis including details about the geometry of the system, applied materials, and the numerical setup is presented and followed by a verification of the applied settings in Sec. 7.2.2. The results of the characterization — EELS spectra, deduced resonances and EELS maps — are presented in Sec. 7.2.3. The results of the EELS analysis and other characterization methods are compared in the subsequent sections. At first, harmonic inversion is employed to analyze the time-domain electric field in Sec. 7.2.4. Next, the mode profile of the electronic eigenmodes is calculated and compared to the EELS maps in Sec. 7.2.5. Finally, the extinction cross section is calculated to verify the declaration of optical bright and dark modes and cross check the resonance energies by EELS and the harmonic inversion in Sec. 7.2.6.

7.2.1 Simulation Setup

We investigate a single split-ring resonator defined by $w = 195\,\mathrm{nm}$, $w_g = 107\,\mathrm{nm}$, $h = 186\,\mathrm{nm}$, $h_g = 137\,\mathrm{nm}$, $r = 18\,\mathrm{nm}$ (cf. Fig. 6.1) and thickness 40 nm located on a membrane of 30 nm thickness. A schematic overview of the system setup (not drawn to scale) is depicted in Fig. 7.1. The silicon nitride membrane is modeled by an isotropic, non-dispersive material of $\epsilon = 4$. Split-ring resonator and membrane are “embedded” in air, i. e., $\epsilon = 1$. The computational domain is terminated by perfectly matched layers of 400 nm width and Silver-Müller boundary conditions mimicking an open space system by absorbing outgoing radiation. As in the previous chapter, the tetrahedral mesh is created using NETGEN and consists of 35k elements. The section of the tetrahedral mesh modeling the split-ring resonator is depicted in Fig. 7.2.

The source of radiation is a relativistic electron modeled by a scattered-field source as described in Sec. 7.1.2. Due to the NRA, the electron's trajectory is known a priori and we are free to define its propagation direction and velocity. For the remainder of the thesis, we choose $\vec{v} = v_0 \cdot \hat{e}_z$, $v_0 = 0.6953c$, where c is the vacuum speed of light, i. e., the electron propagates parallel to the z-Axis. This corresponds to a kinetic energy of 200 keV.

As in the experiment, we raster-scan the sample by modification of the electron's initial coordinates x_0 and y_0. Note, however, that a single simulation provides the loss probability associated with a single pair (x_0, y_0). Therefore, in general a large number of independent simulations must be performed to obtain a full scan. Due to the symmetry of the physical system with respect to the yz-plane, only the domain in positive x-direction must be scanned which reduces the number of calculations by a factor of 2. The loss probability in the halfspace $x < 0$ is determined by

$$P\big(-x, y; \omega\big) = P\big(x, y; \omega\big)\,. \tag{7.5}$$

The total amount of simulations depends on the area of interest and the desired resolution. In case of the single split-ring resonator, a region of 220 nm×440 nm resolved in a 10 nm grid requires approximately 1000 simulations. Employing the symmetry property Eq. (7.5), the loss probability in an area of 440 nm×440 nm is obtained. The system is simulated in third order spatial discretization for 133 fs.

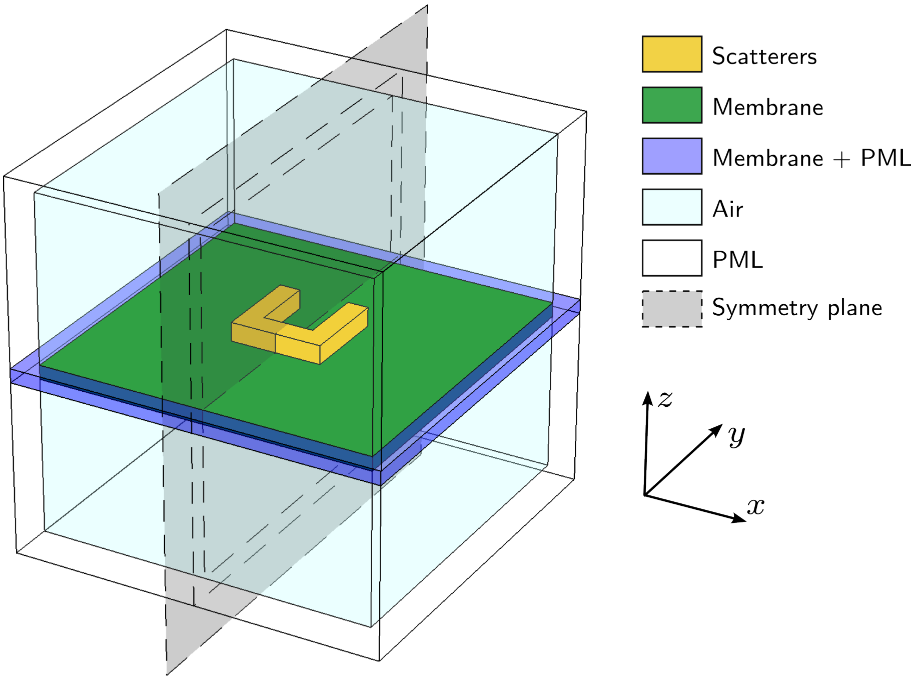


Figure 7.1: *Sketch of the EELS simulation setup. The scatterers (gold), either the single split-ring resonator or configurations of split-ring resonators (Sec. 7.3), are located on a silicon nitride membrane (green). Both, scatterers and membrane are embedded in air (cyan). The computational domain is terminated by PMLs in all directions (white). The blue region models colocated membrane and PML properties. The plane of symmetry at* $x = 0$ *(gray) is explicitly modeled for the calculation of the harmonic inversion and the mode profile only.*

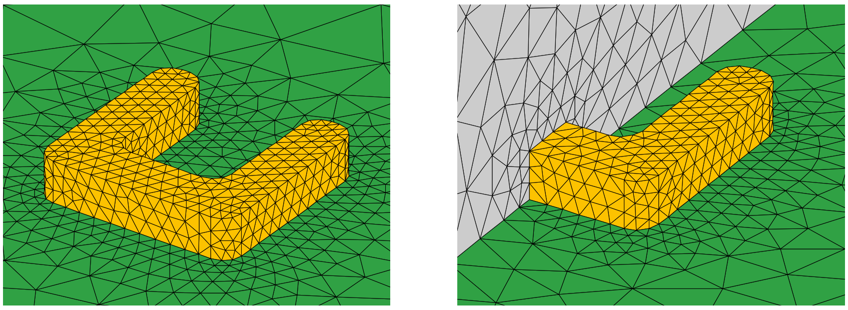


Figure 7.2: *Section of tetrahedral meshes modeling the single split-ring resonator (gold) on a silicon nitride membrane (green) excluding and including the symmetry plane (gray) in the left and right panel, respectively. The first is employed for the EELS analyses while the latter is used for the calculation of the mode profile and harmonic inversion.*

	Gold
ϵ_∞	6.21
ω_D	$13.4 \times 10^{15}\,\mathrm{Hz}$
γ_D	$1.04 \times 10^{14}\,\mathrm{Hz}$
$\Delta\epsilon_L$	1.00
ω_L	$4.02 \times 10^{15}\,\mathrm{Hz}$
γ_L	$5.81 \times 10^{14}\,\mathrm{Hz}$

Table 7.1: *Optimized Drude-Lorentz parameters for gold in the range* $500 - 2000\,$*nm. The associated permittivity is depicted in Fig. 7.3.*

Due to the large number of simulations, we employ the GPU implementation of the DGTD method described in Chap. 5. Consequently, the calculation is performed in single precision and we must verify that the employed floating-point precision yields sufficiently accurate results. Therefore, we recalculate several simulations using a regular CPU implementation with double precision. A comparison of both floating-point precision calculations demonstrates that the analysis presented in the following sections of this thesis suffers neither qualitative nor quantitative differences.

In order to calculate the loss probability according to Eq. (7.3), it is sufficient to record the z-component of the induced field $E_z^{\mathrm{ind}}\big(\vec{\mathrm{r}}_{\mathrm{e}}(t)\,,t\big)$ as the electron propagates parallel to the z-axis. In a postprocessing step, we obtain $E_z^{\mathrm{ind}}\big(\vec{\mathrm{r}}_{\mathrm{e}}(t)\,,\omega\big)$ by a fast Fourier transform of the time-domain field.

An important detail is the material model which is used to describe the dispersive properties of gold. Here, a Drude-Lorentz model is employed whose free parameters were optimized to the experimental data of Johnson and Christy [38]. The parameters optimized in the range of $500 - 2000\,\mathrm{nm}$ by means of the procedure described in Sec. 2.2.4 are summarized in Tab. 7.1. The resulting dispersion relation along with the experimental data is depicted in Fig. 7.3.

7.2.2 Validation of Applied Analysis and Simulation Setup

The principle setup of the EELS simulation was already validated in Sec. 5.3.3, when we compared analytically and numerically determined electron energy-loss spectra. However, we must ensure that the slightly modified setup including the split-ring resonator works correctly.

Thus, before continuing with the electron energy loss spectroscopy analysis, we take a closer look at the recorded time-domain data in order to validate the simulation setup and the excitation of plasmons by the electron source. Fig. 7.4 depicts the modulus of the electric field in the time-domain. In this setup, the electron penetrates the right arm of the SRR as indicated by the inset of Fig. 7.4. The SRR and the membrane span $z_{\mathrm{SRR}} \in \big[0, 0.04\big]$ and $z_{\mathrm{mem}} \in \big[-0.030, 0\big]$, respectively.

Strong field oscillations are observed once the electron is in proximity of the SRR. These correspond to plasmonic oscillations of the SRR excited by the passing electron. Surface plasmons are identified on both sides of the SRR. The field oscillations decay on several different characteristic time-scales associated with the life-time of the plasmonic oscillations.

In order to calculate the loss probability Eq. (7.3), we perform a Fourier transform of the time-dependent electric field to obtain the spectrum along the trajectory $E_z^{\mathrm{ind}}\big(\vec{\mathrm{r}}_{\mathrm{e}}(t)\,,\omega\big)$. This corresponds to an integration along a horizontal line in Fig. 7.4 whereby we are bound to the limited simulated

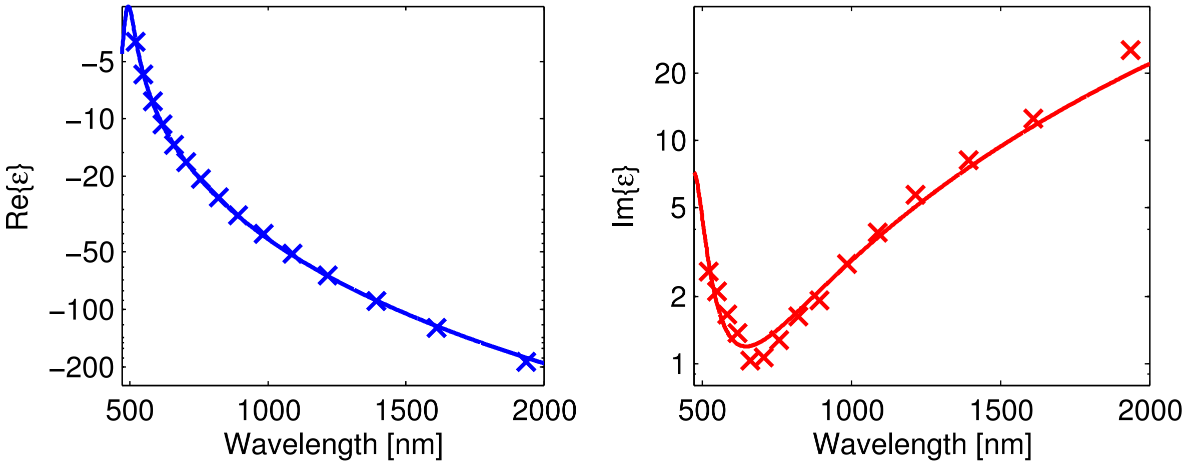


Figure 7.3: *Permittivity of gold in the range of* $500-2000\,nm$*. The plots show the real (left panel) and imaginary (right panel) part of the permittivity* ϵ *for gold based on the optimized parameters (solid lines) in Tab. 7.1. Crosses indicate experimental data taken from Ref. [38]. Note the logarithmic scale due to the large change in permittivity.*

time. In order to guarantee a sufficiently accurate approximation of the Fourier integral, the dominant contributions of $E_z^{\mathrm{ind}}\left(\vec{r}_{\mathrm{e}}(t)\,,t\right)$ must be provided by the simulation. This condition is fulfilled as the field decays by several orders of magnitude during the simulated time (cf. Fig. 7.4).

Due to the limited computational domain, the integration along the trajectory in Eq. (7.3) must be truncated as well. As the modulus of the field and, hence, of the spectrum is several orders of magnitude smaller near the limits of the recorded trajectory compared to the dominant contributions near the SRR, the integration is accurately approximated.

7.2.3 Characterization by EELS Spectra and Maps

In this section, we characterize the single split-ring resonator by the loss probability and visualize the results by means of EELS spectra and EELS maps. An energy loss of only few eV is attributed to plasmonic resonances [80]. Therefore, we analyze the loss probability in the range of $0.4-2.5\,\mathrm{eV}$.

Fig. 7.5 shows EELS spectra, i. e., the loss probability as a function of the energy loss ΔE, for three electron trajectories as indicated by the inset. Several resonance energies can be identified by the Lorentzian peaks of the individual spectra in the range $0.4-1.9\,\mathrm{eV}$. Above that range, several more resonances can be anticipated but are no longer distinguishable due to their spectral overlap. The first four resonance energies deduced from the EELS spectra, indicated by the gray dashed guides in Fig. 7.5, are listed in Tab. 7.3 on page 97.

In the spectral region considered here, the features of the EELS spectrum are associated with the electromagnetic eigenmodes of the SRR [78] which are dominated by the collective oscillations of the free electrons, also referred to as plasmons. Both surface plasmons, i. e., oscillations bound to the surface of the structure, and bulk plasmons, i. e., collective oscillations in the entire volume of the structure, contribute to the spectrum. Plasmons have a limited life-time associated to the width of a resonance. In the case of the single SRR, plasmons of different life-times can be identified in Fig. 7.5.

The spatial dependence of the loss probability is trivially apparent upon comparison of the three EELS spectra. The fundamental resonance at $\Delta E = 0.69\,\mathrm{eV}$, for example, is contained only in the

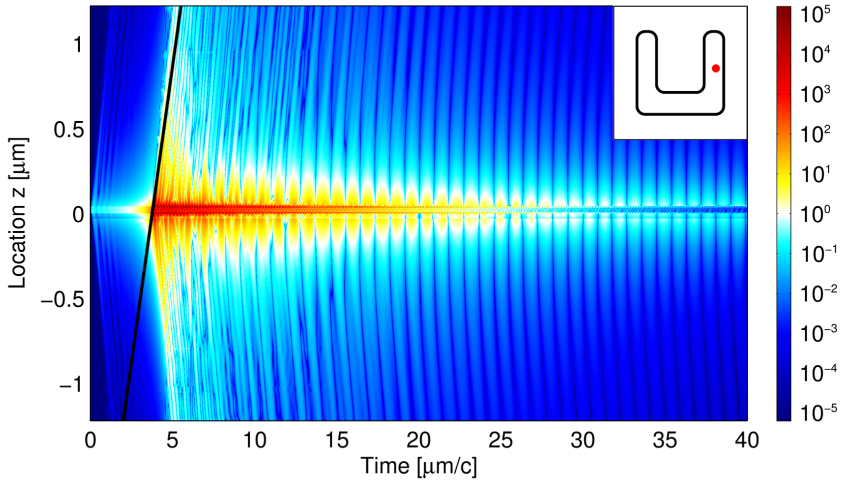


Figure 7.4: *Pseudo-color plot of* $\left|E_z^{ind}(z,t)\right|$ *in dependence of the (normalized) simulated time and the coordinate* z *for* (x,y) *as indicated by the red dot in the inset. The black line indicates the position of the electron moving at* $\vec{v} = 0.6953c \cdot \hat{e}_z$*. The inset also visualizes the trajectory of the electron in the xy-plane. The bottom plane of the SRR and the top plane of the membrane are both located at* $z = 0$.

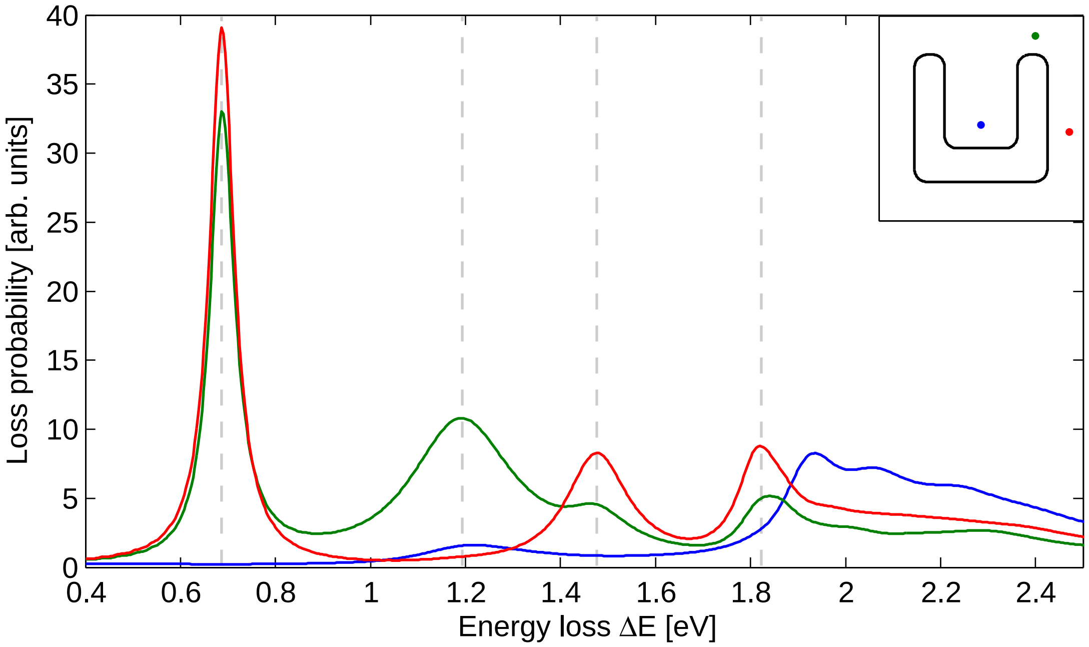


Figure 7.5: *EELS spectra of the single split-ring resonator for distinct electron trajectories centered above the base (blue), above the right arm (green) and right of the SRR (red). The inset visualizes the trajectory of the electron in the xy-plane.*

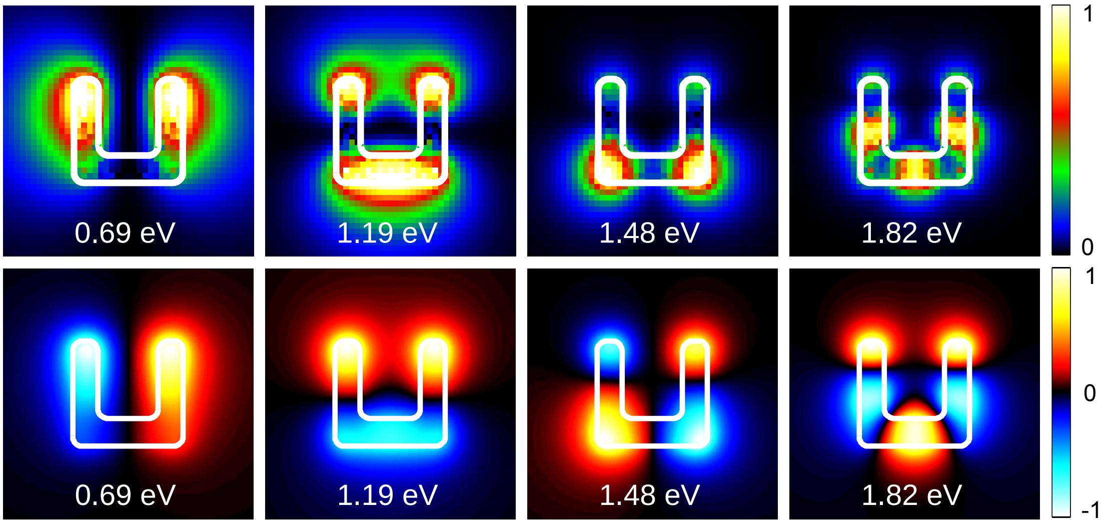


Figure 7.6: *EELS map (top row) and field distribution of the E_z-component (bottom row) for the first four resonance energies of the EELS spectrum calculated for the single split-ring resonator. The white contour represents the boundary of the SRR. The resonance energies correspond to the dashed guides in Fig. 7.5. The colorbar is individually normalized for each image, i. e., colors in different images indicate different absolute values.*

spectra taken on top (green) and right (red) of the SRR's right arm. This is clearly attributed to the modal profile of the plasmonic structure. In fact, the experimentally and numerically accessible spatial dependence is exactly what renders EELS an excellent tool for the characterization of plasmonic systems.

The spatial dependence of the loss probability for a constant energy loss, i. e., $P(x, y; \omega)$, where x, y represent the electron's initial coordinates, is visualized in EELS maps. The EELS maps for the first four resonance energies, indicated by gray dashed guides in Fig. 7.5 and listed in Tab. 7.3, are summarized in the top row of Fig. 7.6. Note that the scale of the colorbar is normalized for each image independently, i. e., identical colors indicate different absolute values in different maps throughout the remainder of the thesis.

The fundamental resonance at $\Delta E = 0.69\,\text{eV}$ shows maximum loss probability at the arms and nearly zero at the center of the base, i. e., it is most likely for the electron to lose $\Delta E = 0.69\,\text{eV}$ at the tip of the SRR's arms. Clearly, a systematic evolution of the probability profile is observed as one "hot spot" and one node is added per next higher resonance. The map of the second resonance at $\Delta E = 1.19\,\text{eV}$ comprises maximum loss probability at the tip of the arms and the base of the SRR. Maximum and moderate loss probability of the third resonance at $\Delta E = 1.48\,\text{eV}$ is observed at the lower and upper corners of the SRR, respectively. At $\Delta E = 1.82\,\text{eV}$ three hot spots of loss probability are found in the center of the arms and the base. With increasing resonance order, the spectral distance between resonances decreases and the extrema of the loss probability are located closer to each other which is in agreement with Refs. [25, 86].

The interpretation of EELS maps is a somewhat delicate issue and a matter of ongoing debate [77, 87]. In 2007, Nelayah *et al.* [78] suggested that the EELS signal maps the eigenmodes' field distribution which is in accordance with the findings of Abajo *et al.* [77, 88], who demonstrate that EELS directly renders the photonic local density of states (LDOS). Based on these results, the EELS signal was used to identify plasmonic modes and associated charge (density) distributions in split-ring resonators [25, 86]. However, recently Hohenester *et al.* [87] showed that there is no direct link between EELS and LDOS, stating that often both quantities qualitatively agree. In the following, we want to gain insight into the interpretation and verify the acquired data by employing a series of alternative characterization methods.

7.2.4 Harmonic Inversion of the Electric Field

In order to verify the resonance energies previously determined from the EELS spectra, we employ HARMINV [89, 90] to analyze the time-dependent electric field recorded at specific points. HARMINV is a free software which employs harmonic inversion to analyze a time-domain signal by assuming the signal to consist of a finite number of decaying sinusoids and determines the frequencies, decay constants, amplitudes, and phases thereof. In addition, the symmetry with respect to the yz-plane of the decaying sinusoids can be determined. Therefore, odd or even symmetry of the excited modes is explicitly enforced by employing PEC or PMC boundary conditions on the plane $x = 0$, respectively. This requires slight modifications of the geometrical and numerical setup.

Simulation Setup

The general setup of the system is not altered compared to Sec. 7.2.1, i. e., the dimensions of the SRR, the membrane, the PMLs as well as the gold material model are identical to the setup of the EELS analysis described in Sec. 7.2.1. In order to take advantage of the geometrical symmetry with respect to the yz-plane, an artificial plane, referred to as the symmetry plane, is introduced at $x = 0$ as visualized in Fig. 7.1. Note that the SRR is centered with respect to the x-axis. Either PEC or PMC boundary conditions are assigned to the symmetry plane to enforce odd or even modes. As a result, only one half of the system must be simulated as the other half must obey the applied symmetry, i. e., we set

$$\begin{aligned} \vec{\mathrm{E}}(x,y,z) &= -\vec{\mathrm{E}}(-x,y,z) \quad \text{for PEC,} \\ \vec{\mathrm{E}}(x,y,z) &= \vec{\mathrm{E}}(-x,y,z) \quad \text{for PMC.} \end{aligned} \tag{7.6}$$

Consequently, one simulation must be performed for each boundary condition. The tetrahedral mesh consists of 23k elements. The section of the mesh modeling the split-ring resonator and the plane of symmetry is depicted in Fig. 7.2. With exception of the symmetry plane at $x = 0$, the computational domain is limited by PMLs and Silver-Müller boundary conditions as before.

In contrast to the EELS simulations, the sample is not scanned, but a single electron trajectory is used to simultaneously excite as many modes as possible. Therefore, the trajectory of the source is determined from the EELS maps where a location is sought that shows high loss probability for all resonances. We choose a trajectory parallel to the z-axis passing the SRR at the tip of its right arm as indicated by the green dot in the inset of Fig. 7.5.

The time-dependent electric field is recorded at three points located 20 nm above the SRR as indicated in the inset of Fig. 7.5. In this region, a strong contribution to the EELS signal can be expected [87]. The simulation is performed in sixth order spatial discretization.

Analysis of the Electric Field by Harmonic Inversion

For all three points, each component of the electric field is analyzed resulting in nine independent sets of resonance energy $\hbar\omega$, quality factor Q and associated boundary BC. Robustness of the analysis is

#	1	2	3	4
$\hbar\omega\,[\mathrm{eV}]$	0.69	1.19	1.48	1.82
Q	11	4	9	13
BC	PEC	PMC	PEC	PMC

Table 7.2: *Resonance energy, quality factor and associated boundary condition for the first four resonances of the single split-ring resonator obtained by employing* HARMINV.

guaranteed by accepting only those sets which reproduce under variation of the input parameters given to HARMINV. In addition, a resonance must be found at least three times in all sets within a tolerance of 0.01 eV in order to be accounted.

The resonance energies, quality factors and associated boundary conditions obtained by this approach are listed in Tab. 7.2. A direct comparison of the resonance energies to the results of other characterization methods is provided in Tab. 7.3 on page 97. The agreement with the resonance energies determined from the EELS spectra is very good. We will make use of the determined mode symmetry in the following section where we compare the modes of the electric field to the EELS maps.

7.2.5 Comparing EELS Maps to the Modes of the Electric Field

We want to gain insight into the interpretation of EELS maps by comparing them to the mode profile of the field. Due to the linearity of the system, we express the time-dependent field as a sum over the modes of the system by

$$\vec{\mathrm{E}}\big(\vec{\mathrm{r}},t\big) = \sum_i \vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega_i\big)\, e^{-\mathrm{i}\omega_i t}, \tag{7.7}$$

where $\vec{\mathrm{E}}\big(\vec{\mathrm{r}},\omega_i\big)$ is the mode profile of the field for the eigenfrequency ω_i. We calculate the mode profile $E_z\big(\vec{\mathrm{r}},\omega_i\big)$ for the resonance frequencies obtained by the EELS analysis (cf. Tab. 7.3) via an on-the-fly Fourier transform [15] of the electric field in a plane 20 nm above the SRR reusing the setup of Sec. 7.2.4. In fact, we deduce both, the time-dependent electric field and the mode profile, from the same simulation.

The EELS analysis of Sec. 7.2.3 does not provide phase information of the mode [77, 86] which is required to construct the profile in the negative halfspace $x < 0$ using Eq. (7.6). Consequently, the phase relation must be obtained by another approach. Employing the symmetry of the system, the harmonic inversion in Sec. 7.2.4 delivers the boundary condition (cf. Tab. 7.2) and, hence, the symmetry associated with each mode.

The magnitude of the z-component of the electric field $|E_z\big(x,y;\omega_i\big)|$ calculated at the resonance frequencies ω_i previously obtained from the EELS simulations is depicted in the bottom row of Fig. 7.6. We observe a similar pattern in the field distribution of the modes compared to the EELS maps in the top row. Due to the symmetry of the system, we obtain alternating odd and even modes.

The first order mode, an odd mode, has one node and two extrema at the tips. The second mode, an even mode, contains two nodes at the center of the arms and three extrema located at the tip of the arms and in the base. For each next higher order one node and one extremum is added to the mode profile. Therefore, the third order mode at $\hbar\omega = 1.48\,\mathrm{eV}$ has extrema at all corners of the SRR with alternating orientation along the SRR. The fourth order mode at $\hbar\omega = 1.82\,\mathrm{eV}$ comprises five extrema and four nodes. In accordance with Refs. [25, 86], the eigenmodes are standing waves along the split-ring resonator.

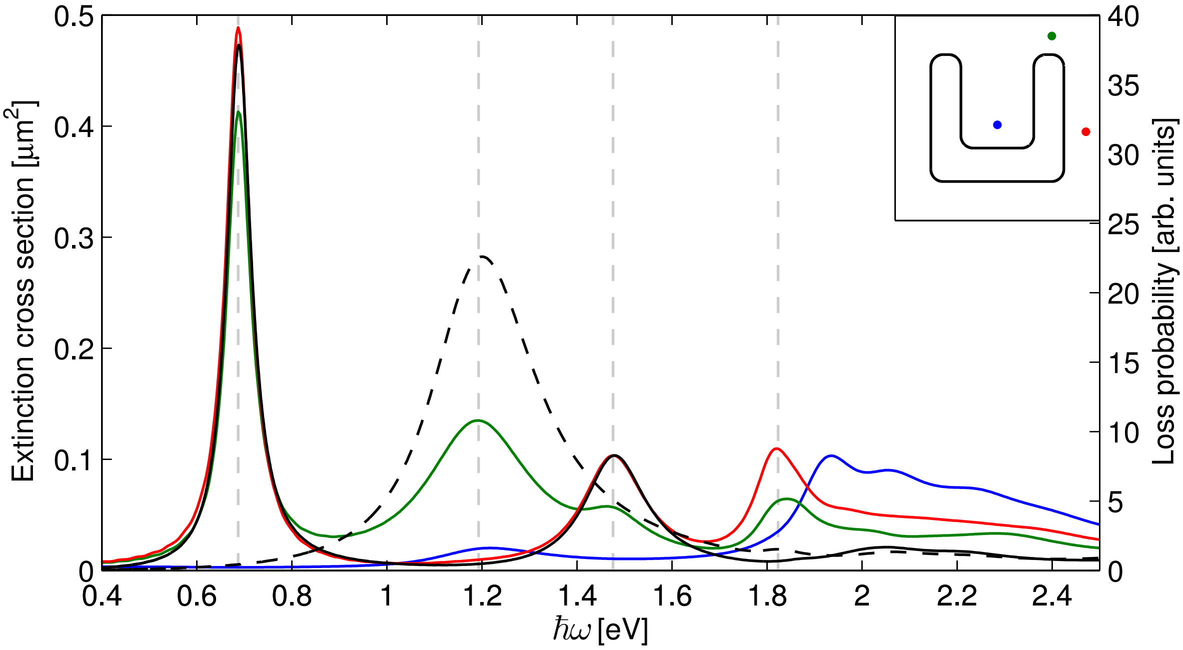


Figure 7.7: *Extinction cross-section spectra and EELS spectra for the single split-ring resonator. The extinction cross sections were obtained using horizontally (solid black) and vertically polarized (dashed black) plane waves, respectively. The EELS spectra are identical to those in Fig. 7.5 and serve for comparison.*

The EELS maps represent the (squared) magnitude of the electromagnetic eigenmodes projected along the trajectory, here the z-axis, as suggested in Refs. [78, 86]. However, minor deviations are observed at the tips of the SRR's arms for the third and fourth resonance in accordance with Ref. [86].

7.2.6 Comparing EELS and Extinction Cross-Section Spectra

The electronic system of a plasmonic structure can support optically bright and dark modes. Modes that can be excited using optical fields, e. g., plane waves, are stated optically bright. In contrast, optical dark modes cannot be (efficiently) excited using optical fields. The relativistic electron excites both optical bright and dark modes. Consequently, both of them contribute to the EELS spectrum. As a result, resonances can spectrally overlap, and the identification of their resonance energies might be hindered. Employing planes waves for excitation instead, only bright modes are excited (and observed) which allows to distinguish between optical bright and dark modes.

Whether a mode is optically bright or dark depends on the symmetry of the mode and the excitation field, and can be determined when the mode profile is available. In case of a single SRR, we expect all odd and even modes to be optically bright and excitable by horizontally and vertically polarized plane waves, respectively. Therefore, we calculate the extinction cross section for the single SRR using the setup of Chap. 6.

The resulting extinction cross-section spectra obtained for horizontal and vertical polarization are compared to the EELS spectra of Sec. 7.2.3 in Fig. 7.7. Note that, in contrast to Chap. 6, the extinction cross section is plotted as a function of energy. The resonance energies determined from the extinction cross section are listed in Tab. 7.3.

Single SRR		#	1	2	3	4
	EELS	$\Delta E\,[\mathrm{eV}]$	0.69	1.19	1.48	1.82
	HARMINV	$\hbar\omega\,[\mathrm{eV}]$	0.69	1.19	1.48	1.82
	extinction	$\hbar\omega\,[\mathrm{eV}]$	0.69	1.20	1.48	1.82

Table 7.3: *Resonance energies of the single split-ring resonator. The values for EELS were obtained from the EELS spectra depicted in Fig. 7.5. The values for* HARMINV *were calculated as described in Sec. 7.2.4. The resonance energies of the extinction cross section were determined from Fig. 7.7.*

The first and third order resonance are excited by a horizontally polarized plane wave. The second order resonance is obtained employing vertical polarization. The fourth order resonance is hard to identify due to a very small extinction cross section. All resonance energies of the extinction cross section match their counterparts in the EELS spectrum. Since all dominant resonances in the EELS spectrum were contained in the extinction cross-section spectrum as well, no dark modes are observed for the single SRR. In the following section on coupled split-ring resonator systems, we will use the extinction cross section to discriminate between optical bright and dark modes.

7.2.7 Summary

We analyzed a single gold split-ring resonator on a silicon nitride membrane using EELS and compared the results to other characterization methods. The resonance energies obtained by EELS, harmonic inversion of the electric field, and the extinction cross section calculation are summarized in Tab. 7.3. Very good agreement in the resonance energy is achieved, rendering EELS reliable and efficient.

In comparison to the field distribution of the plasmonic eigenmodes, we confirmed the findings of Refs. [25, 86] and relate the EELS signal to the (squared) magnitude of the mode profile and the charge (density) distribution. The eigenmodes are found to be standing waves along the structure.

Finally, we compared the EELS spectra of several trajectories to the extinction cross-section spectra employing horizontally and vertically polarized plane waves. All resonances were determined to be optically bright as was expected for an individual structure.

7.3 Coupled Split-Ring Resonator Systems

Based on the interference of plasmons, metallic nanostructures composed of several metaatoms may yield significant field enhancement giving rise to nonlinear interaction processes. The extinction cross section can be enhanced, too, which is applied in, e. g., surface enhanced Raman spectroscopy. Arrays of split-ring resonators are employed in the field of metamaterials, i. e., materials with artificial properties not occurring in nature which may exhibit, e. g., negative refraction.

The analysis of systems composed of several identical metaatoms was performed for dimers of gold [24] and silver spheres [82] as well as nanorods [83] and demonstrated the formation of dark plasmons. The formation of optical dark modes is well understood in the picture of the plasmon hybridization model [91, 92] as demonstrated for a dimer of spheres by Nordlander *et al.* [93].

The plasmon hybridization model is considered an electromagnetic analog of molecular orbital theory which describes the plasmon response of complex nanostructures of arbitrary shape [91]. The plasmons of the individual structures composing a system interact and form new eigenstates whose associated

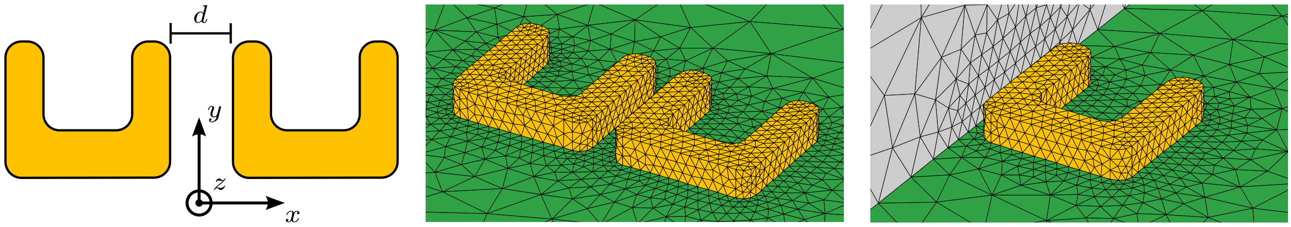


Figure 7.8: *Sketch and sections of tetrahedral meshes of the 1 × 2 configuration of split-ring resonators. The split-ring resonators are separated by $d = 35\,\mathrm{nm}$ and centered around $x = 0$ (left panel). All other parameters are equivalent to the single SRR case (cf. Sec. 7.2.1 and Fig. 6.1). The central and right panel visualize sections of the tetrahedral meshes modeling coupled SRRs (gold) on a silicon nitride membrane (green) excluding and including the symmetry plane (gray), respectively.*

eigenmodes are combinations of the eigenmodes of the metaatom obeying characteristic phase relations. Due to the coupling of the plasmons, each resonance of the metaatom is split into energetically lower and higher resonances whereby the number of split resonances is directly linked to the number of interacting plasmons. In case of a dimer of metallic spheres, two contributing plasmons lead to a splitting into one energetically lower and higher resonance for each fundamental resonance of the sphere. The split resonances are referred to as symmetric or “bonding” for in-phase related plasmons and antisymmetric or “antibonding” for antiphase related plasmons, respectively [91]. In-phase related plasmons can be optically excited, i. e., they form optical bright modes while antiphase related plasmons form optical dark modes.

7.3.1 Coupled Split-Ring Resonators in 1x2 Configuration

Simulation Setup

The first coupled system we analyze is composed of two split-ring resonators arranged side by side as illustrated in Fig. 7.8. The simulation setup of the *1 × 2* configuration of split-ring resonators is conceptually identical to the one presented for the single split-ring resonator in Sec. 7.2.1. The geometry of the individual SRRs, the membrane and the perfectly matched layers as well as the material model and boundary conditions are equal to the single SRR case.

The two SRRs are vertically aligned and horizontally separated by 35 nm as sketched in Fig. 7.8 (left panel). The tetrahedral mesh consists of approximately 47k elements (cf. Fig. 7.8, central panel). The “scanned” area is increased to 330 nm × 440 nm which corresponds to more than 1500 simulations to obtain a 10 nm resolution. Note that similar to the case of the single SRR, only the positive halfspace $x > 0$ is scanned and the symmetry of the geometry is employed to construct the EELS signal for $x < 0$ which yields a total determined area of 660 nm × 440 nm.

Characterization by EELS Spectra and Maps

EELS spectra for selected electron trajectories are depicted in Fig. 7.9. As alluded to in Sec. 7.2.3 several trajectories must be considered in order to obtain all resonances. For each of the first three fundamental resonances of the single SRR, we observe a splitting into an energetically lower and higher resonance for the coupled system. These correspond to the new eigenmodes of the coupled system which are created from the fundamental mode with in-phase and antiphase relation. The associated

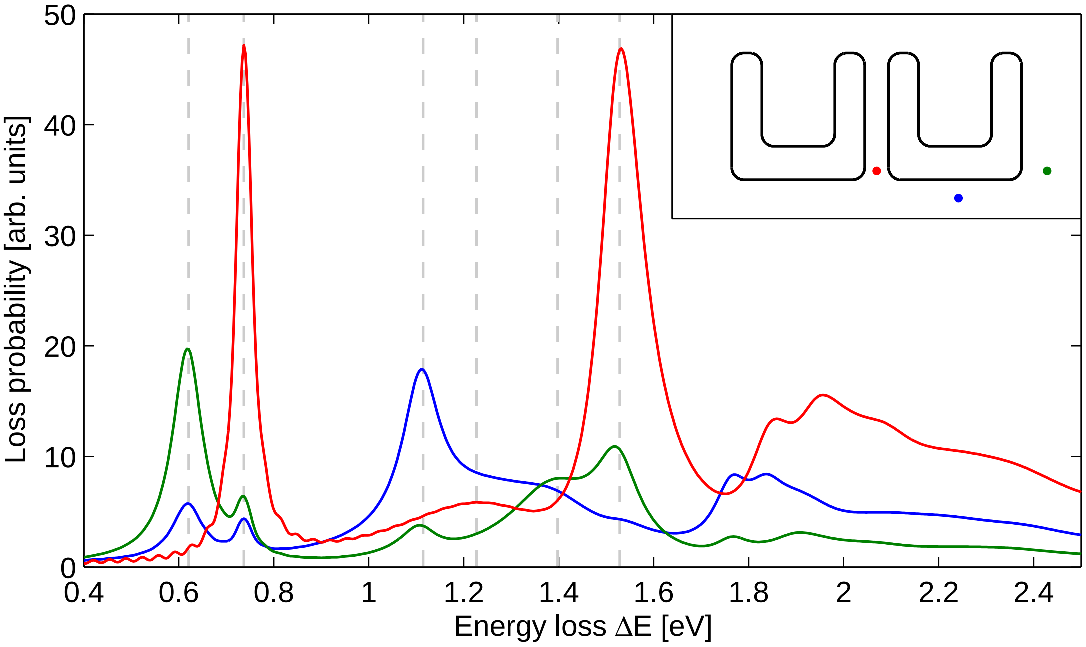


Figure 7.9: *EELS spectra of the 1×2 configuration of SRRs for distinct electron trajectories centered below the base (blue) and right of the right SRR (green) and in between the SRRs (red). The inset visualizes the trajectory of the electron in the xy-plane.*

resonance energies, indicated by the gray dashed guides in Fig. 7.9, are summarized and compared to the values of the single SRR case in Tab. 7.5 on page 105. The energy shift from the fundamental to the lower and higher resonances is asymmetric. Note that the fourth and fifth resonances at $\Delta E = 1.23\,\mathrm{eV}$ and $\Delta E = 1.40\,\mathrm{eV}$, respectively, are difficult to obtain from the EELS spectra as their individual loss probability is spectrally overlapped by other resonances.

EELS maps at the six resonance energies are depicted in the left column of Fig. 7.10. Maximum loss probability for the first resonance of the 1×2 configuration at $\Delta E = 0.62\,\mathrm{eV}$ is observed at the outer arms while the inner arms provide quasi zero loss probability. The second resonance at $\Delta E = 0.74\,\mathrm{eV}$ forms a counter part with maximum and little loss probability at the inner and outer arms, respectively. The maps of the third and fourth resonance show a significant similarity to the second resonance of the single SRR (cf. Fig. 7.6) with pronounced features at the tips and bases of the SRRs. For the fifth resonance the highest loss probability is found at the lower outer corners while the remaining corners show moderate values. High loss probability is observed at all lower corners of the SRRs for the sixth resonance.

In analogy to the single SRR case, harmonic inversion, a comparison to the field distribution of the modes, and an extinction cross section calculation are performed to confirm and compare the results of the EELS analysis.

Harmonic Inversion of the Electric Field

In analogy to Sec. 7.2.4, the time-dependent electric field is analyzed to cross check the resonance energies determined by the EELS analysis using a setup conceptually equivalent to the single SRR case.

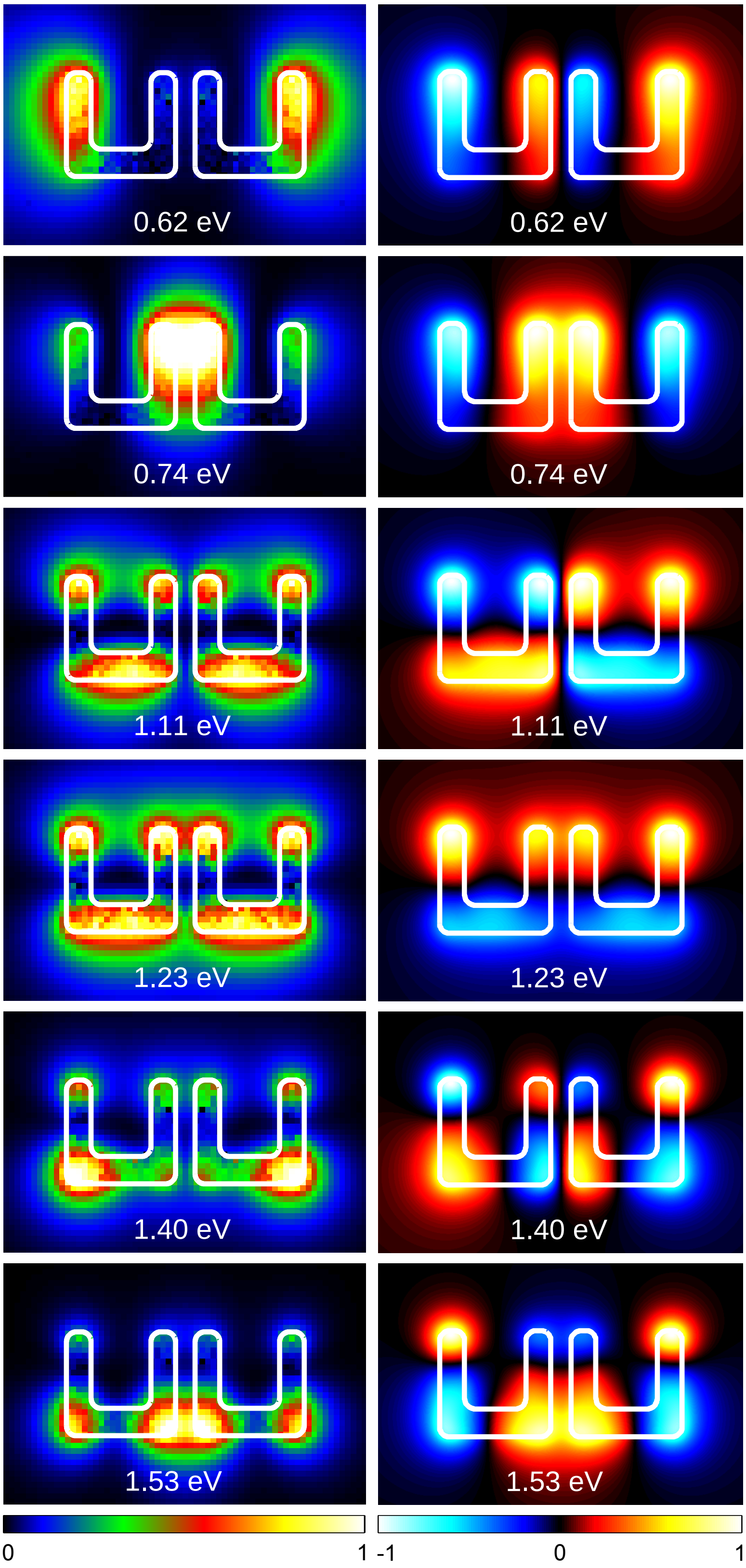


Figure 7.10: *EELS map (left column) and field distribution of the E_z-component (right column) of the 1×2 configuration of SRRs at resonances indicated in Fig. 7.9.*

#	1	2	3	4
$\hbar\omega\,[\mathrm{eV}]$	0.62	0.74	1.11	1.53
Q	8	19	12	14
BC	PEC	PMC	PEC	PMC

Table 7.4: *Resonance energy, quality factor and associated boundary condition for resonances of the 1×2 configuration of split-ring resonators obtained by employing* HARMINV.

Nevertheless, the mesh must be adapted to account for the two SRRs and consists of 27k tetrahedral elements (cf. Fig. 7.8, right panel). As before, the electron propagates parallel to the z-axis and passes the SRRs above the tip of the right[1] SRR's right arm.

We record the field at points 20 nm above the SRRs indicated in the inset of Fig. 7.9. The results of the analysis are listed in Tab. 7.4 and a compilation of the resonance energies obtained by other characterization methods is given in Tab. 7.5 on page 105.

In contrast to the EELS analysis, only four of the first six resonances can be obtained (cf. Tab. 7.5). In particular, the resonances at $\Delta E = 1.23\,\mathrm{eV}$ and $\Delta E = 1.40\,\mathrm{eV}$ are not found in a robust way. These resonances are spectrally broad and their contribution to the EELS spectrum is small (cf. Fig. 7.9). Hence, their contribution to the time-domain signal is small, too, which hinders the determination of the resonances.

The four resonances obtained by HARMINV accurately reproduce the values of the EELS analysis. In analogy to the single SRR case, the associated mode symmetries are employed to construct the mode profiles in the following section.

Comparing EELS Maps to the Modes of the Electric Field

We calculate the mode profile for the *1 × 2* configuration of split-ring resonators in analogy to Sec. 7.2.5, i. e., the setup of the harmonic inversion of the *1 × 2* configuration is reused. The resulting mode profiles calculated for the six resonance energies obtained from the EELS analysis (cf. Tab. 7.5) are depicted in the right column of Fig. 7.10. Here, the associated boundary conditions determined by the harmonic inversion in the previous section were employed to construct the mode profiles in the halfspace $x < 0$ using Eq. (7.6). The symmetry of those modes lacking an associated symmetry is consistently attributed from their "sibling", i. e., the other resonance forming the pair of a split fundamental resonance.

The coupled mode of the first resonance at $\hbar\omega = 0.62\,\mathrm{eV}$ is composed of its "parent" mode profile, i. e., the first resonance of the single SRR at $\hbar\omega = 0.69\,\mathrm{eV}$ (cf. Fig. 7.6). The field on both SRRs is in phase. This is not surprising as the modes of the composed system are constructed from the corresponding, here first, mode of the metaatom with in-phase and antiphase relation, as alluded to in Sec. 7.3. Consequently, the second coupled mode at $\hbar\omega = 0.74\,\mathrm{eV}$ is constructed by the single SRR first order mode in antiphase.

Only in-phase related modes can be excited by optical fields and are, hence, considered optical bright modes. In contrast, optical dark modes cannot be excited by optical fields. Therefore, the modes at $\hbar\omega = 0.62\,\mathrm{eV}$ and $\hbar\omega = 0.74\,\mathrm{eV}$ are optically bright and dark, respectively.

The third and fourth coupled modes are constructed from the second mode of the single SRR with antiphase and in-phase relations, rendering them optically dark and bright, respectively. The behavior

[1] For clarity's sake, the resonator is referred to by its position in the configuration although it is in fact the *only one* in the setup due to the explicit application of the system symmetry (cf. Sec. 7.2.4).

of the fifth and sixth coupled mode is analogous to the first and second one, i. e., they are constructed from the third order mode of the single SRR with in-phase and antiphase relation resulting in optical bright and dark modes, respectively.

In summary, the modes of the *1 × 2* configuration are standing waves along the path of each split-ring resonator coupled by in-phase or antiphase relations. These observations are a consistent extension of the results in Ref. [86].

Comparing EELS maps and mode profiles yields some interesting features. The EELS map of the first resonance at $\Delta E = 0.62\,\text{eV}$ reveals quasi zero loss probability at the inner arms while the corresponding mode profile clearly shows in-phase related first order mode profiles of the single SRR. These "blind spots" have already been reported in Ref. [87]. The second to sixth resonances remain to represent the (squared) electric field along the trajectory with minor deviations at the tips of the sixth resonance as described in Ref. [86]. Obviously, EELS maps do not represent unique characteristics of individual modes as the maps of the third and fourth can barely be distinguished.

Comparing EELS and Extinction Cross-Section Spectra

At last, we perform an optical characterization method, i. e., a method that excites the system by an optical field and probes the system using the far-field. With these constraints only optically bright modes are excited and measured. Hence, the spectrum merely contains optically bright resonances which allows to cross-check previously obtained resonance energies and considerations of optical darkness and brightness of the modes. The characterization quantity of choice is the extinction cross section already employed and described in Chap. 6.

Fig. 7.11 shows the extinction cross-section spectra for both horizontal and vertical polarization together with the EELS spectrum. The resonance energies determined from the extinction cross section are listed in Tab. 7.5 on page 105. The first maximum of the extinction cross section coincides well with the resonance energy of the first EELS resonance of the coupled system at $\hbar\omega = 0.62\,\text{eV}$. This confirms the declaration of this resonance as optically bright. Furthermore, it demonstrates that the loss probability is not "polluted" by other modes as the resonance energy is met precisely. The second resonance of the EELS spectrum is confirmed to be a dark mode as a resonance cannot be observed for both excitation polarizations. The same result is obtained for the fifth and sixth EELS resonances which are confirmed by an extinction cross section resonance at $\hbar\omega = 1.40\,\text{eV}$ to be optically bright and dark, respectively.

The extinction cross section resonance of the vertical polarization at $\hbar\omega = 1.27\,\text{eV}$ does not match the fourth EELS resonance energy exactly. However, we still can confirm the third and fourth EELS resonances as optically dark and bright, respectively. Here, the spectral overlap of several EELS resonances leads to a shift of the "measured" maximum of the loss probability with respect to the true position of the resonance. This effect clearly demonstrates advantages and disadvantages of the simultaneous excitation (and measurement) of dark and bright modes which gives access to a lot of information on the one hand but may be difficult to interpret and deduce exact quantities from on the other hand.

Separation Dependence Analysis by Electron Energy Loss Spectroscopy

In this section, we study the dependence of the resonance energies with respect to the separation distance d (cf. Fig. 7.8) of the SRRs by EELS. Similar investigations were previously performed by means of optical characterization, in particular, the extinction cross section, in Refs. [14, 23]. In accordance with

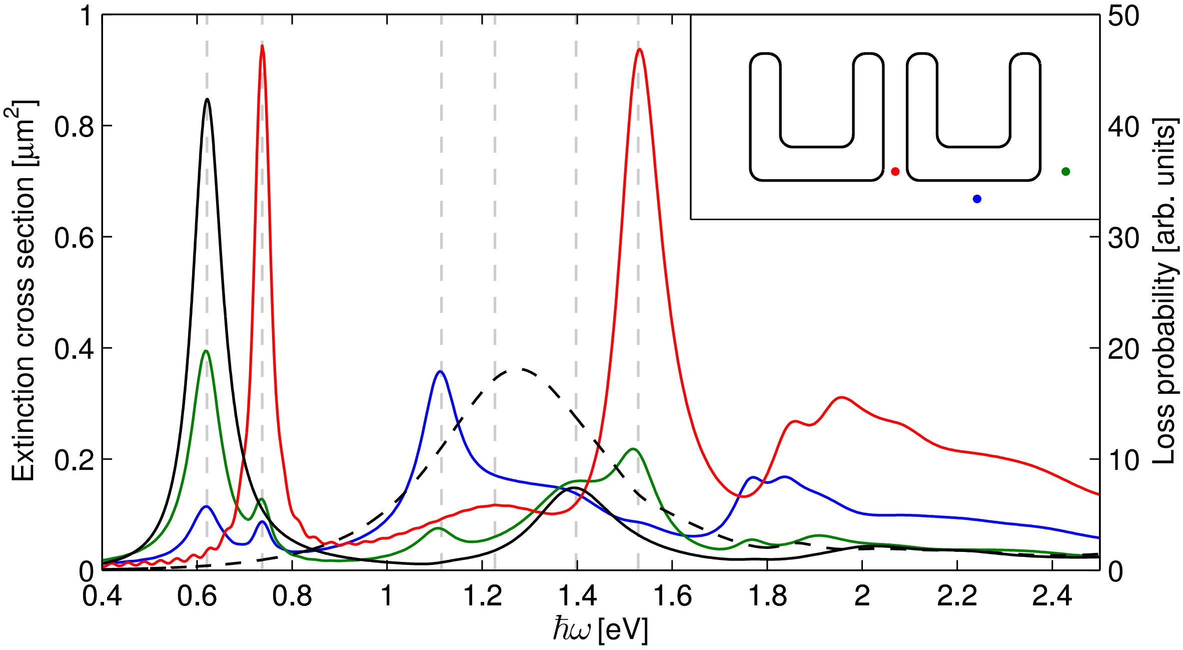


Figure 7.11: *Extinction cross-section spectra and EELS spectra for the 1 × 2 configuration of SRRs. The extinction cross sections were obtained using horizontally (solid black) and vertically polarized (dashed black) plane waves, respectively. The EELS spectra are identical to those in Fig. 7.9 and serve for comparison.*

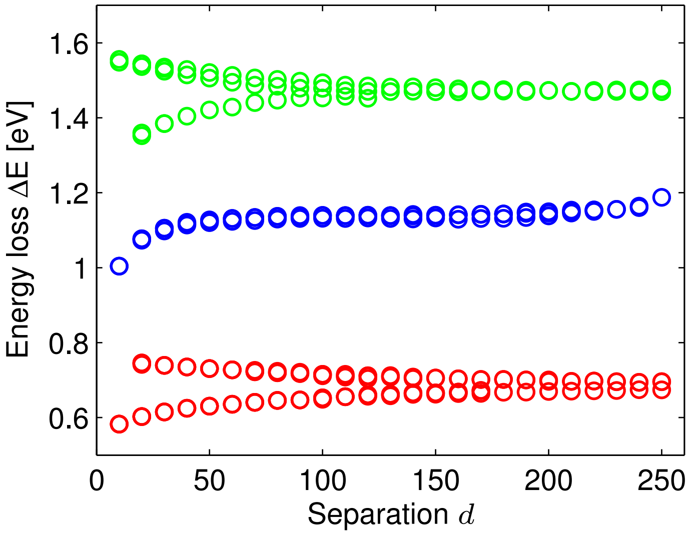


Figure 7.12: *Energy loss ΔE at the resonances of the 1×2 configuration in dependence of the separation distance d. The colors emphasize the resonances' parent resonance of the single split-ring resonator, whereby red, blue, and green reference the first, second, and third resonance, respectively.*

their findings, we expect a strong resonance shift in dependence of the separation distance d, i. e., the smaller the separation the stronger the interaction and the larger the splitting of the resonance energies. For large separation, the resonance energies should approach those of the single, individual SRR. In contrast to Refs. [14, 23], we expect to probe both contributions to the resonance splitting as bright and dark modes are accessible in EELS. However, in their terminology, we investigate the side-by-side configuration only.

The simulation setup is identical to the one presented at the beginning of Sec. 7.3.1. In order to model the varying separation, a tetrahedral mesh must be created for each distance. We model separation distances from 10 nm to 250 nm in steps of 10 nm whereby the separation is defined as sketched in Fig. 7.8. For each distance an EELS analysis for several trajectories is performed and the maxima of the EELS spectra are extracted in the vicinity of the expected resonance positions according to Tab. 7.5 on page 105.

The energy loss at the resonance positions in dependence of the separation distance d is depicted in Fig. 7.12. The plot includes resonances found in all trajectories considered, i. e., several energy loss values at a single separation distance emerge from the resonances found for all trajectories. We observe energetically lower and higher branches of resonances for the children of the single SRR's first and third resonances. The reduction of the separation distance leads to super-linear growth of the energy splitting. For large separation distances, the energy splitting shrinks and the resonance energies of the single SRR are approached (cf. Tab. 7.5).

The blue curve, indicating the resonances emerging from the second resonance of the single SRR, includes the low energy branch of the splitting only, although a high energy branch is expected, too. As before, the resonance around 1.23 eV cannot be robustly determined due to the strong spectral overlap of neighboring resonances. However, the trend of the low energy split branch is still in accordance with previous observations.

Single SRR		#	1		2		3	
	EELS	$\Delta E\,[\mathrm{eV}]$	0.69		1.19		1.48	
Configuration 1×2		#	1	2	3	4	5	6
	EELS	$\Delta E\,[\mathrm{eV}]$	0.62	0.74	1.11	1.23	1.40	1.53
	HARMINV	$\hbar\omega\,[\mathrm{eV}]$	0.62	0.74	1.11	-	-	1.53
	extinction	$\hbar\omega\,[\mathrm{eV}]$	0.62	-	-	1.28	1.40	-

Table 7.5: *Resonance energies of the 1 × 2 configuration of split-ring resonators. The values for EELS were obtained from the EELS spectra depicted in Fig. 7.9. The resonance energies of the extinction cross section were determined from Fig. 7.11. "-" indicates indeterminable resonances. Resonance energies of the single SRR serve for comparison.*

Summary

We have analyzed the *1 × 2* configuration of split-ring resonators by use of EELS, harmonic inversion of the electric field, calculation of the mode profile and the extinction cross section for horizontal and vertical polarization. The resonance energies of the first six resonances are summarized in Tab. 7.5. The splitting of the resonances of the metaatom were observed as expected by the plasmon hybridization model, i. e., for each resonance of the metaatom an energetically lower and higher resonance is determined for the coupled system. The plasmons of the coupled SRR system are composed of in-phase and antiphase related combinations of the single SRR modes. Optical bright and dark plasmons were consistently identified by the results of the harmonic inversion and the extinction cross section spectra.

"Blind spots" (cf. Ref. [87]) in the EELS maps were observed for the first order resonance at $\Delta E = 0.62\,\mathrm{eV}$. The spectral overlap of resonances, promoted by the simultaneous excitation and measurement of optical bright and dark modes, can lead to difficulties in the extraction of exact values from the EELS spectra as the comparison to the extinction cross-section spectra reveals. Furthermore, due to the missing phase information, EELS maps cannot be uniquely attributed to a single resonance (cf. resonances at $\Delta E = 1.11\,\mathrm{eV}$ and $\Delta E = 1.23\,\mathrm{eV}$).

7.3.2 Coupled Split-Ring Resonators in 2x1 Configuration

Simulation Setup

The simulation setup of the *2 × 1* configuration of split-ring resonators is conceptually identical to the one presented for the single split-ring resonator in Sec. 7.2.1 and the *1 × 2* configuration. The geometry of the individual SRRs, the membrane and the perfectly matched layers as well as the material model and boundary conditions are equal to the single SRR case.

The two SRRs are horizontally aligned and vertically separated by 35 nm as sketched in Fig. 7.13 (left panel). The tetrahedral mesh consists of approximately 46k elements (cf. Fig. 7.13, central panel), and the scanned area is increased to 220 nm × 620 nm which corresponds to more than 1300 simulations to obtain a 10 nm resolution. As before, only the halfspace in positive x-direction $x > 0$ is scanned. Employing the symmetry of the physical system, the loss probability in the region $x < 0$ is determined using Eq. (7.5).

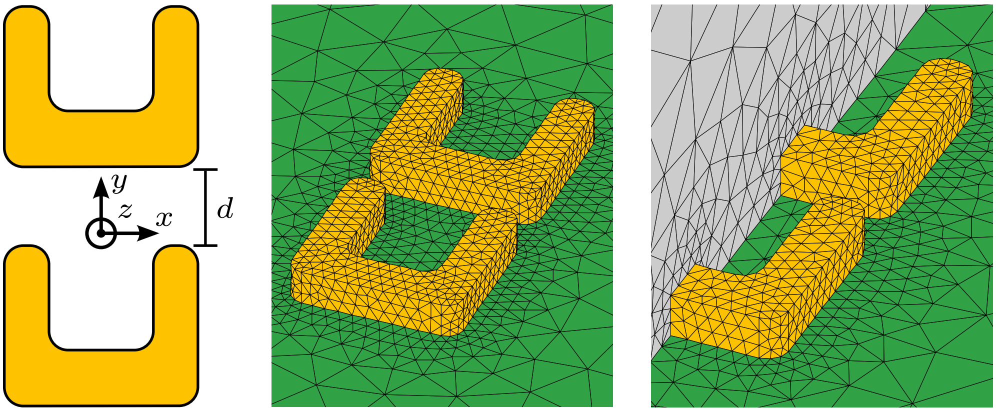


Figure 7.13: *Sketch and sections of tetrahedral meshes of the 2 × 1 configuration of split-ring resonators. The split-ring resonators are separated by $d = 35$ nm and centered around $x = 0$ (left panel). All other parameters are equivalent to the single SRR case (cf. Sec. 7.2.1 and Fig. 6.1). The central and right panel visualize sections of the tetrahedral meshes modeling the coupled system of SRRs (gold) on a silicon nitride membrane (green) excluding and including the symmetry plane (gray), respectively.*

Characterization by EELS Spectra and Maps

The EELS spectra for the *2 × 1* configuration of split-ring resonators for distinct electron trajectories are depicted in Fig. 7.14. Similar to the previously analyzed *1 × 2* configuration, the splitting of the first three fundamental resonances of the single SRR is clearly observed. Resonances at higher energies can be anticipated but are in general strongly overlapping with other resonances. The resonance energies of the first six resonances of the *2 × 1* configuration are summarized in Tab. 7.7 on page 110 together with previously determined resonances for comparison. Again, each resonance of the metaatom is split into an energetically lower and higher resonance.

EELS maps of the *2 × 1* configuration at the resonances listed in Tab. 7.7 are depicted in Fig. 7.15. The first two resonances at $\Delta E = 0.65\,\text{eV}$ and $\Delta E = 0.70\,\text{eV}$ show strong loss probability at the arms of the SRRs with maximum values at the lower and upper SRR, respectively. The loss probability drops to zero between the SRRs for the first resonance while a non-zero loss probability "connects" the two metaatoms in case of the second resonance. The strongest features are observed at the tip of the arms of the upper and at the base of the lower SRR for the resonance at $\Delta E = 1.03\,\text{eV}$. The situation is inverted for the fourth resonance at $\Delta E = 1.31\,\text{eV}$, which shows high loss probability at the tip of the arms of the lower SRR, at the base of the upper SRR, and in between them. Here, moderate loss probability at the hot spots of the third resonance is observed due to spectral overlap of both resonances mediated by the broad third resonance (cf. blue spectrum in Fig. 7.14). The resonances at $\Delta E = 1.42\,\text{eV}$ and $\Delta E = 1.53\,\text{eV}$ show strong loss probability at the SRRs' lower corners and moderate loss probability at the tips of both SRRs. Clear characteristic differences are difficult to state which is, again, due to the spectral overlap of the contributing resonances.

Harmonic Inversion of the Electric Field

The time-dependent electric field is analyzed in analogy to Sec. 7.2.4 to confirm the resonance energies of the EELS analysis. Thus, the mesh is adapted to account for the explicit application of the system's

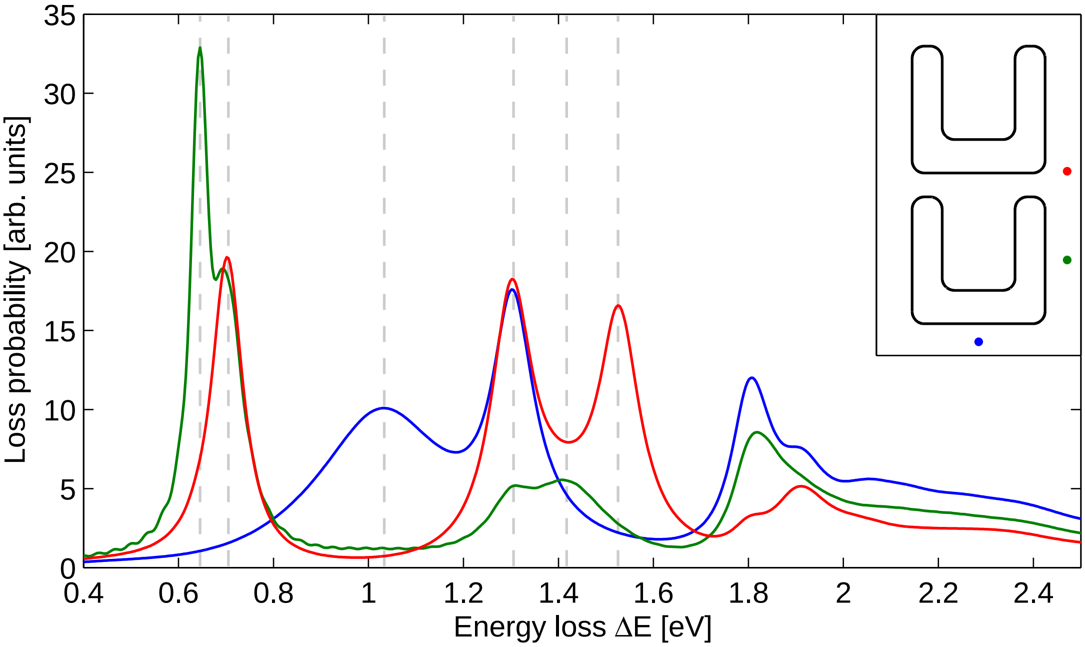


Figure 7.14: *EELS spectra of the 2×1 configuration of SRRs for distinct electron trajectories centered below the base (blue) and right of the lower SRR (green) and right of the upper SRR's base (red). The inset visualizes the trajectory of the electron in the xy-plane.*

#	1	2	3	4	5
$\hbar\omega$ [eV]	0.65	0.71	1.03	1.30	1.53
Q	16	9	3	13	14
BC	PEC	PEC	PMC	PMC	PEC

Table 7.6: *Resonance energy, quality factor and associated boundary condition for resonances of the 2×1 configuration of split-ring resonators obtained by employing* HARMINV.

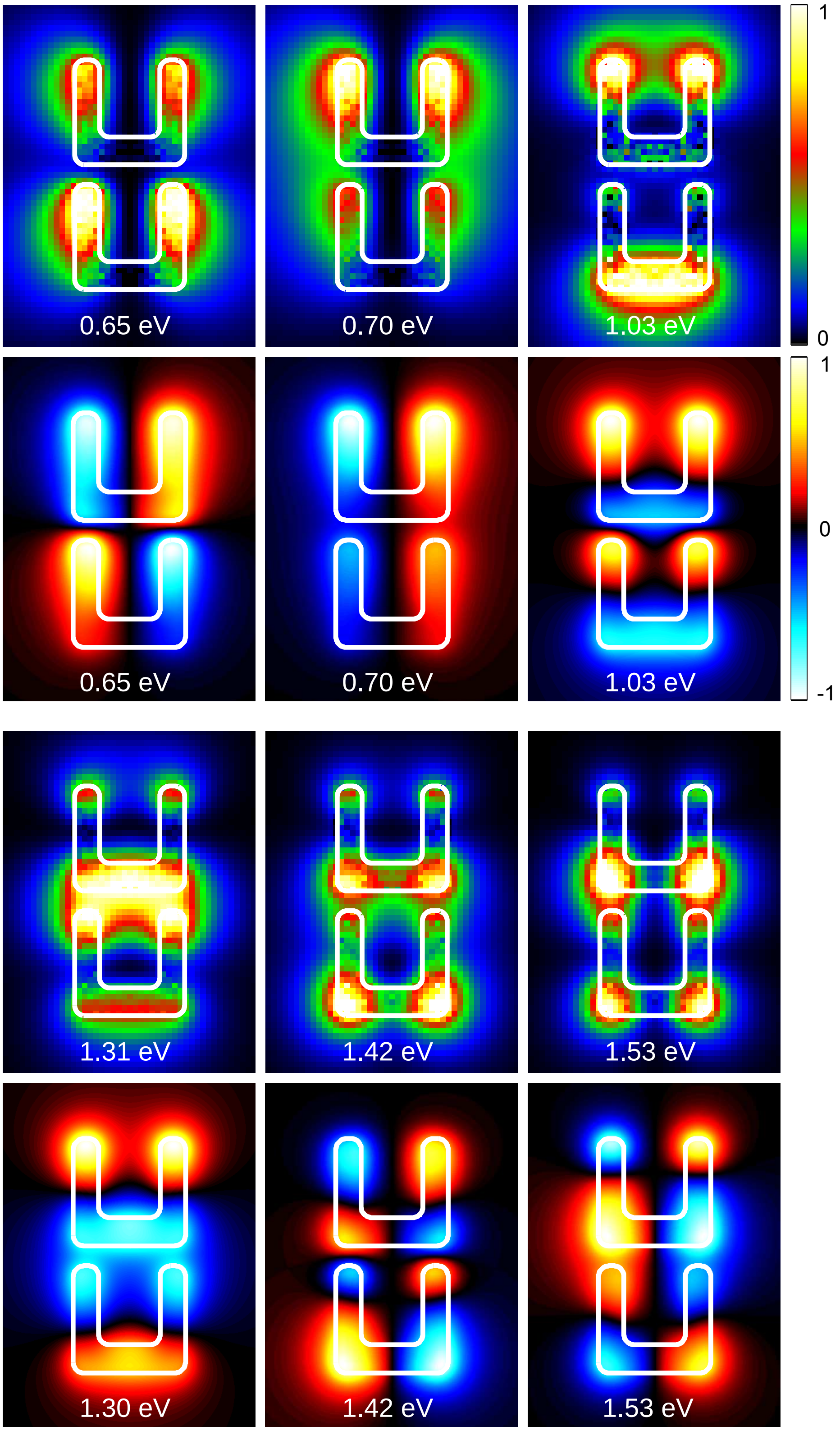


Figure 7.15: *EELS map (first and third row) and field distribution of the E_z-component (second and fourth row) of the 2×1 split-ring resonator configuration at resonance energies indicated by the dashed guides in Fig. 7.14.*

symmetry and consists of 26k elements (cf. Fig. 7.13, right panel). The electron passes the *2 × 1* configuration near the tip of the right arm of the upper SRR. The field is recorded at three points located 20 nm above the SRRs as indicated in the inset of Fig. 7.14. Resulting resonance energies, quality factors and associated boundary conditions are listed in Tab. 7.6 and compared to the data of the EELS analysis in Tab. 7.7 on page 110.

As in the case of the *1 × 2* configuration of SRRs, HARMINV struggles to determine all resonances and their energies in the spectral range of interest. Consequently, the resonance $\Delta E = 1.42\,\text{eV}$ cannot be confirmed while the remaining five resonance energies match the values of the EELS analysis. As usual, the associated boundary conditions are used to construct the mode profile in the following section.

Comparing EELS Maps to the Modes of the Electric Field

The calculation of the mode profile is performed in analogy to Sec. 7.2.5 employing the symmetries of the resonances obtained in the previous section. The missing symmetry for the resonance at $\hbar\omega = 1.42\,\text{eV}$ is attributed as consistency requires. The mode profiles for the six resonances listed in Tab. 7.7 are depicted in Fig. 7.15.

We find the first two coupled modes at $\hbar\omega = 0.65\,\text{eV}$ and $\hbar\omega = 0.70\,\text{eV}$ to consist of the single SRR's first order mode in antiphase and in-phase relation, rendering them optically dark and bright, respectively. The modes at $\hbar\omega = 1.03\,\text{eV}$ and $\hbar\omega = 1.31\,\text{eV}$ are composed of the second order mode of the single SRR and determined to be optically bright and dark which is equivalently true for the fifth and sixth mode of the coupled system. In analogy to the *1 × 2* configuration, the modes of the *2 × 1* configuration are standing waves along the path of each split-ring resonator coupled by in-phase or antiphase relation.

The comparison of EELS maps and mode profiles yields qualitatively consistent results compared to the *1 × 2* configuration. The maps of all except the third resonance seem to represent the magnitude of the (squared) electric field projected along the electron's trajectory. Similar to the first resonance of the *1 × 2* configuration, the EELS map of the third resonance of the *2 × 1* configuration differs considerably from its mode profile due to the "blind spot" at the center of the structure. Due to the absent phase information, the EELS maps of the fifth and sixth resonance cannot be distinguished which emphasizes the fact that EELS maps do not uniquely represent a resonance.

Comparing EELS and Extinction Cross-Section Spectra

We continue the analysis with the calculation of the extinction cross section using both horizontal and vertical polarization. The resulting spectra are depicted together with the EELS spectra in Fig. 7.16. The resonance energies determined from the extinction cross section are listed in Tab. 7.7 on page 110.

The first extinction cross section maximum coincides with the higher resonance of the split first order single SRR resonance, i. e., the second EELS resonance at $\hbar\omega = 0.70\,\text{eV}$. Further extinction cross section maxima at $\hbar\omega = 1.05\,\text{eV}$ and $\hbar\omega = 1.43\,\text{eV}$ are found slightly shifted compared to the EELS spectra. Again, we attribute the shift to the spectral overlap of the resonances in the EELS spectrum. Finally, the assignment of optical bright and dark resonances previously stated are confirmed by the extinction cross-section spectra.

Summary

We characterized the *2 × 1* configuration of split-ring resonators employing EELS, harmonic inversion of the electric field, the mode profiles and the extinction cross-section spectra for horizontal and vertical

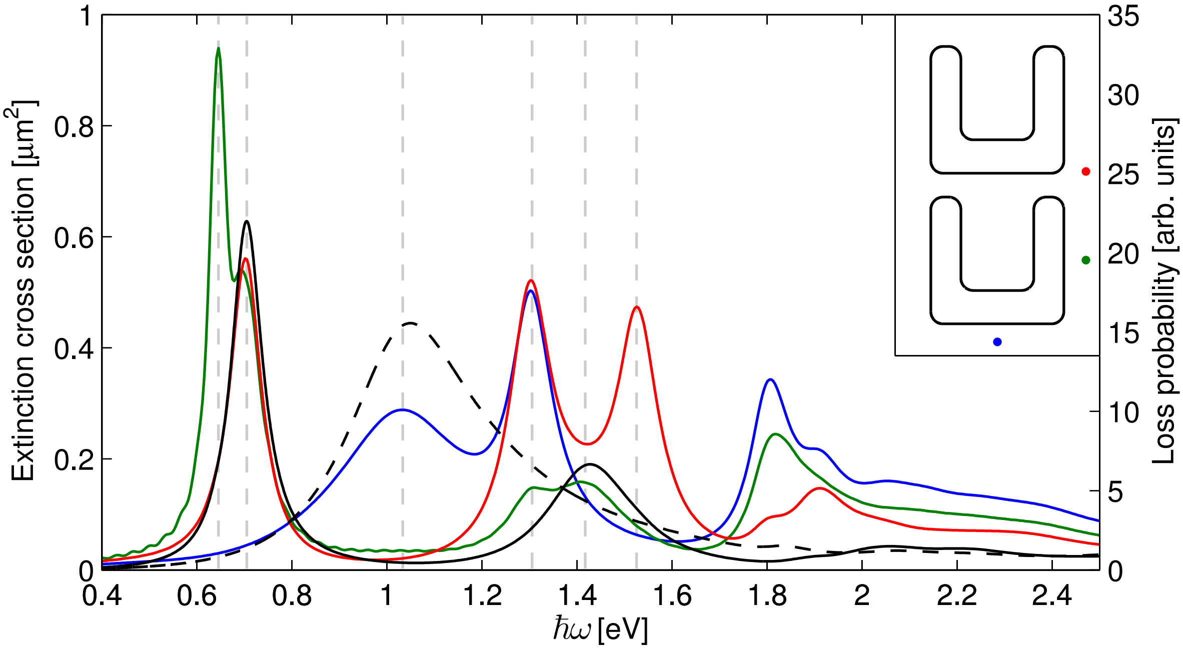


Figure 7.16: *Extinction cross-section spectra and EELS spectra for the 2×1 configuration of SRRs. The extinction cross sections were obtained using horizontally (solid black) and vertically polarized (dashed black) plane waves, respectively. The EELS spectra are identical to those in Fig. 7.14 and serve for comparison.*

Single SRR		#	1		2		3	
	EELS	$\Delta E\,[\mathrm{eV}]$	0.69		1.19		1.48	
Configuration 1×2		#	1	2	3	4	5	6
	EELS	$\Delta E\,[\mathrm{eV}]$	0.62	0.74	1.11	1.23	1.40	1.53
Configuration 2×1		#	1	2	3	4	5	6
	EELS	$\Delta E\,[\mathrm{eV}]$	0.65	0.70	1.03	1.31	1.42	1.53
	HARMINV	$\hbar\omega\,[\mathrm{eV}]$	0.65	0.71	1.03	1.30	-	1.53
	extinction	$\hbar\omega\,[\mathrm{eV}]$	-	0.71	1.05	-	1.43	-

Table 7.7: *Resonance energies of the 2×1 configuration of split-ring resonators. The values for EELS were obtained from the EELS spectra depicted in Fig. 7.14. The resonance energies of the extinction cross section were determined from Fig. 7.16. "-" indicates indeterminable resonances. Resonance energies of the single SRR and 1×2 configuration serve for comparison.*

polarization. The resonance energies determined by these methods are summarized in Tab. 7.7 together with other systems for comparison. In general, qualitatively similar results are obtained compared to the *1 × 2* configuration.

The EELS spectra provide quick access to the resonances of the structure, however, exact values are often difficult to retrieve due to the spectral overlap of optical bright and dark plasmons. The comparison to the extinction cross section shows that the first few resonance energies are correctly determined from the EELS analysis. With the exception of the third order resonance for which a blind spot is observed in the center of the structure, EELS maps represent the magnitude of the (squared) electric field.

The eigenmodes of the composed structure are standing waves along each SRR connected by in-phase and antiphase relations. These were consistently assigned optically bright and dark by harmonic inversion and the extinction cross-section spectra.

7.3.3 Coupled Split-Ring Resonators in 1x4 Configuration

In this section, we investigate an coupled system of four SRRs. Due to the increase of the number of metaatoms, we expect the fundamental resonance of the single SRR to be split into four resonances whereby the number of energetically higher and lower resonances should be equal.

Simulation Setup

The simulation setup of the *1 × 4* configuration of split-ring resonators is conceptually identical to the one presented for the single split-ring resonator in Sec. 7.2.1. The geometry of the individual SRRs, the membrane and the perfectly matched layers as well as the material model and boundary conditions are equal to the single SRR case.

The SRRs are vertically aligned and horizontally separated by 35 nm from each other as sketched in Fig. 7.17 (top panel). The tetrahedral mesh consists of approximately 80k elements (cf. Fig. 7.17, central panel), and the scanned area is increased to 530 nm × 440 nm which corresponds to more than 2400 simulations to obtain a 10 nm resolution. As before, only the positive x-direction is scanned. The loss probability in the negative halfspace $x < 0$ is constructed by Eq. (7.5), resulting in a total accessible area of 1060 nm × 440 nm.

Characterization by EELS Spectra and Maps

EELS spectra for several electron trajectories of the *1 × 4* configuration of split-ring resonators are depicted Fig. 7.18. We can clearly identify the four expected resonances emanated from the first order resonance of the single SRR at $\Delta E = 0.69\,\mathrm{eV}$. The corresponding resonance energies are listed in Tab. 7.9 on page 116 together with values of the single SRR and *1 × 2* configuration for comparison. As expected by the plasmon hybridization model, the number of energetically higher and lower resonances is found to be equal. Interestingly, the first two resonances of the *1 × 4* configuration form a lower and higher pair with respect to the *1 × 2* configuration resonance at $\Delta E = 0.62\,\mathrm{eV}$. The same holds for the higher energy pair. Split resonances of the second fundamental single SRR mode can be anticipated, however, the resonances overlap strongly.

The EELS maps at the four resonance energies listed in Tab. 7.9 are summarized in the left column of Fig. 7.19. At $\Delta E = 0.55\,\mathrm{eV}$, maximum and moderate loss probability is found at the outer arms of the outer and inner SRRs, respectively. Noteworthy, quasi zero loss probability is observed at the inner arms of the respective metaatoms. The following resonances show a clear evolution by forming

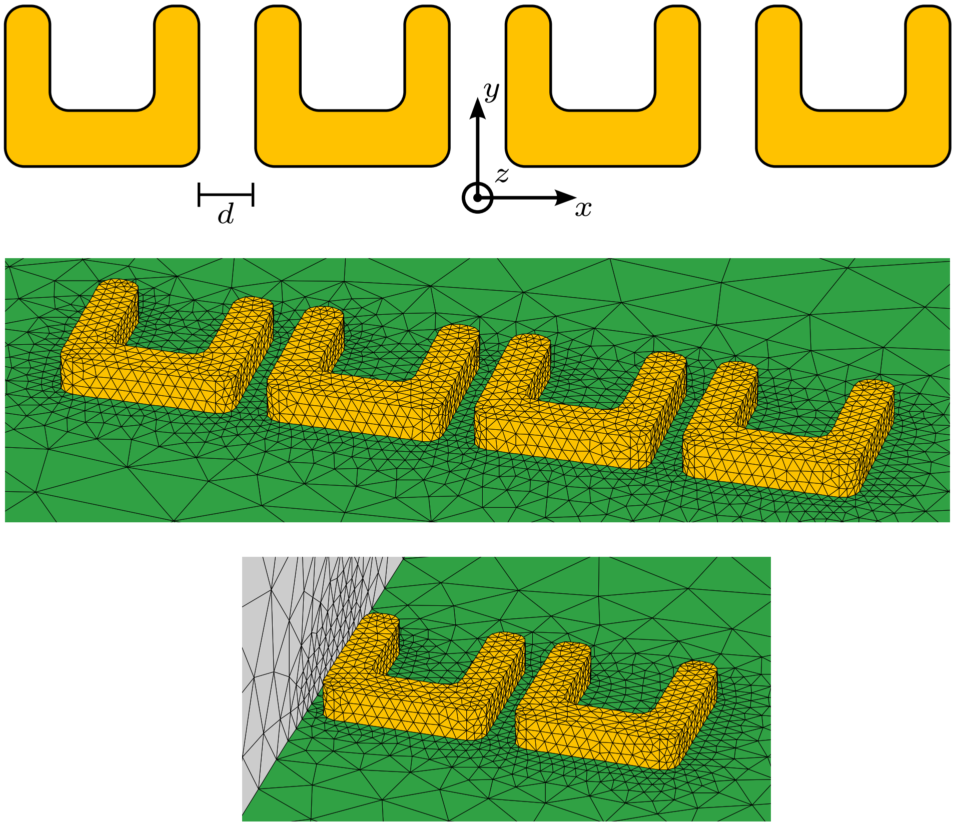


Figure 7.17: *Sketch and sections of tetrahedral meshes of the* 1×4 *configuration of split-ring resonators. The split-ring resonators are equidistantly separated by* $d = 35\,\text{nm}$*, and the configuration is centered around* $x = 0$ *(top panel). All other parameters are equivalent to the single SRR case (cf. Sec. 7.2.1 and Fig. 6.1). The central and bottom panel visualize sections of the tetrahedral meshes modeling the coupled system of SRRs (gold) on a silicon nitride membrane (green) excluding and including the symmetry plane (gray), respectively.*

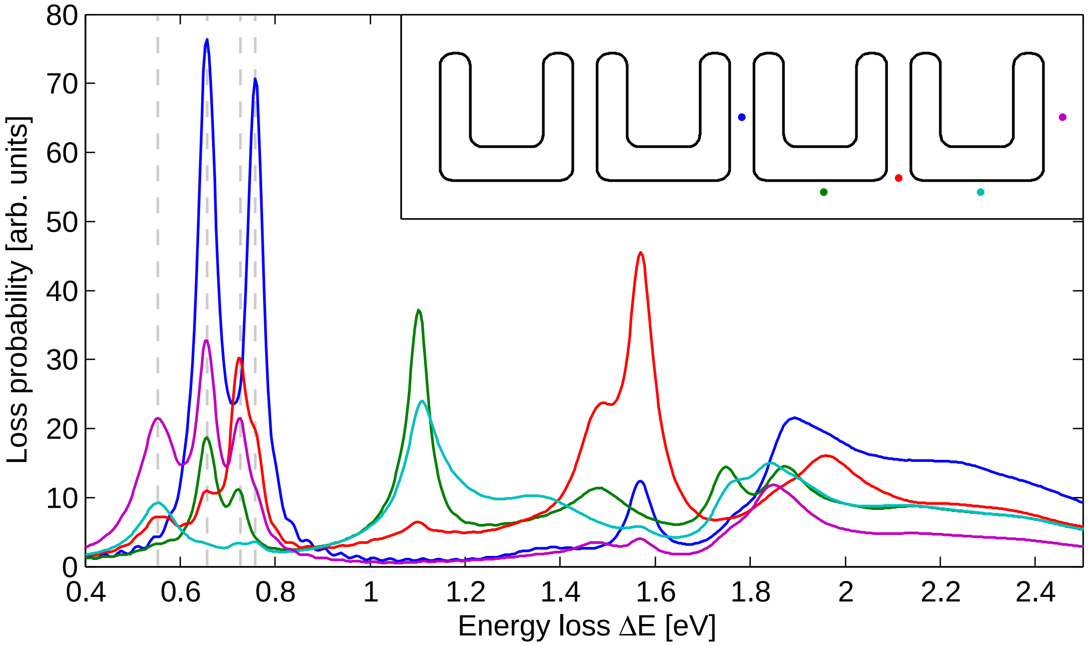


Figure 7.18: *EELS spectra of the 1 × 4 split-ring resonator configuration for distinct electron trajectories, indicated by identical colors in the inset.*

one, two and three locations of maximum loss probability. The resonance at $\Delta E = 0.66\,\text{eV}$ shows significant loss probability at the outermost arms and at the center of the coupled system spread over the arms of the inner SRRs and the central gap. The loss probability of the third order resonance at $\Delta E = 0.73\,\text{eV}$ is dominated by the contributions located at the outer gaps and the arms around them. At $\Delta E = 0.76\,\text{eV}$, all three gaps and their enclosing arms show strong loss probability.

Harmonic Inversion of the Electric Field

The time-dependent electric field is analyzed in analogy to Sec. 7.2.4 to confirm the resonance energies of the EELS analysis. The tetrahedral mesh is adapted to the *1 × 4* configuration and consists of 40k elements (cf. Fig. 7.17, right panel). The system is excited by an electron passing the SRR configuration at the tip of the right arm of the rightmost SRR. We record the electric field at points located 20 nm above the SRRs as indicated in the inset of Fig. 7.18. The resonance energies, quality factors and associated boundary conditions for the first four resonances are listed in Tab. 7.8 and compared to the data of the EELS analysis in Tab. 7.9 on page 116.

The resonance energies confirm the results of the EELS analysis and help to construct the mode profiles of the resonances in the following section.

Comparing EELS Maps to the Modes of the Electric Field

In analogy to previous sections, the analysis is continued by the calculation of the mode profiles. The resulting mode profiles of the four resonance energies listed in Tab. 7.9 were constructed by use of Eq. (7.6) together with the associated boundary conditions determined by HARMINV. The mode profiles are depicted in the right column of Fig. 7.19.

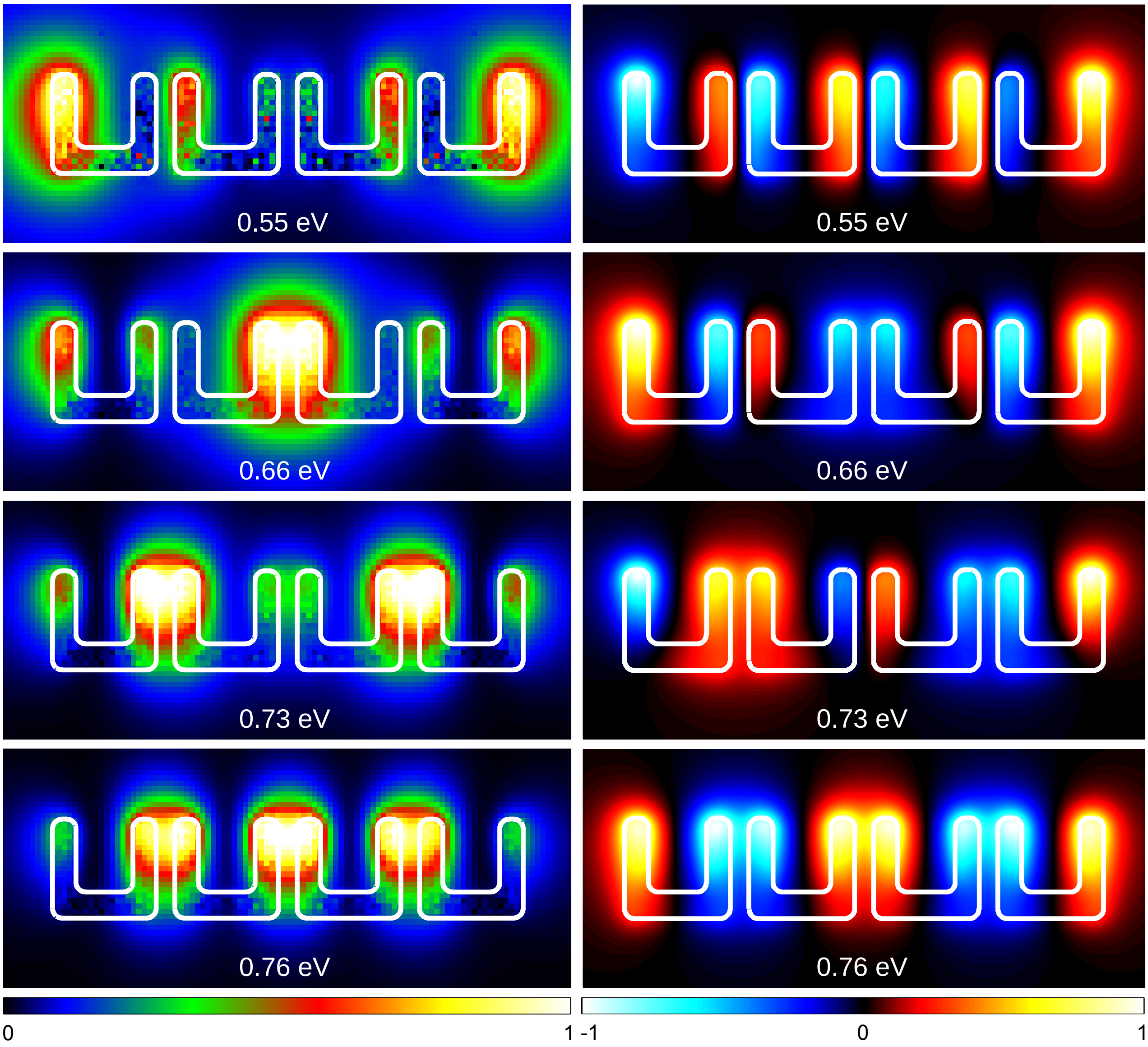


Figure 7.19: *EELS map (left column) and field distribution of the E_z-component (right column) of the 1×4 split-ring resonator configuration at resonance energies indicated by the dashed guides in Fig. 7.18.*

#	1	2	3	4
$\hbar\omega$ [eV]	0.56	0.66	0.73	0.76
Q	6	13	18	20
BC	PEC	PMC	PEC	PMC

Table 7.8: *Resonance energy, quality factor and associated boundary condition for resonances of the 1×4 configuration of split-ring resonators obtained by employing* HARMINV.

The first mode at $\hbar\omega = 0.55\,\mathrm{eV}$ comprises solely in-phase related first order modes of the single SRR which renders the mode optically bright. A similar evolution as compared to the EELS maps is observed for the coupled mode profiles, too. The two rightmost and leftmost SRRs are in phase for the second mode at $\hbar\omega = 0.66\,\mathrm{eV}$, resulting in an antiphase relation of the inner SRRs. In contrast, at $\hbar\omega = 0.73\,\mathrm{eV}$, the two rightmost and leftmost SRRs exhibit an antiphase relation, resulting in an in-phase behavior of the inner SRRs. The fourth eigenmode at $\hbar\omega = 0.76\,\mathrm{eV}$ consists of mutually antiphase related first order modes of the single SRR.

The comparison of the mode profile and the EELS map of the first resonance at 0.55 eV reveals a blind spot at the arms enclosing the central gap, i. e., while the eigenmode shows significant field strength in these arms the loss probability is very small. In addition, a novel feature is observed. An asymmetric distribution of the loss probability in the enclosing arms of the rightmost (and leftmost) gap is observed. More precisely, the field strength of the mode profile in both arms enclosing the rightmost gap is significantly different from zero while the loss probability shows significant values in one arm only.

A similar situation occurs for the second resonance, too. Consequently, antiphase related fields in neighboring arms of the mode profile are not necessarily connected to a uniform loss probability distribution. Another interesting fact is observed for the third resonance at 0.73 eV. While the third order mode crosses zero at $x = 0$ (middle of the central gap, cf. Fig. 7.17), the loss probability remains non-zero. This supports the interpretation of the EELS map to render a LDOS(-like quantity).

Comparing EELS and Extinction Cross-Section Spectra

At last, we analyze the extinction cross section of the *1×4* configuration employing horizontally and vertically polarized plane waves. The resulting extinction cross-section spectra are depicted in Fig. 7.20, thereof determined resonance energies are listed in Tab. 7.9.

The resonance energies obtained by the EELS analysis are confirmed. As expected by the analysis of the mode profile, the first resonance of the coupled system at $\hbar\omega = 0.55\,\mathrm{eV}$ is optically bright. Interestingly, we find the third resonance at $\hbar\omega = 0.73\,\mathrm{eV}$ to be optically bright, too, while the second and fourth resonance at $\hbar\omega = 0.66\,\mathrm{eV}$ and $\hbar\omega = 0.76\,\mathrm{eV}$ are optically dark.

Summary

The *1×4* configuration of split-ring resonators was analyzed by EELS, harmonic inversion of the electric field, the mode profile and the extinction cross section. The resonance energies obtained by these approaches are summarized in Tab. 7.9 and agree well.

We obtain the expected splitting into four resonances for the first resonance of the metaatom from the EELS spectrum. Higher resonances can be anticipated but strongly overlap spectrally. The EELS maps show blind spots, similar to the case of the *1×2* and *2×1* configuration. A novel effect was observed which is characterized by an asymmetric loss probability distribution although the associated mode is (anti)symmetric at this location. The maps also support the interpretation of EELS to render a LDOS(-like quantity).

The first and third resonance are optically bright modes as identified by the extinction cross-section spectra. Consequently, the second and fourth resonance are optically dark modes.

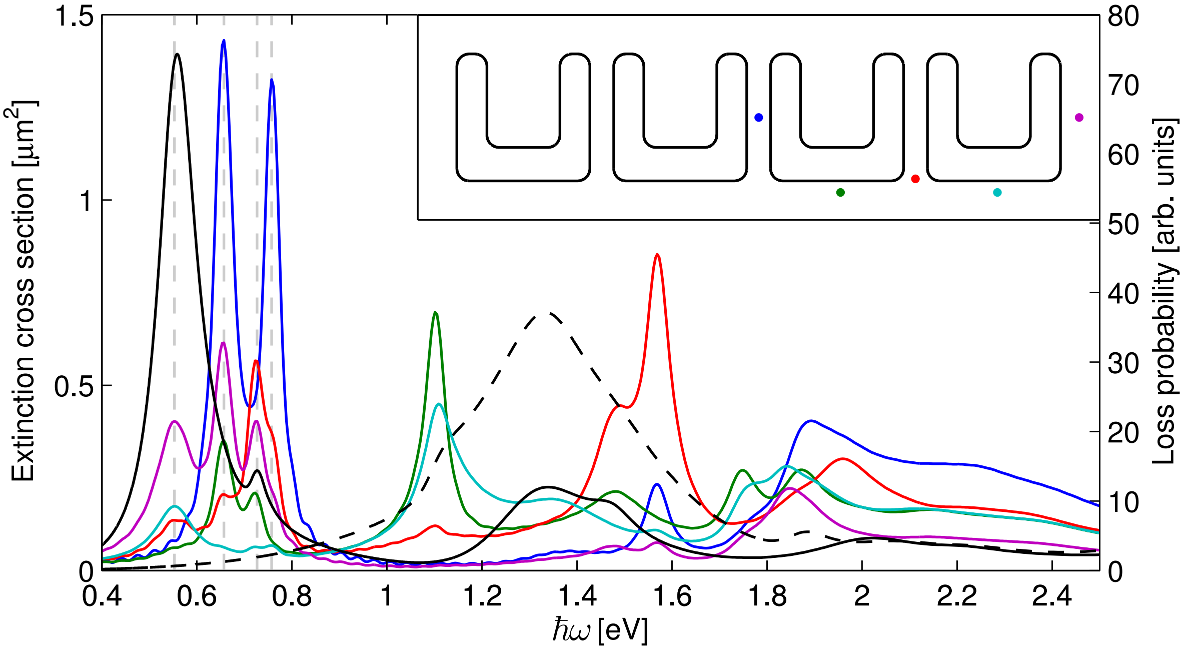


Figure 7.20: *Extinction cross-section spectra and EELS spectra for the 1×4 configuration of SRRs. The extinction cross sections were obtained using horizontally (solid black) and vertically polarized (dashed black) plane waves, respectively. The EELS spectra are identical to those in Fig. 7.18 and serve for comparison.*

Single SRR		#	1			
	EELS	ΔE [eV]	0.69			
Configuration 1×2		#	1		2	
	EELS	ΔE [eV]	0.62		0.74	
Configuration 1×4		#	1	2	3	4
	EELS	ΔE [eV]	0.55	0.66	0.73	0.76
	HARMINV	$\hbar\omega$ [eV]	0.56	0.66	0.73	0.76
	Extinction	$\hbar\omega$ [eV]	0.56	-	0.73	-

Table 7.9: *Resonance energies of the 1×4 configuration of split-ring resonators. The values for EELS were obtained from the EELS spectra depicted in Fig. 7.18. The resonance energies of the extinction cross section were determined from Fig. 7.20. Resonance energies of the single SRR and 1×2 configuration serve for comparison.*

7.4 Summary

In this chapter, we analyzed a single split-ring resonator and various coupled split-ring resonators in *1*×*2*, *2*×*1* and *1*×*4* configuration by electron energy loss spectroscopy, harmonic inversion of the electric field, the field distribution of the electromagnetic modes and the extinction cross sections.

The comparison of the resonance energies determined by EELS, harmonic inversion and the extinction cross-section spectra shows that EELS efficiently and accurately determines the resonance energies of the system's eigenmodes. However, due to the simultaneous excitation and probing of optically bright and dark modes, resonances may overlap strongly. As a consequence, the resonance energies determined from the EELS spectra can differ from the eigenenergies of the electromagnetic modes as observed for the *1*×*2* configuration. The splitting of resonances of the metaatom was consistently observed for all coupled systems in accordance with the plasmon hybridization model [91].

EELS maps, i. e., the loss probability as a function of space for a constant energy loss, were calculated at the resonance energies and compared to the field distribution of the modes projected along the electron's trajectory. The interpretation that EELS maps roughly represent the magnitude of the squared electric field [86] is confirmed for the single SRR. For coupled systems of SRRs, however, we observe blind spots as reported in Ref. [87]. Furthermore, the *1*×*4* configuration shows a locally asymmetric loss probability distribution although the associated mode is (anti)symmetric at that site. At any rate, the interpretation of EELS maps to render to magnitude of the (squared) electric field seems crude for complex systems.

In agreement with Ref. [86], the electromagnetic modes of the single SRR are standing waves along the split-ring resonator. The eigenmodes of the coupled systems are composed of the metaatom's eigenmodes with in-phase or antiphase relation as predicted by the plasmon hybridization model [91]. In addition, the eigenmodes were attributed optically bright and dark modes which was deduced from the harmonic inversion of the electric field and confirmed by the extinction cross-section spectra.

Electron energy loss spectroscopy offers a detailed characterization of plasmonic systems including optically bright *and* dark plasmons. However, due to the absent phase information in the EELS analysis, further characterization methods are required to distinguish between optically bright and dark modes. On a numerical level, the capabilities of EELS are paid for by increased computational effort since every pixel in an EELS maps requires an independent simulation. Our GPU implementation on a NVidia® GeForce® GTX580 outperforms an equivalent well established CPU implementation on a single Intel® Xeon® X5670 Core$^{\mathrm{TM}}$ @ 2.93 GHz by a factor of 50-60 for the simulations performed for the EELS analyses whereby both calculations were executed in single precision. The influence of the limited single precision was inspected by comparison against double precision calculations. DGTD on GPUs provides a feasible method to encounter the numerical effort and provides a suitable tool to perform EELS analyses on plasmonic nanostructures.

Chapter 8

8 Summary and Outlook

Today, increasingly complex dielectric and metallic nanostructures are investigated by researchers. Likewise, characterization methods become more and more sophisticated. These demanding analyses require flexible and highly powerful numerical tools which are capable of handling arbitrarily shaped geometries to a high degree of accuracy at reasonable time- and cost-efficiency.

Various applications have demonstrated that the discontinuous Galerkin time-domain method is well suited for the numerical analysis of photonic and plasmonic nanostructures [22, 24, 52, 94]. An unstructured mesh of tetrahedral elements supports a flexible representation of the physical system's geometry while high-order polynomial basis functions ensure an accurate approximation of the electromagnetic fields. However, flexibility and high accuracy come at the price of increased algorithmic complexity.

In this thesis, we implemented the discontinuous Galerkin time-domain method on graphic processing units. This implementation allows for a time- and cost-efficient treatment of almost arbitrarily shaped dielectric and metallic nanostructures rendering the DGTD method on GPUs a versatile and powerful numerical tool.

8.1 Summary

In Chap. 5, we demonstrated that the DGTD method is well suited for implementation on GPUs. Quantifying the performance in absolute values, we find that the local-element differentiation and the lift operation, i. e., the computationally most demanding sections of the method, are performed by up to 800 GFlops/s[1] and 600 GFlops/s, respectively. This amounts to roughly 50% of the theoretically available number of floating-point operations per second. Based on bare numbers, our implementation outperforms the one presented in Ref. [27] which exploits 30% of the peak performance.[2]

In comparison to a well-established in-house CPU implementation executed on a single core[3], the GPU code shows speedups of the order of 30-50. Comparing the performance in terms of cost-efficiency[4], a net-speedup of $5-10$ over CPU systems can be achieved. Thus, individual simulations are evaluated $5-10$ *times* faster at similar cost.

[1] One GFlops/s corresponds to 10^9 single-precision floating point operations per second.

[2] This comparison is based on values determined on different GPUs of the same hardware architecture. Nevertheless, this comparison should not be overestimated.

[3] We compare the calculation times of simulations performed in single precision on a NVidia® GeForce® GTX580 (GPU system) and an Intel® Core™ i7 CPU 970 @ 3.20 GHz (CPU system).

[4] A single NVidia® GeForce® GTX580 roughly corresponds to 6-10 CPU cores.

Due to the dramatic reduction of simulation time, this approach is particularly beneficial for rapid prototyping. For example, the structures investigated in Chap. 6, nanorods and split-ring resonators, can be simulated in one to two hours which would amount to roughly half a day on a multi-core CPU system. Furthermore, computationally very intense investigations, such as the electron energy loss spectroscopy analysis of Chap. 7, can be achieved in a reasonable amount of time. Here, more than 6000 individual simulations were performed — each of $30-60$ minutes simulation time. Calculated on a single NVidia® GeForce® GTX580, this amounts to (at least) 125 days. At similar cost, our Intel® Core™ i7 CPU 970 @ 3.20 GHz multi-core CPU system would have required (at least) 625 days.[5]

As a first application of the GPU implementation, we calculated the scattering, absorption, and extinction cross-section spectra in the near-infrared for a series of gold nanorods and split-ring resonators (SRRs) in order to assist the verification of a newly developed experimental characterization method [12] (Chap. 6). The comparison between experimental and simulated results exhibits overall good agreement which confirms the anticipated behavior for both the nanorods and the split-ring resonators.

Two major sources of deviations between simulated and experimental results were identified: Due to rough surfaces and grain boundaries of the samples, the experimental gold does not quite agree with the data of Johnson and Christy [38], which we employed to determine the free parameters of a Drude model in the near-infrared regime. In addition, the thickness of the nanostructures is deduced from fabrication characteristics and not measured explicitly. As examined in Ref. [14], the thickness strongly influences the resonance wavelength as well as the scattering and the absorption cross section. In conclusion, the deviations between simulation and experiment can be understood in dependence of these parameters.

At last, an extensive study of coupled systems split-ring resonators employing electron energy loss spectroscopy and further analysis methods was performed in Chap. 7. Our analysis of a single split-ring resonator is in accordance with Refs. [24, 78, 86], i. e., the spatial distribution of the loss probability for a constant energy loss (EELS map) roughly renders the squared modulus of the projected electric field, and the eigenmodes are standing waves along the "U"-like shape. Our analysis of coupled split-ring resonator systems is in accordance with Ref. [87], i. e., we observe the formation of blind spots in EELS maps. Furthermore, the eigenmodes are composed of the single SRR's eigenmodes in in-phase and antiphase relation which is in accordance with the plasmon hybridization model [91, 93].

In addition, we examined the energy splitting due to the interaction of two vertically aligned SRRs in dependence of their separation with comparatively little effort. On an experimental basis, such an investigation is costly and time-consuming since several tens of samples would have to be fabricated and measured in order to yield similar results. Furthermore, experimental artifacts due to imperfections of the fabrication methods can be easily distinguished from intended physical effects. The benefits outlined here are not due to the properties of particular systems but generic features of our implementation.

8.2 Outlook

Finally, we address some issues of potential future work. We have not yet exploited the full potential of the DGTD method. Curvilinear elements [43], for example, yield several advantages compared to planar-faced elements. These elements represent round geometries by curved faces. This immediately increases accuracy and allows the application of larger and less elements. Both of these aspects may

[5] This example is meant to emphasize the value of the speedup. In practice, we employ multiple GPUs in parallel in order to evaluate large amounts of simulations.

accelerate the method. Less elements result in a reduced computational effort. Larger elements allow for a larger time-step, since the time-step is usually limited by those elements which model curved geometries. Therefore, we can profit twice from the implementation of curvilinear elements: It reduces the total number of elements and it potentially improves the time-step. A proof-of-principle implementation, that did however not employing the memory efficient approach by Warburton, was applied for two-dimensional slot-waveguide resonators in Ref. [94].

Another yet unexploited potential is given by *hp*-adaptivity which allows for a dynamic adaption of both the element-size h and the polynomial order p *during* runtime. A dynamic increase of both parameters in regions of high field gradients may increase accuracy while a reduction in low field gradient regions results in decreased computational effort. In combination with the DGTD method's inherent parallelism and increasing functionality of GPUs, accuracy as well as time- and cost-efficiency of this numerical approach will strongly increase in the near future.

These improvements would be especially beneficial for the analysis of large structures with smooth surfaces, e. g., whispering-gallery mode resonators. These resonators exhibit an exceptionally large quality factor and were already applied to detect single molecule events [95].

Another interesting topic closely related to this work, is electron energy gain spectroscopy (EEGS) [96]. In contrast to electron energy loss spectroscopy, the electron gains energy from an initially excited plasmonic structure. While the energetic resolution of EELS experiments is limited by the zero-loss peak, i. e., the energy distribution of the electrons prior to the scattering event, the resolution in EEGS is fundamentally limited by the bandwidth of the illumination of the plasmonic structure leading to a sub-meV resolution [96].

8.3 Conclusion

In conclusion, we have implemented the DGTD method for Maxwell's equations on GPUs. This powerful tool was successfully employed in computationally demanding physical investigations of metallic nanostructures. Both applications of the GPU implementation of this thesis demonstrate the need and the capabilities of sophisticated numerical methods in the field of nano-plasmonics.

We believe that the discontinuous Galerkin time-domain method is one of the most versatile and efficient simulation tools for arbitrarily shaped photonic and plasmonic nanostructures. In conjunction with graphic processing units, the DGTD method provides a time-, and cost-efficient approach to handle the complexity of plasmonic nanostructures.

Acknowledgments

At last, I would like to acknowledge the people who contributed to this thesis.

First of all, I would like to thank *Prof. Dr. Kurt Busch*, my thesis advisor, for giving me the opportunity to work in his excellent group and supporting me throughout difficult times.

I want to express my gratitude to *Prof. Dr. Martin Wegener* who kindly agreed to co-review this thesis and with whom a successful collaboration within the last year of my thesis was established.

My utmost gratitude goes to *Dr. Jens Niegemann* who was supervising my work. Having moved to ETH Zürich in 2011, he continued supervising my work, answering my questions, and sharing my concerns with enduring conscientiousness. Jens, I thank you very much for everything you have done for me, and I wish you all the best for your future scientific work.

I am indebted to *Dr. Michael König* for his everlasting support. He was explaining me countless major and minor puzzling issues. Your happy and motivating mindset has facilitated my work many times. All the best for your future.

I would like to thank *Prof. Dr. Stefan Linden* and *Dipl.-Phys. Felix von Cube* for a successful collaboration.

Special thanks go to *Dipl.-Phys. Martin Husnik*. Despite a successful collaboration, he offered me insight into experimental mindedness.

Many thanks go to the proofreaders of this thesis: *Dr. Jens Niegemann*, *Dr. Michael König*, *Dipl.-Phys. Matthias Moeferdt*, *Dipl.-Phys. Martin Husnik*, *M. Sc. Florian Mauch*, and *M. A. Linda Caggegi*. Thank you very much!

Dr. Andreas Pönicke fixed innumerable problems on the computers for which I am very grateful.

The *Karlsruhe School of Optics and Photonics* (KSOP) deserves my special gratitude. KSOP granted me a scholarship for the first year of my research and provided funds for computers and travel expenses. Immaterial benefits such as management modules and seminars are also gratefully acknowledged.

I do not want to end without acknowledging *Dr. Antje Bergmann*, *Dipl.-Phys. Jens Küchenmeister*, *Dr. Thomas Zebrowski*, and *Dr. Paolo Longo* for maintaining a healthy and friendly spirit throughout the last couple of months. Many things have gone bad ways during that time, and I would like to thank you for supporting me.

Last but not least, I would like to thank my parents, my sister, and my girlfriend Linda for their great care throughout the last years.

Thank you very much.

Publications

1. 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.*, 7:1572–1580, 2010. doi: 10.1166/jctn.2010.1521.

2. J. Niegemann, R. Diehl, and K. Busch. Efficient low-storage Runge–Kutta schemes with optimized stability regions. *J. Comput. Phys.*, 231(2):364–372, 2012. doi: 10.1016/j.jcp.2011.09.003.

3. M. Husnik, S. Linden, R. Diehl, J. Niegemann, K. Busch, and M. Wegener. Quantitative Experimental Determination of Scattering and Absorption Cross-Section Spectra of Individual Optical Metallic Nanoantennae. *Phys. Rev. Lett.*, (accepted for publication), 2012.

4. F. von Cube, S. Irsen, R. Diehl, J. Niegemann, K. Busch, and S. Linden. From Isolated Metaatoms to Photonic Metamaterials: Evolution of the Plasmonic Near-field. (submitted).

Bibliography

[1] J. C. Maxwell. A Dynamical Theory of the Electromagnetic Field. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 155:pp. 459–512, 1865.

[2] Y. Sonnefraud, A. L. Koh, D. W. McComb, and S. A. Maier. Nanoplasmonics: Engineering and observation of localized plasmon modes. *Laser & Photon. Rev.*, 6(3):277–295, 2012. doi: 10.1002/lpor.201100027.

[3] S. Kim, J. Jin, Y.-J. Kim, I.-Y. Park, Y. Kim, and S.-W. Kim. High-harmonic generation by resonant plasmon field enhancement. *Nature*, 453(7196):757–760, Jun. 2008. doi: 10.1038/nature07012.

[4] M. E. Stewart, C. R. Anderton, L. B. Thompson, J. Maria, S. K. Gray, J. A. Rogers, and R. G. Nuzzo. Nanostructured Plasmonic Sensors. *Chem. Rev.*, 108(2):494–521, Jan. 2008.

[5] D. W. Pohl, W. Denk, and M. Lanz. Optical stethoscopy: Image recording with resolution lambda/20. *Appl. Phys. Lett.*, 44(7):651–653, 1984. doi: 10.1063/1.94865.

[6] B. Hecht, B. Sick, U. P. Wild, V. Deckert, R. Zenobi, O. J. F. Martin, and D. W. 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.

[7] B. Novotny, L. ; Hecht. *Principles of nano-optics.* Cambridge University Press, Cambridge [u.a.], 1st edition, 2007.

[8] J. B. Pendry, A. J. Holden, D. J. Robbins, and W. J. Stewart. Magnetism from conductors and enhanced nonlinear phenomena. *IEEE T. Microw. Theory*, 47(11):2075 –2084, Nov. 1999. doi: 10.1109/22.798002.

[9] S. Linden, C. Enkrich, M. Wegener, J. Zhou, T. Koschny, and C. M. Soukoulis. Magnetic Response of Metamaterials at 100 Terahertz. *Science*, 306(5700):1351–1353, Nov. 2004. doi: 10.1126/science.1105371.

[10] G. Dolling, M. Wegener, C. M. 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.

[11] V. M. Shalaev, W. Cai, U. K. Chettiar, H.-K. Yuan, A. K. Sarychev, V. P. Drachev, and A. V. Kildishev. Negative index of refraction in optical metamaterials. *Opt. Lett.*, 30(24):3356–3358, Dec. 2005. doi: 10.1364/OL.30.003356.

[12] M. Husnik, S. Linden, R. Diehl, J. Niegemann, K. Busch, and M. Wegener. Quantitative Experimental Determination of Scattering and Absorption Cross-Section Spectra of Individual Optical Metallic Nanoantennae. *Phys. Rev. Lett.*, (accepted for publication), 2012.

[13] F. von Cube, S. Irsen, R. Diehl, J. Niegemann, K. Busch, and S. Linden. From Isolated Metaatoms to Photonic Metamaterials: Evolution of the Plasmonic Near-field. (submitted).

[14] M. König. *Discontinuous Galerkin Methods in Nanophotonics.* PhD thesis, Institut für Theoretische Festkörperphysik, Karlsruhe Institute of Technology, 2011.

[15] K. Busch, M. König, and J. Niegemann. Discontinuous Galerkin methods in nanophotonics. *Laser & Photon. Rev.*, 5(6):773–809, Nov. 2011. doi: 10.1002/lpor.201000045.

[16] K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, 14(3):302 –307, May 1966. doi: 10.1109/TAP.1966.1138693.

[17] S. C. Taflove, A. ; Hagness. *Computational electrodynamics : the finite-difference time-domain method.* Artech House antennas and propagation library. Artech House, Boston [u.a.], 3$^{\text{rd}}$ edition, 2005.

[18] J.-M. Jin. *The finite element method in electromagnetics.* Wiley, New York [u.a.], 2$^{\text{nd}}$ edition, 2002.

[19] P. Monk. *Finite element methods for Maxwells equations.* Numerical mathematics and scientific computationOxford science publications. Clarendon Press, Oxford, 2003.

[20] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report, LA-UR-73-479, Los Alamos National Laboratory, 1973.

[21] J. S. 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.

[22] J. Niegemann, M. König, K. Stannigel, and K. Busch. Higher-order time-domain methods for the analysis of nano-photonic systems. *Phot. Nano. Fund. Appl.*, 7(1):2–11, Feb. 2009. doi: 10.1016/j.photonics.2008.08.006.

[23] 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. doi: 10.1364/OE.18.006545.

[24] C. Matyssek, J. Niegemann, W. Hergert, and K. Busch. Computing electron energy loss spectra with the Discontinuous Galerkin Time-Domain method. *Phot. Nano. Fund. Appl.*, 9(4):367–373, 2011. doi: 10.1016/j.photonics.2011.04.003.

[25] F. von Cube, S. Irsen, J. Niegemann, C. Matyssek, W. Hergert, K. Busch, and S. Linden. Spatio-spectral characterization of photonic meta-atoms with electron energy-loss spectroscopy. *Opt. Mater. Express*, 1(5):1009–1018, Sep. 2011. doi: 10.1364/OME.1.001009.

[26] NVIDIA Corporation. *OpenCL Programming Guide for the CUDA Architecture*, Feb. 2011. Version 4.0.

[27] A. Klöckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *J. Comput. Phys.*, 228(21):7863 – 7882, 2009. ISSN 0021-9991. doi: 10.1016/j.jcp.2009.06.041.

[28] D. Komatitsch, G. Erlebacher, D. Göddeke, and D. Michéa. High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster. *J. Comput. Phys.*, 229(20):7692 – 7714, 2010. ISSN 0021-9991. doi: 10.1016/j.jcp.2010.06.024.

[29] W. Xian and A. Takayuki. Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster. *Parallel Comput.*, 37(9):521 – 535, 2011. doi: 10.1016/j.parco.2011.02.007.

[30] J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 3$^{\mathrm{rd}}$ edition, 1999.

[31] J. A. Stratton. *Electromagnetic Theory*. Wiley, 2007.

[32] C. F. Bohren and D. R. Huffman. *Absorption and scattering of light by small particles*. A Wiley-Interscience publication. Wiley, New York [u.a.], 1983.

[33] P. Drude. Zur Elektronentheorie der Metalle. *Ann. Phys.*, 306(3):566–613, 1900. doi: 10.1002/andp.19003060312.

[34] P. Drude. Zur Elektronentheorie der Metalle; II. Teil. Galvanomagnetische und thermomagnetische Effecte. *Ann. Phys.*, 308(11):369–402, 1900. doi: 10.1002/andp.19003081102.

[35] N. W. Ashcroft and N. D. Mermin. *Solid state physics*. Saunders College, Fort Worth [u.a.], 1976.

[36] T. Coleman and Y. Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM J. Optim.*, 6(2):418–445, 1996. doi: 10.1137/0806023.

[37] T. F. Coleman and Y. Li. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Math. Program.*, 67:189–224, 1994. doi: 10.1007/BF01582221.

[38] P. B. Johnson and R. W. Christy. Optical Constants of the Noble Metals. *Phys. Rev. B*, 6: 4370–4379, Dec. 1972. doi: 10.1103/PhysRevB.6.4370.

[39] A. Vial, A.-S. Grimault, D. Macías, D. Barchiesi, and M. L. 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*, 71:085416, Feb. 2005. doi: 10.1103/PhysRevB.71.085416.

[40] E. D. Palik. *Handbook of Optical Constants of Solids*. Academic Press, New York, 1985.

[41] M. Husnik, M. W. 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 (10):614–617, Oct. 2008. doi: 10.1038/nphoton.2008.181.

[42] H. C. van de Hulst. *Light scattering by small particles*. Structure of matter series. Wiley [u.a.], New York, 1957.

[43] T. Warburton. A low storage curvilinear discontinuous Galerkin time-domain method for electromagnetics. In *URSI International Symposium on Electromagnetic Theory*, pages 996 –999, Aug. 2010. doi: 10.1109/URSI-EMTS.2010.5637392.

[44] J. Niegemann. *Higher-Order Methods for Solving Maxwell's Equations in the Time-Domain*. PhD thesis, Institut für Theoretische Festkörperphysik, Universitä Karlsruhe (TH), Jan. 2009.

[45] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods.* Springer, 2007.

[46] T. Warburton. An explicit construction of interpolation nodes on the simplex. *J. Engrg. Math.*, 56:247–262, 2006. ISSN 0022-0833. doi: 10.1007/s10665-006-9086-6.

[47] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comput.*, 6:345–390, 1991. ISSN 0885-7474. doi: 10.1007/BF01060030.

[48] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Society for Industrial and Applied Mathematics, 2004. doi: 10.1137/1.9780898717938.

[49] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* Wiley [u.a.], New York, 2003.

[50] J. H. Williamson. Low-storage Runge-Kutta schemes. *J. Comput. Phys.*, 35(1):48 – 56, 1980. doi: 10.1016/0021-9991(80)90033-9.

[51] M. H. Carpenter and C. A. Kennedy. Fourth-Order 2N-Storage Runge-Kutta Schemes. Technical Report 109112, NASA Technical Memorandum, Jun. 1994.

[52] 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. doi: 10.1364/OE.17.014934.

[53] J. Niegemann, R. Diehl, and K. Busch. Efficient low-storage Runge–Kutta schemes with optimized stability regions. *J. Comput. Phys.*, 231(2):364–372, 2012. doi: 10.1016/j.jcp.2011.09.003.

[54] R. Diehl. Advanced Time Stepping Methods for Discontinuous Galerkin Time-Domain Simulators. Master's thesis, Institut für Theoretische Festkörperphysik, Universität Karlsruhe (TH), Oct. 2009.

[55] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Texts in applied mathematics 37. Springer, Berlin, 2^{nd} edition, 2007.

[56] H. O. 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.

[57] 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.*, 7:1572–1580, 2010. doi: 10.1166/jctn.2010.1521.

[58] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, Oct. 1994.

[59] J.-P. Bérenger. Perfectly Matched Layer (PML) for Computational Electromagnetics. *Synth. Lect. Comput. Electromagn.*, 2(1):1–117, 2007. doi: 10.2200/S00030ED1V01Y200605CEM008.

[60] W. C. Chew and W. H. Weedon. A 3D perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microw. Opt. Techn. Let.*, 7(13):599–604, 1994. ISSN 1098-2760. doi: 10.1002/mop.4650071304.

[61] 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. doi: 10.1364/OE.19.004618.

[62] Oct. 2012. URL `http://www.openacc-standard.org/`.

[63] NVIDIA Corporation. CUDA toolkit 4.2 CUBLAS library, Feb. 2012. URL `http://developer.nvidia.com/cuda/cublas`.

[64] Oct. 2012. URL `http://www.culatools.com/`.

[65] Khronos OpenCL Working Group. *The OpenCL Specification*, Nov. 2010. Version 1.1, Revision 33.

[66] NVIDIA Corporation. *NVIDIA's Next Generation CUDA^{TM} Compute Architecture: Fermi^{TM}*. V1.1.

[67] NVIDIA Corporation. *NVIDIA's Next Generation CUDA^{TM} Compute Architecture: Kepler^{TM} GK110*, 2012. V1.0.

[68] Oct. 2012. URL `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-580/specifications`.

[69] NVIDIA Corporation. *Optimization OpenCL Programming Best Practices Guide*, May 2010.

[70] C. A. Balanis. *Advanced engineering electromagnetics.* Wiley, New York [u.a.], 1989.

[71] F. J. García de Abajo. Optical excitations in electron microscopy. *Rev. Mod. Phys.*, 82:209–275, Feb. 2010. doi: 10.1103/RevModPhys.82.209.

[72] F. J. García de Abajo. Relativistic energy loss and induced photon emission in the interaction of a dielectric sphere with an external electron beam. *Phys. Rev. B*, 59(4):3095–3107, Jan. 1999. doi: 10.1103/PhysRevB.59.3095.

[73] L. R. Hirsch, R. J. Stafford, J. A. Bankson, S. R. Sershen, B. Rivera, R. E. Price, J. D. Hazle, N. J. Halas, and J. L. West. Nanoshell-mediated near-infrared thermal therapy of tumors under magnetic resonance guidance. *Proc. Natl. Acad. Sci. USA*, 100(23):13549–13554, Nov. 2003. doi: 10.1073/pnas.2232479100.

[74] J. Schöberl. NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1(1):41–52, 1997. doi: 10.1007/s007910050004.

[75] P. B. Johnson and R. W. Christy. Optical constants of transition metals: Ti, V, Cr, Mn, Fe, Co, Ni, and Pd. *Phys. Rev. B*, 9:5056–5070, Jun. 1974. doi: 10.1103/PhysRevB.9.5056.

[76] R. F. Egerton. Electron energy-loss spectroscopy in the TEM. *Rep. Prog. Phys.*, 72(1):016502, 2009. doi: 10.1088/0034-4885/72/1/016502.

[77] F. J. García de Abajo and M. Kociak. Probing the Photonic Local Density of States with Electron Energy Loss Spectroscopy. *Phys. Rev. Lett.*, 100(4):106804, Mar. 2008. doi: 10.1103/PhysRevLett.100.106804.

[78] J. Nelayah, M. Kociak, O. Stephan, F. J. García de Abajo, M. Tence, L. Henrard, D. Taverna, I. Pastoriza-Santos, L. M. Liz-Marzan, and C. Colliex. Mapping surface plasmons on a single metallic nanoparticle. *Nature Phys.*, 3(5):348–353, May 2007. doi: 10.1038/nphys575.

[79] A. L. Koh, D. W. McComb, S. A. Maier, H. Y. Low, and J. K. W. Yang. Sub-10 nm patterning of gold nanostructures on silicon-nitride membranes for plasmon mapping with electron energy-loss spectroscopy. *J. Vac. Sci. Technol. B*, 28(6):C6O45–C6O49, 2010. doi: 10.1116/1.3501351.

[80] R. F. Egerton. *Electron energy-loss spectroscopy in the electron microscope.* Plenum Press, New York [u.a.], 2$^{\text{nd}}$ edition, 1996.

[81] R. Vogelgesang and A. Dmitriev. Real-space imaging of nanoplasmonic resonances. *Analyst*, 135 (6):1175–1181, 2010. doi: 10.1039/C000887G.

[82] A. L. Koh, K. Bao, I. Khan, W. E. Smith, G. Kothleitner, P. Nordlander, S. A. Maier, and D. W. McComb. Electron Energy-Loss Spectroscopy (EELS) of Surface Plasmons in Single Silver Nanoparticles and Dimers: Influence of Beam Damage and Mapping of Dark Modes. *ACS Nano*, 3(10):3015–3022, 2009. doi: 10.1021/nn900922z.

[83] M.-W. Chu, V. Myroshnychenko, C. H. Chen, J.-P. Deng, C.-Y. Mou, and F. J. García de Abajo. Probing Bright and Dark Surface-Plasmon Modes in Individual and Coupled Noble Metal Nanoparticles Using an Electron Beam. *Nano Lett.*, 9(1):399–404, 2009. doi: 10.1021/nl803270x.

[84] H. Duan, A. I. Fernández-Domínguez, M. Bosman, S. A. Maier, and J. K. W. Yang. Nanoplasmonics: Classical down to the Nanometer Scale. *Nano Lett.*, 12(3):1683–1689, 2012. doi: 10.1021/nl3001309.

[85] B. S. Guiton, V. Iberi, S. Li, D. N. Leonard, C. M. Parish, P. G. Kotula, M. Varela, G. C. Schatz, S. J. Pennycook, and J. P. Camden. Correlated Optical Measurements and Plasmon Mapping of Silver Nanorods. *Nano Lett.*, 11(8):3482–3488, 2011. doi: 10.1021/nl202027h.

[86] G. Boudarham, N. Feth, V. Myroshnychenko, S. Linden, J. García de Abajo, M. Wegener, and M. Kociak. Spectral Imaging of Individual Split-Ring Resonators. *Phys. Rev. Lett.*, 105:255501, Dec. 2010. doi: 10.1103/PhysRevLett.105.255501.

[87] U. Hohenester, H. Ditlbacher, and J. R. Krenn. Electron-Energy-Loss Spectra of Plasmonic Nanoparticles. *Phys. Rev. Lett.*, 103:106801, Aug. 2009. doi: 10.1103/PhysRevLett.103.106801.

[88] F. J. García de Abajo, A. G. Pattantyus-Abraham, N. Zabala, A. Rivacoba, M. O. Wolf, and P. M. Echenique. Cherenkov Effect as a Probe of Photonic Nanostructures. *Phys. Rev. Lett.*, 91:143902, Sep. 2003. doi: 10.1103/PhysRevLett.91.143902.

[89] S. G. Johnson. URL `http://ab-initio.mit.edu/wiki/index.php/Harminv`.

[90] V. A. Mandelshtam. FDM: the filter diagonalization method for data processing in NMR experiments. *Prog. Nucl. Mag. Res. Sp.*, 38(2):159 – 196, 2001. doi: 10.1016/S0079-6565(00)00032-7.

[91] E. Prodan, C. Radloff, N. J. Halas, and P. Nordlander. A Hybridization Model for the Plasmon Response of Complex Nanostructures. *Science*, 302(5644):419–422, Oct. 2003. doi: 10.1126/science.1089171.

[92] E. Prodan and P. Nordlander. Plasmon hybridization in spherical nanoparticles. *J. Chem. Phys.*, 120(11):5444–5454, 2004. doi: 10.1063/1.1647518.

[93] P. Nordlander, C. Oubre, E. Prodan, K. Li, and M. I. Stockman. Plasmon Hybridization in Nanoparticle Dimers. *Nano Lett.*, 4(5):899–903, 2004. doi: 10.1021/nl049681c.

[94] K. R. 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.

[95] A. M. Armani, R. P. Kulkarni, S. E. Fraser, R. C. Flagan, and K. J. Vahala. Label-Free, Single-Molecule Detection with Optical Microcavities. *Science*, 317(5839):783–787, Aug. 2007. doi: 10.1126/science.1145002.

[96] F. J. Garcá de Abajo and M. Kociak. Electron energy-gain spectroscopy. *New J. Phys.*, 10(7):073035, 2008.