
FDTD Simulation von Elektronenstreuung an einer elektromagnetischen Welle

FDTD Simulation of electron scattering in an electromagnetic wave

David Symhoven



München 2017

FDTD Simulation von Elektronenstreuung an einer elektromagnetischen Welle

FDTD Simulation of electron scattering in an electromagnetic wave

David Symhoven

Masterarbeit
am Lehrstuhl Computational & Plasma Physics
der Ludwig-Maximilians-Universität
München

vorgelegt von
David Symhoven
aus Witten

München, den 17.07.2017

Gutachter: Prof.Dr. Hartmut Ruhl

Introduction

With the increase of computational power over the last decade computer simulations become more and more powerful. Not just that they allow us to work with circumstances, which are nearly impossible to create in the lab, like ultra cold experiments near absolute zero, they can also be used to make fundamental physics visible or help us to understand the consequences of fundamental physical laws. Simulations can be used to help interpret experiments or designing new ones. Furthermore, they can provide detailed information about patterns and behaviors we - as humans - could never see in reality. For instance, on subatomic scales dynamics and their consequences are way too fast for us to see, like the radiation of relativistic electrons.

A computer simulation of N interacting particles can be conducted by solving Newton's equation for each particle in the system. This is called the *particle-particle* or Molecular Dynamic (MD) approach. The main computational cost is the calculation of the force \vec{F} acting on a particle. This requires a summation over all other particles in the system

$$\vec{F} = \sum_{i,j} \vec{F}_{ij},$$

where \vec{F}_{ij} is the interaction force of particle i with particle j . Once the force is calculated, the underlying equation of motion can be used to calculate the velocity at a new time step and with that the new position of the particle. With the new position the forces need to be updated and the cycle starts over again. For each particle there are $N - 1$ other particles to interact with. Considering that each pair interaction only needs to be calculated once this results in a total of the order of $N(N - 1)/2$ evaluations. For a small amount of particles, the MD approach is feasible. In simulations with more than 10^8 particles, however, the computational costs become too high. Even the more efficient *Barnes-Hut* or *tree* algorithm [1] - where the interaction of one particle with a group of particles is reduced to the interaction with the multipole fields of that group - has numeric complexity of $\mathcal{O}(N \log(N))$ and is not sufficient. For situations for which particle-particle interactions are not too strong another simulation method called the *Particle-in-Cell* (PIC) algorithms exists and scales with order of $\mathcal{O}(N)$.

Because of its simplicity PIC is the standard simulation technique for plasma physics

today [5, 7, 8]. PIC makes use of the concept of quasi-particles and a Maxwell grid to calculate electric currents at discrete grid points. The Maxwell grid is used to solve discrete Maxwell equations. This way an interaction with the quasi-particles and the fields on the Maxwell grid is possible. Whereas first simulations were limited to one dimensional problems due to a lack of computational power, nowadays three dimensional simulations are feasible [3, 10, 16]. For instance, a MD simulation with 10^8 particles for 100000 time steps requires about three years on a modern 100 TFLOP/s super computer. The same simulation using PIC only requires three seconds [18].

Christian Herzing developed in his PhD-thesis an hybrid numerical technique to improve MD simulations such that their numerical load compares to the one of PIC in many situations [11]. By introducing the concept of near and far fields Herzing has been able to reduce the numeric complexity of MD simulations with radiation to almost $\mathcal{O}(N)$. His framework is written in C++. Unfortunately this is not compatible with the *Plasma Simulation Code* (PSC), which requires C code input. PSC is a massively parallel code framework for the simulation of plasma, laser-plasma, high field physics in plasma and vacuum, and MD, developed by members of the Computational & Plasma physics chair of Prof. Dr. Hartmut Ruhl. The goal of the present work is to translate the framework of Herzing into C and to create a solid and well documented code base others can build upon. To show that the framework is applicable to a real world case the scattering of an electron in an electromagnetic wave is simulated.

The thesis is structured in three parts. The first part deals with the fundamental concepts. One key aspect of this thesis are the Liénard-Wiechert fields, which will be derived analytically. Starting with the derivation of the inhomogeneous wave equation using potential equations, we will afterwards solve the inhomogeneous wave equation using Green's function approach, which will lead to the Liénard-Wiechert fields. In the second part the necessary numeric background is presented, i.e. the underlying equations of motion and how to solve them numerically. Thereto several integration schemes are introduced. Next, the hybrid method by Herzing and the *Finite Difference Time Domain* (FDTD) method, which is used to solve Maxwell equations on a staggered grid, are explained. Finally, the theory and implementation of „Uniaxial Perfectly Matched Layer“ are discussed [2]. A summary and outlook are given in the third part. The framework of Herzing [11] has been rewritten in C entirely and has been extended by new features, like a reverse Boris push, a trilinear interpolation of fields, an effective storage of precalculated fields and UPMLs. In the Appendix additional calculations are presented and topics like normalization and the software stack used are covered.

Acknowledgments

The last ten month have been a great time for me. Both challenging and exhausting but great and exciting. In my opinion we should soften the distinction between the different departments even further. Interdisciplinarity becomes more and more important as the complexity of modern research increases. Computational physics requires good software engineering. Likewise, Software Engineers can benefit from the knowledge of physicists or mathematicians. This is the reason why I liked the idea of writing a simulation framework so much. Although it's not a common thing to do in a physics master thesis, I learned a lot about programming. I really want to thank you - Prof. Ruhl - for sharing this opinion with me and for giving me the opportunity to write this thesis and of course for your input all along the way. It was always fun and inspiring to discuss with you about physics and non-physics related topics like climate change, electro mobility and the education system, just to mention a few.

Especially the last three month have been hard, because I already started my regular job in April. Thanks for your understanding and support!

Also big thanks to Ute Tobiasch for your reliability and scheduling all the appointments.

Thanks to my friends Horos Cuesdeanu, Alexander-Cornelius Heinrich and Manuel Salzberger for proofreading.

Finally I want to thank my colleagues Julian Manke and Manuel Linder whom I shared an office with. Thanks for your support, the long discussions and your help!

Symbols and Constants

Vacuum permittivity	ϵ_0	$8.85418781762 \cdot 10^{-12} \text{ As V}^{-1} \text{ m}^{-1}$
Vacuum permeability	μ_0	$2566370614 \cdot 10^{-6} \text{ N A}^{-2}$
Speed of Light	c	$299792458 \text{ m s}^{-1}$
Electron charge	q	$1.6021766208 \cdot 10^{-19} \text{ C}$
Electron mass	m_e	$9.10938356 \cdot 10^{-31} \text{ kg}$
Electrical flux density	\vec{D}	$[\text{A s m}^{-2}]$
Magnetic flux density	\vec{B}	$[\text{T}]$
Magnetic field strength	\vec{H}	$[\text{A m}^{-1}]$
Electric field strength	\vec{E}	$[\text{V m}^{-1}]$
Electromagnetic Field Tensor	F_ν^μ	
Poynting Vector	\vec{S}	$[\text{N m}^{-1} \text{ s}^{-1}]$
Position	\vec{r}	$[\text{m}]$
Velocity	\vec{v}	$[\text{m s}^{-1}]$
Normalized Velocity	$\vec{\beta} = \frac{\vec{v}}{c}$	$[]$
Normalized Acceleration	$\dot{\vec{\beta}}$	$[\text{s}^{-1}]$
Wavelength	λ	$[\text{m}]$
Wavevector	\vec{k}	$[\text{m}^{-1}]$
Force	\vec{F}	$[\text{N}]$
Mass	m	$[\text{kg}]$
Energy	E	$[\text{J}]$
Power	W	$[\text{W}]$
Time	t	$[\text{s}]$
discretized Time	h	$[\text{s}]$
Frequenzy	ω	$[\text{s}^{-1}]$
Nabla - Operator	$\vec{\nabla}$	$\left(\frac{\partial}{\partial r_1}, \dots, \frac{\partial}{\partial r_n} \right)$
Laplace - Operator	Δ	$\sum_{i=1}^n \frac{\partial^2}{\partial r_i^2}$
d'Alembert - Operator	$\hat{\square}$	$\Delta - \frac{1}{c^2} \frac{\partial^2}{\partial t^2}$

Contents

Acknowledgments	vii
I. Fundamentals	2
1. Electromagnetic radiation	4
1.1. Maxwell equations	4
1.2. Colomb and Lorentz gauge	5
1.3. Liénard-Wiechert potentials	6
1.3.1. Solution of the inhomogeneous wave equation	6
1.3.2. Special case: Moving point charge	7
1.3.3. Energy emission	10
II. Numerics	15
2. Integration of equation of motion	17
2.1. Equations of motion	17
2.2. Euler-scheme	18
2.3. Leap-Frog-scheme	21
2.4. Boris-method	22
3. Interpolations	25
3.1. Linear interpolation of trajectories	25
3.2. Trilinear interpolation of fields	27
4. Hybrid field approach	29
4.1. FDTD	29
4.2. Numeric dispersion relation	31
4.3. Near- and far fields	34
4.4. Field push	35
4.4.1. pushEFieldInsideBoxes()	39

CONTENTS

4.4.2. setHFieldOnBorders()	39
4.4.3. adjustHField()	39
4.4.4. pushEFieldOnBorders()	39
4.5. Particle push and near field update	41
4.6. Particle history	43
4.7. Far field setup before simulation	45
4.8. Electron scattering in an electromagnetic wave	48
5. Uniaxial Perfectly Matched Layer	64
5.1. Introduction	64
5.2. Derivation	65
5.3. Update equations	69
5.4. Implementation of UPML in FDTD	71
III. Summary & Outlook	73
IV. Appendix	75
A. Gauge Transformations	76
A.1. Gauge invariance of fields	76
A.2. Gauge invariance of potential equations	77
B. Retarded potential equations fulfill Laurentz gauge	78
C. Softwarestack and Documentation	80
D. Normalization	81
Bibliography	86
Declaration	88

Part I.

Fundamentals

Besides the numerical aspect of this thesis, the scattering of an electron in an electromagnetic wave also requires an understanding in electromagnetism. In this first part the theoretical basics this thesis is build upon are presented. Starting with the foundation of electromagnetism - the Maxwell equations - the corresponding potential equations are addressed afterwards. Introducing Lorentz gauge leads to the inhomogeneous wave equation, which is solved using Green's function approach. This leads to the Liénard-Wiechert potentials, which are a key aspect of this thesis. Lastly, a discussion about both, the non relativistic and relativistic phenomenon of energy radiation follows, where only the case is considered, in which a particle is accelerated parallel to its direction of motion.

Electromagnetic radiation

1.1 Maxwell equations

The Maxwell equations are the foundation of the classical electromagnetism theory. They describe how the electric field strength $\vec{E} \in \mathbb{R}^3$ and the magnetic field strength $\vec{H} \in \mathbb{R}^3$ are generated by charges and currents respectively and how they evolve over time in space in presence of one another. In presence of matter, however, the interaction of the fields with the material need to be taken into account. The effect of microscopic dipoles, formed by bound charge carriers, are summarized in macroscopic entities called polarisation $\vec{P} \in \mathbb{R}^3$ and magnetization $\vec{M} \in \mathbb{R}^3$. These dipoles align in the external field such that the resulting electric and magnetic field are described by

$$\begin{aligned}\vec{D} &:= \epsilon_0 \vec{E} + \vec{P}, \\ \vec{H} &:= \frac{1}{\mu_0} \vec{B} - \vec{M},\end{aligned}\tag{1.1}$$

where μ_0 and ϵ_0 are the vacuum permeability and permittivity respectively. The macroscopic Maxwell equations then read

$$\vec{\nabla} \cdot \vec{B} = 0,\tag{1.2}$$

$$\vec{\nabla} \cdot \vec{D} = \rho,\tag{1.3}$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t},\tag{1.4}$$

$$\vec{\nabla} \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{j},\tag{1.5}$$

where ρ denotes the charge density of the electric source and \vec{j} the electric current density. In the case of vacuum, the material equations (1.1) reduce to

$$\begin{aligned}\vec{D} &= \epsilon_0 \vec{E}, \\ \vec{H} &= \frac{1}{\mu_0} \vec{B}.\end{aligned}\tag{1.6}$$

1.2 Colom and Lorentz gauge

In the following section the Maxwell equations in vacuum are considered. The electric and magnetic fields can also be described by a scalar and a vector potential respectively. From (1.2) one can conclude that

$$\exists \vec{A} \in \mathbb{R}^3 : \vec{B} = \vec{\nabla} \times \vec{A}, \quad (1.7)$$

where \vec{A} is called the vector potential. Plugging in (1.7) into (1.4) yields

$$\begin{aligned} \vec{\nabla} \times \left(\vec{E} + \frac{\partial \vec{A}}{\partial t} \right) &= 0. \\ \Rightarrow \exists \varphi \in \mathbb{R} : \vec{E} + \frac{\partial \vec{A}}{\partial t} &= -\vec{\nabla} \varphi. \end{aligned} \quad (1.8)$$

φ is called scalar potential. Plugging in (1.8) into (1.3) and also (1.7) and (1.8) into (1.5) gives the potential equations

$$\begin{aligned} -\Delta \varphi - \vec{\nabla} \left(\frac{\partial \vec{A}}{\partial t} \right) &= \frac{\rho}{\epsilon_0}, \\ \underbrace{\left(\Delta - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) \vec{A} - \vec{\nabla} \left(\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} \right)}_{:= \hat{\square}} &= -\mu_0 \vec{j}, \end{aligned} \quad (1.9)$$

where the *Graßmann* identity: $\vec{\nabla} \times (\vec{\nabla} \times \vec{A}) = \vec{\nabla}(\vec{\nabla} \cdot \vec{A}) - \Delta \vec{A}$ is used.

The description of the fields by the aforementioned potentials φ and \vec{A} are not unique. This is called gauge freedom. The potentials can be specifically adjusted to the problem at hand. The gauge transformations look like

$$\begin{aligned} \vec{A} &\mapsto \vec{A}' = \vec{A} + \vec{\nabla} \psi, \\ \varphi &\mapsto \varphi' = \varphi - \frac{\partial \psi}{\partial t}, \end{aligned} \quad (1.10)$$

where $\psi: \mathbb{R}^3 \times \mathbb{R} \mapsto \mathbb{R}$. In Appendix A is shown that ψ does indeed not change the physics.

If one chooses ψ such that

$$\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} = 0, \quad (1.11)$$

then it is called a *Lorentz* gauge and the potential equations (1.9) decouple into two separate wave equations

$$\begin{aligned}\hat{\square}\varphi &= -\frac{\rho}{\epsilon_0}, \\ \hat{\square}\vec{A} &= -\mu_0\vec{j}.\end{aligned}\tag{1.12}$$

1.3 Liénard-Wiechert potentials

1.3.1 Solution of the inhomogeneous wave equation

The *Liénard-Wiechert* potentials are the solution of equation (1.12). Following [15] the general case

$$\hat{\square}\psi(\vec{r}, t) = f(\vec{r}, t),\tag{1.13}$$

is solved, where $\psi, f : \mathbb{R}^3 \times \mathbb{R} \mapsto \mathbb{R}$. The function f is called source function and will be specified later. Equation (1.13) is a linear inhomogeneous partial differential equation. Those kind of equations are solved with the *Green's* function approach. If one can find the Green's function G fulfilling

$$\hat{\square}G(\vec{r} - \vec{r}', t - t') = -\delta(\vec{r} - \vec{r}')\delta(t - t'),\tag{1.14}$$

then one can calculate the solution of (1.13) via

$$\psi(\vec{r}, t) = \int \int G(\vec{r} - \vec{r}', t - t')f(\vec{r}, t)d\vec{r}' dt'.\tag{1.15}$$

To find G one uses the following Fourier transformations

$$\begin{aligned}G(\vec{r} - \vec{r}', t - t') &= \frac{1}{4\pi^2} \int \int G(\vec{k}, \omega) \exp(i\vec{k}(\vec{r} - \vec{r}')) \exp(-i\omega(t - t')) d\omega d\vec{k}, \\ \delta(\vec{r} - \vec{r}') &= \frac{1}{4\pi^2} \int \exp(i\vec{k}(\vec{r} - \vec{r}')) d\vec{k}, \\ \delta(t - t') &= \frac{1}{4\pi^2} \int \exp(-i\omega(t - t')) d\omega.\end{aligned}\tag{1.16}$$

Plugging in (1.16) into (1.14) yields

$$\int \int \exp(i\vec{k}(\vec{r} - \vec{r}')) \exp(-i\omega(t - t')) \left[G(\vec{k}, \omega) \left(-k^2 + \frac{\omega^2}{c^2} \right) + \frac{1}{4\pi^2} \right] = 0,$$

which determines G to be

$$G(\vec{k}, \omega) = \frac{1}{4\pi^2} \frac{1}{k^2 - \frac{\omega^2}{c^2}}.$$

Using residue theorem one finds the Fourier back transform of G to be

$$G_{ret}(\vec{r} - \vec{r}', t - t') = \frac{\delta(t' - t_{ret})}{4\pi |\vec{r} - \vec{r}'|}, \quad (1.17)$$

where

$$t_{ret} := t - \frac{|\vec{r} - \vec{r}'|}{c} \quad (1.18)$$

is the retarded time. G_{ret} is also called the *retarded* Green's function in contrast to the *advanced* Green's function G_{av} with the advanced time

$$t_{av} := t + \frac{|\vec{r} - \vec{r}'|}{c}. \quad (1.19)$$

The latter does not fulfill the causal principle of physics. One expects the reason for a change in a signal at \vec{r} at time t , due to a perturbation, to be in the past, not in the future. Because $t_{ret} < t$ the retarded Green's function is the one to work with. Plugging in (1.17) into (1.15) yields

$$\psi(\vec{r}, t) = \frac{1}{4\pi} \int \frac{f(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}', \quad (1.20)$$

which finally leads to the retarded potential equations

$$\varphi(\vec{r}, t) = \frac{1}{4\pi\epsilon_0} \int \frac{\rho(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}', \quad (1.21)$$

$$\vec{A}(\vec{r}, t) = \frac{\mu_0}{4\pi} \int \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}'. \quad (1.22)$$

Lastly, in Appendix B is shown that they indeed fulfill the Laurentz gauge (1.11).

1.3.2 Special case: Moving point charge

Now the special case of a particle with charge q , moving along the trajectory \vec{r}_p with velocity $\vec{v} := \dot{\vec{r}}_p$, where the dot donates the time derivative, is discussed. As mentioned

before, the source functions for the problem at hand are defined. For the charge density one has

$$\rho(\vec{r}, t) = q\delta(\vec{r} - \vec{r}_p(t)) \quad (1.23)$$

and for the current density

$$\vec{j}(\vec{r}, t) = q\vec{v}\delta(\vec{r} - \vec{r}_p(t)). \quad (1.24)$$

Identifying

$$\begin{aligned} \psi(\vec{r}, t) &= \varphi(\vec{r}, t) \text{ and } f(\vec{r}', t') = \frac{\rho(\vec{r}', t')}{\epsilon_0}, \\ \vec{\psi}(\vec{r}, t) &= \vec{A}(\vec{r}, t) \text{ and } \vec{f}(\vec{r}', t') = \mu_0 \vec{j}(\vec{r}', t') \end{aligned}$$

and using (1.15) with (1.17) yields after integrating over \vec{r}'

$$\begin{aligned} \varphi(\vec{r}, t) &= \frac{1}{4\pi\epsilon_0} \int \frac{\delta\left(\frac{|\vec{r}-\vec{r}_p|}{c} - t + t'\right)}{|\vec{r} - \vec{r}_p|} dt', \\ \vec{A}(\vec{r}, t) &= \frac{q\mu_0}{4\pi} \int \vec{v}(t') \frac{\delta\left(\frac{|\vec{r}-\vec{r}_p|}{c} - t + t'\right)}{|\vec{r} - \vec{r}_p|} dt'. \end{aligned} \quad (1.25)$$

For the t' - integration the following property of the δ - distribution is needed

$$\delta[g(t')] = \sum_{i=1}^n \frac{\delta(t' - t_i)}{\left| \left(\frac{dg}{dt'} \right) \Big|_{t'=t_i} \right|}, \quad (1.26)$$

where

$$g(t') := \frac{|\vec{r} - \vec{r}_p|}{c} - t + t', \quad (1.27)$$

and t_i are the roots of g . To find the roots of g , consider

$$\begin{aligned} \frac{dg}{dt'} &= 1 + \frac{1}{c} \frac{d}{dt'} |\vec{r} - \vec{r}_p| = 1 - \frac{\vec{v}(t')}{c} \underbrace{\frac{\vec{r} - \vec{r}_p}{|\vec{r} - \vec{r}_p|}}_{|\cdot|=1} \\ &\implies 1 - \frac{\vec{v}(t')}{c} \leq \frac{dg}{dt'} \leq 1 + \frac{\vec{v}(t')}{c} \\ &\stackrel{v < c}{\implies} \frac{dg}{dt'} > 0, \end{aligned} \quad (1.28)$$

which means that g is a monotonically increasing function and therefore has at most one root, which is $t' = t_{ret}$ as can be seen from equation (1.27). In case that g has no roots at all, one can conclude that $\varphi \equiv 0$ and $\vec{A} \equiv 0$.

With $t_{ret} := t - \frac{|\vec{r} - \vec{r}_p|}{c}$ the integration over t' yields

$$\begin{aligned}\varphi(\vec{r}, t) &= \frac{1}{4\pi\epsilon_0} \frac{q}{|\vec{r} - \vec{r}_p(t_{ret})| - \frac{\vec{v}(t_{ret})}{c}(\vec{r} - \vec{r}_p(t_{ret}))}, \\ \vec{A}(\vec{r}, t) &= \frac{1}{4\pi} \frac{\mu_0 q \vec{v}(t_{ret})}{|\vec{r} - \vec{r}_p(t_{ret})| - \frac{\vec{v}(t_{ret})}{c}(\vec{r} - \vec{r}_p(t_{ret}))},\end{aligned}\quad (1.29)$$

which are the *Liénard-Wiechert* potentials. Upon defining

$$\begin{aligned}\vec{\beta}(t) &:= \frac{\vec{v}(t)}{c}, \\ \vec{R}(\vec{r}, t) &:= \vec{r} - \vec{r}_p(t), \\ \vec{n}(\vec{r}, t) &:= \frac{\vec{R}(\vec{r}, t)}{R(\vec{r}, t)},\end{aligned}\quad (1.30)$$

one can rewrite the Liénard-Wiechert potentials in a more compact and final form

$$\begin{aligned}\varphi(\vec{r}, t) &= \frac{1}{4\pi\epsilon_0} \frac{q}{(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t')) R(\vec{r}, t')} \Big|_{t'=t_{ret}}, \\ \vec{A}(\vec{r}, t) &= \frac{\mu_0}{4\pi} \frac{q \vec{\beta}(t')}{(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t')) R(\vec{r}, t')} \Big|_{t'=t_{ret}}.\end{aligned}\quad (1.31)$$

The latter potentials are the generalization of the Coulomb potential. With the help of (1.8) the electric and magnetic fields can be derived from the *Liénard-Wiechert* potentials (1.31). Since this is a quite lengthy calculation it is referred to [15] and the final result is just presented

$$\begin{aligned}\vec{E}(\vec{r}, t) &= \frac{q}{4\pi\epsilon_0} \left(\frac{\vec{n}(\vec{r}, t') - \vec{\beta}(t)}{\gamma^2 (1 - \vec{\beta}(t) \cdot \vec{n}(\vec{r}, t'))^3 R^2(\vec{r}, t')} \right. \\ &\quad \left. + \frac{1}{c} \frac{\vec{n}(\vec{r}, t') \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t)) \times \dot{\vec{\beta}}(t)}{(1 - \vec{\beta}(t) \cdot \vec{n}(\vec{r}, t'))^3 R(\vec{r}, t')} \right) \Big|_{t'=t_{ret}}, \\ \vec{B}(\vec{r}, t) &= \frac{1}{c} \left(\vec{n}(\vec{r}, t') \times \vec{E}(\vec{r}, t) \right) \Big|_{t'=t_{ret}}.\end{aligned}\quad (1.32)$$

This is the electric and magnetic field of a moving charged particle. Due to the finite speed of light, the fields need time to travel from their source to the observation point. That means that the fields at time t , were actually produced at an earlier time t_{ret} . As can be seen in Figure 3.1, the retarded time is determined by the intersection point of the particle trajectory with the backward lightcone of the observation point.

1.3.3 Energy emission

Now the energy flow and emission is discussed. Therefore, the Poynting vector needs to be analyzed

$$\vec{S} = \frac{1}{\mu_0} \vec{E} \times \vec{B} = \frac{1}{\mu_0 c} \left[\vec{n}_{ret} E^2 - (\vec{n}_{ret} \vec{E}) \vec{E} \right], \quad (1.33)$$

where $\vec{n}_{ret} := \vec{n}(\vec{r}, t_{ret})$. Taking a closer look at (1.32) reveals that both \vec{E} and \vec{B} can be separated into two parts, one of which is not dependent on the particle acceleration

$$\begin{aligned} \vec{E}_{(v)}(\vec{r}, t) &= \frac{q}{4\pi\epsilon_0} \frac{1}{\gamma^2 (1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t'))^3 R^2(\vec{r}, t')} \left. \left(\vec{n}(\vec{r}, t') - \vec{\beta}(t') \right) \right|_{t' = t_{ret}}, \\ \vec{B}_{(v)}(\vec{r}, t) &= \frac{q}{4\pi\epsilon_0 c} \frac{1}{\gamma^2 (1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t'))^3 R^2(\vec{r}, t')} \underbrace{\left[\vec{n} \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t')) \right]}_{= \frac{1}{c} (\vec{v} \times \vec{n})} \Big|_{t' = t_{ret}}, \end{aligned} \quad (1.34)$$

whilst the other is through $\dot{\vec{\beta}}$

$$\begin{aligned} \vec{E}_{(a)}(\vec{r}, t) &= \frac{q}{4\pi\epsilon_0 c} \frac{\vec{n}(\vec{r}, t') \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t')) \times \dot{\vec{\beta}}(t')}{\left(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t') \right)^3 R(\vec{r}, t')} \Big|_{t' = t_{ret}}, \\ \vec{B}_{(a)}(\vec{r}, t) &= \frac{q}{4\pi\epsilon_0 c^2} \frac{\vec{n}(\vec{r}, t') \times [\vec{n}(\vec{r}, t') \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t')) \times \dot{\vec{\beta}}(t')]}{\left(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t') \right)^3 R(\vec{r}, t')} \Big|_{t' = t_{ret}}. \end{aligned} \quad (1.35)$$

Since $E_{(v)} \propto \frac{1}{R^2}$ whereas $E_{(a)} \propto \frac{1}{R}$, the field contributions of $E_{(a)}$ will dominate in the far field, i.e. for large distances from the source. Plugging in (1.32) in (1.33) yields several

terms with different orders of $\frac{1}{R}$, namely

$$\begin{aligned}\vec{S}_{vv} &\propto \left| \vec{E}_{(v)} \times \vec{B}_{(v)} \right| \propto \frac{1}{R^4}, \\ \vec{S}_{va} &\propto \left| \vec{E}_{(v)} \times \vec{B}_{(a)} \right| \propto \frac{1}{R^3}, \\ \vec{S}_{aa} &\propto \left| \vec{E}_{(a)} \times \vec{B}_{(a)} \right| \propto \frac{1}{R^2}.\end{aligned}$$

By evaluating the energy flux through an arbitrary surface with differential surface area $d\vec{\sigma}$, as illustrated in Figure 1.1, one can see that it will suffice to only consider the $\frac{1}{R^2}$ terms, because the $\frac{1}{R^3}$ or higher terms do not contribute to the energy emission, since

$$\oint \vec{S} d\vec{\sigma} \rightsquigarrow \oint \frac{1}{R^3} r^2 d\Omega \rightsquigarrow \oint \frac{1}{r} d\Omega \rightsquigarrow \frac{1}{r} \xrightarrow{r \rightarrow \infty} 0, \quad (1.36)$$

where $d\Omega := \sin(\vartheta) d\vartheta d\varphi$.

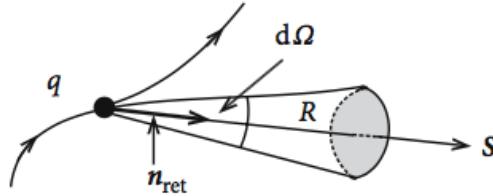


Figure 1.1.: Illustration of how to calculate the energy emission of a moving particle with charge q [15].

That means one can focus on equation (1.35) when calculating (1.33)

$$\begin{aligned}\vec{S} = \frac{1}{\mu_0 c} \frac{q^2}{16\pi^2 \epsilon_0^2 c^2} & \left. \frac{\vec{n}(\vec{r}, t') \left[\vec{n}(\vec{r}, t') \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t')) \times \dot{\vec{\beta}}(t') \right]^2}{(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t'))^6 R^2(\vec{r}, t')} \right|_{t' = t_{ret}} \\ & + \mathcal{O}\left(\frac{1}{R^3}\right). \quad (1.37)\end{aligned}$$

The energy flux has the direction of \vec{n}_{ret} , i.e. it flows from the particle position \vec{r}_p at time $t' = t_{ret}$ to the observation point \vec{r} . It can also be seen that only accelerated particles

$(\dot{\vec{\beta}} \neq 0)$ loose energy through radiation, which leads to a reduction in kinetic energy. Although an uniformly moving particle produces E and B fields, it does not loose energy through radiation.

Lastly, the radiation power is discussed. The emitted energy per time dt into the solid angle $d\Omega$ is described by

$$\frac{dW}{d\Omega} = (\vec{S} \cdot \vec{n}(\vec{r}, t')) R^2(\vec{r}, t') \Big|_{t'=t_{ret}}. \quad (1.38)$$

Usually the observed amount of energy per unit time dt is not the same as the emitted energy of the particle per unit time dt_{ret} . To learn more about the emission within the particles system, equation (1.38) needs to be expressed in terms of dt_{ret} , which yields

$$\frac{dW}{d\Omega} = (\vec{S} \cdot \vec{n}(\vec{r}, t')) R^2(\vec{r}, t') \left(\frac{dt}{dt'} \right) \Big|_{t'=t_{ret}}. \quad (1.39)$$

From the definition of $t_{ret} := t' - \frac{|\vec{r} - \vec{r}'|}{c}$ one finds

$$\left(\frac{dt}{dt'} \right) \Big|_{t'=t_{ret}} = \left(1 + \frac{1}{c} \frac{d}{dt'} |\vec{r} - \vec{r}_p(t')| \right) \Big|_{t'=t_{ret}} = \left(1 - \frac{1}{c} \vec{n}(\vec{r}, t') \vec{v}(t') \right) \Big|_{t'=t_{ret}}. \quad (1.40)$$

Finally, one obtains

$$\frac{dW}{d\Omega} = \frac{q^2}{16\pi^2\epsilon_0 c} \frac{\left[\vec{n}(\vec{r}, t') \times (\vec{n}(\vec{r}, t') - \vec{\beta}(t')) \times \dot{\vec{\beta}}(t') \right]^2}{\left(1 - \vec{\beta}(t') \cdot \vec{n}(\vec{r}, t') \right)^5} \Big|_{t'=t_{ret}}. \quad (1.41)$$

Now two cases are discussed. The non relativistic case ($\beta \approx 0$) and the relativistic case ($\beta \approx 1$).

$\beta \approx 0$: In this case (1.41) simplifies to

$$\begin{aligned}
 \frac{dW}{d\Omega} &= \frac{q^2}{16\pi^2\epsilon_0 c} \left[\vec{n}(\vec{r}, t') \times \left(\vec{n}(\vec{r}, t') \times \dot{\vec{\beta}}(t') \right) \right]^2 \Big|_{t'=t_{ret}} \\
 &= \frac{q^2}{16\pi^2\epsilon_0 c} \left[\underbrace{\left(\vec{n}(\vec{r}, t') \cdot \dot{\vec{\beta}}(t') \right) \vec{n}(\vec{r}, t')}_{=\dot{\beta}n \cos(\vartheta)} - \underbrace{\left(\vec{n}(\vec{r}, t') \cdot \vec{n}(\vec{r}, t') \right) \dot{\vec{\beta}}(t')}_{|\cdot|=1} \right]^2 \Big|_{t'=t_{ret}} \\
 &= \frac{q^2 \dot{\beta}_{ret}^2}{16\pi^2\epsilon_0 c} (\cos^2(\vartheta) - 2\cos^2(\vartheta) + 1) \\
 &= \frac{q^2 \dot{\beta}_{ret}^2}{16\pi^2\epsilon_0 c} \sin^2(\vartheta),
 \end{aligned}$$

where ϑ is the angle between $\dot{\vec{\beta}}$ and \vec{n} . The subindex in β_{ret} means again the evaluation at $t' = t_{ret}$. One concludes that the direction of maximum radiation is perpendicular to the particles direction of motion, in contrast to

$\beta \approx 1$: where one will see shortly that the maximum radiation will be primarily in the direction of motion of the particle. In the following only the simple case, where $\dot{\vec{\beta}} \parallel \vec{\beta}$, is conducted. Following the same steps as above one obtains

$$\frac{dW}{d\Omega} = \frac{q^2 \dot{\beta}_{ret}^2}{16\pi^2\epsilon_0 c} \frac{\sin^2(\vartheta)}{(1 - \beta(t') \cos(\vartheta))^5}. \quad (1.42)$$

To get the angle, where the radiation is maximal, one calculates

$$\begin{aligned}
 \frac{d}{d(\cos(\vartheta))} \left(\frac{dW}{d\Omega} \right) &\stackrel{!}{=} 0 \\
 \iff -2\cos(\vartheta)(1 - \beta_{ret} \cos(\vartheta))^5 + 5\beta_{ret}(1 - \cos^2(\vartheta))(1 - \beta_{ret} \cos(\vartheta))^4 &\stackrel{!}{=} 0 \\
 \iff (1 - \beta_{ret} \cos(\vartheta))^4 [-2\cos(\vartheta)(1 - \beta_{ret} \cos(\vartheta)) + 5\beta_{ret}(1 - \cos^2(\vartheta))] &\stackrel{!}{=} 0 \\
 \iff -2\cos(\vartheta)(1 - \beta_{ret} \cos(\vartheta)) + 5\beta_{ret}(1 - \cos^2(\vartheta)) &\stackrel{!}{=} 0 \\
 \iff \cos^2(\vartheta) + \frac{2}{3\beta_{ret}} \cos(\vartheta) - \frac{5}{3} &\stackrel{!}{=} 0 \\
 \implies \cos(\vartheta)_{max} &= \frac{1}{3\beta_{ret}} \left(\sqrt{1 + 15\beta_{ret}^2} - 1 \right).
 \end{aligned}$$

Obviously the angle decreases monotonically with increasing velocity, i.e. increasing β_{ret} . $\beta_{ret} = 1$ yields $\cos(\vartheta) = 1$ or equivalently $\vartheta = 0$. In Figure 1.2 the results are summarized.

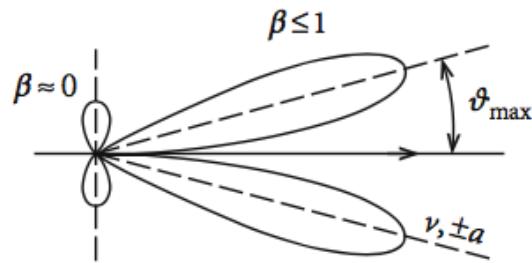


Figure 1.2.: Radiation characteristics of a moving charged particle [15].

Part II.

Numerics

The following section deals with the numeric aspects of this thesis. An explanation of the underlying equations of motion and their history is given. After that several methods are presented with which one can solve differential equations. Over the course of the last decades numerous methods were invented each of which has its own strengths and weaknesses. Some of them are very easy to implement, like the *Euler* scheme, since it only uses first order derivation to approximate the next position. This usually leads to unprecise results, especially when the solution of the differential equation changes rapidly. Then the approximated solution by the solver and the actual solution differ drastically. Others are quite complicated and sophisticated to implement, like the *Runge Kutta* 4th order scheme, since it uses several derivations at different time steps to approximate the next position. This makes Runge Kutta very accurate, though. Following this, the numerical complexity is defined and calculated for some chosen algorithms.

2

Integration of equation of motion

2.1 Equations of motion

As is known from mechanics the dynamics of particles is determined by the forces acting on them. In case of an electron scattering process there is a force due to electromagnetic fields. That can be external fields, but also fields due to moving particles, as explained in Section 1.3.

The dynamics of our system is described by the *Lorentz-Newton* equations

$$\begin{aligned} \frac{dx^\mu}{d\tau} &= u^\mu, \\ \frac{du^\mu}{d\tau} &= F_\nu^\mu u^\nu + g^\mu. \end{aligned} \quad (2.1)$$

The term F_ν^μ describes the electromagnetic field strength tensor

$$F_\nu^\mu = \begin{pmatrix} 0 & E_x & E_y & E_z \\ E_x & 0 & B_z & -B_y \\ E_y & -B_z & 0 & B_x \\ E_z & B_y & -B_x & 0 \end{pmatrix}. \quad (2.2)$$

The damping term g^μ considers the fact that charged particles radiate fields when they are moving, which leads to a loss in their kinetic energy. Within the context of classical electrodynamics Max *Abraham* and Hendrick *Lorentz* discussed radiation damping in their same-named equation first. In 1938 *Dirac* generalized the equation whilst taking special relativity into account [9]. It turned out, however, that this equation has unphysical solutions, which violate the causality principle. *Landau* and *Lifschitz* solved this problem using perturbation theory [14]. The Landau-Lifschitz equation reads

$$\begin{aligned} g^\mu = \frac{\mu_0 q^2}{6\pi c} &\left(\frac{q}{m_e} \frac{\partial F^{\mu\nu}}{\partial x^\xi} u^\nu u^\xi - \frac{e^2}{m_e^2} F^{\mu\xi} F_{\nu\xi} u^\nu \right. \\ &\left. + \frac{q^2}{m_e^2 c^2} (F_{\nu\xi} u^\xi) (F^{\nu\pi} u_\pi) u^\mu \right). \end{aligned} \quad (2.3)$$

In this thesis the damping term g^μ is neglected. In [12] a rough estimation shows that radiation damping of an electron gyrating in a constant B field with frequency ω_0 can be neglected if

$$\omega_0 \Delta \tilde{t} \ll 1 \quad (2.4)$$

holds. The parameter $\Delta \tilde{t} = \frac{q^2}{4\pi\epsilon_0 m_e c^3} = 9.40 \cdot 10^{-24}$ s is the normalized time. In Section 4.8 is shown that equation (2.4) is indeed fulfilled in the simulations of the present work. In the following different methods are presented of how to solve the Lorentz-Newton equation (2.1) numerically.

2.2 Euler-scheme

The most simple method is the explicit *Euler*-method. It's easy to implement but not very accurate, as can be seen later. But before going into the details of the explicit Euler-scheme prerequisites are addressed, which all following methods have in common. Starting point will always be a first order system of the kind

$$\begin{aligned} \frac{dx^\mu}{d\tau} &= u^\mu, \\ \frac{du^\mu}{d\tau} &= f^\mu(x^\nu, u^\nu), \\ x^\mu(\tau_0) &= x_0^\mu, \\ u^\mu(\tau_0) &= u_0^\mu. \end{aligned} \quad (2.5)$$

Systems of higher order can always be reduced to a first order system.

In order to solve the equation of motion numerically the domain needs to be discretized. Therefore, the time interval is divided into N equidistant partial intervals h , by defining

$$h := \Delta\tau = \tau_{i+1} - \tau_i.$$

The idea is to calculate each point along the trajectory $x_i^\mu = x^\mu(\tau_i)$ iteratively, starting from the initial values x_0^μ and u_0^μ . But to calculate these points all differential operators in (2.5) need to be discretized as well. That is, where all methods differ. Each method has its own way to discretize the differential operators.

The basis of the Euler-scheme is a first order Taylor expansion of the integration variable x^μ in τ around τ_i

$$x^\mu(\tau_{i+1}) = x^\mu(\tau_i) + \left. \frac{dx^\mu}{d\tau} \right|_{\tau=\tau_i} \underbrace{(\tau_{i+1} - \tau_i)}_{=h} + \mathcal{O}(h^2). \quad (2.6)$$

Analogously for u^μ and solving for $\frac{dx^\mu}{d\tau}$ and $\frac{du^\mu}{d\tau}$ respectively yields

$$\begin{aligned} \frac{x_{i+1}^\mu - x_i^\mu}{h} &= u_i^\mu, \\ \frac{u_{i+1}^\mu - u_i^\mu}{h} &= f^\mu(x_i^\nu, u_i^\nu). \end{aligned} \quad (2.7)$$

This way of discretizing allows a very easy calculation of x_i^μ according to

$$\begin{aligned} x_{i+1}^\mu &= x_i^\mu + h u_i^\mu, \\ u_{i+1}^\mu &= u_i^\mu + h f^\mu(x_i^\nu, u_i^\nu). \end{aligned} \quad (2.8)$$

In order to calculate the goodness of this approximation one needs to introduce the *procedural error* and the *order of consistency* [13].

Definition 2.2.1 (Procedural error and order of consistency) Let $I \subseteq \mathbb{R}$ be an interval, $f : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, $y : I \rightarrow \mathbb{R}^d$ a solution of the initial value problem

$$\begin{aligned} \frac{d}{d\tau} y(\tau) &= f(\tau, y(\tau)), \\ y(\tau_0) &= y_0. \end{aligned} \quad (2.9)$$

(a) The term

$$\eta(\tau, h) := y(\tau) + h f(\tau, y(\tau)) - y(\tau + h) \quad \text{for } \tau \in I, 0 < h \leq b - \tau \quad (2.10)$$

is called local procedural error of the one-step-scheme at τ for the increment h .

(b) The one-step-scheme has an order of consistency $p \geq 1$, if the local procedural error fulfills

$$||\eta(\tau, h)|| \leq Ch^{p+1} \quad \text{for } \tau \in I, 0 < h \leq b - \tau, \quad (2.11)$$

with a constant $C \geq 0$, which is independent of τ and h .

Descriptively, the procedural error is the difference between the exact solution $y(\tau + h)$ and the result, which one gets from the one-step-scheme starting from the exact solution at the earlier time step $y(\tau)$. Figure 2.1 illustrates the situation.

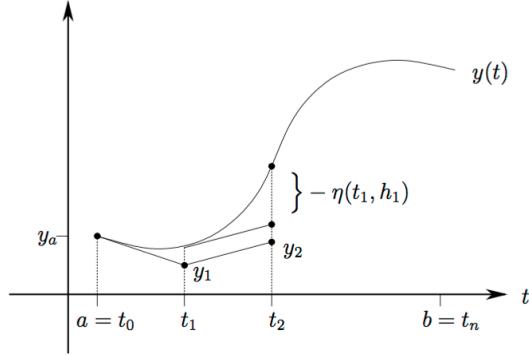


Figure 2.1.: Illustration of the procedural errors of an one-step-scheme [13].

The definitions 2.2.1 is now used to calculate the order of consistency of the Euler-scheme. Starting point is the system (2.1). Thereby, the focus is on the equation for u^μ since x^μ can be easily integrated from u^μ . Following definition 2.2.1 yields

$$y = u^\mu. \quad (2.12)$$

One gets

$$\eta(\tau, h) = u^\mu(\tau_i) + h f^\mu(x^\nu, u^\nu) - u^\mu(\tau_{i+1}). \quad (2.13)$$

The last term can be calculated with a Taylor-expansion analogously to (2.6)

$$u^\mu(\tau_{i+1}) = u^\mu(\tau_i) + h \left. \frac{du^\mu}{d\tau} \right|_{s=\tau_i} + h^2 \left. \frac{d^2u^\mu}{d\tau^2} \right|_{s=\tau_i}. \quad (2.14)$$

Plugging in (2.14) in (2.13) yields

$$\begin{aligned} \eta(\tau, h) &= u^\mu(\tau_i) + h \frac{du^\mu}{d\tau} - u^\mu(\tau_{i+1}) \\ \stackrel{(2.14)}{\implies} \eta(\tau, h) &= u^\mu(\tau_i) + h \frac{du^\mu}{d\tau} - u^\mu(\tau_i) - h \frac{du^\mu}{d\tau} - h^2 \frac{d^2u^\mu}{d\tau^2} \\ \iff \eta(\tau, h) &= \frac{d^2u^\mu}{d\tau^2} h^2, \end{aligned} \quad (2.15)$$

since $\frac{du^\mu}{d\tau} = f^\mu(x^\nu, u^\nu)$ holds for the Euler-scheme. Thus,

$$|\eta(\tau, h)| \leq Ch^2 \quad \text{with } C := \frac{1}{2} \max_{\tau \in \mathcal{D}(u^\mu)} \left| \frac{d^2 u^\mu}{d\tau^2} \right|. \quad (2.16)$$

$\mathcal{D}(u^\mu)$ denotes the domain of u^μ . Therefore, the Euler-scheme has an order of consistency of one which is the lowest possible value. The higher the order of consistency is, the better the algorithm.

2.3 Leap-Frog-scheme

A definitely better method is the so called *Leap-Frog*-scheme. One can easily proof that it has an order of consistency of two. In contrast to the explicit Euler-scheme this method has several advantages. For one, it is time reversible, i.e. it is possible to reach any previous point in time from every point later in the trajectory. On the other hand, the Leap-Frog-scheme is symplectic, meaning it conserves the phase space volume from which energy and momentum conservation follows. However, one disadvantage is that it's only suited for systems in which the acting force exclusively depends on the current position but not on the velocity of the particle. This would lead to an implicit equation system which is numerically way more expensive to solve. Thus, the differential equation should be of the form

$$\frac{d^2 x^\mu}{d\tau^2} = \frac{du^\mu}{d\tau} = f^\mu(x^\nu). \quad (2.17)$$

As already mentioned, the various methods discretize the differential operators differently. The Leap-Frog-scheme uses

$$\begin{aligned} \frac{x_{i+1}^\mu - x_i^\mu}{h} &= u_{i+\frac{1}{2}}^\mu, \\ \frac{u_{i+\frac{1}{2}}^\mu - u_{i-\frac{1}{2}}^\mu}{h} &= f^\mu(x_i^\nu). \end{aligned} \quad (2.18)$$

Solving for the new time step yields

$$\begin{aligned} x_{i+1}^\mu &= x_i^\mu + hu_{i+\frac{1}{2}}^\mu, \\ u_{i+\frac{1}{2}}^\mu &= u_{i-\frac{1}{2}}^\mu + hf^\mu(x_i^\nu). \end{aligned} \quad (2.19)$$

As one can see, position and velocity are calculated at different times. They are shifted against each other in time by $h = \frac{1}{2}$. But what if one has a system in which the force depends on the velocity? Is one stuck with expensive implicit methods? Fortunately not. One can use the *Boris*-method.

2.4 Boris-method

This method was invented in 1970 by J.P. Boris [4] and is the standard method for pushing particles in plasma simulations today. Goal is to solve the Lorentz-Newton equation

$$\frac{v_{i+\frac{1}{2}} - v_{i-\frac{1}{2}}}{h} = \frac{q}{m} \left(\vec{E} + \frac{v_{i+\frac{1}{2}} + v_{i-\frac{1}{2}}}{2} \times \vec{B} \right). \quad (2.20)$$

Boris noticed that upon defining

$$\begin{aligned} \vec{v}_- &:= v_{i-\frac{1}{2}} + \frac{h}{2} \frac{q\vec{E}}{m} \quad \text{and} \\ \vec{v}_+ &:= v_{i+\frac{1}{2}} + \frac{h}{2} \frac{q\vec{E}}{m}, \end{aligned} \quad (2.21)$$

one can eliminate the electric field. Plugging in (2.21) into (2.20) yields

$$\begin{aligned} \frac{\vec{v}_+ - \vec{v}_-}{h} &= \frac{q}{m} (\vec{v}_+ + \vec{v}_-) \times \vec{B} \\ \implies |\vec{v}_+ - \vec{v}_-| &= \frac{qh}{m} |\vec{v}_+ + \vec{v}_-| B \underbrace{\sin(\vartheta)}_{=1}, \end{aligned} \quad (2.22)$$

where ϑ is the angle between \vec{B} and \vec{v} . By splitting up the accelerating effect of the electric field in two parts with the rotational effect of the magnetic field in between, the accuracy is increased without having computational overhead. Hence, the steps are

1. Obtain \vec{v}_- by starting from $v_{i-\frac{1}{2}}$ and adding half electric impulse.
2. Use rotation in (2.22) to obtain \vec{v}_+ and finally
3. obtain $v_{i+\frac{1}{2}}$ by adding yet another half electric impulse.

In the code used in the present work the slightly more sophisticated but relativistically correct form with $u = \gamma v$ is used. Up to now one question remains unanswered. How does one solve for \vec{v}_+ in (2.22)? To answer this consider Figure 2.2.

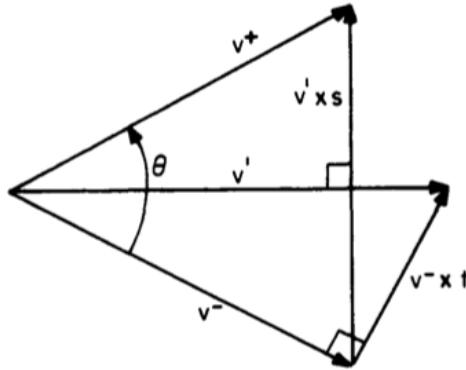


Figure 2.2.: Velocity space showing the rotation from \vec{v}_- to \vec{v}' . The velocities shown are projections of the total velocities onto the plane perpendicular to \vec{B} [5].

Goal is to find a vector which is parallel to $\vec{v}_+ - \vec{v}_-$. Its magnitude is yet to be determined. Consider the vector \vec{v}' which is perpendicular to $\vec{v}_+ - \vec{v}_-$ then $\vec{v}' \times \vec{s}$, where $\vec{s} \propto \vec{B}$, is the desired vector. Upon defining

$$\vec{t} := -\frac{\vec{B}}{B} \tan\left(\frac{\theta}{2}\right) = \frac{q\vec{B}h}{2m} \quad (2.23)$$

one obtains \vec{v}' from

$$\vec{v}' = \vec{v}_- + \vec{v}_- \times \vec{t}. \quad (2.24)$$

Now, \vec{v}' is perpendicular to $\vec{v}_+ - \vec{v}_-$. Finally, a vector $\vec{s} \propto \vec{B}$ is needed which magnitude is determined by the constraint $|\vec{v}_+| = |\vec{v}_-|$. Using half angle formulas one finds

$$\vec{s} = \frac{2\vec{t}}{1+t^2}. \quad (2.25)$$

Hence,

$$\vec{v}_+ = \vec{v}_- + \vec{v}' \times \vec{s}. \quad (2.26)$$

Finally, a quick error analysis is conducted. The angle of rotation θ is expected to be close to

$$\omega_c h = \frac{2\pi h}{T} = \frac{qBh}{m},$$

where T is the period.

Following the geometrical analysis in [5], illustrated in 2.3, and using equation (2.22) one finds

$$\begin{aligned} \left| \tan\left(\frac{\theta}{2}\right) \right| &= \frac{\frac{1}{2}|\vec{v}_+ - \vec{v}_-|}{\frac{1}{2}|\vec{v}_+ + \vec{v}_-|} = \frac{qB}{m} \frac{h}{2} = \frac{\omega_c h}{2} \\ \Rightarrow \theta &= 2 \arctan\left(\frac{\omega_c h}{2}\right) = \omega_c h \left(1 - \frac{\omega_c^2 h^2}{12} + \dots\right). \end{aligned} \quad (2.27)$$

For $\omega_c h < 0.35$ the error is smaller than 1%.

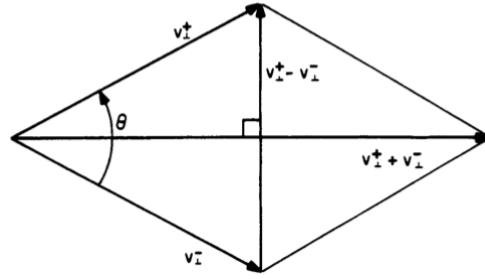


Figure 2.3.: Determination of $|\tan\left(\frac{\theta}{2}\right)|$ from boris rotation [5].

To sum it up, equation (1.32) allows the calculation of Liénard-Wiechert fields at a given observation point. With the Boris-scheme an effective and precise pushing of particles in an electromagnetic field is possible. The next step is to combine those two and present a way of how to set up a simulation.

3

Interpolations

3.1 Linear interpolation of trajectories

The methods, previously presented, calculate the particle trajectory solely at discrete points in time $x_i^\mu(\tau)$. Calculating Liénard-Wiechert fields according to equation (1.32), however, requires the intersection point of the trajectory with the backward lightcone of the observation point. In most cases the calculated points of the trajectory are not lying directly on the lightcone, so a procedure to calculate the intersection point exactly is required. The simplest solution is a linear interpolation between the last point inside and the first point outside the lightcone. Figure 3.1 illustrates the situation. Thereto, let $x_j^\mu \in \mathbb{R}^{3+1}$ be the last point inside and $x_{j+1}^\mu \in \mathbb{R}^{3+1}$ the first point outside the lightcone. Further let $x_c^\mu \in \mathbb{R}^{3+1}$ be the intersection point of interest then one gets

$$x_c^\mu = x_j^\mu + \lambda \left(x_{j+1}^\mu - x_j^\mu \right), \quad (3.1)$$

where $\lambda \in [0, 1]$. Due to the finite speed of light the intersection point x_c^μ needs to fulfill

$$|\vec{x}_o(t) - \vec{x}_c(t_{ret})| = c (t - t_{ret}) \iff (x_o - x_c)_\mu (x_o - x_c)^\mu = 0. \quad (3.2)$$

Thereby, $x_o^\mu \in \mathbb{R}^{3+1}$ denotes the observation point, where the fields shall be calculated. Note, that on the left hand side of (3.2) only spatial components of the respective four vectors are used.

Plugging in (3.1) in (3.2) yields

$$\begin{aligned} \lambda^2 (x_{j+1} - x_j)_\mu (x_{j+1} - x_j)^\mu + \lambda 2(x_{j+1} - x_j)_\mu (x_j - x_o)^\mu + (x_j)_\mu (x_j)^\mu \\ + (x_o)_\mu (x_o)^\mu - 2(x_j)_\mu (x_o)^\mu = 0. \end{aligned} \quad (3.3)$$

Equation (3.3) suggests the following definitions

$$\begin{aligned} a &:= (x_{j+1} - x_j)_\mu (x_{j+1} - x_j)^\mu, \\ b &:= 2(x_{j+1} - x_j)_\mu (x_j - x_o)^\mu, \\ c &:= (x_j)_\mu (x_j)^\mu + (x_o)_\mu (x_o)^\mu - 2(x_j)_\mu (x_o)^\mu. \end{aligned}$$

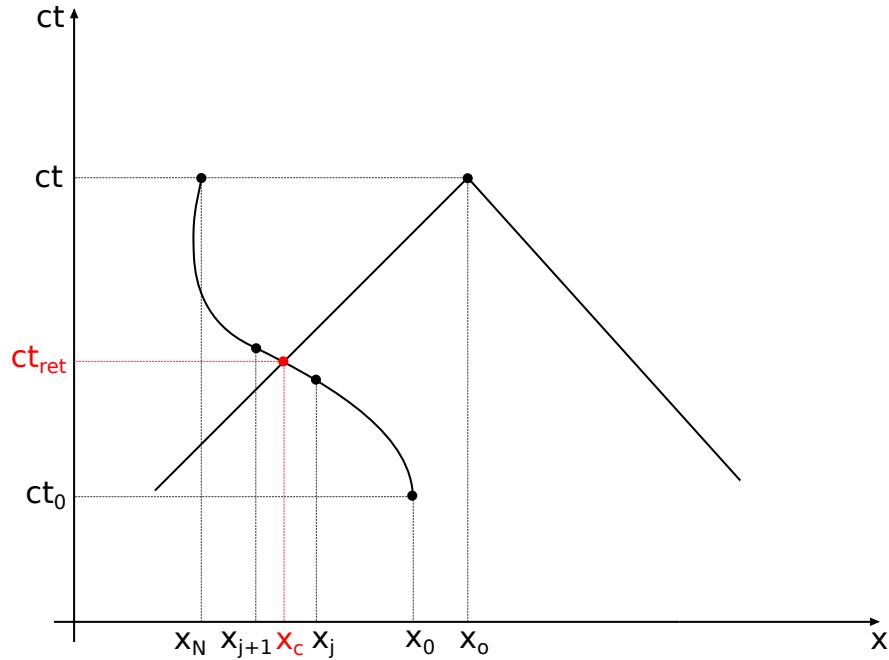


Figure 3.1.: Minkowski space showing the particle trajectory with starting point $x_0^\mu(t_0)$ and last point $x_N^\mu(t)$. The observation point $x_o^\mu(t)$ with its backward lightcone is also shown. To calculate the Liénard-Wiechert fields at the observation point $x_o^\mu(t)$ the intersection point $x_c^\mu(t_{ret})$ of the trajectory with the backward lightcone is required.

In general the quadratic equation (3.3) in λ has two solutions

$$\lambda_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

One solution denotes the intersection point with the backward lightcone, the other one with the forward lightcone. Since $\lambda \in [0, 1]$ one is only interested in the larger one

$$\lambda_{1/2} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$

Plugging in λ in (3.1) gives the desired intersection point.

3.2 Trilinear interpolation of fields

Another interpolation method used in this thesis is the trilinear interpolation to calculate the field values at the particle position more accurately. As is explained in more detail later (see Chapter 4) field values will either be stored on grid points or calculated analytically. However, if one wants to consider interactions between multiple particles correctly one needs the field values at the particle position, which does usually not coincide with the grid point, where the field values are stored. Matters are complicated further by the fact that the grid is staggered, due to numerical stability issues. This is explained in more detail in Section 4.1. Consider a cube with the points A, B, C, ..., H, as is illustrated in the following Figure 3.2

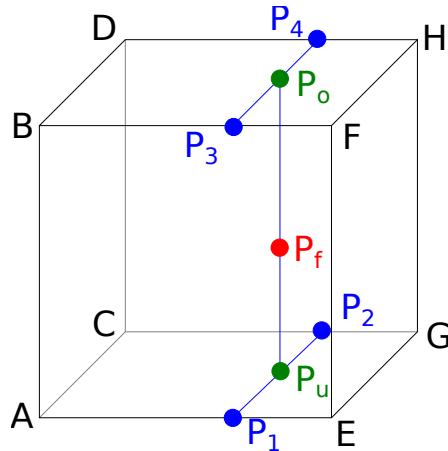


Figure 3.2.: Cube with its eight corner points A, B, C, ..., H and interpolation point P_f . The points P_1 , P_2 , P_3 and P_4 are the result of an interpolation along x direction of the lines \overline{AE} , \overline{CG} , \overline{BF} and \overline{DH} . P_u and P_o are obtained by interpolating along y - axis of the lines $\overline{P_1P_2}$ and $\overline{P_3P_4}$. Finally, P_f can be obtained by interpolating along z direction of the line $\overline{P_uP_o}$.

These eight points are part of the grid and contain the field values for both \vec{E} and \vec{H} . To get the correct field values at the particle position P_f a 3D interpolation is necessary. The trilinear interpolation simply consists of seven linear interpolations. First, one interpolates the x - value along \overline{AE} , \overline{CG} , \overline{BF} and \overline{DH} to get the points P_1 , P_2 , P_3 and P_4 . Thereto, let $x_p \in \mathbb{R}$ be the particle's x position and $x_0 \in \mathbb{R}$ and $x_1 \in \mathbb{R}$ be the corresponding x - values at A and E respectively. Same holds for the aforementioned

tuples as well. Then one can write

$$\begin{aligned}
 u &:= \frac{x_p - x_0}{x_1 - x_0} \\
 \implies P_1 &= A + u(E - A) \\
 \implies P_2 &= C + u(G - C) \\
 \implies P_3 &= B + u(F - B) \\
 \implies P_4 &= D + u(H - D).
 \end{aligned} \tag{3.4}$$

In the next step one interpolates along the y - axis. Analogously, one defines y_p , y_0 and y_1 . One concludes

$$\begin{aligned}
 v &:= \frac{y_p - y_0}{y_1 - y_0} \\
 \implies P_u &= P_1 + v(P_2 - P_1) \\
 \implies P_o &= P_3 + v(P_4 - P_3).
 \end{aligned} \tag{3.5}$$

Finally, with z_p , z_0 and z_1 one gets

$$\begin{aligned}
 w &:= \frac{z_p - z_0}{z_1 - z_0} \\
 \implies P_f &= P_u + v(P_o - P_u).
 \end{aligned} \tag{3.6}$$

4

Hybrid field approach

In this section the concept of hybrid fields is introduced. Christian Herzing has developed this concept in his dissertation [11] and has shown how this approach reduces the numerical complexity from N^2 to N . Usually, the numerical complexity of multi particle simulation is N^2 , due to the interaction between each particle. In the present work N push operations for the particles are needed. One push for each particle. In order to do that, however, equation (2.1) needs to be solved and therefore all Liénard-Wiechert fields from the other $N - 1$ particles are needed. This results in $N(N - 1)$ calculations for each time step. Calculating the Liénard-Wiechert fields require all positions from all particles to be stored as explained in Section 3.1. For a few particles such simulations are effortlessly feasible. But with increasing particle numbers such simulations may not just require more and more memory capacity but also become so time consuming that at some point they become impossible. That's where the hybrid field model comes in. Instead of calculating the Liénard-Wiechert fields at every time step for all particles at all grid points the fields are stored onto the grid and propagated through the grid using Maxwell equations. How that works in detail is explained in the following sections.

4.1 FDTD

This section presents how to solve the Maxwell equations (1.2) and how to propagate the fields on a numerical grid. The focus is thereby on the Maxwell equations in vacuum, i.e. $\rho = \vec{j} = 0$. As can be seen in the Liénard-Wiechert formula (1.32) there is a singularity for the field values at the particle position itself. Thus, fields are only propagated far away from the source, which is explained in more detail later.

To push the fields on the grid, i.e solving for the fields at the next time step, one uses the Yee-scheme introduced by *Kane Yee* in 1966 [22] or equivalently *Finite Difference Time Domain* method to discretize the Maxwell equations. Each point on the discretized grid is represented as a tuple (i, j, k) . One index for each dimension (x, y, z) . The distances between the grid points are $(\Delta x, \Delta y, \Delta z)$ respectively. The parameter h denotes the time discretization as before. In order to have finite central differences rather than plain finite

differences the evaluation of \vec{E} and \vec{H} components are shifted in time about $\frac{h}{2}$ against each other. To have the same benefits for the rotation the \vec{E} and \vec{H} components are shifted in space as well, as it's illustrated in Figure 4.1.

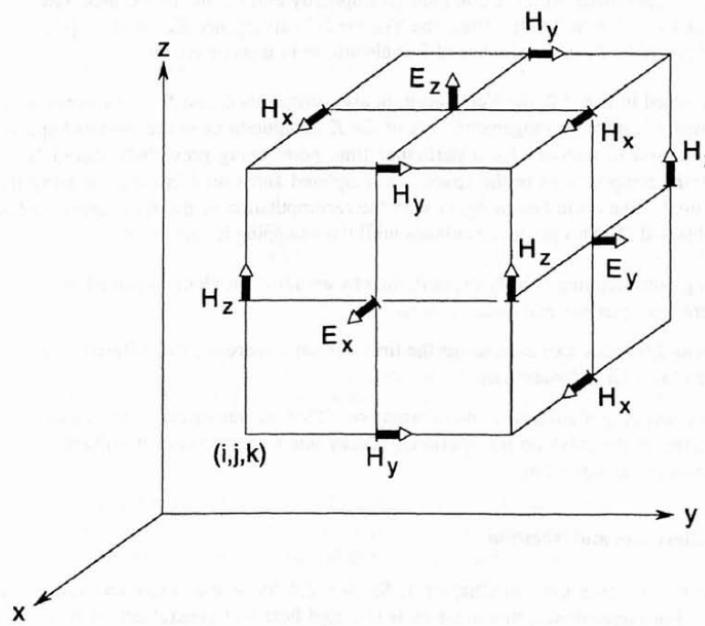


Figure 4.1.: Illustration of a so called Yee-Box which is used to solve the curl Maxwell equations. Shown are the positions of the electric and magnetic field vector components about a cubic unit cell of the Yee space lattice. The Yee algorithm centers its E and H components in three dimensional space so that every E component is surrounded by four circulating H components, and every H component is surrounded by four circulating E components [20].

The discrete Maxwell equations then read

$$\begin{aligned} \frac{\vec{E}|_{(i,j,k)}^{n+\frac{1}{2}} - \vec{E}|_{(i,j,k)}^{n-\frac{1}{2}}}{h} &= \vec{\nabla}^- \times \vec{H}|_{(i,j,k)}^n, \\ \frac{\vec{H}|_{(i,j,k)}^{n+1} - \vec{H}|_{(i,j,k)}^n}{h} &= \vec{\nabla}^+ \times \vec{E}|_{(i,j,k)}^{n+\frac{1}{2}}. \end{aligned} \quad (4.1)$$

The operators $\vec{\nabla}^-$ and $\vec{\nabla}^+$ act on a discretized vector field $F|_{(i,j,k)} : \mathbb{R}^3 \rightarrow \mathbb{R}$ as follows

$$\vec{\nabla}^- F|_{(i,j,k)} := \left(\frac{F|_{(i,j,k)} - F|_{(i-1,j,k)}}{\Delta x}, \frac{F|_{(i,j,k)} - F|_{(i,j-1,k)}}{\Delta y}, \frac{F|_{(i,j,k1)} - F|_{(i,j,k-1)}}{\Delta z} \right)^T, \quad (4.2)$$

$$\vec{\nabla}^+ F|_{(i,j,k)} := \left(\frac{F|_{(i+1,j,k)} - F|_{(i,j,k)}}{\Delta x}, \frac{F|_{(i,j+1,k)} - F|_{(i,j,k)}}{\Delta y}, \frac{F|_{(i,j,k+1)} - F|_{(i,j,k)}}{\Delta z} \right)^T \quad (4.3)$$

where T denotes the transpose as usual. The spatial positions, where \vec{E} and \vec{H} components will be calculated, are given by

$$\begin{aligned} \vec{E}|_{(i,j,k)} &= \begin{pmatrix} E_x [(i + \frac{1}{2}) \Delta x, j \Delta y, k \Delta z] \\ E_y [i \Delta x, (j + \frac{1}{2}) \Delta y, k \Delta z] \\ E_z [i \Delta x, j \Delta y, (k + \frac{1}{2}) \Delta z] \end{pmatrix}, \\ \vec{H}|_{(i,j,k)} &= \begin{pmatrix} H_x [i \Delta x, (j + \frac{1}{2}) \Delta y, (k + \frac{1}{2}) \Delta z] \\ H_y [(i + \frac{1}{2}) \Delta x, j \Delta y, (k + \frac{1}{2}) \Delta z] \\ H_z [(i + \frac{1}{2}) \Delta x, (j + \frac{1}{2}) \Delta y, \Delta z] \end{pmatrix}. \end{aligned} \quad (4.4)$$

4.2 Numeric dispersion relation

By discretizing the numerical grid the dispersion relation of light also changes. To study this effect the discretized Maxwell equations (4.1) are solved. Without loss of generality choose the ansatz of a TMz-mode, i.e. $H_z = 0$ and $E_z \neq 0$:

$$\begin{aligned} E_z|_{(i,j,k)}^n &= E_{z0} \exp \left\{ \hat{i} \left(\tilde{k}_x i \Delta x + \tilde{k}_y i \Delta y - \omega n h \right) \right\}, \\ H_x|_{(i,j,k)}^n &= H_{x0} \exp \left\{ \hat{i} \left(\tilde{k}_x i \Delta x + \tilde{k}_y i \Delta y - \omega n h \right) \right\}, \\ H_y|_{(i,j,k)}^n &= H_{y0} \exp \left\{ \hat{i} \left(\tilde{k}_x i \Delta x + \tilde{k}_y i \Delta y - \omega n h \right) \right\}, \end{aligned} \quad (4.5)$$

where \hat{i} denotes the imaginary unit and \tilde{k}_x , \tilde{k}_y the x- and y-components of the wave vector. Using TMz-mode equation (4.1) also simplifies to

$$\begin{aligned} \frac{E_z|_{(i,j,k)}^{n+\frac{1}{2}} - E_z|_{(i,j,k)}^{n-\frac{1}{2}}}{h} &= \left(\frac{H_y|_{(i,j,k)}^n - H_y|_{(i-1,j,k)}^n}{\Delta x} - \frac{H_x|_{(i,j,k)}^n - H_x|_{(i,j-1,k)}^n}{\Delta y} \right), \\ \frac{H_x|_{(i,j,k)}^{n+1} - H_x|_{(i,j,k)}^n}{h} &= - \left(\frac{E_z|_{(i,j+1,k)}^{n+\frac{1}{2}} - E_z|_{(i,j,k)}^{n+\frac{1}{2}}}{\Delta y} \right), \\ \frac{H_y|_{(i,j,k)}^{n+1} - H_y|_{(i,j,k)}^n}{h} &= \left(\frac{E_z|_{(i+1,j,k)}^{n+\frac{1}{2}} - E_z|_{(i,j,k)}^{n+\frac{1}{2}}}{\Delta x} \right). \end{aligned} \quad (4.6)$$

Upon substituting (4.5) in (4.6) and after some manipulation one finds

$$E_{z0} \sin\left(\frac{\omega h}{2}\right) = h \left[\frac{H_{x0}}{\Delta y} \sin\left(\frac{\tilde{k}_y \Delta y}{2}\right) - \frac{H_{y0}}{\Delta x} \sin\left(\frac{\tilde{k}_x \Delta x}{2}\right) \right], \quad (4.7)$$

$$H_{x0} = \frac{h E_{z0}}{\Delta y} \frac{\sin\left(\frac{\tilde{k}_y \Delta y}{2}\right)}{\sin\left(\frac{\omega h}{2}\right)}, \quad (4.8)$$

$$H_{y0} = \frac{h E_{z0}}{\Delta x} \frac{\sin\left(\frac{\tilde{k}_x \Delta x}{2}\right)}{\sin\left(\frac{\omega h}{2}\right)}. \quad (4.9)$$

Plugging in (4.8) and (4.9) in (4.6) one obtains the numerical dispersion relation

$$\left[\frac{1}{h} \sin\left(\frac{\omega h}{2}\right) \right]^2 = \left[\frac{1}{\Delta x} \sin\left(\frac{\tilde{k}_x \Delta x}{2}\right) \right]^2 + \left[\frac{1}{\Delta y} \sin\left(\frac{\tilde{k}_y \Delta y}{2}\right) \right]^2. \quad (4.10)$$

For the one dimensional case equation (4.10) reduces to

$$\omega h = 2 \arcsin \left(\frac{\Delta t}{\Delta x} \sin\left(\frac{\tilde{k}_x \Delta x}{2}\right) \right), \quad (4.11)$$

which is still more complicated than the analytic dispersion relation

$$\omega = ck. \quad (4.12)$$

In Figure 4.2 both, the analytical and the numerical dispersion relation are plotted. Equation (4.11) has a maximum at

$$\begin{aligned} \tilde{k}_x^{max} &= \frac{\pi}{\Delta x} \\ \iff \lambda_x^{max} &= 2\Delta x. \end{aligned} \quad (4.13)$$

The largest possible wavevector, \tilde{k}_x^{max} , is called *Nyquist Limit* for a numerical grid with resolution Δx , or Δy , Δz respectively. At this wavevector the group velocity

$$v_g = \frac{\partial \tilde{k}_x}{\partial \omega} \Big|_{\tilde{k}=\tilde{k}^{max}} = 0 \quad (4.14)$$

vanishes instead of propagating with speed of light. That means that the Yee-scheme is only valid for wavelengths $\lambda \ll \lambda^{max}$.

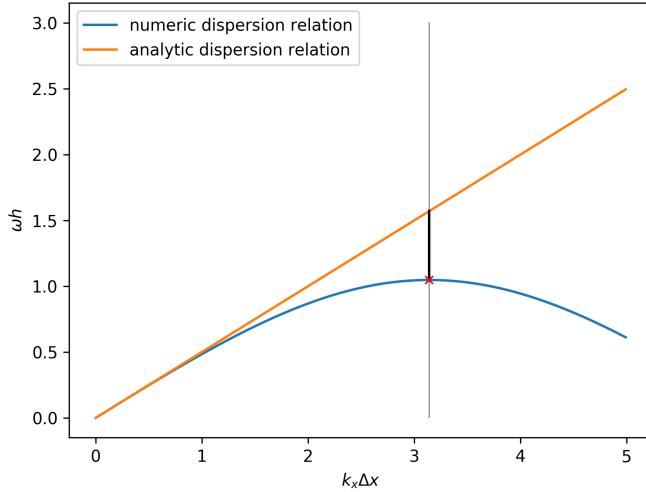


Figure 4.2.: Comparison of numerical and analytical dispersion relation of discretized Maxwell equations with Yee-scheme. $c \equiv 1$ is used and $\frac{\Delta t}{\Delta x} \equiv \frac{1}{2}$. At $\tilde{k}_x^{max} = \frac{\pi}{\Delta x}$ the group velocity vanishes instead of propagating with speed of light. Therefore, only waves with $\lambda \ll \lambda^{max}$ can be resolved properly on the grid.

After having discussed the fundamentals the practical section of the present work follows. Section 4.1 has explained how to store and propagate the fields on a numerical grid. In Section 1.3 an analytical expression has been derived of how to calculate the fields of a moving charged particle. It has also been observed that there is a singularity in (1.32) at the particle position itself. That means that the fields vary strongly in a vicinity of the particle and can not be resolved correctly on the grid, as discussed in Section 4.2. Hence, it seems plausible to separate the simulation area into appropriate near- and far field areas. Within the near field area of a particle its fields are **not** stored on the grid. With this the singularity is no problem any longer. Only far fields from other particles, propagated into the near field area are further propagated and stored on the grid. Possible interactions with other particles within the near field area are calculated analytically from (1.32). Far fields are stored on the grid and propagated with the Yee-scheme (4.1).

4.3 Near- and far fields

Before the simulation area is separated into near and far fields a grid needs to be initialized. The grid is divided in `numberBoxesInX`, `numberBoxesInY` and `numberBoxesInZ` boxes, each of which has `numberOfGridPointsForBoxInX`, `numberOfGridPointsForBoxInY` and `numberOfGridPointsForBoxInZ` length, which yields a total of

```
numberOfGridPointsInX =
numberOfGridPointsForBoxInX * numberOfBoxesInX ;
numberOfGridPointsInY =
numberOfGridPointsForBoxInY * numberOfBoxesInY ;
numberOfGridPointsInZ =
numberOfGridPointsForBoxInZ * numberOfBoxesInZ ;
```

Upon defining the resolution `dx`, `dy`, `dz` the size of the simulation area is then given by

```
lengthOfSimulationBoxInX = numberOfGridPointsInX * dx ;
lengthOfSimulationBoxInY = numberOfGridPointsInY * dy ;
lengthOfSimulationBoxInZ = numberOfGridPointsInZ * dz ;
```

The near field is then defined as the $3 \cdot 3 \cdot 3 = 27$ boxes with the box, containing the particle, in the center. In Figure 4.3 the near fields of one and two particles are shown. The near fields can of course overlap. The near field box at (18, 18) belongs to both particles.

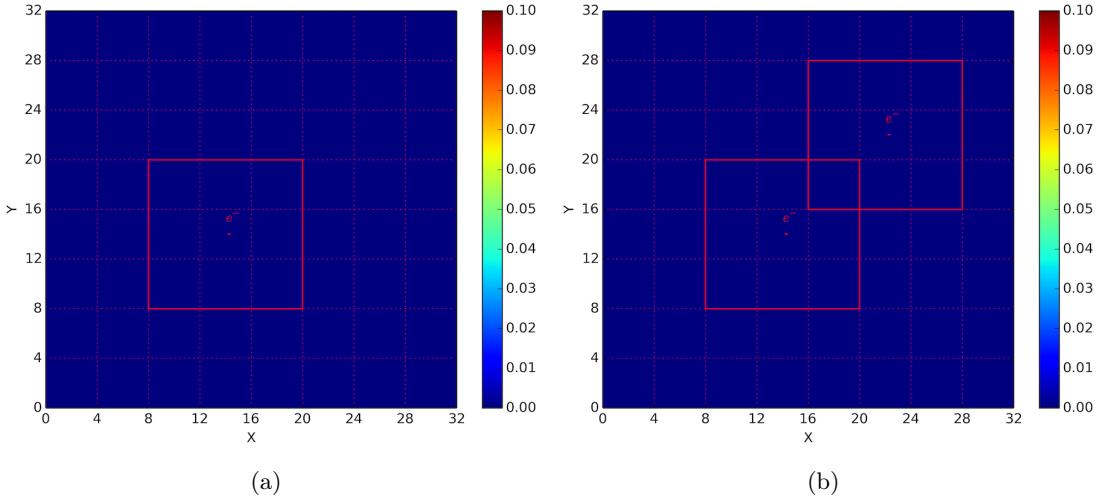


Figure 4.3.: Near fields of (a) one and (b) two particles. The box at $(18, 18)$ belongs to the near field of both particles, marked with e^- . Notice the staggered red grid in the background. In this example 8 boxes with 20 grid points each and a resolution of $dx = dy = dz = 0.2$ are used. The plane plotted was chosen according to the z-coordinate of the particle.

4.4 Field push

How a field push actually works is explained in the following. Since fields are only propagated in the far field area, as explained in Section 4.1, special care needs to be taken at the boundaries between far field and near field. To explain this consider the following plane of a grid illustrated in Figure 4.4. The red dots are considered to be the near field area of a particle. The black dots are therefore the far field area. The grid points within the near field does not contain any field values. Fields are only stored and propagated on the grid points in the far field. If the value for H_z is now pushed at $(2, 4, k)$ (recall that each grid point contains the field values for either H_x, H_y, H_z or E_x, E_y, E_z due to the staggered grid, explained in Section 4.1), where $k \in (1, 2, 3, \dots, \text{numberOfGridPointsInZ} - 1)$, one needs to solve

$$\frac{\partial H_z}{\partial t} = - \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right). \quad (4.15)$$

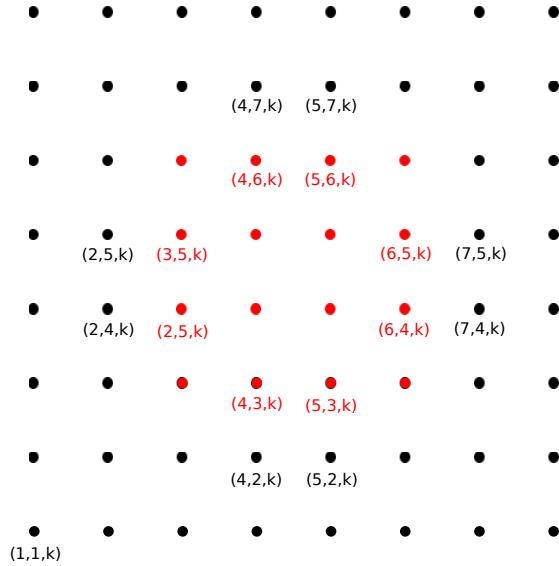


Figure 4.4.: One plane of a grid showing the **near field** and far field of a particle. The **near field** does not contain any field values of the particle. The variable k denotes the plane at an height z and can be chosen arbitrarily.

Using the discretized Maxwell equations (4.1) one gets

$$\vec{H}\Big|_{z(2,4,k)}^{n+1} = \vec{H}\Big|_{z(2,4,k)}^n - h \left(\frac{\mathcal{E}\Big|_{y(3,4,k)}^{n+\frac{1}{2}} - \mathcal{E}\Big|_{y(2,4,k)}^{n+\frac{1}{2}}}{\Delta x} - \frac{\mathcal{E}\Big|_{x(2,5,k)}^{n+\frac{1}{2}} - \mathcal{E}\Big|_{y(2,4,k)}^{n+\frac{1}{2}}}{\Delta y} \right). \quad (4.16)$$

$\vec{H}\Big|_{z(2,4,k)}^{n+1}$ is stored on a far field grid point. Its calculation, however, requires a field value $\mathcal{E}\Big|_{y(3,4,k)}^{n+\frac{1}{2}}$ which is stored on a near field grid point and therefore contains no field values from the particle (it can, however, contain field values from other particles propagated from the outside into the near field area). Hence, the Liénard-Wiechert field (1.32) of the particle needs to be calculated analytically at $(3, 4, k)$ $\mathcal{E}\Big|_{yp(3,4,k)}^{n+\frac{1}{2}}$ and added to the

field value already stored at this grid point. Finally, the correct calculation is

$$\begin{aligned} \vec{H}|_{z(2,4,k)}^{n+1} &= \vec{H}|_{z(2,4,k)}^n \\ &- h \left(\frac{\left(E|_{y(3,4,k)}^{n+\frac{1}{2}} + E|_{yp(3,4,k)}^{n+\frac{1}{2}} \right) - E|_{y(2,4,k)}^{n+\frac{1}{2}}}{\Delta x} - \frac{E|_{x(2,5,k)}^{n+\frac{1}{2}} - E|_{y(2,4,k)}^{n+\frac{1}{2}}}{\Delta y} \right). \end{aligned} \quad (4.17)$$

On the contrary, if value for H_z is pushed at $(6, 4, k)$ one needs to analytically calculate the Liénard-Wiechert field (1.32) of the particle at $(7, 4, k)$ $E|_{yp(7,4,k)}^{n+\frac{1}{2}}$ and subtract it from the field value already stored at this grid point. $(7, 4, k)$ already contains the field contribution of the particle since it is a far field grid point. And because $(6, 4, k)$ is a near field grid point only contributions from other particles need to be taken into account which are about to propagate into the near field area. After subtracting the particles own contributions the correct push can be calculated via

$$\begin{aligned} \vec{H}|_{z(6,4,k)}^{n+1} &= \vec{H}|_{z(6,4,k)}^n \\ &- h \left(\frac{\left(E|_{y(7,4,k)}^{n+\frac{1}{2}} - E|_{yp(7,4,k)}^{n+\frac{1}{2}} \right) - E|_{y(6,4,k)}^{n+\frac{1}{2}}}{\Delta x} - \frac{E|_{x(6,5,k)}^{n+\frac{1}{2}} - E|_{y(6,4,k)}^{n+\frac{1}{2}}}{\Delta y} \right). \end{aligned} \quad (4.18)$$

This and more examples as well as a more detailed explanation can be found in [11]. All points along the border between near- and far field need to be adjusted accordingly.

How this works out in code is explained below. Figure 4.5 shows three boxes, two of which are near field boxes and one is a far field box. For reasons of clarity only the grid points along the edges are shown.

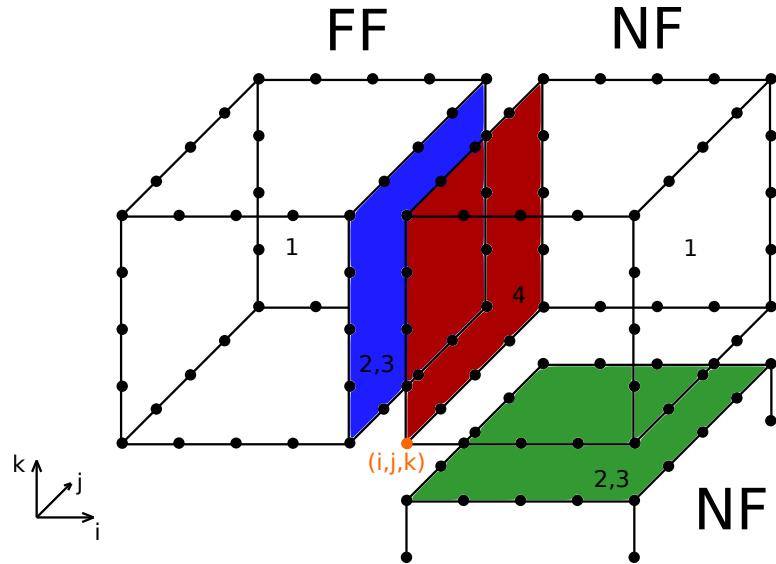


Figure 4.5.: Section of a grid with three boxes, two of which are near field boxes (NF) and one being a far field box (FF). The colored planes represent those grid points, where adjustments are necessary, as explained above. One \vec{E} -Field push consists of four steps, or equivalently four subroutines in code: 1. `pushEFieldInsideBoxes()`, 2. `setHFieldOnBorders()`, 3. `adjustHFields()`, 4 `pushEFieldAtBorders()`. For reasons of clarity the grid points within the boxes are left out.

One complete field push consists out of four steps. Even though the field pushes for \vec{H} and \vec{E} are conceptionally identical they differ in the grid points they are accessing to calculate the rotations $\vec{\nabla}^+$ and $\vec{\nabla}^-$ in (4.1). The steps (subroutines) for one \vec{E} -field push are

1. `pushEFieldInsideBoxes()`
2. `setHFieldOnBorders()`
3. `adjustHField()`
4. `pushEFieldOnBorders()`

Since this is the core of the simulation these steps will be explained in more detail below.

4.4.1 pushEFieldInsideBoxes()

This routine simply calculates the discretized Maxwell equations (4.1). It loops through the entire grid and stores the necessary field values but leaves out those grid points which are on the left plane (yz -plane) of each box. In Figure 4.5 marked with red. Now all \vec{E} values in all boxes are pushed except those on the red plane. To push those it needs to access the field values on the blue and green plane which are in the far field and therefore need to be adjusted.

4.4.2 setHFieldOnBorders()

In order to push \vec{E} the routine needs values of \vec{H} . Therefore, this method loops through all boxes and stores all values of \vec{H} in the plane left, in front and below of the current box (the plane in front is not shown for reasons of clarity). As can be seen from the Yee-scheme only H_y and H_z are needed from the left plane. H_z and H_x are needed from the plane in front and H_y and H_x are needed from the plane below. If the current box is a box where to the left does not exist then the respective values for H are set to zero.

4.4.3 adjustHField()

This routine consists of three subroutines `adjustHyz_im1()`, `adjustHxz_jm1()` and `adjustHxy_km1()`. The first adjusts H_y and H_z on plane to the left, the second adjusts H_x and H_z on the plane in front (not shown) and the latter adjusts H_x and H_y on the plane below. All of those three subroutines do conceptionally the same. They loop through all boxes and check whether the current box is part of the near field of the particle and the box to the left is not. This is the situation shown in Figure 4.5. Then the respective Liénard-Wiechert fields get calculated on the blue plane and subtracted from the values stored by `setHFieldOnBorders()`. Notice, that the red plane is part of a near field box and the blue plane is part of a far field box. Hence, the routine needs to subtract the field contributions of the particle on the blue plane. On the contrary, if the case is that the current box is a far field box and the box to left is a near field box, then the red plane is part of a far field box and the blue plane is part of a near field box. Hence, the routine calculates the Liénard-Wiechert fields on the blue plane and adds them to the respective values stored by `setHFieldOnBorders()`.

4.4.4 pushEFieldOnBorders()

This method finally pushes the \vec{E} -field with the adjusted values. It also loops through the entire grid, but only gets active if the current grid point lies on the red plane. If so, then it gets the values stored by `setHFieldOnBorders()` and adjusted by `adjustHField()` and uses them to calculate the discretized Maxwell equation (4.1).

The routines for pushing the \vec{H} -field are analogously:

1. `pushHFieldInsideBoxes()`
2. `setEFieldOnBorders()`
3. `adjustEField()`
4. `pushHFieldOnBorders()`

while using the planes to the right, behind and above.

As an example an electron with $\gamma = 1.1$ is simulated in a constant external B_z field. Figure 4.6 shows the result.

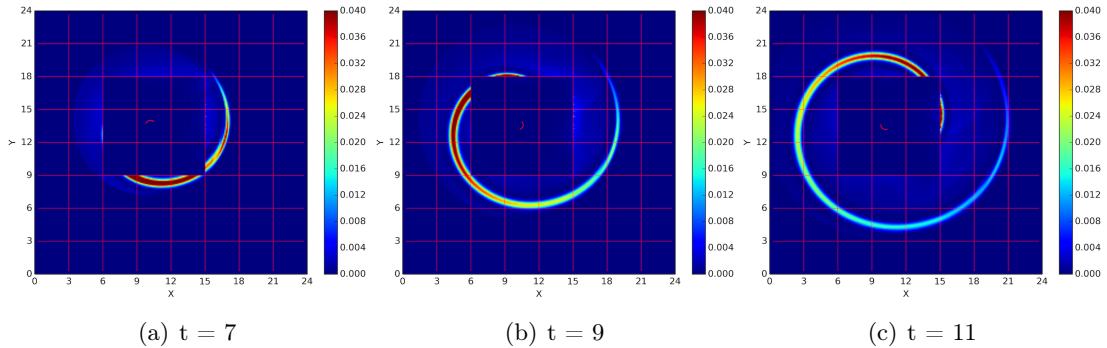


Figure 4.6.: Simulation of an electron with $\gamma = 1.1$ in a constant $B_z = 1$ field at (a) $t = 7$, (b) $t = 9$ and (c) $t = 11$. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2 / (4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

The electron moves in a circular fashion in the xy -plane. The red line in the center of the near field area shows the particle's trajectory. The initial conditions are chosen such that the particle does not leave its initial box. That means that the near field does not change during runtime. The Liénard-Wiechert fields are only calculated at the border between near and far field and only if the retardation constraint $t_{ret} > 0$ is met. As already

mentioned before, Figure 4.7 finally shows the expected behavior of two particle's far fields propagating into each other's near field regions.

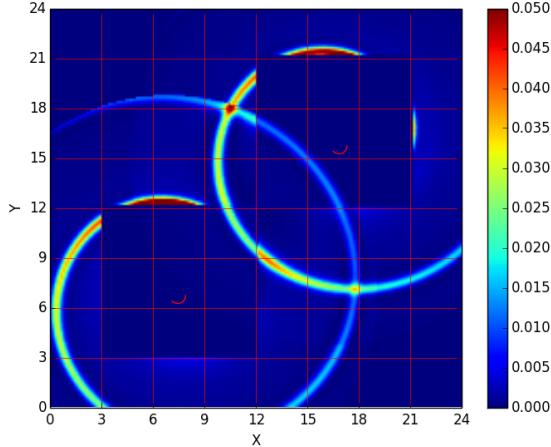


Figure 4.7.: Gyration of two electrons with $\gamma = 1.1$ each. This shows the expected behavior of two particle's far fields propagating into each other's near field regions. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$ and the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

But what if the particle does leave its initial box ?

4.5 Particle push and near field update

In order to push the particle properly not just the external fields but also the field contributions from other particles propagated into the near field region need to be taken into account. To get an even better result trilinear interpolation is used (see Section 3.2) to interpolate the field value at the particle position from the surrounding eight grid points. If, however, another particle2 enters the near field region of the observed particle1 equation (1.32) is used to calculate the field emitted by particle2 at the position of particle1 analytically.

If a particle changes its box during one push from time t to $t + dt$ the near field of the particle changes as illustrated in Figure 4.8

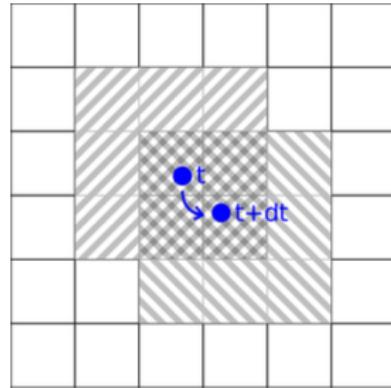


Figure 4.8.: Particle push from time t to $t+dt$. Since the particle changes its box the near field region also changes. The boxes marked with $\backslash\backslash$ do no longer belong to the near field, whereas the boxes marked with $\backslash\backslash\backslash$ do now belong to the near field area. Boxes illustrated as \boxtimes remain as near field boxes [11].

There are three situations which need to be taken care of. First, the boxes which do not longer belong to the near field area marked with $\backslash\backslash$ in Figure 4.8. Since those boxes were part of the near field they didn't contain any field contributions of the particle. But because they are no longer part of the near field region those field contributions need to be added. The routine `addLWFIELDSInBox()` calculates the Liénard-Wiechert field (1.32) of the particle at all grid points in the specified box and adds them to the respective grid point. Second, all boxes marked with $\backslash\backslash\backslash$ are now part of the near field region and therefore the Liénard-Wiechert field of the particle at all grid points in the specified box needs to be subtracted. The routine `subLWFIELDSInBox()` does exactly that. Lastly, there are boxes which remain as part of the near field area marked with \boxtimes . Nothing needs to be done here. As an example an electron with $\gamma = 1.2$ is simulated in a crossed \vec{E} and \vec{B} field. The results are shown in Figure 4.9

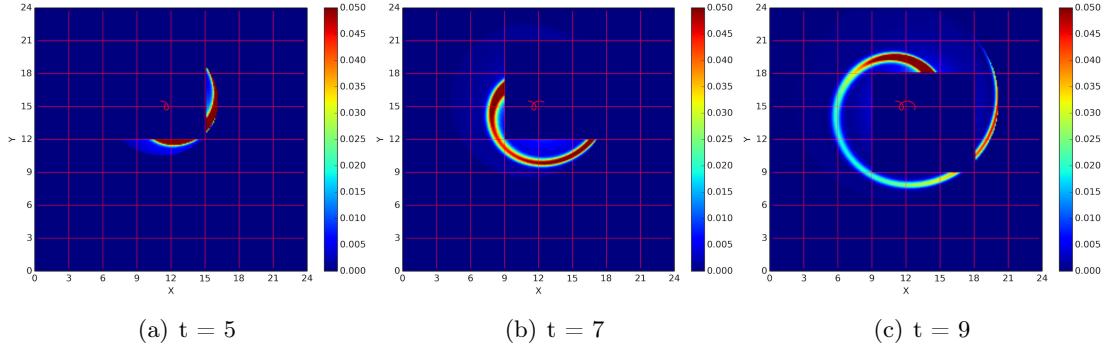


Figure 4.9.: Simulation of an electron with $\gamma = 1.2$ in a crossed electromagnetic field with $B_z = 1$ and $E_y = 0, 2$ at (a) $t = 5$, (b) $t = 7$ and $t = 9$. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2 / (4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$

4.6 Particle history

In order to calculate the Liénard-Wiechert Field according to (1.32) at an observation point, i.e. a grid point, the intersection point of the particles trajectory with the backward lightcone of that observation point is needed. For a more detailed explanation see Section 3.1. That means that first all positions of the particle have to be stored. Second, they have to be tracked along the trajectory to check whether the backward lightcone was crossed or not. To store the entire trajectory of a few particles is not very memory consuming. However, if one wants to simulate 10^{10} or even 10^{20} particles one would need a huge amount of memory just to store the trajectories. As the following rough calculation shows this is not feasible:

One position contains of four Doubles. One for time, three for spacial components. A Double has 8 B. That makes a total of 32 B memory for one particle position. A simulation for $t = 10$ with a resolution of $dx = dy = dz = 0.1$ using Yee-scheme has $\frac{10}{0.5 \cdot 0.1} = 200$ steps. That yields $200 \cdot 32 \text{ B} = 6.4 \text{ kB}$. For 10^{10} particles this sums up to 64 TB. Not to mention that there is a lot more memory necessary to store all the calculated fields on the grid.

The problem is that the positions which are already used to find an intersection point

can't simply be deleted because the intersection point depends on the particle's velocity. The faster the particle gets, the longer the history needs to be. It can be seen why that is in the following picture 4.10

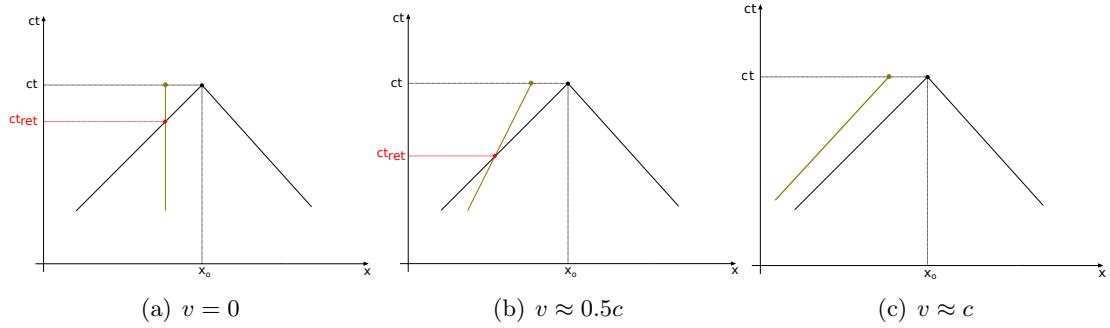


Figure 4.10.: Observation point x_o at time ct . The particle trajectory is shown in ocher. The intersection point is marked red. It can be seen that the faster the particle gets, the longer the particle history needs to be in order to find the intersection point. At the speed of light there would be no intersection point since the backward lightcone and the particle history would be parallel.

The Figure shows an observation point x_o and its backward lightcone in a Minkowsky space. The ocher line illustrates the particle's trajectory, crossing the backward lightcone at an intersection point, is marked as a red dot. With increasing velocity the trajectory tends more and more towards a 45° angle. At $v = c$ the trajectory would be parallel to the backward lightcone and thus an intersection would be impossible. Therefore, one can conclude that a plausible criteria for finding the position after which all following positions could be neglected has to be velocity dependent. Due to the aforementioned arguments one can moreover conclude that the dependence would be proportional to $\frac{1}{1-\beta}$. But this is not good enough. Since the velocity can change during runtime, an a priori approach is required to reduce the memory consumption. As discussed in Section 1.3.3 the intensity declines with a $\frac{1}{r^2}$ dependance. This allows to define a distance over which the emitted fields would be attenuated so much that they could be neglected. Unfortunately, a concrete formulation of that idea could not be developed in the present work.

4.7 Far field setup before simulation

So far a particle (or multiple particles) can be initialized and can be moved on the grid according to the Lorentz-Newton equation (2.1) without damping. If there are external fields or interaction forces between multiple particles they get accelerated and start to radiate Liénard-Wiechert fields. As soon as the retardation constraint is met ($t_{ret} > 0$) the fields will appear at the border between near and far field region and get propagated via (4.1). This works fine but unfortunately this initial setup is far from realistic. The particles were not created the second the simulation starts. It is not of interest to simulate the creational process but the emitted fields due to interactions between particles which do already exist. Therefore, the particles will already have a long trajectory history and some kind of Liénard-Wiechert fields coming with them. The question is: How do setup the fields on the grid before the simulation starts? The following approach is used:

Boris pusher is used (see Section 2.4) to push the particles from t to $t + dt$, starting with the initial conditions, given prior to the simulation. This algorithm can also be used to push backwards in time since it is symplectic and therefore time reversible. By selecting the simulation parameter $t = t_0 > 0$ the trajectory is calculated backwards in time for t_0 time units. This can be achieved by reversing the external fields and the initial velocity vector of the particle. Once one gets t_0 back in time one reverses the external fields and the velocity vector again, sets the particles proper time to zero and starts pushing the particle from there. In order to store all new trajectory points in the history it needs to be extended by $\frac{t_0}{dt}$ elements first. With this approach the simulation starts with the particle being at the specified initial conditions by the user, but the particle will already have a trajectory history, from which the initial Liénard-Wiechert fields can be calculated with (1.32) in all far field boxes of the grid. This, however, is computationally quite expensive. Therefore, a routine was developed which serializes the initial conditions of the simulation in a file and only calculates the fields if they were not calculated and stored before. The flow chart in Figure 4.12 shows which functions get called when the user sets $t = t_0 > 0$. In Figure 4.11 some examples of initial fields for different times are shown. Notice, that in all cases the simulation starts with the particle being at the initial position ¹ the user entered prior to the simulation. Only the distance over which the fields are initialized differ due to different particle history lengths.

¹The initial position used in the simulation is `Particle->x[1] = 11.1` and `Particle->x[2] = 15.5`

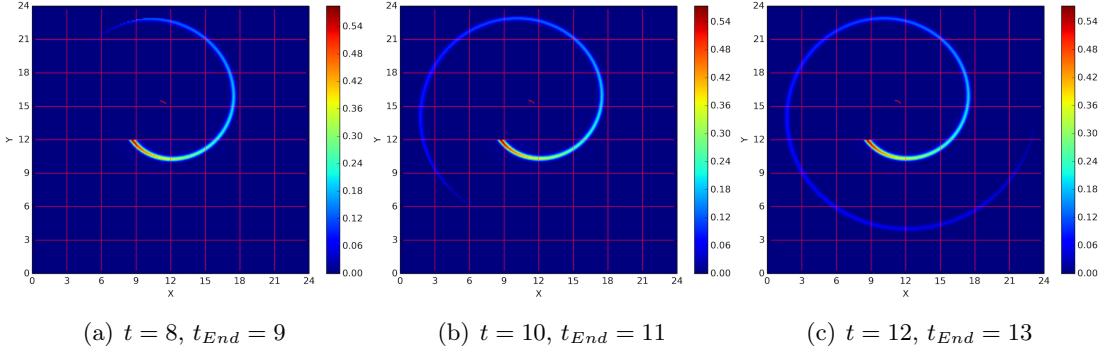


Figure 4.11.: The plots show the radiation fields after simulating for $t = 1$ with a start time of (a) $t = 8$, (b) $t = 10$ and (c) $t = 12$. If $t > 0$ the history gets extended with a reverse Boris push such that the initial conditions are still met at the beginning of the simulation. The Liénard-Wiechert fields are calculated from the extended particle history. The electron has an energy of $\gamma = 1.2$. A constant $B_z = 1$ is used as external field. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2 / (4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$

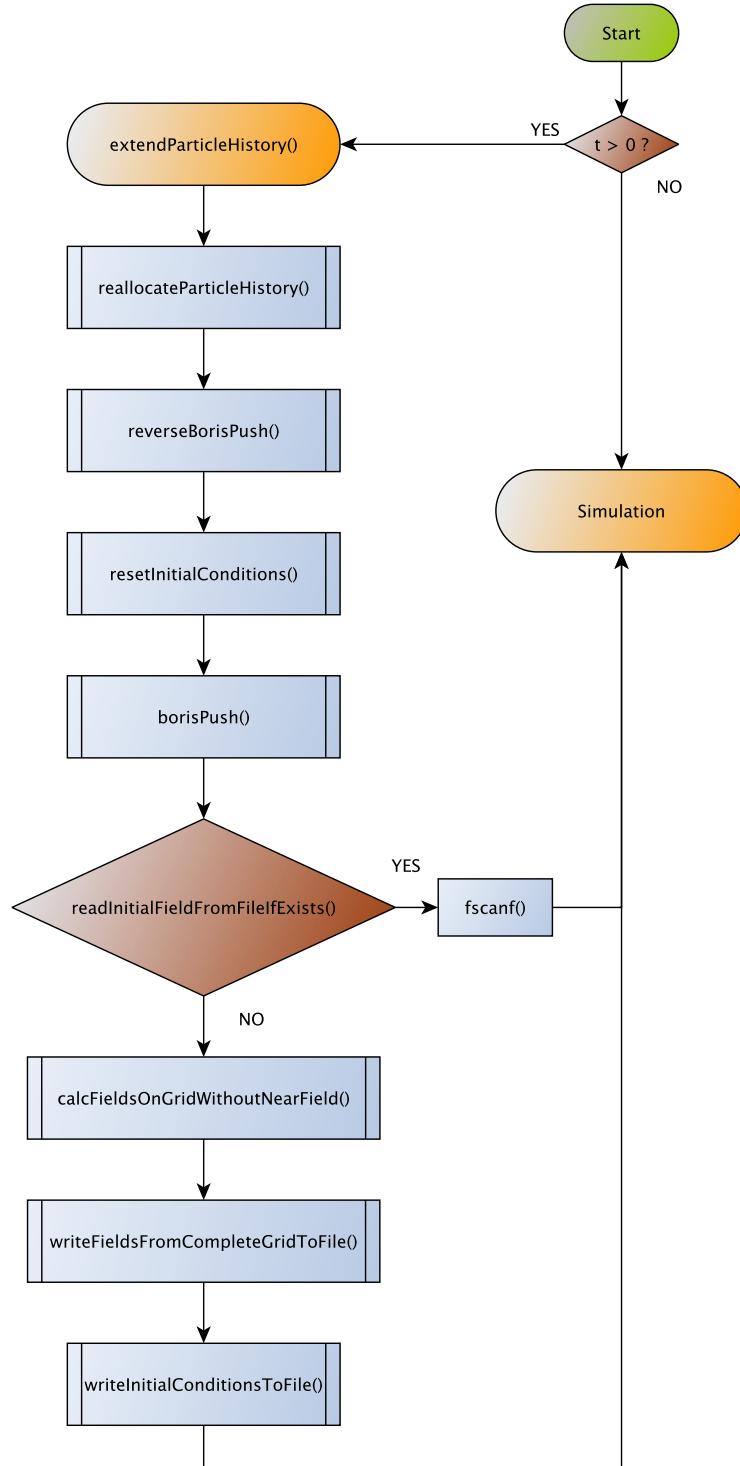


Figure 4.12.: Flow chart showing the relevant function calls when the user sets $t = t_0 > 0$. In this case the radiation fields are calculated from the extended particle history and are set onto the numerical grid. If $t = t_0 = 0$ the simulation starts right away. This way of setting up the simulation is only correct if there is only one particle or the particles do not interact during the initialization process.

4.8 Electron scattering in an electromagnetic wave

Finally, to proof the framework's proper functionality an electron scattering process with a laser pulse for realistic scales is simulated. Making use of the normalization outlined in Appendix D the following dimensionless simulation parameters are used

$$\begin{aligned} a_0 &= \frac{eE_{0L}}{m_e\omega_L c} = 100, \\ \lambda_L &= 10^{-6} \text{m} = 1 \mu\text{m}, \\ \implies \omega_L &= 1.88 \cdot 10^{15} \text{s}^{-1}, \end{aligned} \quad (4.19)$$

where a_0 is the laser amplitude, λ_L the laser wavelength and ω_L the laser frequency. Due to these parameters and with the help of

$$\omega = \frac{4\pi\epsilon_0 m_e c^3}{q^2} = 1.06 \cdot 10^{23} \text{s}^{-1}, \quad (4.20)$$

$$E_0 = \frac{\omega m_e c}{q} = 1.77 \cdot 10^{20} \text{V m}^{-1}, \quad (4.21)$$

from Appendix D one finds

$$\frac{\omega_L}{\omega} = 1.77 \cdot 10^{-8}. \quad (4.22)$$

With the help of (4.21) and (4.22) one obtains

$$\begin{aligned} E_{0L} &\equiv \frac{a_0 m_e \omega_L c}{q} \\ &= 100 \frac{m_e c}{q} \cdot 1.77 \cdot 10^{-8} \omega \\ &= 1.77 \cdot 10^{-6} E_0. \end{aligned} \quad (4.23)$$

The same holds for B_{0L} . Finally, the dimensionless grid and time resolution dx and dt need to be determined. In order to resolve the fields properly on the grid the condition

$$\Delta t_L \omega_L \ll 1 \quad (4.24)$$

has to hold. Making use of (4.22) one obtains

$$\Delta t_L \ll \frac{1}{\omega_L} \stackrel{(4.22)}{=} \frac{1}{1.77 \cdot 10^{-8} \omega}, \quad (4.25)$$

Since $dt = \Delta t_L \omega$ and $dt = 0.5dx$ (see Section 4.1) choosing $dx = 10^7$ is sufficiently small.

For the results being discussed shortly the following settings are used

```

Resolution->dx = pow(10,7);
Resolution->dy = pow(10,7);
Resolution->dz = pow(10,7);

Box->numberOfGridPointsInX = 32;
Box->numberOfGridPointsInY = 32;
Box->numberOfGridPointsInZ = 32;

Grid->numberOfBoxesInX = 22;
Grid->numberOfBoxesInY = 10;
Grid->numberOfBoxesInZ = 10;

int numberOfParticles = 1;

double dt = 0.5 * Resolution.dx;
double t = 400 * pow(10, 7);
double tEnd = 1100 * pow(10, 7);

Particle1->mass = pow(10, 10);
Particle1->charge = pow(10, 10);

Particle1->x[0] = 0;
Particle1->x[1] = 6.0 * pow(10, 9);
Particle1->x[2] = 1.6 * pow(10, 9);
Particle1->x[3] = 1.6 * pow(10, 9);

```

For the external laser pulse a cosine function is modulated with a sin squared function in x direction and a gaussian in y direction

$$\begin{aligned}
E_y(x, t) &= E_0 \cdot \sin^2 \left(\frac{\omega_L(t - x/c)}{\varphi} \right) \cdot \cos(\omega_L(t - x/c)) \cdot \exp \left(-\frac{(\omega_L y)^2}{c^2 \sigma^2} \right), \\
H_z(x, t) &= H_0 \cdot \sin^2 \left(\frac{\omega_L(t - x/c)}{\varphi} \right) \cdot \cos(\omega_L(t - x/c)) \cdot \exp \left(-\frac{(\omega_L y)^2}{c^2 \sigma^2} \right),
\end{aligned} \tag{4.26}$$

where the parameters φ and σ are chosen such that the pulse has a width of half the simulation box both in x and y directions. All other field components of E and H are zero. Figure 4.13 shows the external pulse.

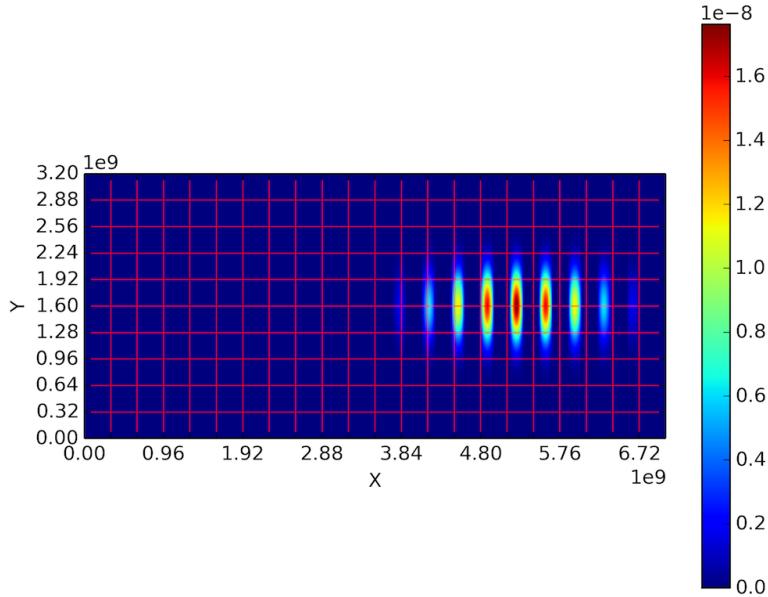


Figure 4.13.: External laser pulse at $t = 700 \cdot 10^7$. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

The idea is to slowly increase γ of the electron while maintaining a small angle between the particle momentum and the k-vector of the incoming external laser pulse. With increasing γ the radiation field emitted should become more and more directed. At the same time the radiation frequency should increase. The factor γ is increased from 1.0 to 1.5 with increments of $\Delta\gamma = 0.1$. To analyze the radiation direction and radiation frequency a Fourier analysis of the electric field components \vec{E} and of the components of the Poynting vector \vec{S} is made. Figures 4.14 - 4.25 show the simulation results, both with and without the near field region of the particle. The ratio of the damping force g^μ and Lorentz force is also calculated at each time step. This way it is easy to show that the damping force g^μ can be neglected.

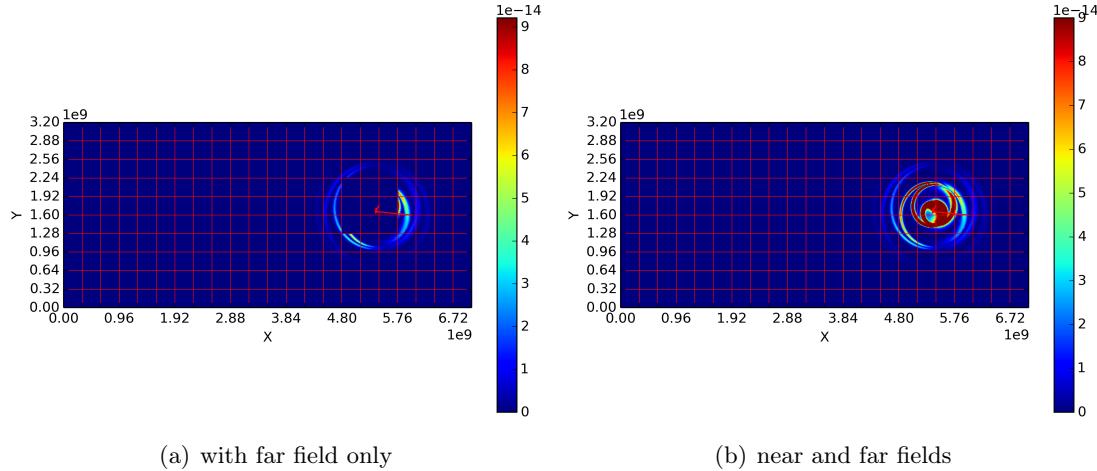


Figure 4.14.: Initial conditions: $u_x = 0.1$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.0$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

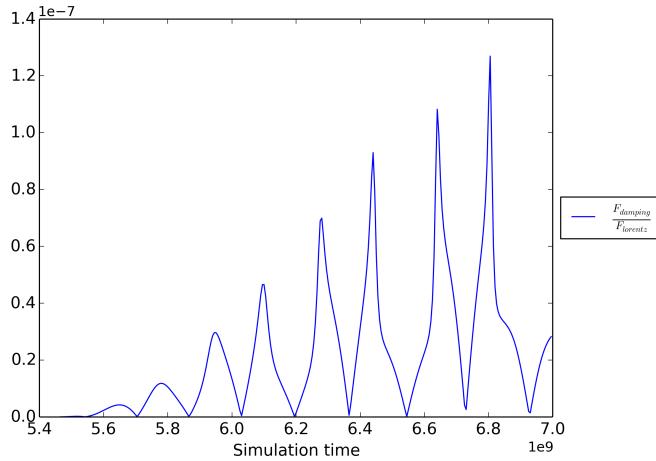


Figure 4.15: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.0$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

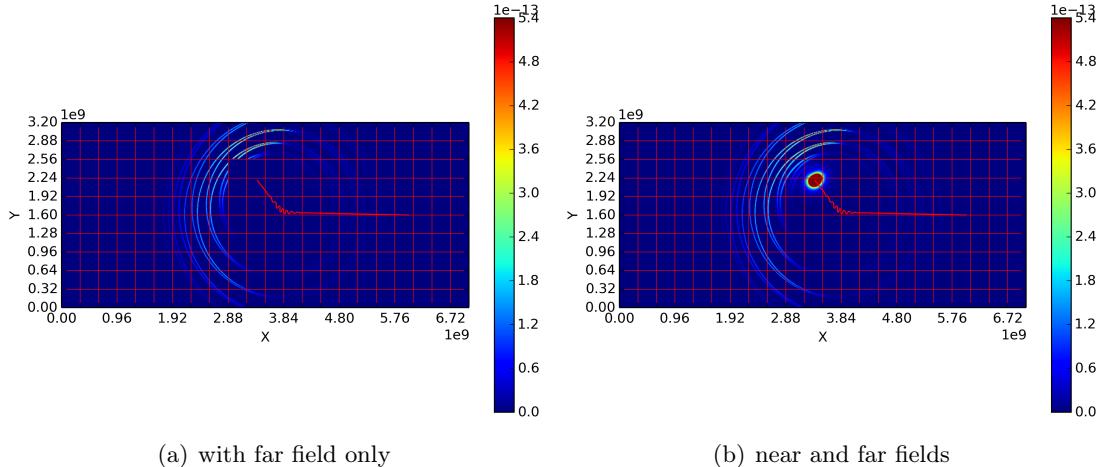
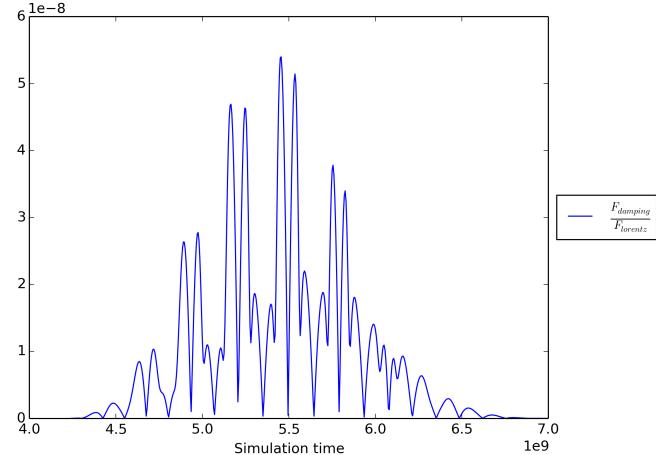


Figure 4.16.: Initial conditions: $u_x = 0.46$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.1$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

Figure 4.17: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.1$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.



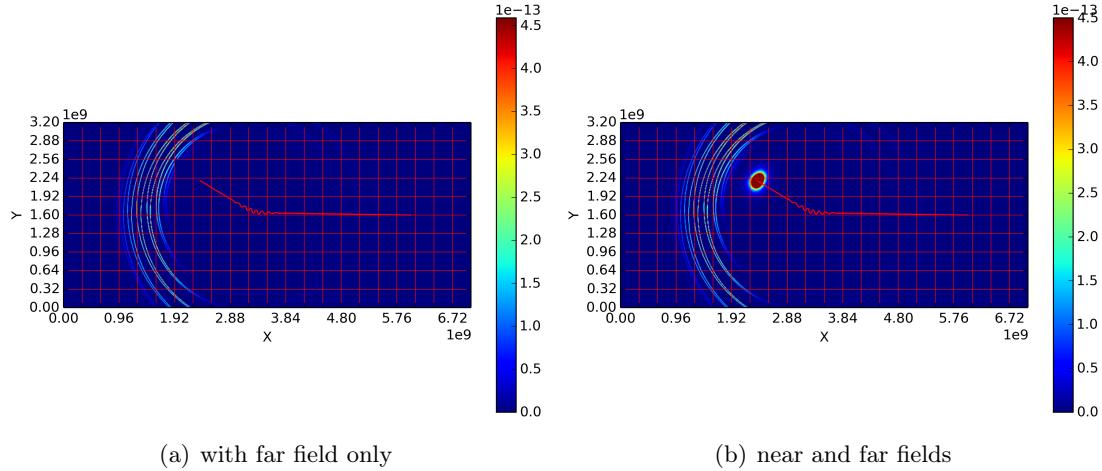


Figure 4.18.: Initial conditions: $u_x = 0.67$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.2$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

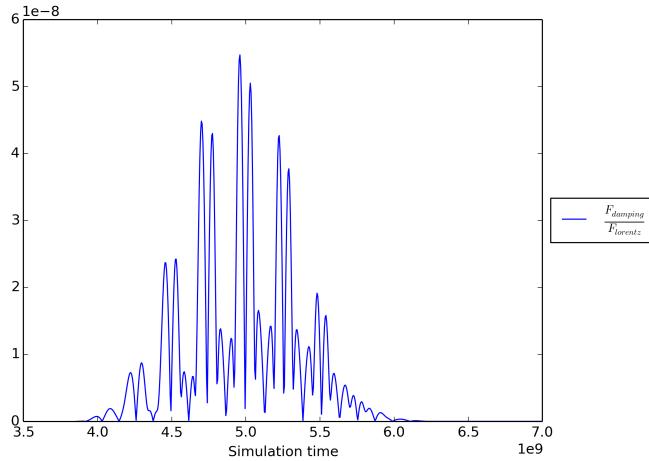


Figure 4.19: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.2$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

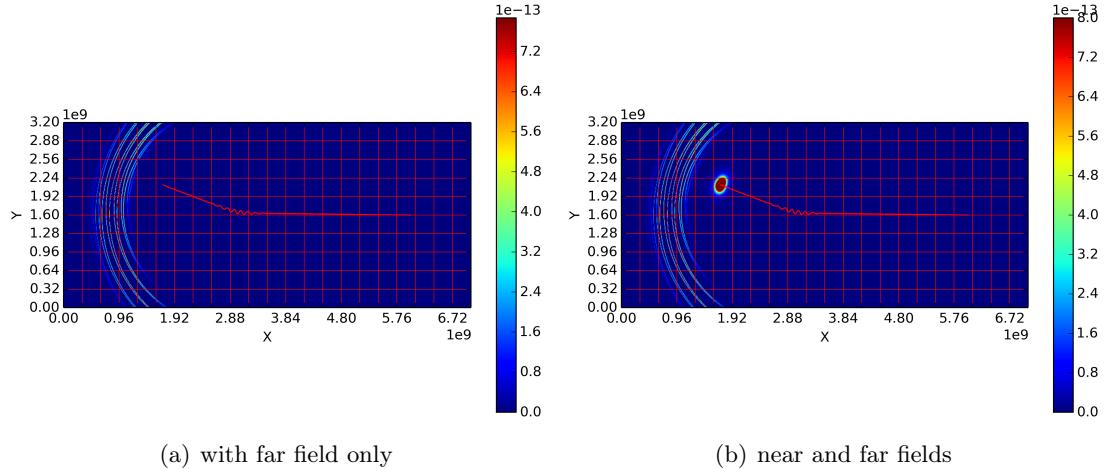
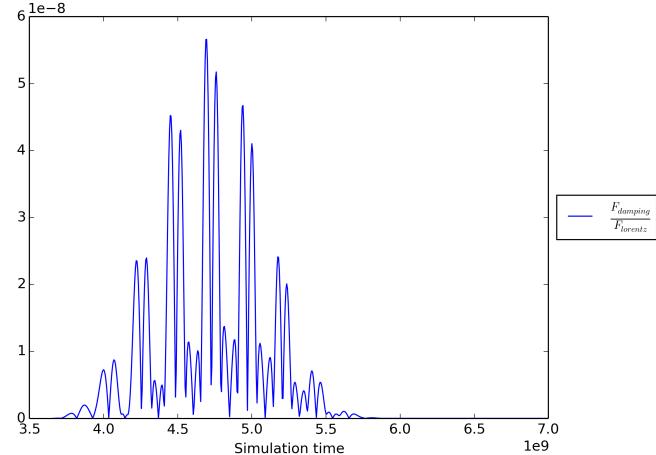


Figure 4.20.: Initial conditions: $u_x = 0.84$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.3$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

Figure 4.21: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.3$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.



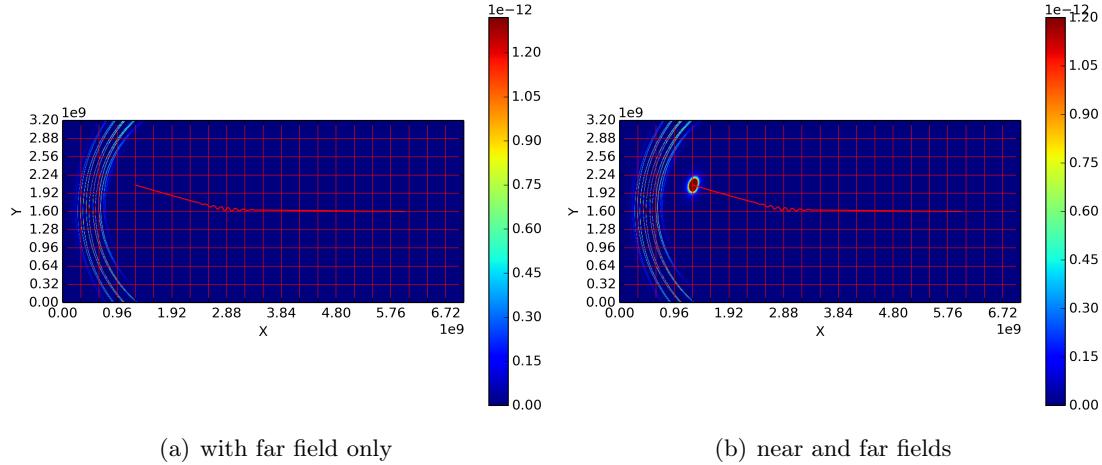


Figure 4.22.: Initial conditions: $u_x = 0.98$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.4$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

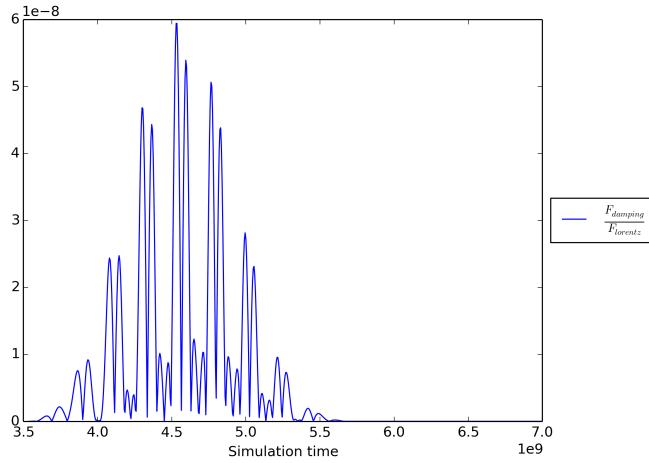


Figure 4.23: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.4$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

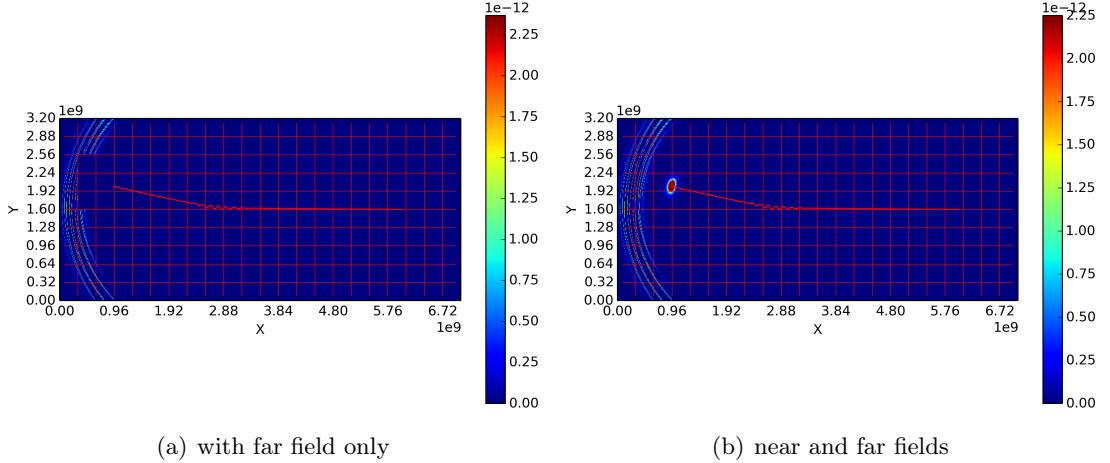
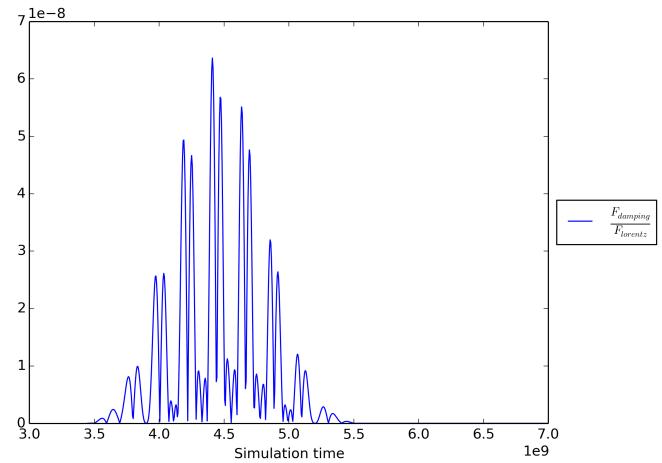


Figure 4.24.: Initial conditions: $u_x = 1.12$, $u_y = 0.01$ and $u_z = 0.0$. The particle history is calculated for $t = 400 \cdot 10^7$ prior to the simulation. The pictures are taken at $t = 1100 \cdot 10^7$. The electron has an energy of $\gamma = 1.5$. The external pulse from above is used as external field and is therefore not shown here. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

Figure 4.25: Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.5$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.



CHAPTER 4. HYBRID FIELD APPROACH

In the next step the electric field components \vec{E} as well as the Poynting vector \vec{S} are 2D Fourier analyzed. The results are shown in figures 4.26 - 4.31

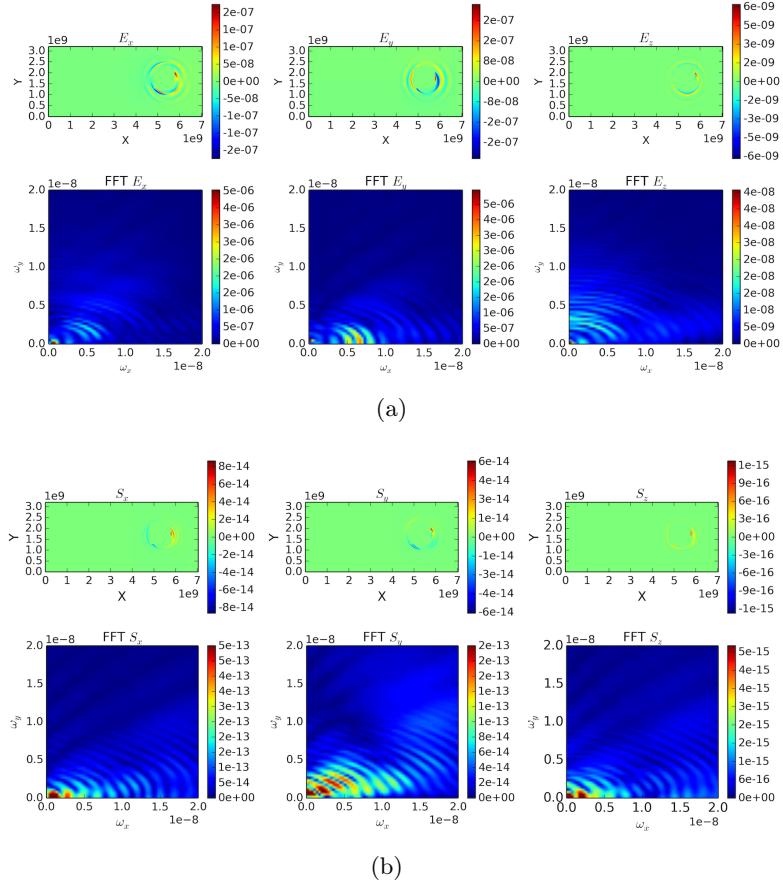


Figure 4.26.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.0$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

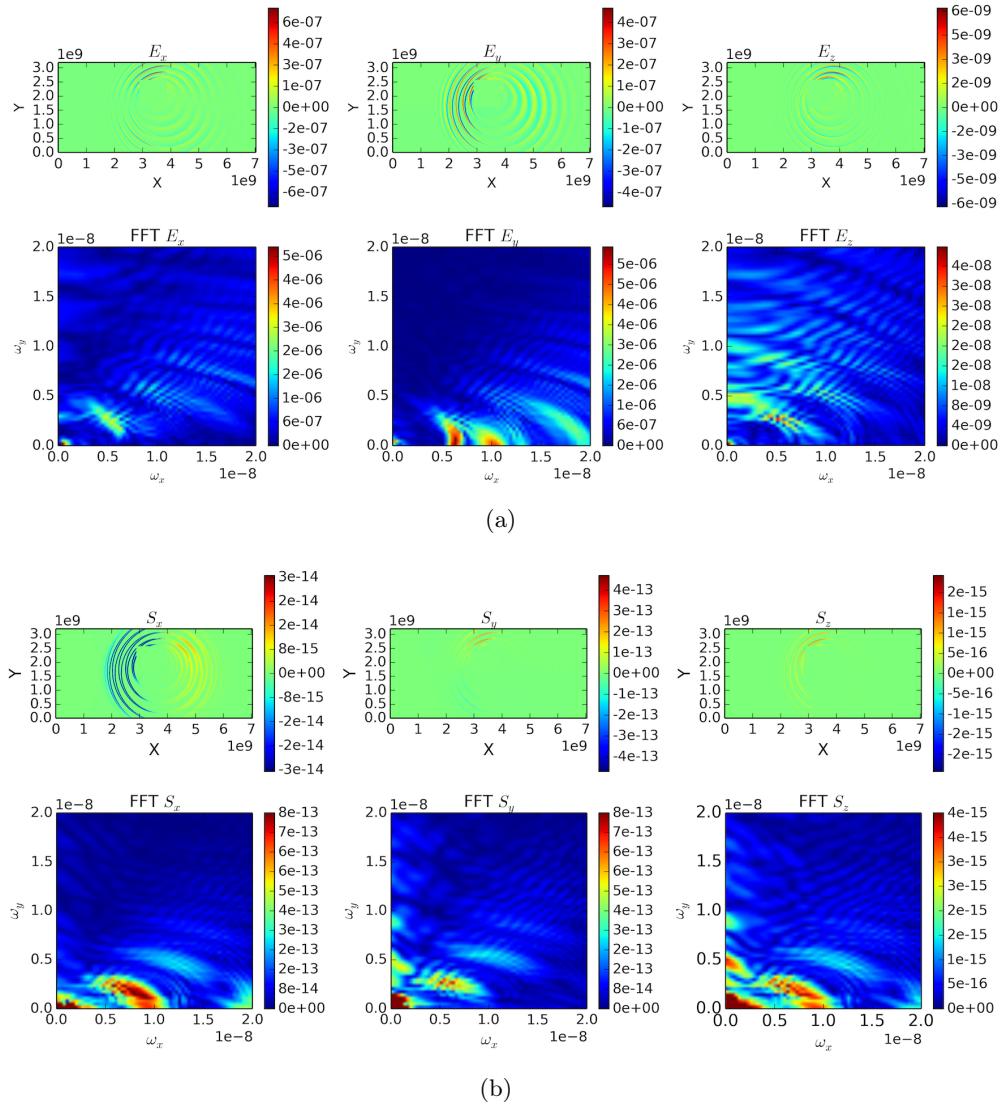


Figure 4.27.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.1$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

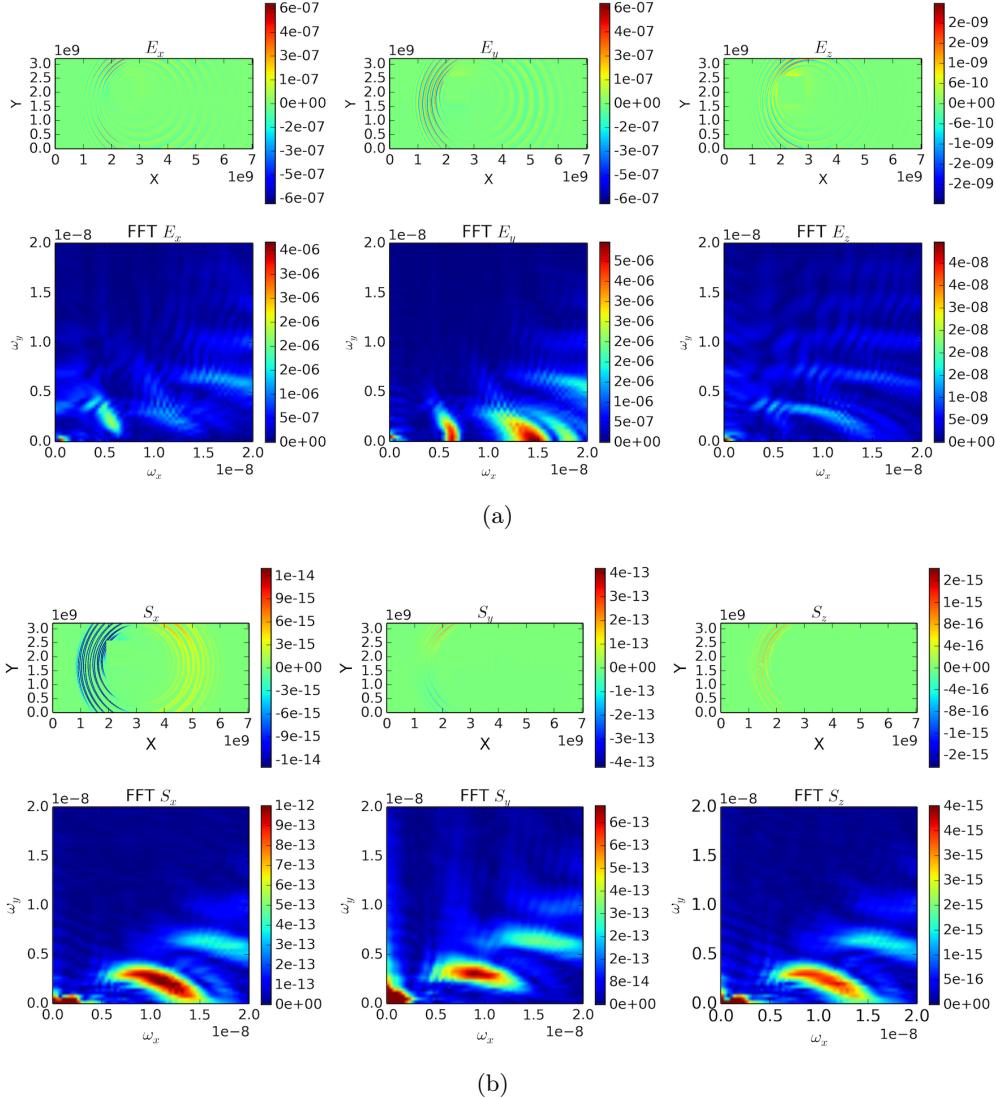


Figure 4.28.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.2$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

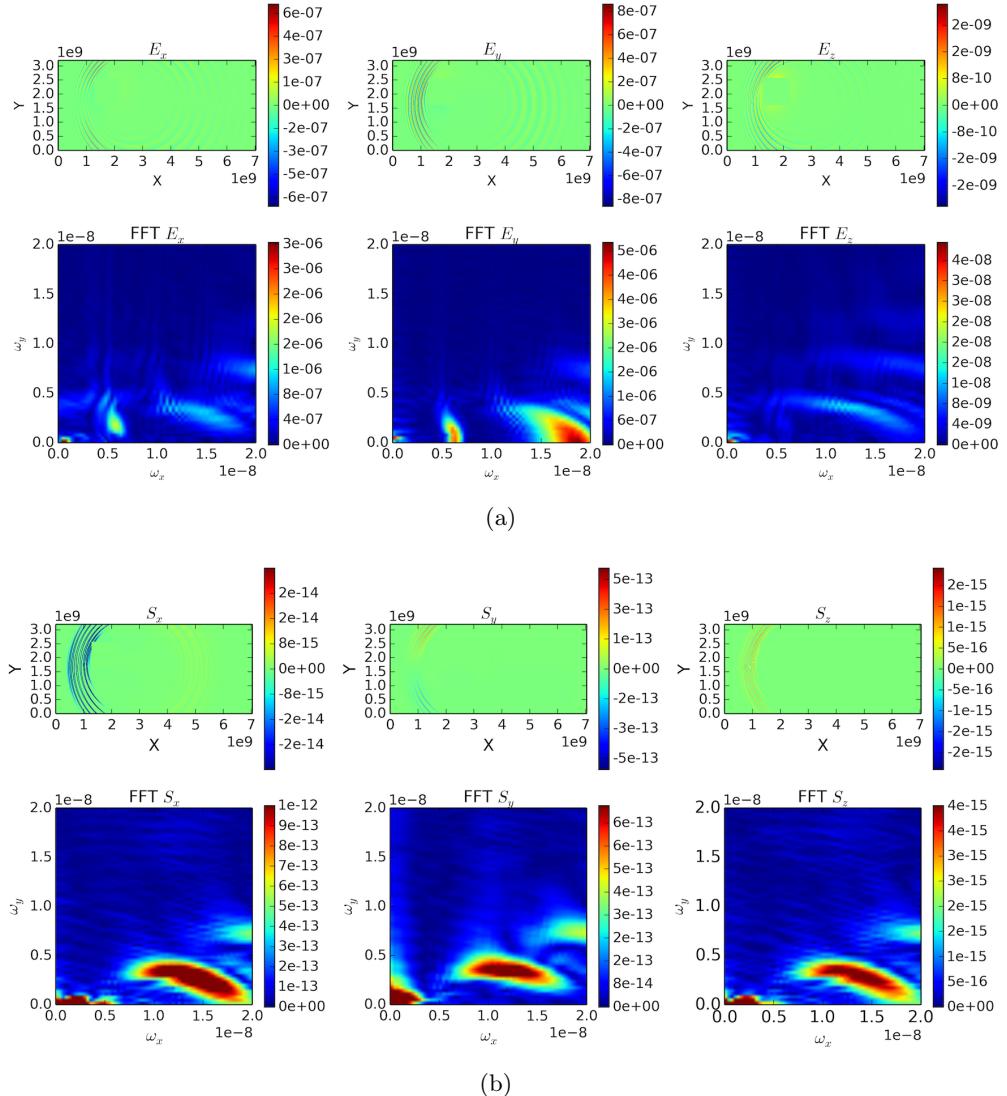


Figure 4.29.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.3$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

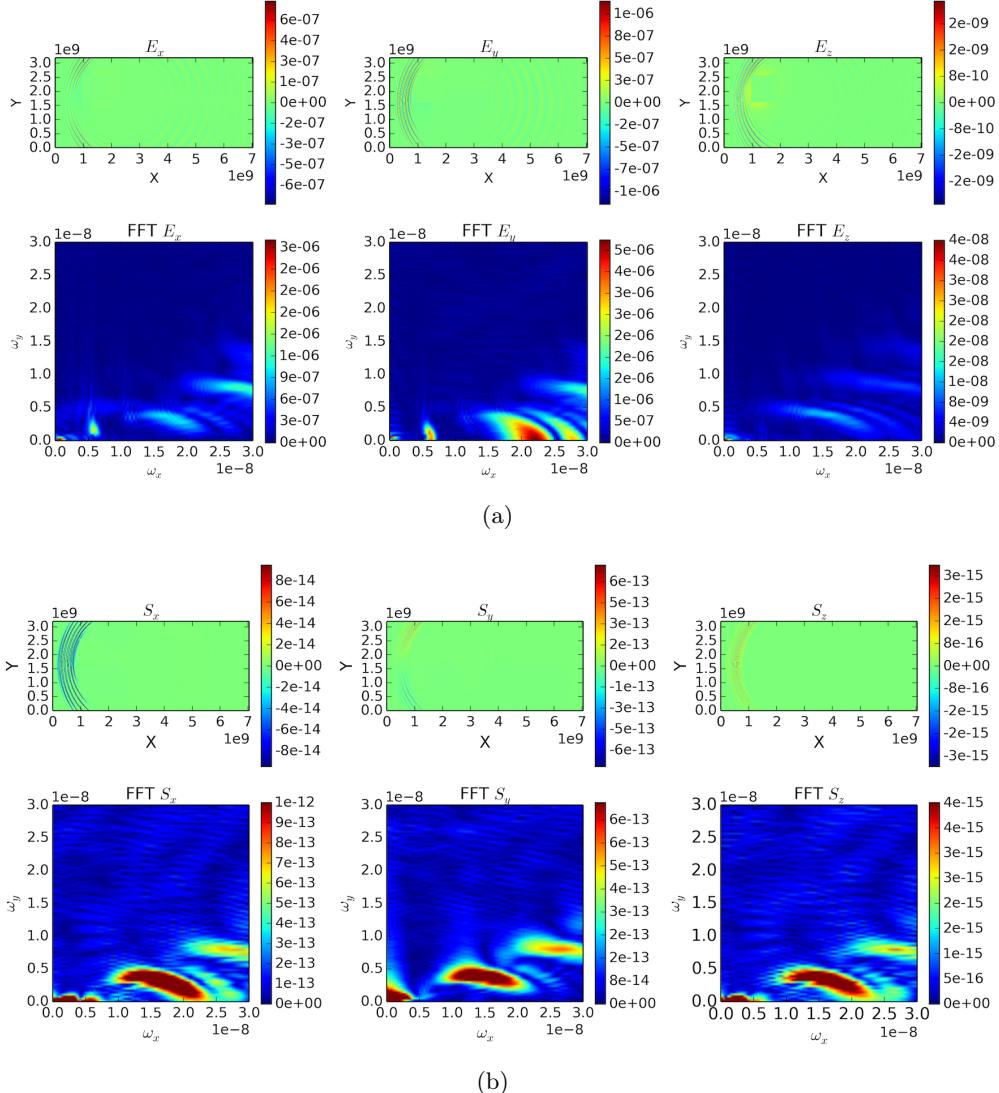


Figure 4.30.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.4$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

CHAPTER 4. HYBRID FIELD APPROACH

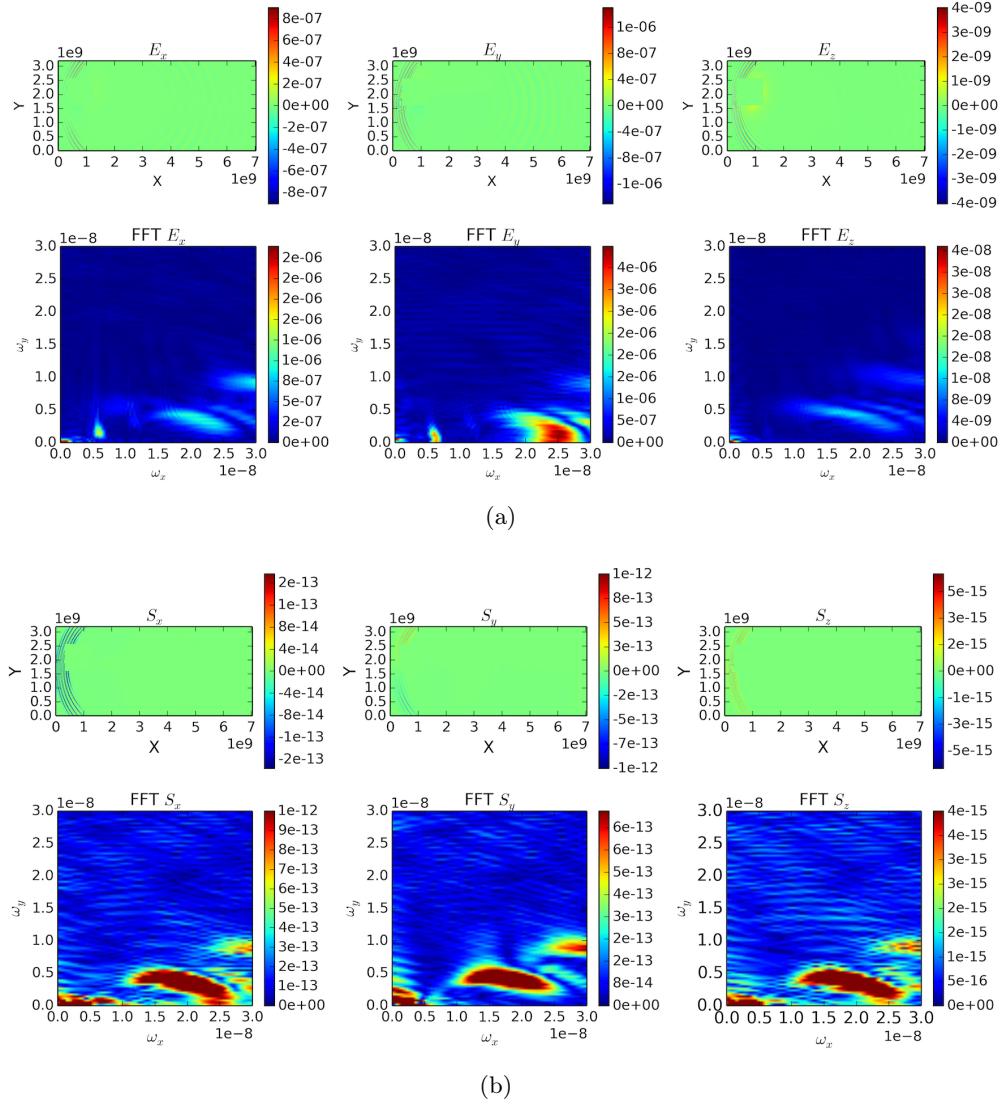


Figure 4.31.: Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.5$. The colorbars in the upper row show the values of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. The colorbars in the lower row show the magnitude of the Fourier components of (a) E_x , E_y , E_z and (b) S_x , S_y and S_z in the xy -plane. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4 / e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2 / (4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$.

From the Fourier analysis of the electric field components, shown in the Figures 4.26 - 4.31, especially from the E_y component, an increase in frequency of the radiated fields can clearly be seen. There is a main peak moving from low frequencies to high frequencies along the ω_x -axis as γ increases. This behavior is as expected from Section 1.3.3. While interacting with the laser pulse the particle is accelerated causing radiation parallel to the direction of acceleration. Whereas at low energies the trajectory of the particle drifts dominantly along the y -direction the particles moves more along the x -direction at higher energies. The reason is that the time the particle interacts with the laser pulse decreases with higher values of γ . The less time the particle interacts with the laser pulse the less it gets deflected in y -direction. Therefore, ω_y frequencies appear more often at lower than at higher values of γ . The same argument holds for the Fourier spectrum of the Poynting vector \vec{S} . As the radiation fields become more and more directed along the x -direction with increasing γ the energy flux \vec{S} becomes more directed as well.

5

Uniaxial Perfectly Matched Layer

5.1 Introduction

With all simulations there is the topic about boundary conditions. This chapter deals with that important aspect. Boundary conditions apply at the edge of the numerical grid and dictate how the solution is treated, when approaching the edge. The Dirichlet boundary condition, for instance, sets the solution to zero, leading to reflections¹. Periodic boundary conditions extend the grid by setting one edge on top of the opposite one. Solutions leaving the grid to the right enter the simulation area to the left again. When solving differential equations numerically one always has to think about, which boundary conditions apply best for the problem at hand. The most commonly used ones are the two aforementioned ones. Dirichlet boundary conditions apply for physical problems involving fast decaying solutions in space so that truncation does not matter, as long as the numerical grid is large enough. Other problems, like periodic structures, can be simulated with periodic boundary conditions. However, one of the most challenging problems to truncate involve wave equations, like in our case, since their solution only decays slowly². Using periodic, Dirichlet or Neumann boundary conditions will lead to unacceptable artifacts from boundary reflections. One approach is the *absorbing boundary condition* [20]. It turns out that this approach works just fine for the one dimensional case, i.e. waves traveling in only one direction. However, most problems - like the one in the present work - include three dimensions. Fortunately in 1994 Berenger came up with a new idea presented in his seminal paper [2]. Instead of having an absorbing boundary condition he rather proposed an absorbing boundary layer. The layer is part of the grid and has a specified width. When a wave enters the layer, it is attenuated by the absorption and decays exponentially. Now the problem remains, that the absorbing layer represents an interface, where the impedance change, which in turn causes reflection and

¹It's like a rope with a fixed connection to a wall. Since the energy cannot be transferred to the wall, all energy is reflected, causing the incident wave to move back.

²Energy conservation requires the energy flux $|S| \propto |E|^2$ to decay inversely proportional to the surface area $\sim r^2$ in three dimensions. Therefore, $|E| \propto \frac{1}{r}$.

transmission. Berenger, however, showed that it's possible to construct a medium, which does not reflect any incident wave, independent of angle, polarization and frequency. A so called PML - *Perfectly Matched Layer*. Aside from Berenger's original formulation -the *split field* PML - some equivalent descriptions of PML came up over the years. For one there are the *uniaxial* PML (UPML), which are used in this thesis and are explained in more detail below. The most elegant and general description is the *stretched coordinate* PML [6, 17, 21], which is based upon an analytic continuation of Maxwell's equations into complex spatial coordinates. Both the split field PML and the UPML can be derived from this approach.

5.2 Derivation

The most common formulation is the UPML. Following [19, 20] an overview of the most important steps is presented to understand why UPML works and how to implement it. It will turn out that the medium inside our PML needs to be anisotropic. Why that is can be seen later. Being able to describe anisotropic phenomenon one needs to incorporate anisotropy into Maxwell equations. One can do this by making the permittivity ϵ_r and permeability μ_r tensors³. The time harmonic Maxwell's Curl equations then read

$$\begin{aligned}\vec{\nabla} \times \vec{E}(\omega) &= -j\omega\mu_0 [\mu_r] \vec{H}(\omega), \\ \vec{\nabla} \times \vec{H}(\omega) &= \sigma \vec{E}(\omega) + j\omega \vec{D}(\omega),\end{aligned}\tag{5.1}$$

with the relation $\vec{D}(\omega) = \epsilon_0 [\epsilon_r] \vec{E}$ and conductivity σ . As usual, ϵ_0 and ϵ_r are the vacuum and relative permittivity respectively. One starts in the frequency domain, because PML is a frequency domain concept and thus it's easier to work with in frequency domain. However, to get the update equations, a back transform to time domain is required. The

³There are three types of anisotropic media:

$$\text{isotropic: } \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix}, \text{ uniaxial: } \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{bmatrix}, \text{ biaxial: } \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}.$$

It is also possible to always choose a coordinate system such that the tensor becomes diagonal.

update equations then need to be implemented. Equation (5.1) can be rewritten as

$$\begin{aligned} \begin{pmatrix} 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix} \begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix} &= j\omega\epsilon_0 \begin{bmatrix} \epsilon_x + \frac{\sigma_x^E}{j\omega} & 0 & 0 \\ 0 & \epsilon_y + \frac{\sigma_y^E}{j\omega} & 0 \\ 0 & 0 & \epsilon_z + \frac{\sigma_z^E}{j\omega} \end{bmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix}, \\ \begin{pmatrix} 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} &= -j\omega\mu_0 \begin{bmatrix} \mu_x + \frac{\sigma_x^H}{j\omega} & 0 & 0 \\ 0 & \mu_y + \frac{\sigma_y^H}{j\omega} & 0 \\ 0 & 0 & \mu_z + \frac{\sigma_z^H}{j\omega} \end{bmatrix} \begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix}, \end{aligned} \quad (5.2)$$

Next reflection and transmission at a surface is discussed. In the case of a wave with arbitrary polarization and angle of incidence, illustrated in Figure 5.1, one can use *Fresnel* equations.⁴ It can be seen that the reflection is a consequence of a change in impedance and also critically angle dependent. It's only possible to match the impedances at one specific angle. However, by introducing anisotropy one can solve this problem. It is known that the impedance is given by

$$\eta = \sqrt{\frac{\mu}{\epsilon}}. \quad (5.4)$$

Thus, to keep the impedances matched throughout the entire grid, including the absorbing layer, one needs $[\epsilon_r] = [\mu_r]$. This is called an doubly anisotropic media. For convenience one defines

$$[s] := [\epsilon_r] = [\mu_r] = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}, \quad (5.5)$$

⁴

$$\begin{aligned} r_{TE} &= \frac{\eta_2 \cos(\theta_1) - \eta_1 \cos(\theta_2)}{\eta_2 \cos(\theta_1) + \eta_1 \cos(\theta_2)}, \\ t_{TE} &= \frac{2\eta_2 \cos(\theta_1)}{\eta_2 \cos(\theta_1) + \eta_1 \cos(\theta_2)}, \\ r_{TM} &= \frac{\eta_2 \cos(\theta_2) - \eta_1 \cos(\theta_1)}{\eta_1 \cos(\theta_1) + \eta_2 \cos(\theta_2)}, \\ t_{TM} &= \frac{2\eta_1 \cos(\theta_1)}{\eta_1 \cos(\theta_1) + \eta_2 \cos(\theta_2)}, \end{aligned} \quad (5.3)$$

where η_i is the impedance in region i with refractive index n_i , θ_1 the angle of incidence and θ_2 the angle of transmission.

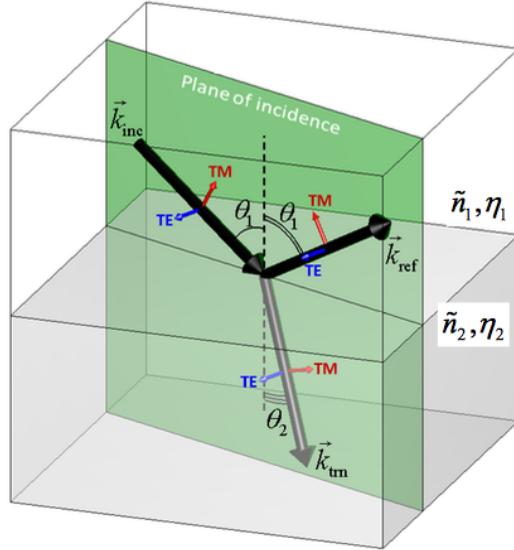


Figure 5.1.: Wave with wavevector \vec{k} , arbitrary polarization and angle of incidence θ at a surface with impedance η_i and refractive index n_i [19].

with a, b, c to be determined. Considering anisotropy the Fresnel equations change. For example, for an incident wave traveling along z-axis the Fresnel equations read

$$\begin{aligned} r_{TE} &= \frac{\sqrt{a} \cos(\theta_1) - \sqrt{b} \cos(\theta_2)}{\sqrt{a} \cos(\theta_1) + \sqrt{b} \cos(\theta_2)}, \\ t_{TE} &= \frac{\sqrt{a} \cos(\theta_1)}{\sqrt{a} \cos(\theta_1) + \sqrt{b} \cos(\theta_2)}, \\ r_{TM} &= \frac{-\sqrt{a} \cos(\theta_1) + \sqrt{b} \cos(\theta_2)}{\sqrt{a} \cos(\theta_1) + \sqrt{b} \cos(\theta_2)}, \\ t_{TM} &= \frac{\sqrt{a} \cos(\theta_1)}{\sqrt{b} \cos(\theta_1) + \sqrt{a} \cos(\theta_2)}, \end{aligned} \quad (5.6)$$

where θ_1 the angle of incidence and θ_2 the angle of transmission. These two are connected via Snell's law

$$\sin(\theta_1) = \sqrt{bc} \sin \theta_2. \quad (5.7)$$

Now conditions need to be found for a, b, c to make the Fresnel equations (5.6) independent of angle and simultaneously let the reflection parameter vanish. By choosing $\sqrt{bc} = 1$ Snell's law simplifies to $\theta_1 = \theta_2$. Thus, the reflection parameters reduce to

$$\begin{aligned} r_{TE} &= \frac{\sqrt{a} - \sqrt{b}}{\sqrt{a} + \sqrt{b}}, \\ r_{TM} &= \frac{-\sqrt{a} + \sqrt{b}}{\sqrt{a} + \sqrt{b}}. \end{aligned} \quad (5.8)$$

With this choice the reflection is independent of angle. Furthermore, if $r_{TE} \stackrel{!}{=} 0$ and $r_{TM} \stackrel{!}{=} 0$ holds, $a \stackrel{!}{=} b$ follows. To sum it up, a, b, c need to fulfill

$$a = b = \frac{1}{c}. \quad (5.9)$$

Therefore, the tensor $[s_z]$ needs to look like this

$$[s_z] = \begin{bmatrix} s_z & 0 & 0 \\ 0 & s_z & 0 \\ 0 & 0 & \frac{1}{s_z} \end{bmatrix}. \quad (5.10)$$

Recall that this is an uniaxial tensor. This is the reason why this approach is called Uniaxial Perfectly Matched Layer. Analogously one gets

$$[s_y] = \begin{bmatrix} s_y & 0 & 0 \\ 0 & \frac{1}{s_z} & 0 \\ 0 & 0 & s_y \end{bmatrix} \text{ and } [s_x] = \begin{bmatrix} \frac{1}{s_z} & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{bmatrix}, \quad (5.11)$$

for waves traveling in y and x direction respectively. These can be combined in a single tensor [20]

$$[s] := [s_x] [s_y] [s_z] = \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix}, \quad (5.12)$$

where

$$\begin{aligned} s_x(x) &= \kappa_x + \frac{\sigma'_x(x)}{j\omega_0\epsilon_0}, \\ s_y(y) &= \kappa_y + \frac{\sigma'_y(y)}{j\omega_0\epsilon_0}, \\ s_z(z) &= \kappa_z + \frac{\sigma'_z(z)}{j\omega_0\epsilon_0}. \end{aligned} \quad (5.13)$$

The parameters σ' and κ will be specified later. Having this, the Maxwell equations (5.1) now read

$$\begin{aligned}\vec{\nabla} \times \vec{E}(\omega) &= -j\omega\mu_0 [\mu_r] [s] \vec{H}(\omega), \\ \vec{\nabla} \times \vec{H}(\omega) &= \sigma \vec{E}(\omega) + j\omega [s] \vec{D}(\omega).\end{aligned}\quad (5.14)$$

Before deriving the update equations the electric field is normalized as follows

$$\begin{aligned}\vec{E} &= \sqrt{\frac{\epsilon_0}{\mu_0}} \vec{E} = \frac{1}{\eta_0} \vec{E} \\ \vec{D} &= \frac{1}{\sqrt{\mu_0 \epsilon_0}} \vec{E} = c_0 \vec{E},\end{aligned}\quad (5.15)$$

which yields

$$\begin{aligned}\vec{\nabla} \times \vec{E}(\omega) &= -j\omega\mu_0 \frac{[\mu_r]}{c_0} [s] \vec{H}(\omega), \\ \vec{\nabla} \times \vec{H}(\omega) &= \eta_0 \sigma \vec{E}(\omega) + \frac{j\omega}{c_0} [s] \vec{D}(\omega).\end{aligned}\quad (5.16)$$

5.3 Update equations

The update equations are now derived from (5.16). For simplicity, only Ampere's law with the additional assumption that the interior of the simulation area has no loss ($\sigma = 0$) is considered. Only the PML will have loss ($\sigma' \neq 0$). Ampere's law rewritten in matrix form then simplifies to

$$\begin{pmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} \end{pmatrix} = j\omega \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \begin{pmatrix} \tilde{D}x \\ \tilde{D}y \\ \tilde{D}z \end{pmatrix}. \quad (5.17)$$

Substituting (5.13) into (5.17) and using the Fourier identity

$$\mathcal{F} \left\{ \frac{\partial^n}{\partial t^n} g(t) \right\} = (j\omega)^n G(\omega), \quad (5.18)$$

where g is a piecewise \mathcal{C}^1 function and $g \in \mathcal{L}^1(\mathbb{R})$ yields the following system of time domain equations

$$\begin{pmatrix} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \\ \frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} \end{pmatrix} = \frac{\partial}{\partial t} \begin{bmatrix} \kappa_x & 0 & 0 \\ 0 & \kappa_y & 0 \\ 0 & 0 & \kappa_z \end{bmatrix} \begin{pmatrix} \tilde{D}x \\ \tilde{D}y \\ \tilde{D}z \end{pmatrix} + \frac{1}{\epsilon_0} \begin{bmatrix} \sigma'_x & 0 & 0 \\ 0 & \sigma'_y & 0 \\ 0 & 0 & \sigma'_z \end{bmatrix} \begin{pmatrix} \tilde{D}x \\ \tilde{D}y \\ \tilde{D}z \end{pmatrix}. \quad (5.19)$$

The system (5.19) can be discretized with Yee-scheme, as explained in Section 4.1 and using central differences. Because it's quite lengthy it is omitted here and instead referred to literature (see [19, 20]). For instance, the D_x update is given by

$$\begin{aligned} \mathbf{D}_x|_{(i+\frac{1}{2},j,k)}^{n+1} = & \left(\frac{2\epsilon_0\kappa_y - \sigma'_y h}{2\epsilon_0\kappa_y + \sigma'_y h} \right) D_x|_{i+\frac{1}{2},j,k}^n + \left(\frac{2\epsilon_0 h}{2\epsilon_0\kappa_y + \sigma'_y h} \right) \\ & \left(\frac{H_z|_{(i+\frac{1}{2},j+\frac{1}{2},k)}^{n+\frac{1}{2}} - H_z|_{(i+\frac{1}{2},j-\frac{1}{2},k)}^{n+\frac{1}{2}}}{\Delta y} - \frac{H_y|_{(i+\frac{1}{2},j,k+\frac{1}{2})}^{n+\frac{1}{2}} - H_y|_{(i+\frac{1}{2},j,k-\frac{1}{2})}^{n+\frac{1}{2}}}{\Delta z} \right). \end{aligned} \quad (5.20)$$

From $\vec{D}(\omega) = \epsilon_0 [\epsilon_r] \vec{E}$ the update equation for \vec{E} can be derived

$$\begin{aligned} E_x|_{(i+\frac{1}{2},j,k)}^{n+1} = & \left(\frac{2\epsilon_0\kappa_z - \sigma'_z h}{2\epsilon_0\kappa_z + \sigma'_z h} \right) E_x|_{i+\frac{1}{2},j,k}^n + \left(\frac{1}{(2\epsilon_0\kappa_z + \sigma'_z h)\epsilon} \right) \\ & \left((2\epsilon_0\kappa_x + \sigma'_x h) \mathbf{D}_x|_{i+\frac{1}{2},j,k}^{n+1} - (2\epsilon_0\kappa_x - \sigma'_x h) D_x|_{i+\frac{1}{2},j,k}^n \right). \end{aligned} \quad (5.21)$$

As can be seen the update equation for $E_x|_{(i+\frac{1}{2},j,k)}^{n+1}$ contains the term $\mathbf{D}_x|_{i+\frac{1}{2},j,k}^{n+1}$. Consequently, the update process of E_x requires an update of D_x first. Meaning, D_x is calculated for the next time step first and then this result is used to update E_x . The same procedure applies for the update equations of B_x and H_x

$$\begin{aligned} B_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{3}{2}} = & \left(\frac{2\epsilon_0\kappa_y - \sigma'_y h}{2\epsilon_0\kappa_y + \sigma'_y h} \right) B_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{1}{2}} + \left(\frac{2\epsilon_0 h}{2\epsilon_0\kappa_y + \sigma'_y h} \right) \\ & \left(\frac{E_z|_{(i,j+1,k+\frac{1}{2})}^{n+1} - E_z|_{(i,j,k+\frac{1}{2})}^{n+1}}{\Delta y} - \frac{E_y|_{(i,j+\frac{1}{2},k+1)}^{n+1} - E_y|_{(i,j+\frac{1}{2},k)}^{n+1}}{\Delta z} \right), \end{aligned} \quad (5.22)$$

$$\begin{aligned} H_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{3}{2}} = & \left(\frac{2\epsilon_0\kappa_z - \sigma'_z h}{2\epsilon_0\kappa_z + \sigma'_z h} \right) H_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{1}{2}} + \left(\frac{1}{(2\epsilon_0\kappa_z + \sigma'_z h)\mu} \right) \\ & \left((2\epsilon_0\kappa_x + \sigma'_x h) B_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{3}{2}} - (2\epsilon_0\kappa_x - \sigma'_x h) B_x|_{(i,j+\frac{1}{2},k+\frac{1}{2})}^{n+\frac{1}{2}} \right). \end{aligned} \quad (5.23)$$

Notice, that if the PML is turned off, i.e. $\sigma' = 0$ and $\kappa = 1$ the prefactors simplify drastically. Plugging in (5.20) into (5.21) and also (5.22) into (5.23) result in the discretized Maxwell equations derived in (4.1). Of course, the update equations of the other field components can be derived analogously.

5.4 Implementation of UPML in FDTD

Finally, this Section shows how to implement UPML into an existing FDTD simulation. Since all the prefactors are constants it's advisable to calculate them beforehand. Here is a snippet of the code used in the present work

```
Grid->upml1E[ j ] = (2*kappa-sigma*dt)/(2*kappa+sigma*dt);
Grid->upml2E[ j ] = (2*dt)/(2*kappa+sigma*dt);
Grid->upml3E[ k ] = (2*kappa-sigma*dt)/(2*kappa+sigma*dt);
Grid->upml4E[ k ] = 1.0/(2*kappa+sigma*dt);
Grid->upml5E[ i ] = (2*kappa+sigma*dt);
Grid->upml6E[ i ] = (2*kappa-sigma*dt);
```

The same applies for the H components. From a theoretical point of view a reflectionless PML is possible. In practice, however, numerical artifacts arise, due to finite spatial sampling [20]. The change from σ in the interior to σ' in the PML is effectively a local discontinuity and thus causes reflections. For best performance, Berenger proposed that the PML losses σ' shall gradually rise from zero at $x = 0$ to σ_{max} at $x = d$ where d is the PML layer width [2]

$$\begin{aligned}\sigma_x(x) &= \left(\frac{x}{d}\right)^m \sigma_{x,max}, \\ \kappa_x(x) &= 1 + (\kappa_{x,max} - 1) \left(\frac{x}{d}\right)^m.\end{aligned}\tag{5.24}$$

The parameter m defines the grading and is typically $3 \leq m \leq 4$ in FDTD simulations. $m = 3.5$ has been found to be a good choice. Through a lot of testing and experimenting σ_{max} was found to be [20]

$$\sigma_x^{opt} = \frac{0.8(m+1)}{\eta\Delta x},\tag{5.25}$$

where $\eta = \sqrt{\frac{\mu}{\epsilon}}$. This of course also applies for the other components of σ . Figure 5.2 shows how a simulation with UPML looks like

CHAPTER 5. UNIAXIAL PERFECTLY MATCHED LAYER

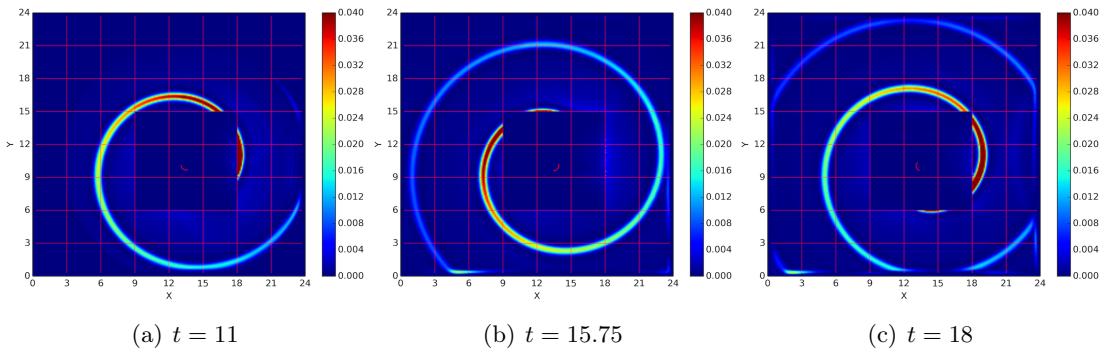


Figure 5.2.: Simulation of an electron in constant B_z field using UPML at (a) $t = 11$, (b) $t = 15.75$ and (c) $t = 18$. The electron has an energy of $\gamma = 1.2$. A constant $B_z = 1$ is used as external field. The colorbar shows the value of $|\vec{E}|^2$. \vec{E} and $c\vec{B}$ are in units of $4\pi\epsilon_0 m_e^2 c^4/e^3 = 1.77 \cdot 10^{20} \text{ V m}^{-1}$, the length is in units of $e^2/(4\pi\epsilon_0 m_e c^2) = 2.82 \cdot 10^{-15} \text{ m}$ and the time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24} \text{ s}$.

Part III.

Summary & Outlook

This thesis is build upon the hybrid field approach presented in [11]. The framework outlined in [11] is written in C++. In the present work the framework has been rewritten in C and furthermore has been extended with new features like a reverse Boris push to extend the particle history to times prior to the start of the simulation, the trilinear interpolation of fields at the particle positions, the effective storage of precalculated fields needed for a more efficient calculation of the MD scheme, and UPMLs. A stable code base has been produced, including Doxygen documentation in order to allow other users to understand and use the code more easily. In order to test the implementation of the MD scheme of the present work, a scattering process of an electron in an electromagnetic wave has been simulated and can be compared with real world experiments. By analyzing the Fourier components of the scattered electric field and the associated Poynting vector, it has been shown that with increasing energy γ of the electron the radiation frequency increases and the radiation becomes more directed.

As already outlined in Section 4.6 the storage of particle positions must be optimized, as soon as particle numbers become large. In the present work it has been shown, that a plausible criterium for finding particles, who's position and velocity has to be stored, will depend on the velocity of the particles.

Another feature of the numerical scheme in the present work, that can be optimized are the UPMLs. Problems of UPMLs occur when Liénard-Wiechert fields are calculated from the extended particle history prior to the simulation. If field values are set into the UPML region, they will stay there or even worse diverge causing strange artefacts at the boundaries. The UPML layer should be excluded from the field initialization. Furthermore, the UPMLs have more specific problems [20]. They, however, are not specifically related to the present work.

Part IV.

Appendix

A

Gauge Transformations

In this chapter it is shown that the gauge fields leave both the electric and magnetic fields and the corresponding potential equations invariant.

A.1 Gauge invariance of fields

It is shown that $\vec{E}' = \vec{E}$ and $\vec{B}' = \vec{B}$.

$$\begin{aligned}\vec{E}' &= -\vec{\nabla}\varphi' - \frac{\partial\vec{A}'}{\partial t} \\ &= -\vec{\nabla}\left(\varphi - \frac{\partial\psi}{\partial t}\right) - \frac{\partial}{\partial t}\left(\vec{A} + \vec{\nabla}\psi\right) \\ &= -\vec{\nabla}\varphi + \vec{\nabla}\left(\frac{\partial\psi}{\partial t}\right) - \frac{\partial\vec{A}}{\partial t} - \frac{\partial}{\partial t}(\vec{\nabla}\psi) \\ &= \vec{E}\end{aligned}$$

and

$$\begin{aligned}\vec{B}' &= \vec{\nabla} \times \vec{A}' \\ &= \vec{\nabla} \times (\vec{A} + \vec{\nabla}\psi) \\ &= \vec{\nabla} \times \vec{A} + \vec{\nabla} \times (\vec{\nabla}\psi) \\ &= \vec{B}.\end{aligned}$$

A.2 Gauge invariance of potential equations

Now the invariance of the potential equations is shown. Therefore, consider

$$\begin{aligned}
 -\Delta\varphi' - \vec{\nabla} \left(\frac{\partial \vec{A}'}{\partial t} \right) &= \frac{\rho}{\epsilon_0} \\
 \iff -\Delta \left(\varphi - \frac{\partial \psi}{\partial t} \right) - \vec{\nabla} \left[\frac{\partial}{\partial t} (\vec{A} + \vec{\nabla}\psi) \right] &= \frac{\rho}{\epsilon_0} \\
 \iff -\Delta\varphi + \Delta \left(\frac{\partial \psi}{\partial t} \right) - \vec{\nabla} \left(\frac{\partial \vec{A}}{\partial t} \right) - \vec{\nabla} \left[\frac{\partial}{\partial t} (\vec{\nabla}\psi) \right] &= \frac{\rho}{\epsilon_0} \\
 \iff -\Delta\varphi - \vec{\nabla} \left(\frac{\partial \vec{A}}{\partial t} \right) &= \frac{\rho}{\epsilon_0}
 \end{aligned}$$

and

$$\begin{aligned}
 \square \vec{A}' - \vec{\nabla} \left(\vec{\nabla} \vec{A}' + \frac{1}{c^2} \frac{\partial \varphi'}{\partial t} \right) &= -\mu_0 \vec{j} \\
 \iff \hat{\square} (\vec{A} + \vec{\nabla}\psi) - \vec{\nabla} \left[\vec{\nabla} (\vec{A} + \vec{\nabla}\psi) + \frac{1}{c^2} \frac{\partial}{\partial t} \left(\varphi - \frac{\partial \psi}{\partial t} \right) \right] &= -\mu_0 \vec{j} \\
 \iff \hat{\square} \vec{A} + \square (\vec{\nabla}\psi) - \vec{\nabla} \left[\vec{\nabla} \vec{A} + \Delta\psi + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} - \frac{1}{c^2} \frac{\partial^2 \psi}{\partial t^2} \right] &= -\mu_0 \vec{j} \\
 \iff \hat{\square} \vec{A} + \square (\vec{\nabla}\psi) - \vec{\nabla} \left[\vec{\nabla} \vec{A} + \square\psi + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} \right] &= -\mu_0 \vec{j} \\
 \iff \hat{\square} \vec{A} - \vec{\nabla} \left(\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} \right) &= -\mu_0 \vec{j}
 \end{aligned}$$

B

Retarded potential equations fulfill Laurentz gauge

It is shown that the retarded potential equations

$$\begin{aligned}\varphi(\vec{r}, t) &= \frac{1}{4\pi\epsilon_0} \int_V \frac{\rho(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}', \\ \vec{A}(\vec{r}, t) &= \frac{\mu_0}{4\pi} \int_V \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}'\end{aligned}$$

with $t_{ret} := t - \frac{|\vec{r} - \vec{r}'|}{c}$ fulfill the Laurentz gauge

$$\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} = 0.$$

Plugging in yields

$$\begin{aligned}\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} &= \frac{\mu_0}{4\pi} \int_V \vec{\nabla}_{\vec{r}} \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}' + \frac{1}{c^2} \frac{1}{4\pi\epsilon_0} \int_V \frac{\partial}{\partial t} \frac{\rho(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}', \\ &= \frac{\mu_0}{4\pi} \int_V \vec{\nabla}_{\vec{r}} \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} \frac{\partial}{\partial t} \frac{\rho(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}',\end{aligned}$$

where $c^2 = \frac{1}{\mu_0\epsilon_0}$ is used. Now consider the first term in the integrand

$$\begin{aligned}\vec{\nabla}_{\vec{r}} \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} &= \vec{j}(\vec{r}', t_{ret}) \vec{\nabla}_{\vec{r}} \left(\frac{1}{|\vec{r} - \vec{r}'|} \right) + \frac{1}{|\vec{r} - \vec{r}'|} \partial_{t_{ret}} \vec{j}(\vec{r}', t_{ret}) \vec{\nabla}_{\vec{r}} t_{ret} \\ &= -\vec{j}(\vec{r}', t_{ret}) \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3} - \frac{1}{c} \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^2} \partial_{t_{ret}} \vec{j}(\vec{r}', t_{ret})\end{aligned}$$

APPENDIX B. RETARDED POTENTIAL EQUATIONS FULFILL
LAURENTZ GAUGE

On the other hand one also has

$$\begin{aligned} \vec{\nabla}_{\vec{r}'} \cdot \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} &= \vec{j}(\vec{r}', t_{ret}) \vec{\nabla}_{\vec{r}'} \left(\frac{1}{|\vec{r} - \vec{r}'|} \right) + \\ &\quad \frac{1}{|\vec{r} - \vec{r}'|} \left(\vec{\nabla}_{\vec{r}'} \cdot \vec{j}(\vec{r}', t_{ret}) \Big|_{t_{ret}} + \partial_{t_{ret}} \vec{j}(\vec{r}', t_{ret}) \vec{\nabla}_{\vec{r}'} \cdot \vec{t}_{ret} \right) \\ &= \vec{j}(\vec{r}', t_{ret}) \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3} + \\ &\quad \frac{1}{|\vec{r} - \vec{r}'|} \left(\vec{\nabla}_{\vec{r}'} \cdot \vec{j}(\vec{r}', t_{ret}) \Big|_{t_{ret}} + \frac{1}{c} \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|} \partial_{t_{ret}} \vec{j}(\vec{r}', t_{ret}) \right), \end{aligned}$$

which means that one can write

$$\vec{\nabla}_{\vec{r}} \cdot \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} + \vec{\nabla}_{\vec{r}'} \cdot \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} = \frac{\vec{\nabla}_{\vec{r}'} \cdot \vec{j}(\vec{r}', t_{ret}) \Big|_{t_{ret}}}{|\vec{r} - \vec{r}'|}.$$

Plugging in this into (B.1) yields

$$\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} = \frac{\mu_0}{4\pi} \int_V -\vec{\nabla}_{\vec{r}'} \cdot \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} + \underbrace{\frac{\vec{\nabla}_{\vec{r}'} \cdot \vec{j}(\vec{r}', t_{ret}) \Big|_{t_{ret}}}{|\vec{r} - \vec{r}'|}}_{=0 \text{ due to continuity equation}} + \frac{\partial}{\partial t} \frac{\rho(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}'.$$

For the remaining first term *Stokes Law* is used

$$\vec{\nabla} \vec{A} + \frac{1}{c^2} \frac{\partial \varphi}{\partial t} = -\frac{\mu_0}{4\pi} \int_V \vec{\nabla}_{\vec{r}'} \cdot \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{r}' = -\frac{\mu_0}{4\pi} \oint_{\partial V} \frac{\vec{j}(\vec{r}', t_{ret})}{|\vec{r} - \vec{r}'|} d\vec{\sigma} = 0,$$

because the current density vanishes at infinity. That was to show.

C

Softwarestack and Documentation

The following Softwarestack was used to create this thesis:

- IDE: XCode Version 8.3.3 (8E3004b)
- Version Control System: Git
- Version Control IDE: Atlassian SourceTree Version 2.5.2(111)
- Repository: <https://github.com/dsymhoven/masterarbeit>
- Package Manager: MacPorts Version 2.3.5
- Plots: Python2.7
- Texteditor: Atom Version 1.17.2
- Documentation: Doxygen Version 1.8.13
- Documentation: <https://dsymhoven.github.io/masterarbeit/>

D

Normalization

This chapter deals with how to make the basic equations from Section 2.1 dimensionless. Even though it's not necessary, it's always advisable in numerics to have a dimensionless systems for reasons of simplicity and clarity. In addition to the transformation equations the dimensionless Lorentz-Newton equation (2.1), the damping term, i.e. Landau-Lifschitz equation (2.3) and of course the Liénard-Wiechart equations (1.32) is also presented.

To make the physical entities dimensionless a characteristic frequency ω is introduced. Dimensionless parameters are denoted with a tilde.

With the following transformations

$$\begin{aligned}
 t &= \frac{1}{\omega} \tilde{t}, \\
 \vec{r} &= \frac{c}{\omega} \tilde{\vec{r}}, \\
 R &= \frac{c}{\omega} \tilde{R}, \\
 d\tau &= \frac{1}{\omega} d\tilde{s}, \\
 F^{\mu\nu} &= \frac{\omega m_e}{q} \tilde{F}^{\mu\nu}, \\
 F^{\mu\nu}_{\text{ext}} &= \frac{\omega m_e}{e} \tilde{F}^{\mu\nu}_{\text{ext}}, \\
 F^{\mu\nu}_{\text{lw}} &= \frac{q}{4\pi\epsilon_0} \frac{\omega^2}{c^3} \tilde{F}^{\mu\nu}_{\text{lw}}, \\
 F^{\mu\nu} &= F^{\mu\nu}_{\text{ext}} + \frac{q^2}{4\pi\epsilon_0} \frac{\omega^2}{c^3} \tilde{F}^{\mu\nu}_{\text{lw}}
 \end{aligned} \tag{D.1}$$

the aforementioned equations become dimensionless, where $F^{\mu\nu}_{\text{ext}}$ denotes the external fields, $F^{\mu\nu}_{\text{lw}}$ the Liénard-Wiechert fields and $F^{\mu\nu}$ the total fields. Defining the charac-

APPENDIX D. NORMALIZATION

teristic frequency ω such, that

$$\begin{aligned} \frac{q^2}{4\pi\epsilon_0} \frac{\omega^2}{c^3} &\stackrel{!}{=} 1 \\ \implies \omega &= 4\pi\epsilon_0 \frac{m_e c^3}{q^2}, \end{aligned} \tag{D.2}$$

both the external and the Liénard-Wiechert fields have the same units

$$\frac{\omega m_e}{q} = 4\pi\epsilon_0 \frac{m_e^2 c^3}{q^3}. \tag{D.3}$$

In the following the basic equations from Section 2.1 are presented in dimensionless form: The dimensionless Lorentz-Newton equation (2.1) reads

$$\frac{du^\mu}{d\tilde{s}} = \left(\tilde{F}^{\mu\nu}_{\text{lw}}(x^\mu) + \tilde{F}^{\mu\nu}_{\text{ext}}(x^\mu) \right) u^\nu + \tilde{g}^\mu. \tag{D.4}$$

The Landau-Lifschitz equation (2.3) becomes

$$\tilde{g}^\mu = \frac{2}{3} \left(\frac{\partial \tilde{F}^{\mu\nu}}{\partial \tilde{x}^\xi} u^\nu u^\xi - \tilde{F}^{\mu\xi} \tilde{F}_{\nu\xi} u^\nu + (\tilde{F}_{\nu\xi} u^\xi)(\tilde{F}^{\nu\pi} u_\pi) u^\mu \right), \tag{D.5}$$

and finally the Liénard-Wiechart equations (1.32)

$$\begin{aligned} \vec{E}(\vec{r}, \tilde{t}) &= \left(\frac{\vec{n}(\vec{r}, \tilde{t}') - \vec{\beta}(\tilde{t})}{\gamma^2 (1 - \vec{\beta}(\tilde{t}) \cdot \vec{n}(\vec{r}, \tilde{t}'))^3 \tilde{R}^2(\vec{r}, \tilde{t}')} \right. \\ &\quad \left. + \frac{\vec{n}(\vec{r}, \tilde{t}') \times (\vec{n}(\vec{r}, \tilde{t}') - \vec{\beta}(\tilde{t})) \times \dot{\vec{\beta}}(\tilde{t})}{(1 - \vec{\beta}(\tilde{t}) \cdot \vec{n}(\vec{r}, \tilde{t}'))^3 \tilde{R}(\vec{r}, \tilde{t}')} \right) \Big|_{\tilde{t}'=\tilde{t}_{ret}}, \\ \vec{B}(\vec{r}, \tilde{t}) &= \left(\vec{n}(\vec{r}, \tilde{t}') \times \vec{E}(\vec{r}, \tilde{t}) \right) \Big|_{\tilde{t}'=\tilde{t}_{ret}}. \end{aligned} \tag{D.6}$$

List of Figures

1.1.	Illustration of how to calculate the energy emission of a moving particle with charge q .	11
1.2.	Radiation characteristics of an moving charged particle	14
2.1.	procedural error	20
2.2.	Velocity space showing the rotation from \vec{v}_- to \vec{v}'	23
2.3.	Determination of $ \tan(\frac{\theta}{2}) $ from boris roation	24
3.1.	Interpolation of Trajectories	26
3.2.	Cube with its eight corner points A, B, C, ..., H and interpolation point P_f .	27
4.1.	Yee-Box	30
4.2.	Comparison of numerical and analytical dispersion relation of discretized Maxwell equations with Yee-scheme	33
4.3.	Near fields of one and two particles	35
4.4.	One plane of a grid showing the near field and far field of a particle. The near field does not contain any field values of the particle	36
4.5.	Section of a grid with three boxes, two of which are near field boxes (NF) and one being a far field box (FF)	38
4.6.	Simulation of an electron with $\gamma = 1.1$ in a constant $B_z = 1$ field	40
4.7.	Gyration of two electrons with $\gamma = 1.1$ each.	41
4.8.	Particle push from time t to $t + dt$. Since the particle changes its box the near field region also changes	42
4.9.	Simulation of an electron with $\gamma = 1.2$ in a crossed electromagnetic field with $B_z = 1$ and $E_y = 0, 2$	43
4.10.	The faster the particle gets, the longer the history needs to be	44
4.11.	Initial fields for different values of simulation parameter t	46
4.12.	Flow chart showing the relevant function calls when the user sets $t = t_0 > 0$	47
4.13.	External laser pulse at $t = 700 \cdot 10^7$	50
4.14.	Initial conditions: $u_x = 0.1$, $u_y = 0.01$ and $u_z = 0.0$	51

LIST OF FIGURES

4.15. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.0$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	51
4.16. Initial conditions: $u_x = 0.46$, $u_y = 0.01$ and $u_z = 0.0$	52
4.17. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.1$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	52
4.18. Initial conditions: $u_x = 0.67$, $u_y = 0.01$ and $u_z = 0.0$	53
4.19. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.2$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	53
4.20. Initial conditions: $u_x = 0.84$, $u_y = 0.01$ and $u_z = 0.0$	54
4.21. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.3$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	54
4.22. Initial conditions: $u_x = 0.98$, $u_y = 0.01$ and $u_z = 0.0$	55
4.23. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.4$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	55
4.24. Initial conditions: $u_x = 1.12$, $u_y = 0.01$ and $u_z = 0.0$	56
4.25. Ratio of the damping force g^μ and the Lorentz force F_L for $\gamma = 1.5$. The force g^μ is calculated at each time step according to (2.3). Obviously, the damping force g^μ can be neglected. Time is in units of $e^2/(4\pi\epsilon_0 m_e c^3) = 9.40 \cdot 10^{-24}$ s.	56
4.26. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.0$	57
4.27. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.1$	58
4.28. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.2$	59
4.29. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.3$	60

LIST OF FIGURES

4.30. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.4$	61
4.31. Fourier analysis of (a) the electric field components of \vec{E} and (b) the components of the Poynting vector \vec{S} for $\gamma = 1.5$	62
5.1. Wave with wavevector \vec{k} , arbitrary polarization and angle of incidence θ at a surface with impedance η_i and refractive index n_i	67
5.2. Simulation of an electron in constant B_z field using UPML	72

Bibliography

- [1] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *nature*, 324:446–449, December 1986.
- [2] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, October 1994.
- [3] M. Bonitz and D. Semkat. *Introduction to Computational Methods in Many Body Physics*. Rinton Press, 2006.
- [4] J. P. Boris. Relativistic plasma simulation-optimization of a hybrid code. *Proceeding of Fourth Conference on Numerical Simulations of Plasmas*, November 1970.
- [5] A. Bruce Langdon Charles K. Birdsall. *Plasma physics via computer simulation*. IOP, 1991.
- [6] Weng Cho Chew and William H. Weedon. A 3-d perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microwave Opt. Tech. Lett.*, 7:599–604, 1994.
- [7] B. I. Cohen, M. A. Mostrom, D. R. Nicholson, A. N. Kaufman, C. E. Max, and A. B. Langdon. Simulation of laser beat heating of a plasma. *Physics of Fluids*, 18:470–474, April 1975.
- [8] John M. Dawson. Particle simulation of plasmas. *Rev. Mod. Phys.*, 55:403–447, Apr 1983.
- [9] P. A. M. Dirac. The quantum theory of the electron. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 117(778):610–624, 1928.
- [10] R. A. Fonseca, L. O. Silva, F. S. Tsung, V. K. Decyk, W. Lu, C. Ren, W. B. Mori, S. Deng, S. Lee, T. Katsouleas, and J. C. Adam. *OSIRIS: A Three-Dimensional, Fully Relativistic Particle in Cell Code for Modeling Plasma Based Accelerators*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

BIBLIOGRAPHY

- [11] C. Herzing. *Die molekulardynamische Simulation strahlender Plasmen*. PhD thesis, Ludwig Maximilians Universität München, Lehrstuhl für Plasma & Computational Physics, 2016.
- [12] John David Jackson. *Classical electrodynamics*. Wiley, New York, NY, 3rd ed. edition, 1999.
- [13] Christian Kanzow. *Numerische Mathematik II*. 2005. [Online; accessed 13-Oktober-2016].
- [14] E.M.Lifschitz L.D.Landau. *Lehrbuch der klassischen Physik*, volume 2 of *Klassische Feldtheorie*. Harri (Deutsch), 12 edition, 1997.
- [15] W. Nolting. *Grundkurs Theoretische Physik 3*: Grundkurs Theoretische Physik / Wolfgang Nolting. Springer, 2004.
- [16] A. Pukhov. Three-dimensional electromagnetic relativistic particle-in-cell code VLPL (Virtual Laser Plasma Lab). *Journal of Plasma Physics*, 61(03):425–433, April 1999.
- [17] C. M. Rappaport. Perfectly matched absorbing boundary conditions based on anisotropic lossy mapping of space. *IEEE Microwave and Guided Wave Letters*, 5(3):90–92, Mar 1995.
- [18] Chuang Ren. *Introduction to Particle-in-Cell Methods in Plasma Simulations*. 2011. [Online; accessed 14-Mai-2017].
- [19] Raymond Rumpf. Electromagnetic analysis using finite-difference time-domain. University Lecture, 2014.
- [20] A. Taflove and S.C. Hagness. *Computational Electrodynamics: The Finite-difference Time-domain Method*. Artech House antennas and propagation library. Artech House, 2005.
- [21] F. L. Teixeira and W. C. Chew. General closed-form pml constitutive tensors to match arbitrary bianisotropic and dispersive linear media. *IEEE Microwave and Guided Wave Letters*, 8(6):223–225, Jun 1998.
- [22] Kane S. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, pages 302–307, 1966.

Declaration

Erklärung:

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

München, Datum der Abgabe

München, 17.07.2017, David Symhoven