Diploma work

Finite-Element-Time-Domain (FETD) Simulation of charged Particles in a Cavity

Marcus Wittberger

Responsibility: Prof. Dr. R. Jeltsch
Supervision: Prof. Dr. P. Arbenz
Dr. A. Adelmann

2nd November 2005 - 1st March 2006

ETH Zürich Computational Science and
PSI SLS-TME-TA-2006-0288

Contents

1 Introduction

1.1 Abstract

The subject of this diploma work is the simulation of a plasm. A plasma is a gas with a very small density composed of free charged particles (ions). An important goal is to understand the dynamic of the particles (here protons) and the electromagnetic fields considering particle accelerators. This work is a part of the collaboration between the Swiss Federal Institute of Technology and the Paul Scherrer Institute in Villigen. I was supervised by Peter Arbenz (ETH, Institute of Computational Science) and Andreas Adelmann (PSI, Large Research Facilities).

The protons are located in a bounded domain which is surrounded by a perfect electric conductor (PEC). This implies that the arising electric field is perpendicular on the boundary. The dynamic of the electromagnetic field is described by the Maxwell equations. The fields are generating forces which act on the particles and change thus the charge density ρ and the current density $\mathbf{j}$. The latter quantities are the origins of the fields and appear in the Maxwell equations on the right hand side as source. Thus the task can be formulated as a boundary value problem, i.e. as partial differential equations (Maxwell equations) provided with boundary conditions.

Boundary value problems can be treated with different methods. One historically important method is called *finite differences* (FD). Required are two structured, hexaedral meshes which are slightly displaced to each other. The values of the field is calculated on the corners of the hexaedras. Via these values the spacial and temporal derivatives are computed in order to a Taylor approximation. This method is the most common for electromagnetic problems. For a long time it was the only accomplishable approach. One major disadvantage is the restriction to very basic geometries. Accessorily, adaptive mesh refinement is provided just in a constricted way.

Since a few decades the method of *Finite Elements* (FE) is an alternative. A relatively abstract formulation provides to include arbitrary geometries and potentiates adaptive mesh refinement. In this diploma work Finite Elements are used. A short description of this method is given in a following section.

An american group of researchers has recently developed an FE-library called *FEMSTER*. The library has been implemented in object oriented C++. It is freely available for the purpose of research. To get a higher accuracy one can either refine the mesh (h) or one can boost the polynomial degree (p). It is conjectured that for a given accuracy the system of equations gets smaller in order to rise the polynomial degree. Unlike most of the Finite Element Frameworks, in FEMSTER it is possible to select Finite Elements of any polynomial degree. An other advancement of FEMSTER is the integration of differential forms as basis for Finite Elements. Differential forms and electromagnetic quantities have some remarkably connections. This approach is fairly recent and is promoted to a large amount by the FEMSTER group and Ralf Hiptmair (ETH Zürich).

An additional quite capacious program library is *IPPL* (*Independent Parallel Particle Layer*). Its main developer is Andreas Adelmann. *IPPL* provides the treatment of large amounts of particles. A main pillar is the support of parallel computing.

The goal of this diploma work is to merge these two program libraries in such a way that electromagnetic fields and particles can be correct described by the Maxwell equations. Given an initial condition, the system should be evolved in order to time domain. Of course the program is requested to run as efficiently as possible. Numerical errors should be estimated.

1.2 Aufgabenstellung und Ziel

In dieser Diplomarbeit geht es um Simulationen von Plasmen. Ein Plasma ist ein Gas mit einer sehr geringen Dichte das aus freien Ladungsträgern (Ionen) besteht. Das Ziel ist ein besseres Verständnis für die Dynamik von Teilchen (Protonen) und elektromagnetischen Feldern in Teilchenbeschleunigern. Die Arbeit ist ein Teilergebnis aus der Zusammenarbeit zwischen der Eidgenössischen Technischen Hochschule Zürich und dem Paul Scherrer Institut in Villigen. Die Betreuer waren Peter Arbenz (ETH, Institute of Computational Science) und Andreas Adelmann (PSI, Large Research Facilities).

Die Protonen befinden sich in einem abgeschlossenen Gebiet das mit einem perfekt leitenden Material umgeben ist. Dies hat zur Folge, dass die sich ergebenden elektrischen Felder senkrecht auf dem Gebietsrand stehen. Die magnetischen Felder auf dem Rand müssen dann tangential zum Rand sein. Die Dynamik der elektromagnetischen Felder wird beschrieben durch die Maxwell-Gleichungen. Die Felder erzeugen Kräfte, welche auf die Teilchen wirken und so die Ladungsdichte ρ und die Stromdichte $\mathbf{j}$ verändern. Die letzteren Grössen sind die Ursache der Felder und stehen in den Maxwell Gleichungen als Quellen auf der rechten Seite. Die Aufgabenstellung kann also formuliert werden durch ein Randwertproblem, das heisst durch partielle Differentialgleichungen (die Maxwell Gleichungen), die mit Randbedingungen versehen sind.

Randwertprobleme können mit verschiedenen Ansätzen gelöst werden. Historisch wichtig ist die Methode der *Finiten Differenzen* (FD). Hier werden zwei leicht verschobene, strukturierte hexaedrale Gitter des Gebietes benötigt. Auf den Hexaeder-Ecken werden dann die Felder berechnet. Mit den diskreten Feldern werden schliesslich die Ableitungen mit Hilfe einer Taylor Näherung berechnet. Diese Methode ist die bekannteste für elektromagnetische Probleme und war lange Zeit auch die einzig parktizierbare. Leider sind nur sehr einfache Geometrien damit berechenbar und Gitterverfeinerungen sind nur in beschränktem Masse möglich.

Seit wenigen Jahrzehnten ist die Methode der *Finiten Elemente* (FE) eine Alternative. Eine relativ abstrakte Formulierung macht das einfachere Einbinden von beliebigen Geometrien und eine adaptive Gitterverfeinerung möglich. In dieser Arbeit wird diese Methode verwendet. Eine kurze Beschreibung dieses Ansatzes findet man in einem späteren Abschnitt.

Eine amerikanische Forschergruppe entwickelte kürzlich eine FE-Bibliothek namens *FEMSTER*. Die Bibliothek wurde in objektorientiertem C++ implementiert und ist zu Forschungszwecken frei verfügbar. In FEMSTER können finite Elemente beliebiger Polynom-Ordnung ausgewählt werden. Man verspricht sich von dieser höheren p-Ordnung - im Gegensatz zur h-Gitterverfeinerung - kleinere Gleichungssysteme für eine festgesetzte Genauigkeit. Neu im Ansatz von FEMSTER ist, dass die finiten Elemente auf Differentialformen basieren. Die Maxwell Gleichungen mit diesem mathematischen Gebilde anzupacken bildet eine neue Theorie, zu dessen Entstehen sowohl die FEMSTER-Gruppe als auch Ralf Hiptmair (ETH Zürich) einen grossen Beitrag geleistet haben.

Eine weitere grosse Programm-Bibliothek ist *IPPL* (*Independent Parallel Particle Layer*). Einer ihrer Hauptentwickler war Andreas Adelmann. Mit dieser Bibliothek können grosse Mengen von Teilchen behandelt werden. Ein wichtiges Standbein ist dabei die Unterstützung von parallelen Computernetzwerken.

Ziel meiner Arbeit ist, die beiden Programm Bibliotheken so zusammen zu fügen, dass die elektromagnetischen Felder und Teilchen durch die Maxwell Gleichungen korrekt beschrieben werden. Ausgehend von einer Anfangsbedingung soll das System dann zeitlich entwickelt werden. Natürlich soll das Programm möglichst effizient arbeiten. Die nicht zu vermeidenden numerischen Fehler sollen abgeschätzt werden.

1.3 Acknowledgement

I would like to thank Prof. Dr. Peter Arbenz and Dr. Andreas Adelmann for giving me the chance to work on such an interesting topic! I acknowledge greatfully that Prof. Dr. Rolf Jeltsch has accepted to take responsibility for this diploma work!

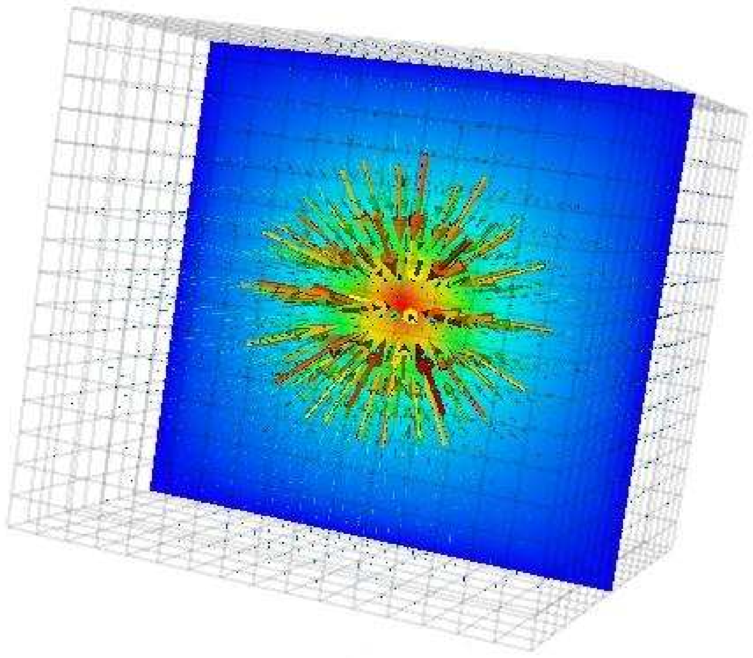


Figure 1.1: Solution of a poisson problem: monopole in a conducting cubic box

1.4 Good to know

- In general, bold letters represent vector fields or matrices whereas ordinary letters correspond to scalars.
- In this text *cell* and *element* are synonyms.

Constants

As the calculations are performed for vacuum, the dielectric and magnetic properties are constant. For the sake of simplicity, ϵ_0 and μ_0 are set equal to 1. This implies that the velocity of light c is equal to 1, too.

Scalar products

In the FEM often integrals over a domain Ω have to be performed. If a and b are functions, we agree on writing for the scalar product

$$(a, b)_\Omega := \int_\Omega a \cdot b \cdot d\mathbf{x}$$

In the context of differential forms this notation is also used

$$(e, f)_\Omega := \int_\Omega e \wedge f$$

Bibliography

The bibliography contains all papers and books I have consulted for the diploma work.

1.5 The finite element method

1.5.1 Variational formulation

For exemple we would like to solve the elliptic boundary value problem

$$-\Delta u(x) = f(x) \qquad \forall x \in \Omega \tag{1.1}$$

$$u(x) = 0 \qquad \forall x \in \partial\Omega \tag{1.2}$$

for a function $u : \mathbb{R}^n \to \mathbb{R}$ in any domain $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega =: \Gamma = \Gamma_D + \Gamma_N$. The finite element method is based on the variational formulation. For that reason we multiply equation (1.1) by any smooth test function v and integrate (by parts) the resulting equation over the whole domain. The variational formulation reads then

Find $u \in V = \{u \in H^1(\Omega) : u|_{\Gamma_D} = 0\}$, so that

$$\underbrace{(\nabla u, \nabla v)_\Omega}_{=:s(u,v)} - \underbrace{\left(\frac{\partial u}{\partial n}, v\right)_\Gamma}_{=0} = \underbrace{(f, v)_\Omega}_{=:l(v)} \qquad \forall v \in V. \tag{1.3}$$

$s(u, v)$ is a bilinear form, $l(v)$ is a linear functional. As the integral is linear, one can split the variational formulation in integrals over parts of the domain (cells Σ) and sum them up.

$$\sum_\Sigma s^\Sigma(u, v) = \sum_\Sigma l^\Sigma(v) \tag{1.4}$$

Now the discretisation takes place as we write the functions as a linear combination of suitable basis functions b on a reference cell.

$$u(x) = \sum_{i=1}^N \alpha_i b_i(x), \qquad v(x) = \sum_{i=1}^N \vartheta_i b_i(x). \tag{1.5}$$

The variational formulation can now be converted into several aequivalent equations

$$\sum_\Sigma s^\Sigma\Big(\sum_j \alpha_j^\Sigma b_j, \sum_i \vartheta_i^\Sigma b_i\Big) = \sum_\Sigma l^\Sigma\Big(\sum_i \vartheta_i^\Sigma b_i\Big) \qquad \forall \vartheta_i^\Sigma \tag{1.6}$$

$$\Longleftrightarrow$$

$$\sum_\Sigma \sum_i \vartheta_i^\Sigma \Big(\sum_j s^\Sigma(b_j, b_i)\alpha_i^\Sigma - l^\Sigma(b_i)\Big) = 0 \qquad \forall \vartheta_i^\Sigma \tag{1.7}$$

$$\Longleftrightarrow$$

$$\sum_\Sigma \Big(\sum_j s^\Sigma(b_j, b_i)\alpha_i^\Sigma - l^\Sigma(b_i)\Big) = 0 \qquad \forall i \tag{1.8}$$

$$\Longleftrightarrow$$

$$\mathcal{A}^\Sigma(\mathbf{s}^\Sigma \alpha^\Sigma - \mathbf{l}^\Sigma) = 0 \tag{1.9}$$

$$\Longleftrightarrow$$

$$\mathcal{A}^\Sigma(\mathbf{s}^\Sigma)\mathcal{A}(\alpha^\Sigma) - \mathcal{A}(\mathbf{l}^\Sigma) = 0 \tag{1.10}$$

$$\Longleftrightarrow$$

$$\mathbf{S}\alpha = \mathbf{l} \tag{1.11}$$

$\mathbf{s}^\Sigma$ is called the *element stiffness matrix.* $\mathbf{S}$ is the (sparse) *global stiffness matrix.* α are the *degrees of freedom* and $\mathbf{l}$ is the *load vector.* The so called assembly process $\mathcal{A}^\Sigma()$ builds a global linear system of equations. We can see now that the discretisation of equation (1.3) leads to a linear system of equations.

A *finite element* is defined by a triple $(\Sigma, \mathcal{P}_\Sigma, \mathcal{F}_\Sigma)$ [11] which consists of

- a geometric domain Σ (either tetrahedron, hexahedron or prism)
- a space of functions $\mathcal{P}_\Sigma$ (usually polynomials, b_i)
- a set of linear functionals on $\mathcal{P}_\Sigma$, the degrees of freedom $\mathcal{F}_\Sigma$

Usually in the FEM, the calculations are performed on a reference element. With an additional coordinate transformation one can get the finite element on any given element. Prisms can be used to connect tetrahedrons and hexahedrons. Two neighbouring elements are expected to fulfil the *patch condition*, i.e. the global function has to be continuous at the transition face.

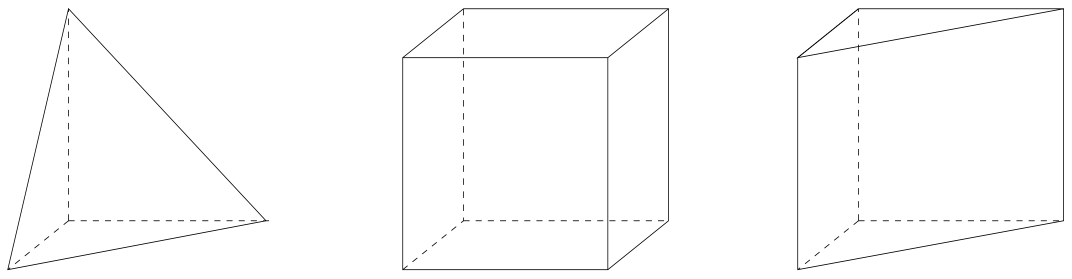


Figure 1.2: Reference elements used in the finite element method

1.5.2 The boundary conditions

It remains to apply the boundary conditions.

Dirichlet boundary conditions

In a weak form Dirichlet boundary conditions are already set by the term $\left(\frac{\partial u}{\partial n}, v\right)_\Gamma$ from equation (1.3) to be zero. But this is not enough. For example to force the Dirichlet boundary conditions (1.2), several degrees of freedom i representing the boundary must be set to zero. This can be achieved by erasing the i-th row and the i-th column of the matrix and by erasing the i-th component of the load vector. The matrix should remain regular in order to get unique degrees of freedom. This can be achieved by setting the i-th diagonal element to 1 (or an arbitrary nonzero value). In the following example α_2 should be zero:

$$\begin{pmatrix} s_{11} & 0 & s_{13} \\ 0 & 1 & 0 \\ s_{31} & 0 & s_{33} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} l_1 \\ 0 \\ l_3 \end{pmatrix}$$

Neumann boundary conditions

It turns out by considering the Galerkin procedure that for applying homogeneous the Neumann boundary conditions there is no need to modify either the system matrix nor the load vector. The calculation of the magnetic field can thus be accomplished in a straight forward way. However, the Neumann matrix contains more nonzero components than the Dirichlet matrix and the system is thus more costly to solve.

1.6 Differential Forms

The calculus of differential forms provides a cohesive and intuitive framework for computational electromagnetism. Within this context, one can classify the various fields from Maxwell's equa-

$$\begin{tikzcd}
0-forms: & & \phi \arrow[d, "d"] & & \\
1-forms: & \mathbf{A} \arrow[r, "-\frac{\partial}{\partial t}"] \arrow[d, "d"] & \mathbf{E} \arrow[d, "d"] & & \mathbf{H} \arrow[d, "d"] \\
2-forms: & \mathbf{B} \arrow[r, "-\frac{\partial}{\partial t}"] \arrow[d, "d"] & 0 & \mathbf{D} \arrow[r, "-\frac{\partial}{\partial t}"] \arrow[d, "d"] & \mathbf{J} \arrow[d, "d"] \\
3-forms: & 0 & & \rho \arrow[r, "-\frac{\partial}{\partial t}"] & 0
\end{tikzcd}$$

tions in a more informative manner than standard vector calculus (see figure above). Roughly speaking, a differential k-form is a quantity appearing under an k-dimensional integral. The mathematically correct definition reads as follows [7]

Definition
A *differential form of degree k* or *exterior k-form* is a mapping

$$\begin{array}{rccc} \omega: & U & \to & \mathrm{Alt}^k(\mathbb{R}^n) \\ \omega(x): & (\mathbb{R}^n)^k & \to & \mathbb{C} \end{array}$$

which assigns to each $x \in U$ an alternating k-times linear ω, i.e.

$$\omega(\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_j, \dots, \mathbf{v}_k) = -\omega(\mathbf{v}_1, \dots, \mathbf{v}_j, \dots, \mathbf{v}_i, \dots, \mathbf{v}_k). \tag{1.12}$$

For a good introduction into differential forms, see [6], [7]. The three main operators in the calculus of differential forms are the *exterior product*, the *exterior derivative* and the *hodge star* operator.

1.6.1 Properties of the operators

Exterior product $\wedge$
An exterior product maps any l-form f and any k-form g to a $(l+k)$-form ($\Psi^l \wedge \Psi^k \to \Psi^{l+k}$).

- $f \wedge g = -g \wedge f$
- $(af + bg) \wedge h = a(f \wedge h) + b(g \wedge h)$

Exterior derivative d
The exterior derivative maps a l-form to a $(l+1)$-form ($d: \Psi^l \to \Psi^{l+1}$).

- $d(f^l \wedge g^m) = df^l \wedge g^m + (-1)^l f^l \wedge dg^m$ (needed for integration by parts)
- $d(df^l) = 0$

Hodge star operator $\star$
For the particular case of three dimensional space, the Hodge star operator is an invertible linear function that maps l-forms to $(3-l)$-forms and is denoted by $\star : \Psi^l \to \Psi^{3-l}$.

1.7 The FEMSTER library

FEMSTER is a library which provides classes to handle the basic mathematical structure of the finite elements $(\Sigma, \mathcal{P}_K, \mathcal{F}_\Sigma)$. There is no front end included. In the following the classes are shortly described. For a more sophisticated explanation the PhD thesis of Rob Rieben [8] can be consulted.

Element3D

One of the triple $(\Sigma, \mathcal{P}_K, \mathcal{F}_\Sigma)$ of the finite element is the geometric domain Σ. The class `Element3D` handles all of the information regarding the topology and geometry of Σ. In the finite element method it is usual to use a (local) reference element $\hat{\Sigma}$ (see Figure 1.2). On these cells bilinear forms are computed. In a second step the result is mapped by a *local to global mapping* $\mathbf{r} = \Gamma(\hat{\mathbf{r}})$ onto the (global) element. Often a Jacobi matrix and its determinant is needed to accomplish the mapping. The class provides methods which calculate the Jacobi matrix and its determinant.

Basis functions: 0-Froms, 1-Forms, 2-Forms, 3-Forms

This class implements the basis functions and its features on the reference elements. One can choose among interpolatory and hierarchical basis functions. The underlaying polynomials are Lagrange interpolation polynomials. Based on the theory of curl- and div- conforming function spaces described by [5] and [11], either 0-form (Ψ^0), 1-form (Ψ^1), 2-form (Ψ^2) or 3-form (Ψ^3) basis functions can be selected. Of course the dimension of such a space is infinite. With the degree p of the polynomials the power of the approximation can be controlled. The number of basis functions is proportional to p^3 (see Table 1.3). Each basis function corresponds to a degree of freedom. The class is organized such that the degrees of freedom $\mathcal{F}_\Sigma$ get the desired properties (unisolvence, invariance and locality). Implemented methods are for example global and local evaluations of the basis functions and the representation of a given function in the basis.

polynomial degree p	1	2	3	4
# 0-form Dofs	8	27	64	125
# 1-form Dofs	12	54	144	300
# 2-form Dofs	6	36	108	240

Figure 1.3: Number of degrees of freedoms for a local hexahedral element

Bilinearform

As we consider the variational formulation of a differential equation, we have to perform integrals of functions over a geometric domain. Those integrals can be seen as bilinear forms. Integrals should be accomplished on $L^2(\Omega)$ functions. With help of the star hodge operator $\star$ different integral types are possible. There are either the bilinear forms $(\star f, g)_\Omega$ and $(\star df, dg)_\Omega$ where f and g are from the same function space or the bilinear forms $(\star df, g)_\Omega$ where df and g are from the same function space. For the former exist classes such as *Bilinear0Form*,*Bilinear1Form*, *Bilinear2Form* and *Bilinear3Form*. Implemented methods of these classes are *getMassMatrix()*, *getMassStiffnessMatrix()*and *getLoadVector()*. The latter types of integrals can be accomplished with the aid of the classes *Bilinear01Form*, *Bilinear12Form* and *Bilinear23Form*. For these classes either the ordinary or the topological derivative matrix can be received.

IntegrationRule

To calculate bilinear forms on polynomial function spaces the *Gauss-Quadrature* rule proves to be a good choice. The corresponding weights and nodes can be demanded for any polynomial degree and for all three reference elements in this class.

Permutation

As the polynomials are evaluated on the reference elements, problems can arise in order to retain the correct orientation on the original elements. For every reference element there exist rules for edge, face and volume orientations expressed by sequences of the element nodes. As in the mesh some nodes of different elements are identified, the orientation of the reference element can not always be enforced. Because of this drawback all locally calculated matrices and vectors have to be permuted. Every column and row corresponds to a degree of freedom. The permutation occurs in such a way that the edge, face and volume degree of freedom are permuted separately.

Mesh3D

The diversity of the geometrical domains which can be treated has its origin in the mesh. The mesh reader class `mesh3D` can handle three different element topologies (tetrahedron, hexahedron, prism). It keeps information of the element types (number of nodes, number of edges, ...), mesh size and boundary faces. Last but not least, *mesh3D* provides a local to global mapping $\mathcal{M}$ for each form which maps degrees of freedoms from the local indices to the global indices. This mapping is needed for assembling global matrices and vectors.

FEMSTER provides a few mesh files for the unity cube. For every of the three element topologies six files are available with different resolutions (see Table 3.8). The FEMSTER group uses mainly the hexaedral elements. Complex hexaedral meshes can be generated with the comercial program *True Grid*. At some places in the library I received the impression that tetraedras have not been tested likewise as hexaedrons. So I decided to work with hexaedrals cube files. (Even though the geometry of the cube is fairly basic, the solutions of the Maxwell equations on that geometrical domain are not!). Nevertheless, in the beginning of the diploma work I computed the static field (poisson problem) of a charge density in the geometry (see Figure 5.1) with tetraedrons. The tetraedral mesh was generated by the program *Netgen*. As the data format differ, I had to write a converter.

Solvers: CG and MINRES

Once the global system is assembled, it has to be solved. Delivered with FEMSTER is also a *conjugate gradient* (CG) solver class. As the mass and stiffness matrices both are symmetric and positive definite it can be applied. The dimensions of the system matrices become usually quite large. The matrices itself are exceedingly sparse. For this end the *compressed row storage* (CRS) format is provided with another class. The CG operations are performed with BLAS routines which are eminently efficient.

As we will see later, a saddle point problem has to be solved. The saddle point matrix is indefinite. Thus the CG algorithm can not be employed. An adequate algorithm is the *minimal residual* (MINRES) algorithm [24] which I implemented with BLAS routines in the same manner as the CG algorithm. Later in the progress of the diploma work it turned out that the saddle point system in our consideration can be split into two separate systems which are both positive definite (see Appendix).

2 Electromagnetic fields

2.1 Maxwell equations

The Maxwell equations are defined in terms of the electric field $\mathbf{E}$, the magnetic induction $\mathbf{B}$, the charge density ρ and the current density $\mathbf{j}$. The dielectric displacement $\mathbf{D}$ and the magnetic field $\mathbf{H}$ are defined by $\mathbf{D} = \epsilon\mathbf{E}$ and $\mathbf{B} = \mu^{-1}\mathbf{H}$. In vacuum the tensors ϵ, μ are the identity mapping. The Maxwell equations (provided with PEC boundary conditions: the electric field is perpendicular on the boundary, the magnetic field tangential) read

$$\begin{aligned}
\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} &= 0 && (2.1)\\
\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} &= \mathbf{J} && (2.2)\\
\nabla \cdot \mathbf{D} &= \rho && (2.3)\\
\nabla \cdot \mathbf{B} &= 0. && (2.4)
\end{aligned}$$

The former two equations describe the dynamics of the fields. The divergence equations characterize the static fields. Electromagnetic fields can exist in the absence of charge if they are time dependent. They then describe electromagnetic waves. The sources of electromagnetic fields are charge and current (moving charge).

2.2 Maxwell equations as a constraint formulation

The ansatz described in the following has its seeds in the book [12] and has been proposed in [1]. If the divergence equations of the Maxwell equations are satisfied by the initial condition, then these conditions are identically satisfied for all time by the dynamic equations (2.1) and (2.2). If one could integrate the equations exactly, the dynamic equations would suffice. But as we can just calculate approximations of the solution, full accuracy can not be achieved. As a consequence of this fact the divergence conditions are violated. A workaround can be reached by writing the Maxwell equations in a constraint formulation. We expand the dynamic equations by the gradient of Lagrange multipliers $\lambda_\varphi(\mathbf{r}, t)$, $\lambda_\vartheta(\mathbf{r}, t)$ to force equality and add the divergence conditions as constraints:

$$\begin{aligned}
\nabla \times \mathbf{E} + \mu^{-1}\frac{\partial \mathbf{H}}{\partial t} + \nabla\lambda_\vartheta &= 0 && (2.5)\\
\nabla \times \mathbf{H} - \epsilon\frac{\partial \mathbf{E}}{\partial t} + \nabla\lambda_\varphi &= \mathbf{J} && (2.6)\\
\epsilon\nabla \cdot \mathbf{E} &= \rho && (2.7)\\
\mu^{-1}\nabla \cdot \mathbf{H} &= 0. && (2.8)
\end{aligned}$$

The Lagrange multipliers in the dynamic equations can be seen as error correcting potentials. The corresponding gradients are corrections to the electric and magnetic fields. In other words, the part in $\mathbf{E}$ and $\mathbf{B}$ which does not satisfy the divergence condition is removed!

2.3 Maxwell equations with differential forms

The Maxwell equations can be stated in terms of differential forms. Consider a domain Ω on which φ and ϑ are 0-forms, E and H are 1-forms, J is a 2-form and ρ is a 3-form. As we consider a perfect vacuum, ϵ and μ are equal to one and we will write $\star$ instead of $\star_\epsilon$ and $\star_\mu$. The Maxwell equations read now

$$dE + \star(\partial_t H + d\vartheta) = 0 \tag{2.9}$$
$$dH - \star(\partial_t E + d\varphi) = J \tag{2.10}$$
$$d(\star E) = \rho \tag{2.11}$$
$$d(\star H) = 0. \tag{2.12}$$

Here φ and ϑ are Lagrangian multipliers for the constraints (2.11) and (2.12), respectively. As perfect electrical conductors (PEC) are considered on the boundary, we have $\mathbf{E}\perp\partial\Omega$ and $\mathbf{H}\|\partial\Omega$. The *trace* of both the electric and magnetic field then vanishes on the boundary. In the calculus of differential forms this conditions can be written as proposed by Hiptmair [2]:

$$\mathbf{t}_{\partial\Omega}E = 0 \tag{2.13}$$
$$\mathbf{t}_{\partial\Omega}H = 0. \tag{2.14}$$

2.3.1 De Rahm diagram

With aid of the de Rahm diagram, one can see how differential forms (corresponding to some function spaces) transform under appliance of the exterior derivative and the Hodge star operator:

$$\begin{tikzcd}
 & \begin{matrix}0-form\\ H^1(\Omega)/\mathbb{R}\end{matrix} \arrow[r, "\overset{\nabla}{d}"] \arrow[d, "\star"] & \begin{matrix}1-form\\ H(curl;\Omega)\end{matrix} \arrow[r, "\overset{\nabla\times}{d}"] \arrow[d, "\star"] & \begin{matrix}2-form\\ H(div;\Omega)\end{matrix} \arrow[r, "\overset{\nabla\cdot}{d}"] \arrow[d, "\star"] & \begin{matrix}3-form\\ L^2(\Omega)\end{matrix} \arrow[r, "\overset{\nabla}{d}"] \arrow[d, "\star"] & 0 \\
0 & \begin{matrix}3-form\\ L^2(\Omega)\end{matrix} \arrow[l, "\underset{\nabla}{d}"'] & \begin{matrix}2-form\\ H(div;\Omega)\end{matrix} \arrow[l, "\underset{\nabla\cdot}{d}"'] & \begin{matrix}1-form\\ H(curl;\Omega)\end{matrix} \arrow[l, "\underset{\nabla\times}{d}"'] & \begin{matrix}0-form\\ H^1(\Omega)/\mathbb{R}\end{matrix} \arrow[l, "\underset{\nabla}{d}"'] &
\end{tikzcd}$$

2.4 Variational formulations

The discrete form approximations are defined as follows:

$$\begin{array}{ll|ll}
\Phi(\mathbf{r},t) = \sum_i \phi_i(t)\mathbf{v}_i(\mathbf{r}) & \mathbf{v}_i \in X_h \subset \Psi^0 & H(\mathbf{r},t) = \sum_i h_i(t)\mathbf{w}_i(\mathbf{r}) & \mathbf{w}_i \in W_h \subset \Psi^1 \\
\Theta(\mathbf{r},t) = \sum_i \theta_i(t)\mathbf{v}_i(\mathbf{r}) & \mathbf{v}_i \in X_h \subset \Psi^0 & J(\mathbf{r},t) = \sum_i j_i(t)\mathbf{f}_i(\mathbf{r}) & \mathbf{f}_i \in F_h \subset \Psi^2 \\
E(\mathbf{r},t) = \sum_i e_i(t)\mathbf{w}_i(\mathbf{r}) & \mathbf{w}_i \in W_h \subset \Psi^1 & \rho(\mathbf{r},t) = \sum_i \rho_i(t)\mathbf{g}_i(\mathbf{r}) & \mathbf{g}_i \in G_h \subset \Psi^3
\end{array}$$

We would like to formulate the Maxwell equations in two second order systems where E and H are separated.

2.4.1 Primal problem: E-formulation

Taking the derivative of equation (2.10) with respect to t yields

$$d\partial_t H - \star(\partial_t^2 E + d\,\underbrace{\partial_t \varphi}_{=:\Phi}) = \partial_t J. \tag{2.15}$$

We rewrite equation (2.9) as

$$\partial_t H = -\star dE - d\vartheta \tag{2.16}$$

and plug it into equation (2.15). Together with the divergence condition (2.11) and keeping in mind that $d \circ d\vartheta = 0$, we get a second order system which is not dependent on the magnetic field H anymore

$$\begin{aligned} d(\star dE) + \star\partial_t^2 E + \star d\Phi &= -\partial_t J \qquad (2.17)\\ d(\star E) &= \rho. \qquad (2.18) \end{aligned}$$

Following the variational approach means that one has to multiply these equations by a test function and then integrate over Ω. Equation (2.17) consists only of 2-form terms, equation (2.18) only of 3-form terms. The differential forms which are well integrable are 3-forms. Therefore, we wedge-multiply the former equation by a 1-form E' and the second equation by a 0-form Φ'. Applying the integration by parts rule to the first term of the former equation yields

$$(d\star dE, E')_\Omega = \underbrace{(\star dE, E')_{\partial\Omega}}_{=0} + (\star dE, dE')_\Omega \tag{2.19}$$

E' has to be in the same space as E. Because of the condition (2.13) the surface integral vanishes. The whole variational formulation now reads

$$\begin{array}{lllllll}
(\star dE, dE')_\Omega & + & \partial_t^2(\star E, E')_\Omega & + & (\star d\Phi, E')_\Omega & = & -\partial_t(J, E')_\Omega \quad \forall E' \in W_h \\
-(\star E, d\Phi')_\Omega & & & & & = & (\rho, \Phi')_\Omega \quad \forall \Phi' \in X_h \\
\mathbf{S}_1\mathbf{e} & + & \mathbf{M}_1\ddot{\mathbf{e}} & + & (\mathbf{M}_1\mathbf{K}_{01})\phi & = & -\mathbf{L}_1(\dot{\mathbf{j}}) \\
(\mathbf{M}_1\mathbf{K}_{01})^T\mathbf{e} & & & & & = & -\mathbf{L}_0(\rho)
\end{array} \tag{2.20}$$

The matrices $\mathbf{S}_1$ (*stiffness matrix* of the 1-1-bilinear form), $\mathbf{M}_1$ (*mass matrix* of the 1-1-bilinear form) and $\mathbf{K}_{01}$ (*derivative matrix* of the 0-1-bilinear form) are constant and do just depend on the mesh topology. $\mathbf{S}_1$ and $\mathbf{M}_1$ are symmetric and positive definite. $\mathbf{K}_{01}$ is rectangular and maps a 0-form into a 1-form.

2.4.2 H and B formulation

To calculate the magnetic field one can proceed in different ways. Either the same procedure as for the electric field can be applied or the magnetic field can be received from the electric field via equation (2.9). The latter method can be accomplished with or without a constraint for the divergence condition of the magnetic field.

Dual problem: Variational formulation
For the dual problem we perform exactly the same way as in the primal problem: Taking the derivative of equation (2.9) with respect to t yields

$$d\partial_t E + \star(\partial_t^2 H + d\underbrace{\partial_t \vartheta}_{=:\Theta}) = 0. \tag{2.21}$$

We multiply equation (2.10) by $\star$ and rewrite it as

$$\partial_t E = -d\varphi + \star dH - \star J \tag{2.22}$$

and fill it into equation (2.21). Together with the divergence condition (2.12) and keeping in mind that $d \circ d\varphi = 0$, we get a second order system which is not dependent on the electric field E anymore

$$\begin{aligned} d(\star dH) + \star\partial_t^2 H + \star d\Theta &= d\star J \\ d(\star H) &= 0. \end{aligned} \tag{2.23}$$

Proceeding as in the primal problem we get the dual problem:

$$\begin{array}{lclclcll} (\star dH, dH')_\Omega & + & \partial_t^2(\star_\mu H, H')_\Omega & + & (\star d\mathbf{\Theta}, H')_\Omega & = & (d\star J, H')_\Omega & \forall H' \in W_h \\ (\star H, d\mathbf{\Theta}')_\Omega & & & & & = & 0 & \forall \Theta' \in X_h \\ & & & & & & & \\ \mathbf{S}_1\mathbf{h} & + & \mathbf{M}_1\ddot{\mathbf{h}} & + & (\mathbf{M}_1\mathbf{K}_{01})\theta & = & \mathbf{L}_1(d\star\mathbf{j}) & \\ (\mathbf{M}_1\mathbf{K}_{01})^T\mathbf{h} & & & & & = & \mathbf{L}_0(0) & \end{array} \tag{2.24}$$

Until here the matrices $\mathbf{S}_1$, $\mathbf{M}_1$, $\mathbf{K}_{01}$ are exactly the same as in the primal problem. In order to apply Neumann boundary conditions the matrices for the magnetic field stay as they are whereas the matrices for the electric field have to be modified in order to get Dirichlet conditions.

Get B directly from E
Once the electric field E is available, one can calculate the magnetic field directly with the aid of the Maxwell equation (2.9). If the time derivative of B is approximated by a second order finite difference we have

$$dE_n = -\partial_t B = \frac{B_{n-\frac{1}{2}} - B_{n+\frac{1}{2}}}{\Delta t} + \mathcal{O}(\Delta t^2). \tag{2.25}$$

It is obvious that the electric and magnetic field discretisations are displaced at about half a time step. The magnetic field is now written as a 2-form. To apply the curl operator in FEMSTER, just a curl matrix $\mathbf{K}_{12}$ has to be multiplied by the vector of the electrical degrees of freedom:

$$\mathbf{b}_{n+\frac{1}{2}} \approx \mathbf{b}_{n-\frac{1}{2}} - \Delta t \cdot (\mathbf{K}_{12})\mathbf{e}_n \tag{2.26}$$

This is quite a fast procedure. One major drawback is that the calculation of B from the temporal variation bears a low accuracy. Besides this grievance there is no divergence condition. Nevertheless, I have implemented the B field calculation with this method to save (cpu-) time. To get an impression of the error the accuracies of the two procedures can be appraised from Figure 3.2.

2.5 Field integrator

This section describes the field integration algorithm for the electric field. Starting from the variational formulation (2.20) it is easy to see that this is a generalized wave equation. $\mathbf{S}_1$ characterizes the second derivative with respect to space. The second order time derivative $\ddot{\mathbf{e}}$ can be approximated by a simple leap frog scheme:

$$\ddot{\mathbf{e}} = \frac{\partial^2 \mathbf{e}}{\partial t^2} \approx \frac{\mathbf{e}^{n+1} - 2\mathbf{e}^n + \mathbf{e}^{n-1}}{\Delta t^2} \tag{2.27}$$

This scheme is of second order and symplectic. The variational formulation furnished with the leap frog approximation reads

$$\begin{array}{lclcl} \mathbf{S}_1\mathbf{e}^n & + & \mathbf{M}_1\frac{\mathbf{e}^{n+1}-2\mathbf{e}^n+\mathbf{e}^{n-1}}{\Delta t^2} & + & (\mathbf{M}_1\mathbf{K}_{01})\phi^{n+1} & = & -\mathbf{L}_1(\dot{\mathbf{j}}) \\ (\mathbf{M}_1\mathbf{K}_{01})^T\mathbf{e}^{n+1} & & & & & = & -\mathbf{L}_0(\rho) \end{array} \tag{2.28}$$

With two foregoing vectors $\mathbf{e}^n$, $\mathbf{e}^{n-1}$ one can calculate the current vector $\mathbf{e}^{n+1}$. This means that for starting the integrator one needs two initial conditions. The system (2.28) with $\mathbf{e}^{n+1}$ and ϕ^{n+1} on the left hand side reads $\forall n = 1, 2, 3, ...$

$$\begin{array}{lclcl} \mathbf{M}_1\mathbf{e}^{n+1} & + & \Delta t^2(\mathbf{M}_1\mathbf{K}_{01})\phi^{n+1} & = & \mathbf{M}_1\left[2\mathbf{e}^n - \mathbf{e}^{n-1}\right] - \Delta t^2\left[\mathbf{S}_1\mathbf{e}^n + \mathbf{L}_1(\dot{\mathbf{j}})\right] \\ (\mathbf{M}_1\mathbf{K}_{01})^T\mathbf{e}^{n+1} & & & = & -\mathbf{L}_0(\rho) \end{array} \tag{2.29}$$

Defining $\tilde{\phi}^{n+1} := \Delta t^2 \phi^{n+1}$, the same system of equations can be written in matrix and vector notation as follows

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_1\mathbf{K}_{01} \\ (\mathbf{M}_1\mathbf{K}_{01})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e}^{n+1} \\ \tilde{\phi}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1\left[2e^n - e^{n-1}\right] - \Delta t^2\left[\mathbf{S}_1\mathbf{e}^n + \mathbf{L}_1(\dot{\mathbf{j}})\right] \\ -\mathbf{L}_0(\rho) \end{pmatrix} \tag{2.30}$$

This is called a *saddle point problem.* The matrix on the left is symmetric and indefinite because of the zero block. Keep in mind that the matrix $\mathbf{K}_{01}$ is rectangular (not square). This system can either be solved by an adequate solver (for example with MINRES) or the saddle point modification (see appendix) can be applied. This modification yields two systems of equations and an update term:

$$\boxed{\begin{array}{lcll} \mathbf{M}_1\mathbf{v}^{n+1} & = & \mathbf{M}_1\left[2\mathbf{e}^n - \mathbf{e}^{n-1}\right] - \Delta t^2\left[\mathbf{S}_1\mathbf{e}^n + \mathbf{L}_1(\dot{\mathbf{j}})\right] & =: \mathbf{f} \\ \mathbf{S}_0\tilde{\phi}^{n+1} & = & \mathbf{K}_{01}^T\mathbf{f} + \mathbf{L}_0(\rho) & \\ \mathbf{e}^{n+1} & = & \mathbf{v}^{n+1} - \mathbf{K}_{01}\tilde{\phi}^{n+1} & \end{array}} \tag{2.31}$$

At every time step this system of equations has to be solved. Again we have posivite definite matrices $\mathbf{M}_1$ and $\mathbf{S}_0$ for which a solution should be found. This can be performed with the conjugate gradient method. It is interesting that $\mathbf{S}_0$ is the Poisson matrix (i.e. the 0-form stiffness matrix $(d\Psi_i^0, d\Psi_i^0)_\Omega$). It remains to claim the Dirichlet boundary conditions which have been discussed in a previous section. The boundary conditions should be applied to the system (2.30). This is the same as erasing the corresponding rows and columns (as described) of $\mathbf{M}_1$, $\mathbf{S}_0$, $\mathbf{K}_{01}$ and accomplish algorithm (2.31).

2.6 Initial conditions

For a temporal integration of the system it is essential to know initial conditions of the electric and magnetic field. We assume that these fields satisfy the Maxwell equations and suppose that the fields and the Lagrange multipliers are given at time $t = 0$:

$$\begin{aligned} \mathbf{e}(0) &= \mathbf{e}_0 & \varphi(0) &= 0 \\ \mathbf{h}(0) &= \mathbf{h}_0 & \vartheta(0) &= 0 \end{aligned}$$

As we would like to discretise the second derivative of $\mathbf{e}$ and $\mathbf{h}$ with respect to time by

$$\ddot{\mathbf{e}} = \frac{\partial^2 \mathbf{e}}{\partial t^2} \approx \frac{\mathbf{e}^{n+1} - 2\mathbf{e}^n + \mathbf{e}^{n-1}}{\Delta t^2} \tag{2.32}$$

we need two initial conditions for each $\mathbf{e}$ and $\mathbf{h}$. To this end an extra variational formulation is built. In the following we show how to compute the initial conditions $\mathbf{e}^{-1/2}$ and $\mathbf{e}^{1/2}$ of the electric field. We take equation (2.10) of the Maxwell equations, multiply it by a test 1-form E' and integrate over the domain Ω:

$$\begin{aligned} \partial_t(\star\mathbf{E}, \mathbf{E}')_\Omega &= (\star\mathbf{B}, d\mathbf{E}')_\Omega - (\star d\varphi, \mathbf{E}')_\Omega - (\mathbf{J}, \mathbf{E}')_\Omega \\ \mathbf{M}_1\dot{\mathbf{e}} &= (\mathbf{M}_2\mathbf{K}_{12})^T\mathbf{b} - (\mathbf{M}_1\mathbf{K}_{01})\varphi - \mathbf{L}_1(\mathbf{j}) \end{aligned} \tag{2.33}$$

The first order discretisation of the first time derivative (for only half time steps) can be written as

$$\dot{\mathbf{e}}^0 = \partial_t\mathbf{e}(0) \approx \frac{2(\mathbf{e}^0 - \mathbf{e}^{-\frac{1}{2}})}{\Delta t} \approx \frac{2(\mathbf{e}^{\frac{1}{2}} - \mathbf{e}^0)}{\Delta t} \tag{2.34}$$

Filled in into (2.33) at time 0 and keeping in mind that $\varphi^0 = 0$ one gets a variational formulation for the initial conditions

$$\begin{aligned} \mathbf{M}_1\mathbf{e}^{\pm 1/2} \pm \tfrac{\Delta t}{2}(\mathbf{M}_1\mathbf{K}_{01})\phi^{\pm 1/2} &= \mathbf{M}_1\mathbf{e}^0 \pm \tfrac{\Delta t}{2}\left[(\mathbf{M}_2\mathbf{K}_{12})^T\mathbf{b}^0 - \mathbf{L}_1(\mathbf{j}^0)\right] \\ \pm\tfrac{\Delta t}{2}(\mathbf{M}_1\mathbf{K}_{01})^T\mathbf{e}^{\pm 1/2} &= \pm\tfrac{\Delta t}{2}\mathbf{L}_0(\rho) \end{aligned} \tag{2.35}$$

In matrix notation the last equation reads

$$\begin{pmatrix} \mathbf{M}_1 & \pm\frac{\Delta t}{2}(\mathbf{M}_1\mathbf{K}_{01}) \\ \pm\frac{\Delta t}{2}(\mathbf{M}_1\mathbf{K}_{01})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{e}^{\pm 1/2} \\ \phi^{\pm 1/2} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1\mathbf{e}^0 \pm \frac{\Delta t}{2}\left[(\mathbf{M}_2\mathbf{K}_{12})^T\mathbf{b}^0 - \mathbf{L}_1(\mathbf{j}^0)\right] \\ \pm\frac{\Delta t}{2}\mathbf{L}_0(\rho) \end{pmatrix} \tag{2.36}$$

Applying the modification of the saddle point problem (see Appendix) (with $\tilde{\phi}^{\pm 1/2} := \pm\frac{\Delta t}{2}\phi^{\pm 1/2}$) one has to solve the following system of equations

$$\begin{aligned} \mathbf{M}_1\mathbf{v}^{\pm 1/2} &= \mathbf{M}_1\mathbf{e}^0 \pm \frac{\Delta t}{2}\left[(\mathbf{M}_2\mathbf{K}_{12})^T\mathbf{b}^0 - \mathbf{L}_1(\mathbf{j}^0)\right] \\ \mathbf{S}_0\tilde{\phi}^{\pm 1/2} &= \mathbf{K}_{01}^T\left[\mathbf{M}_1\mathbf{e}^0 \pm \frac{\Delta t}{2}\left[(\mathbf{M}_2\mathbf{K}_{12})^T\mathbf{b}^0 - \mathbf{L}_1(\mathbf{j}^0)\right]\right] - \mathbf{L}_0(\rho) \\ \mathbf{e}^{\pm 1/2} &= \mathbf{v}^{\pm 1/2} - \mathbf{K}_{01}\tilde{\phi}^{\pm 1/2} \end{aligned}$$

The boundary conditions are applied exactly as in (2.31).

2.7 Code fragments

Of course the whole C++ implementation can not be printed in this report. But the thread for the field integration algorithm is shown here.

```
// Initializing:         - read in mesh

                         - choose basis
                         - bilinear forms
                         - initialize permutations

                         - applying boundary conditions to matrices

                         - read in initial condition

// Assembly process:     for all cells do {
                                 getMassMatrix();
                                 getStiffnessMatrix();

                                 permuteMatrices();

                                 assemble_global_MassMatrix
                                 assemble_global_StiffnessMatrix
                         }

                         apply matrix boundary conditions


// Time integration      for each time step do{
                                  calculate new magnetic field B
                                  //right hand side update:
                                      R1 = M[2e_2-e_1]-deltaT**2[Se_2+L(j)]
                                      R2 = K^T*f - L(rho)
                                  e_2 -> e_1
                                  set vector boundary conditions
                                  solve:   M*v   = R1
                                  solve:   S*phi = R2
                                  e_2 = v - K*phi

                                  write solution to file
                         }
```

3 Testing the field integrator

For a cube with no charge density in its interior we can expand the solution in terms of eigenmodes, which are exact solutions. Every arbitrary solution can be expressed in terms of such eigenmodes. These exact solutions allow us to evaluate the error made by temporal numerical integration. Another test is to compute the frequencies (by accomplishing a discrete Fourier transform) and compare them with exact frequencies.

3.1 Exact solutions

The eigenmodes of a cubic cavity can be split into two families: the transversal electric (TE) and transversal magnetic (TM) modes [14].

3.1.1 Hertz vectors

With the aid of polarization potentials - the Hertz vectors - one can get solutions of the Maxwell equations for simple but arbitrary geometrical domains. In materials the relations

$$\mathbf{D} = \epsilon\mathbf{E} + \mathbf{P}_{ext}, \qquad \mathbf{B} = \mu\mathbf{H} + \mathbf{M}_{ext} \tag{3.1}$$

hold. In the case of vacuum $\mathbf{P}_{ext}$ and $\mathbf{M}_{ext}$ are zero and the tensors ϵ and μ are identity mappings. In [13] the *Hertz vectors* $\mathbf{\Pi}_e$ and $\mathbf{\Pi}_m$ are defined by the vector and scalar potential:

$$\mathbf{A} = \frac{\partial \mathbf{\Pi}_e}{\partial t} + \nabla \times \mathbf{\Pi}_m, \qquad \Phi = -\nabla \cdot \mathbf{\Pi}_e. \tag{3.2}$$

Plugged into the wave equations for $\mathbf{A}$ and Φ, derived by the Maxwell equations, and taking into account a few implications leads to wave equations for $\mathbf{\Pi}_e$ and $\mathbf{\Pi}_m$

$$\begin{array}{rcl} \partial_t^2 \mathbf{\Pi}_e - \Delta \mathbf{\Pi}_e &=& 0 \\ \partial_t^2 \mathbf{\Pi}_m - \Delta \mathbf{\Pi}_m &=& 0 \end{array} \tag{3.3}$$

which have to be satisfied in the considered domain. Once the Hertz vectors are fixed, the electromagnetic field can be calculated by the following formulas

$$\mathbf{E} = \nabla(\nabla \cdot \mathbf{\Pi}_e) - \partial_t^2 \mathbf{\Pi}_e - \nabla \times \partial_t \mathbf{\Pi}_m \tag{3.4}$$

$$\mathbf{H} = \nabla \times \partial_t \mathbf{\Pi}_e + \nabla \times \nabla \times \mathbf{\Pi}_m. \tag{3.5}$$

3.1.2 TM- and TE-waves

To simplify matters we are considering a cube with edge length equal to one. Possible solutions of $\mathbf{\Pi}_e$ and $\mathbf{\Pi}_m$ for the z-direction in the unity cube are

$$\mathbf{\Pi}_e = \begin{pmatrix} 0 \\ 0 \\ C_e \sin(k_x x) \cdot \sin(k_y y) \cdot \cos(k_z z) \cdot e^{i\omega t} \end{pmatrix} \tag{3.6}$$

which generate the TM modes, and

$$\mathbf{\Pi}_m = \begin{pmatrix} 0 \\ 0 \\ C_e \cos(k_x x) \cdot \cos(k_y y) \cdot \sin(k_z z) \cdot e^{i\omega t} \end{pmatrix} \tag{3.7}$$

which generate the TE modes. These two solutions can be seen as a basis for the z component of an arbitrary function. The wave equations (3.3) are satisfied if

$$k_x^2 + k_y^2 + k_z^2 = k^2 = \omega^2 \tag{3.8}$$

The components of $\mathbf{k}$ have to satisfy the conditions

$$k_x = n\pi \qquad k_y = m\pi \qquad k_z = q\pi \tag{3.9}$$

for some integers n, m, $q \in \mathbb{Z}_+$. From (3.4) and (3.5) one gets the electromagnetic field by taking the real part. Recall that a multiplication by i corresponds to a phase shift of $\frac{\pi}{2}$ in the time domain.

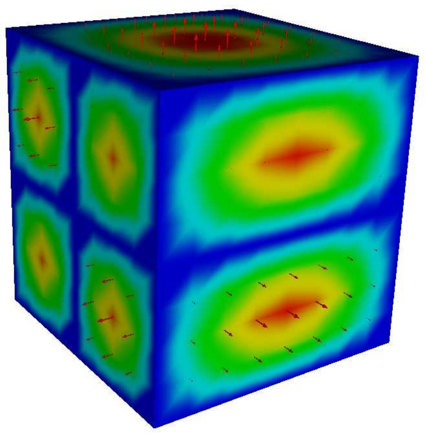
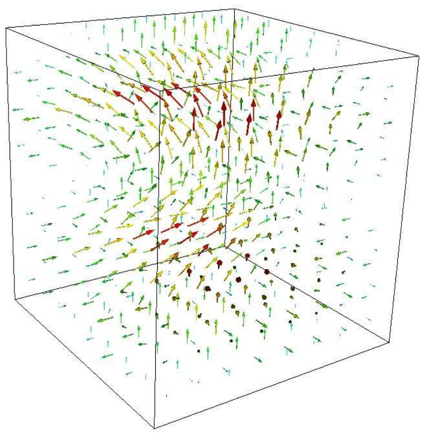
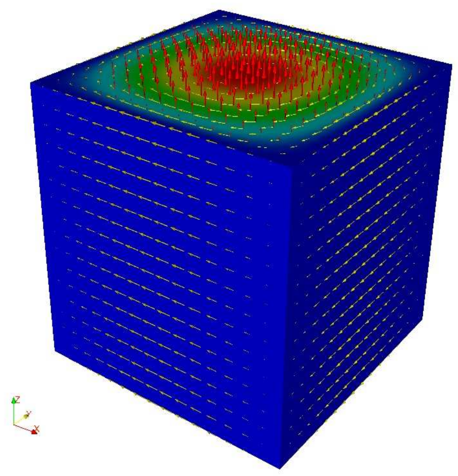
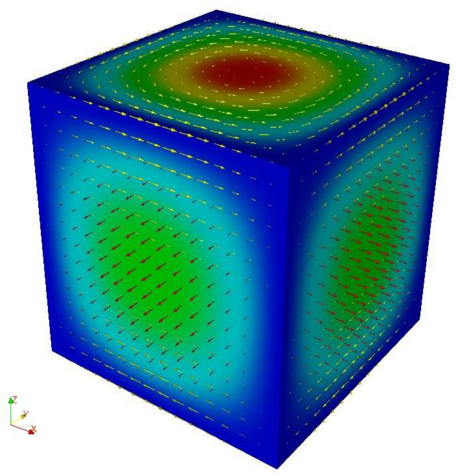



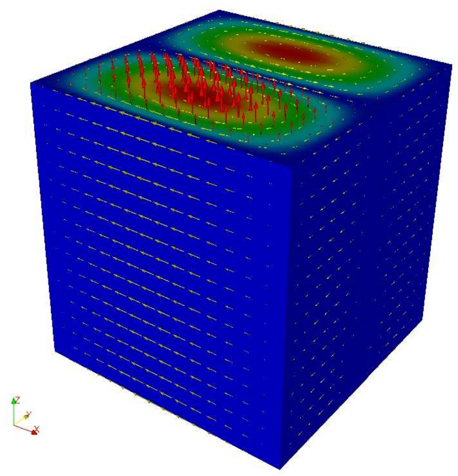
Figure 3.1: Interaction of three different modes: TM_{110}+TM_{022}+TM_{102}

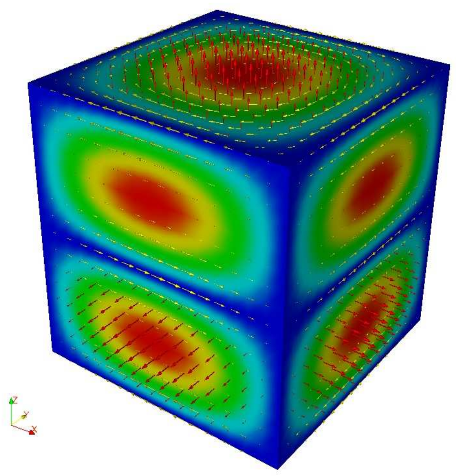
For the TM-waves we get[1]:

$$
\begin{aligned}
\mathbf{E}_{TM}(\mathbf{x},t) &= C_e\cos(\omega t) \cdot \begin{pmatrix} -k_x k_z & \cos(k_x x)\sin(k_y y)\sin(k_z z) \\ -k_y k_z & \sin(k_x x)\cos(k_y y)\sin(k_z z) \\ (k_x^2+k_y^2) & \sin(k_x x)\sin(k_y y)\cos(k_z z) \end{pmatrix} \\
\mathbf{H}_{TM}(\mathbf{x},t) &= C_e\omega\cos(\omega t+\tfrac{\pi}{2}) \cdot \begin{pmatrix} +k_y & \sin(k_x x)\cos(k_y y)\cos(k_z z) \\ -k_x & \cos(k_x x)\sin(k_y y)\cos(k_z z) \\ \multicolumn{2}{c}{0} \end{pmatrix}
\end{aligned}
\tag{3.10}
$$


(a) TM_{110} mode


(b) TM_{111} mode

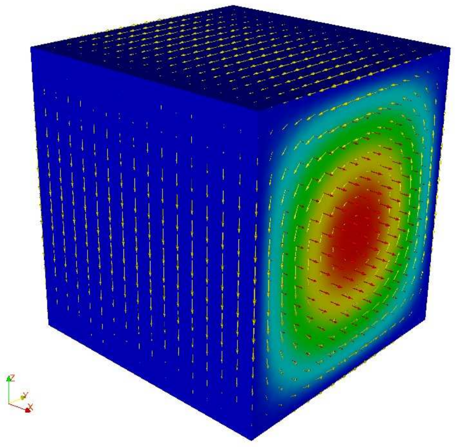

(c) TM_{120} mode

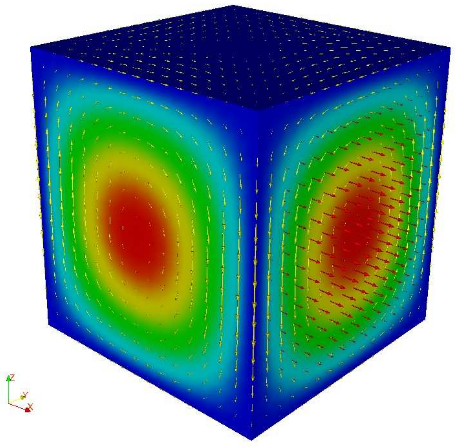

(d) TM_{112} mode

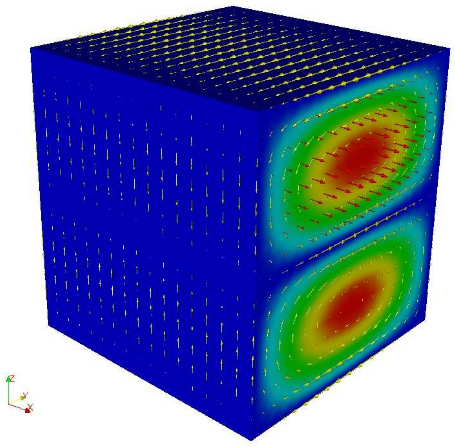
[1]Surface colours and red arrows: electric field. Yellow arrows: magnetic field

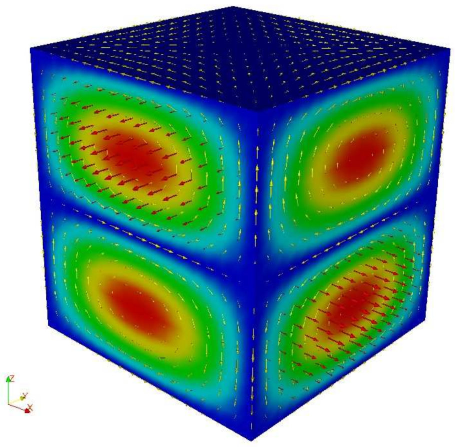
For the TE-waves it holds

$$
\begin{aligned}
\mathbf{E}_{TE}(\mathbf{x},t) &= C_m\omega\cos(\omega t + \tfrac{\pi}{2}) \cdot \begin{pmatrix} +k_y & \cos(k_x x)\sin(k_y y)\sin(k_z z) \\ -k_x & \sin(k_x x)\cos(k_y y)\sin(k_z z) \\ & 0 \end{pmatrix} \\
\mathbf{H}_{TE}(\mathbf{x},t) &= C_m\cos(\omega t) \cdot \begin{pmatrix} -k_x k_z & \sin(k_x x)\cos(k_y y)\cos(k_z z) \\ -k_y k_z & \cos(k_x x)\sin(k_y y)\cos(k_z z) \\ (k_x^2 + k_y^2) & \cos(k_x x)\cos(k_y y)\sin(k_z z) \end{pmatrix}
\end{aligned}
\tag{3.11}
$$


(e) TE$_{011}$ mode


(f) TE$_{111}$ mode


(g) TE$_{012}$ mode


(h) TE$_{112}$ mode

3.2 Electromagnetic Energy

The electromagnetic field energy density is defined as

$$\omega_{em} = \frac{1}{2} \star (\mathbf{E}^2 + c^2\mathbf{B}^2) \tag{3.12}$$

As $\mathbf{E}(\mathbf{x}, t) = \sum_i \mathbf{e}_i(t)\mathbf{w}_i(\mathbf{x})$, it follows for the overall electric energy in the domain Ω

$$\begin{aligned} W_e &= \frac{1}{2}(\mathbf{E}, \mathbf{E})_\Omega = \frac{1}{2}\Big(\sum_i \mathbf{e}_i\mathbf{w}_i, \sum_j \mathbf{e}_j\mathbf{w}_j\Big)_\Omega & (3.13)\\ &= \frac{1}{2}\sum_{i,j} \mathbf{e}_i\mathbf{e}_j(\mathbf{w}_i, \mathbf{w}_j)_\Omega & (3.14)\\ &= \frac{1}{2}\mathbf{e}^T\mathbf{M}_1\mathbf{e} & (3.15) \end{aligned}$$

With (3.10) we get the electrical field energy density for TM$_{nmq}$ modes by

$$\begin{aligned} W_e &= \frac{1}{2}\int_{[0,1]^3} \mathbf{E}^2 d\mathbf{x} & (3.16)\\ &= \frac{C_e^2}{16}\omega_{mnq}^2(m^2 + n^2) \cdot \cos^2(\omega_{mnq}t) & (3.17) \end{aligned}$$

For the magnetic field energy density one gets the same amplitude as for the electric field. The phase is shifted such that the total energy density is conserved.

$$\begin{aligned} W_m &= \frac{1}{2}\int_{[0,1]^3} \mathbf{H}^2 d\mathbf{x} & (3.18)\\ &= \frac{C_e^2}{16}\omega_{mnq}^2(m^2 + n^2) \cdot \cos^2(\omega_{mnq}t + \frac{\pi}{2}) & (3.19) \end{aligned}$$

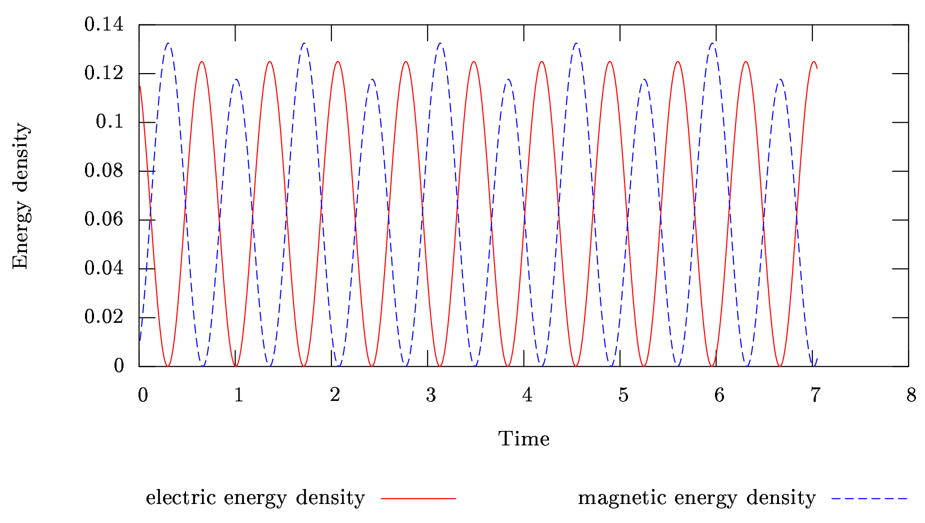


Figure 3.2: Electric and magnetic energy density

3.3 Fourier transformation

Electromagnetic fields can be decomposed in terms of plain waves $e^{i\mathbf{k}\cdot\mathbf{r}-i\omega t}$. This can be done by the use of the Fourier transform. The transform can be accomplished either by means of the wave vector $\mathbf{k}$

$$\mathbf{E}(\mathbf{r},t) = (2\pi)^{-3/2}\int_{\mathbb{R}^3}\mathbf{A}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{r}-i\omega(\mathbf{k})t}d\mathbf{k} \tag{3.20}$$

or by means of the angular frequency ω

$$\mathbf{E}(\mathbf{r},t) = (2\pi)^{-1/2}\int_{\mathbb{R}}\mathbf{A}(\omega)e^{i\mathbf{k}\cdot\mathbf{r}-i\omega(\mathbf{k})t}d\omega. \tag{3.21}$$

In computational science a fast version of the discrete Fourier transform (FFT) is usual. $\mathbf{E}(\mathbf{r},t)$ is sampled at aequidistant points with spacing Δ (either spacial of temporal). The FFT is then performed as follows:

$$A_n = \sum_{j=0}^{n-1} E_j e^{-2\pi i\cdot\frac{jk}{n}}, \qquad n = 2^q. \tag{3.22}$$

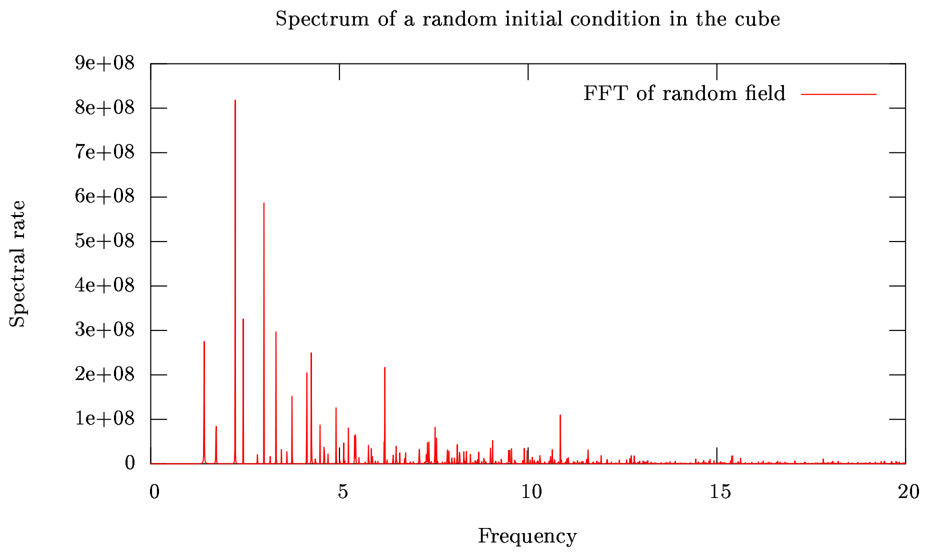
Due to Nyquist's critical frequency $f_c = \frac{1}{2\Delta}$, A_n embrace aequidistant values in $[-f_c, f_c]$. For the cavity problem one could resolve the distance between two faces. As the resolution of E_j gets higher, larger wave numbers appear in A_n. But the spacing of A_n is always the lowest wave number. If, instead, one would like to get a better resolution in the transformation A_n one proceeds by sampling E_j at more times.

3.3.1 Testing the spectrum of the unity cube cavity

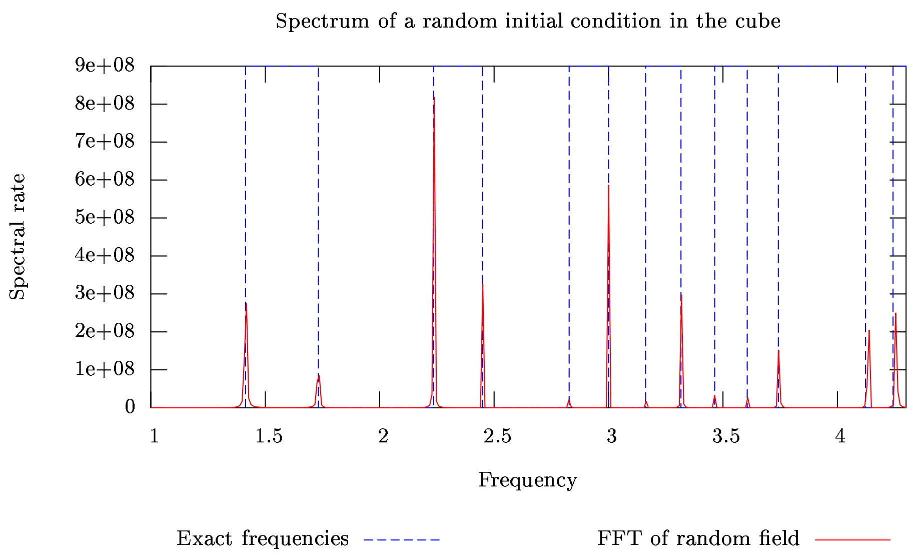
The exact eigenfrequencies of the unity cube depend on the integers n,m and q from (3.9) and are

$$2\pi f_{mnq} = \omega_{nmq} = \pi\sqrt{n^2+m^2+q^2}. \tag{3.23}$$

The FFT has been performed for a random initial condition in the unity cube cavity. The spectrum is shown in the figure below (in units of $2f$).



The polynomial degree of the basis function has been set to $p = 2$. 40'000 FE integration time steps have been accomplished with a time increment of 0.005 on a hexaedral mesh with $512 = 8^3$ cells. In a periodical interval of a few steps the components of the electric field are taken at different locations in the cube.



The calculated frequency peaks coincide quite well with the exact delta peaks. This is an indication that wrong frequencies do not arise during the integration. This fact argues for the integration method.

3.4 Errors

3.4.1 The Lagrange multiplier

As there are no particles up to now the divergence of both the electric and magnetic field should vanish. Then, for an exact solution of the Maxwell equations the Lagrange multiplier is zero in every position. To this end the Lagrange multiplier is an indicator for the error made by

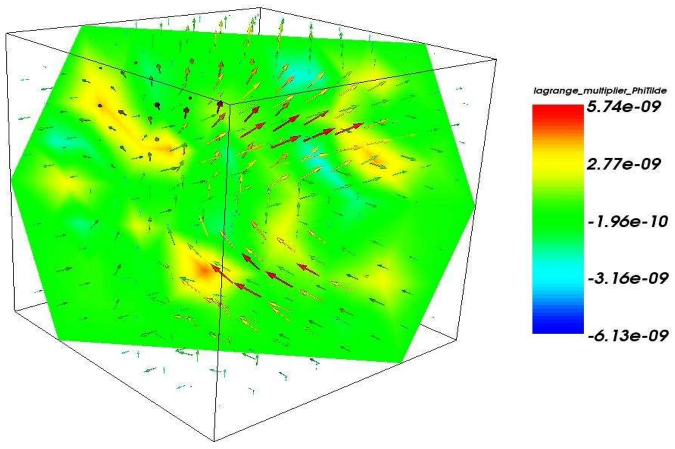


Figure 3.3: The colours on the plane correspond to the Lagrange multiplier and are thus indicators for the local error

numerical temporal integration. Figure 3.3 shows that the local absolute value of the Lagrange multiplier is less than the solver accuracy (norm of residual) which for this simulation is 10^{-8}.

Another property of the Lagrange multiplier is that the divergence bearing part of the field is eliminated. The calculated solution is therefor always a solution of the Maxwell equations. Without Lagrange multipliers this wouldn't be the case. To show the improvement of the

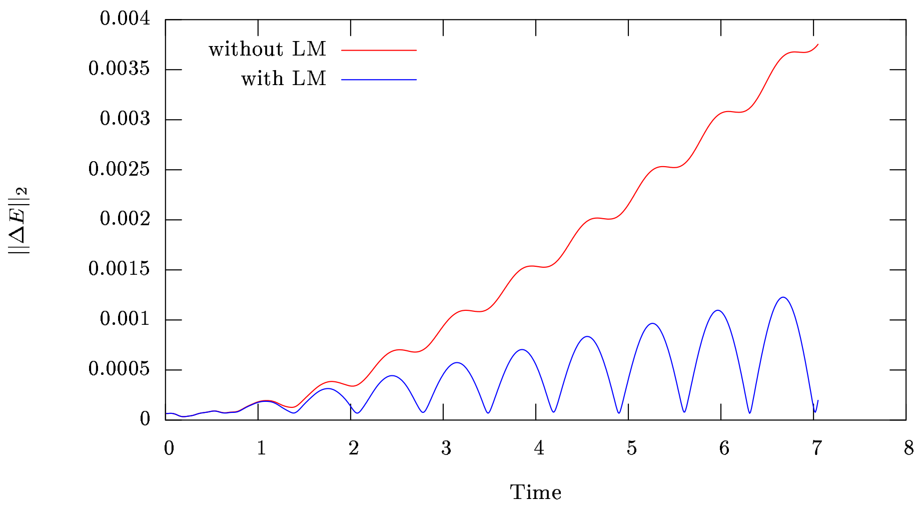


Figure 3.4: $||E_{ex} - E_{comp}||_2$ for runs with and without Lagrange multiplier

method I have performed two runs with the TM_{110} mode - first with Lagrange multipliers and then without - and plotted the L_2-norm of the error (see Figure 3.4).

It is nice to see that the LM-solution shows a lower error norm as the ordinary solution.

3.4.2 Limits for Δt

Next, the performance of the field integrator should be tested for different time steps Δt. On the basis of this behaviour further tests for different modes and for varying the mesh size h and the polynomial degree p can be accomplished.

Upper limit

In order to get stable solutions there is an upper limit for Δt. In [10] the first equation of algorithm (2.31) is considered (called the *central difference discretization*) in an iterative sense. With the theory of discrete system analysis a condition on Δt for convergent iterations is derived and reads

$$\Delta t \leq \frac{2}{\sqrt{\lambda_{max}}}. \tag{3.24}$$

λ_{max} is the spectral radius of $\mathbf{M}_1^{-1}\mathbf{S}_1$, that means the largest eigenvalue of the generalized eigensystem

$$\mathbf{S}_1\mathbf{x} = \lambda\mathbf{M}_1\mathbf{x}. \tag{3.25}$$

Even though the stability condition is predicted for a central difference discretization without additional Lagrange multiplier, it is true for (2.31) in a generous scale. I have calculated the largest eigenvalue for a cube mesh with 64 cells and polynomial degree $p = 3$ with *Matlab* to $\lambda_{max} = 0.023$. With this parameters *Matlab* had some difficulties with reading the matrix file (size: 45 megabytes). For higher degrees or finer meshes this matrix export would be infeasable. Then the estimation of the largest eigenvalue has to be in the C++ program itself. It can be accomplished by a modification of the Lanczos algorithm. I would have done this if I had had more time. An interesting question then could be answered: how does λ_{max} depend on the mesh?

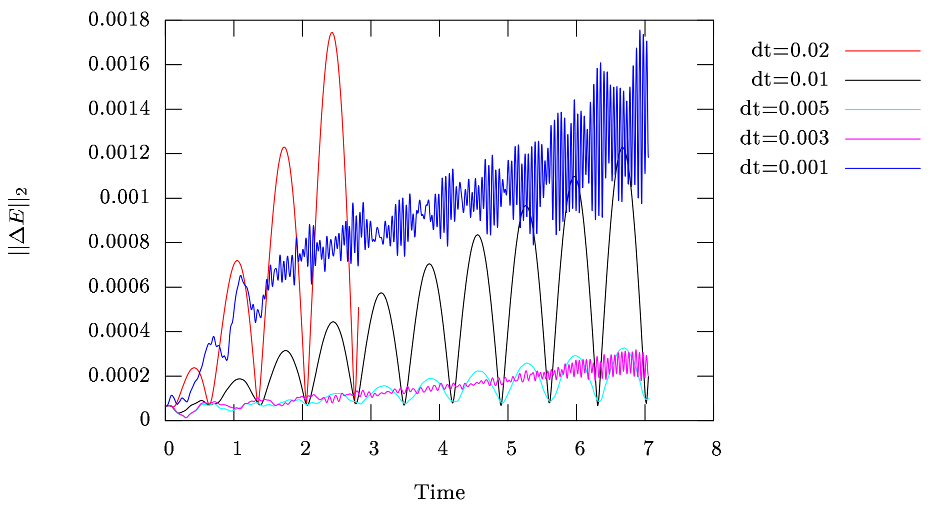


Figure 3.5: $||E_{exact} - E_{comp}||_2$ for different time increments (TM110 mode, 64 cells)

Lower limit

I tested the stability condition (3.24). As expected, for $\Delta t = 0.04$ the algorithm diverged (after 3 steps). For a series of error calculating runs I have chosen Δt in the interval $[0.001, 0.02]$. Simulated is the TM_{110} mode with a period of $\sqrt{2}$ and $p = 3$. The absolute errors of these runs are shown in Figure 3.5, the relative errors in Figure 3.6. It is an interesting fact that in order to keep the error small, Δt possesses an optimal value. As Δt gets smaller than this value the error begins to wiggle. I am not sure what the cause for this behaviour is. I guess that it has to do with the solver accuracy. A test with $\Delta t = 0.003$ and a modified solver accuracy (from 10^{-8} to 10^{-12}) approves that the wiggling vanishes. Anyway, the Lagrange multiplyers stay regular for all time increments as Figure 3.7 shows.

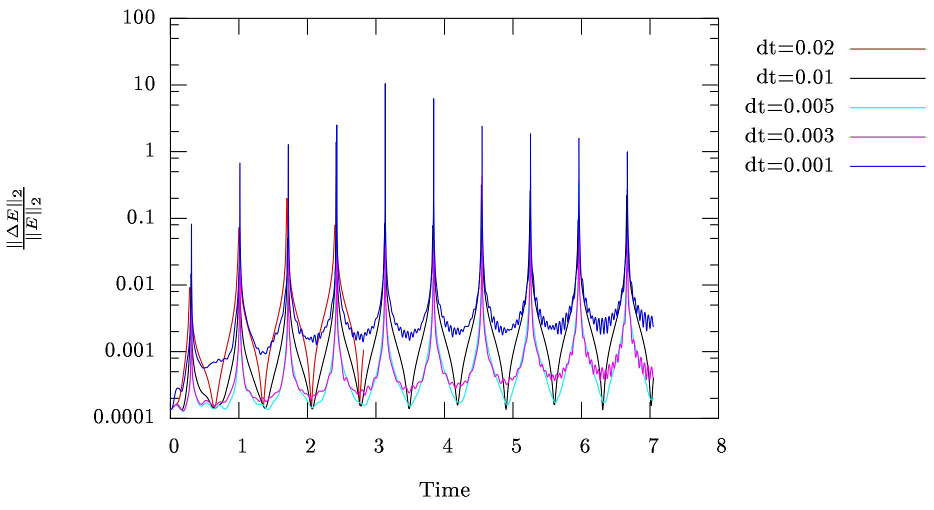


Figure 3.6: $\frac{\|\Delta E\|_2}{\|E\|_2}$ for different time increments (TM110 mode, 64 cells)

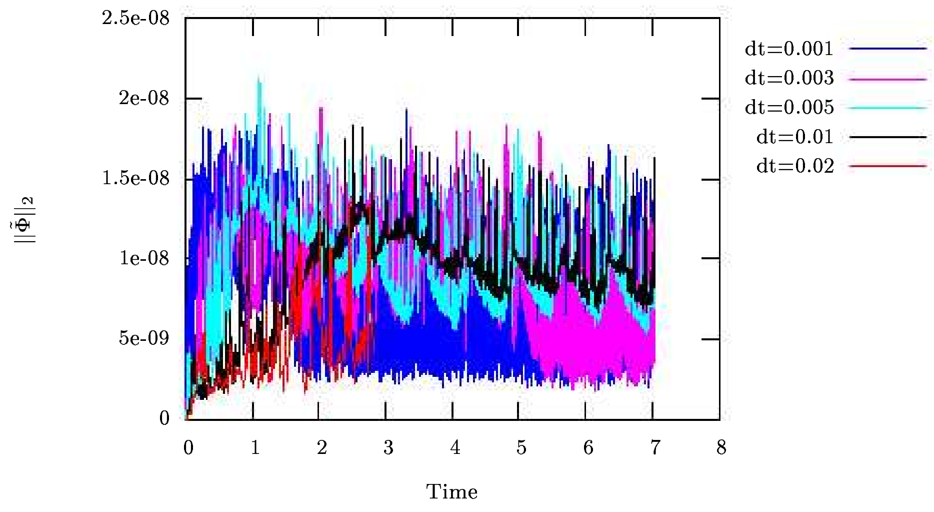
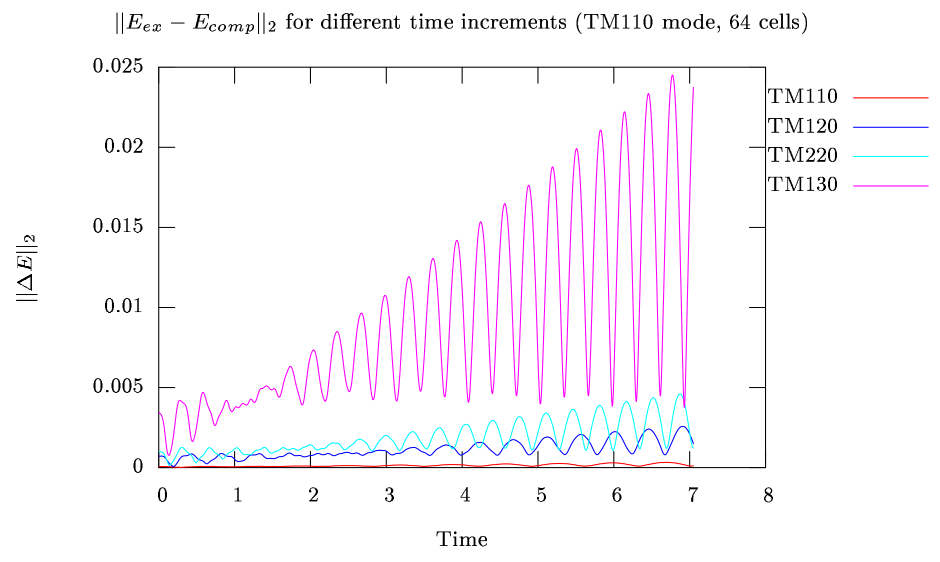


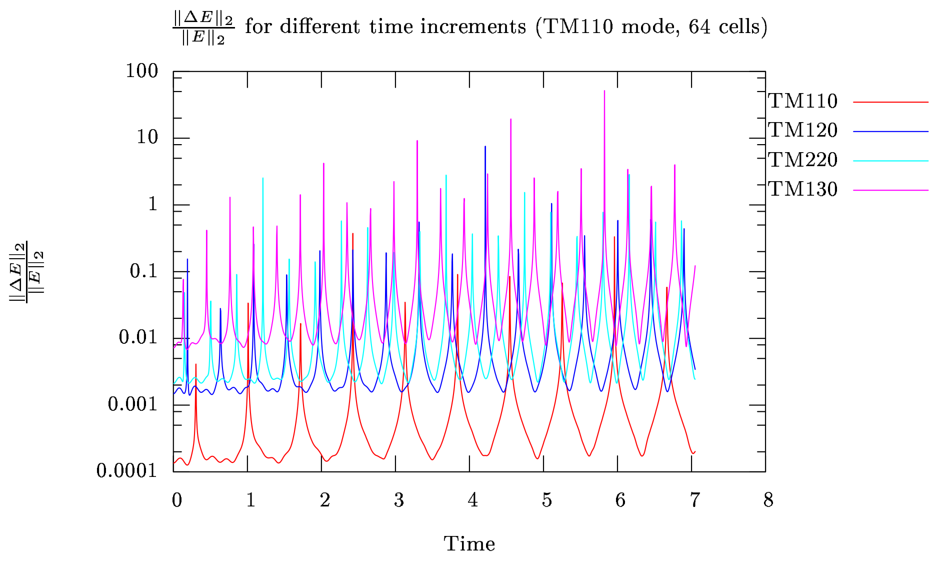
Figure 3.7: $\|\tilde{\Phi}\|_2$ for different time increments (TM110 mode, 64 cells)

3.4.3 Different modes

Now we are considering runs for different modes. The time increment is assigned to be $\Delta t = 0.01$ and the polynomial degree is set to $p = 3$ (which is the first nontrivial degree concerning some properties of the degrees of freedom). The mesh is again the one from the previous runs (hexaedral mesh for the unity cube with 64 cells). It is obvious that the error (absolute and



relative) grows rapidly for higher modes. On the one side this has to do with the higher frequency (see (3.23)). The signal is then modified faster and the temporal resolution gets worse. On the other hand the energy density depends on the square of ω whereby errors grow as well. Whether the accuracy satisfies the needs, will be seen when simulating particles. Particles are exiting plenty of modes. We can still hope that the higher modes possess a lower amplitude.



3.4.4 hp-FEM

The quality of the finite element approximation can be improved by either refine the mesh with elements of a smaller diameter h or by rising the polynomial degree p of the basis functions. In both cases the number of total degrees of freedoms rises extremely rapid (consider the table below).

Mesh size h		Polynomial degree p			
h	# cells	1	2	3	4
2^{-1}	8	27	125	343	729
		54	300	882	1'944
2^{-2}	64	125	729	2'197	4'913
		300	1'944	6'084	13'872
2^{-3}	512	729	4'913	15'625	35'937
		1'944	13'872	45'000	104'544
2^{-4}	4'096	4'913	35'937	117'649	274'625
		13'872	104'544	345'744	811'200
2^{-5}	32'768	35'937	274'625	912'673	2'146'689
		104'544	811'200	2'709'792	6'390'144
2^{-6}	262'144	274'625	2'146'689	7'189'057	16'974'593
		811'200	6'390'144	21'455'424	50'725'632

Figure 3.8: Number of total degrees of freedom for a unity cube in terms of p and h

It is important to know how the computed errors depend on p, h and the time increment Δt. The FEMSTER group observed that the Helmholtz equation (eigenvalue problem) could be solved more efficiently for a given error by rising p (instead of lowering h).

This behaviour should also be the case for FETD as the system to be solved is similar. My own evaluation of the hp-dependent behaviour of the accuracy is in progress. Because of lack of time no plots are given here. It is to append that for 512 cells and $p = 3$ already a lot of memory is allocated and the computer is allready behaving unacceptably slow (cpu: 3GHz, memory: 1GB).

4 Particles

As there were many more problems to be solved than expected in the beginning in order to implement the field integrator, there was a lack of time to implement the particles completely. At the editorial deadline unphysical effects can be observed at the transition of a particle from one local cell to the neighbouring cell.

4.1 A particle integrator: Velocity Verlet

For the sake of simplicity I have used a particle integrator which I have utilized allready in a semester work: the Velocity Verlet algorithm. The Velocity Verlet algorithm is an improvement to the Verlet integrator inasmuch the forces are calculated allready with the updated positions.

$$\begin{aligned}
\tilde{\mathbf{v}} &:= \mathbf{v}_{n-1} + \frac{\Delta t}{2m}\mathbf{f}_{n-1} \\
\mathbf{r}_n &= \mathbf{r}_{r-1} + \Delta t \tilde{\mathbf{v}} \\
\mathbf{f}_n &= \mathbf{F}(\mathbf{r}_n) = q\big[\mathbf{E}(\mathbf{r}_n) + \tilde{\mathbf{v}} \times \mathbf{B}(\mathbf{r}_n)\big] \\
\mathbf{v}_n &= \tilde{\mathbf{v}} + \frac{\Delta t}{2m}\mathbf{f}_n
\end{aligned} \tag{4.1}$$

4.2 Cooperation of FEMSTER and IPPL

Once IPPL was linked to FEMSTER, two major adaptions had to be implemented: supply each particle with its fields and calculate the terms $\mathbf{L}_1(\dot{\mathbf{j}})$ and $\mathbf{L}_0(\rho)$ from the right hand side of (2.31). As the particles can be seen as infinitesimally small, $\mathbf{j}$ and ρ become delta functions. For the delta distribution we have

$$\int_\Omega \delta(\mathbf{x} - \mathbf{x}')f(\mathbf{x})d\mathbf{x} = f(\mathbf{x}'). \tag{4.2}$$

I have reimplemented the FEMSTER routine computing the load vectors. This new routine for the delta function in not yet properly tested. Maybe a weight, depending on the position and analogous to the Gauss quadrature weights, is missing.

In [12] it is suggested that for certain configurations it is not necessary to update the particles for every time step of the field integrator. This should be taken into account when my diploma work program is used to develop particle traces.

Another difficulty is to find valid initial conditions of suddenly appearing and moving particles. In the program either the zero-field or the static solution (Poisson problem) can be chosen as initial condition. For a suddenly generated current an electromagnetic wave loosens itself from the particle. The propagation of this wave can yet be obtained with a visualization program called *ParaView*. For a moving particle a close divergence field and a far curl field is identifiable.

5 Conclusion

5.1 Review

The goal of this diploma work was to merge the finite element library *FEMSTER* and the particle framework *IPPL* to a Finite-Element-Time-Domain particle tracking program. The particles were asked to follow relativistic laws. The full set of the Maxwell equations should be incorporated. FEMSTER is based on a calculus using differential forms and provides higher order finite elements. To integrate the particles a simple *leap frog* scheme is designated. The program is desired to be validated and benchmarked. Different geometrical domains should be tested. Finally some ideas were asked to be developed for parallelizing the program.

First of all I decided to treat the time domain integration of the fields without any particles. After reading a lot about differential forms, FEMSTER, FETD and solvers I began to put up the variational formulation. After treating some static problems with FEMSTER (Poisson problem) first steps could be made for implementing Assous' approach [1] with Lagrange multipliers. In the beginning I solved the resulting saddle point problem with an own implemented MINRES algorithm. Later on Peter Arbenz suggested to treat the problem with the saddle point modification (described in the appendix). But as was not exactly clear weather the approach is applicable or not and as I didn't know that the assembling process of the derivative matrix $\mathbf{K}$ differs from assembling other matrices, much time passed by having an integrator which did not converge! Once the modified saddle point problem was understood, the correct implementation became easy. To visualize the data I wrote a function which generates files in the VTK format. These files can be visualized with the program *ParaView*. Then the field integrator had been tested - with success! The spectrum, calculated with an FFT, contains the correct frequencies and errors do behave as expected. The Lagrange multiplier approach is applicable for this topic. Then, remaining time had become marginal. I tried to include the particle framework IPPL. Two major innovations had to be done: provide the particles with its fields E and H (FEMSTER $\rightarrow$ IPPL) and provide the field integrator with the distribution of charge density and current density (IPPL $\rightarrow$ FEMSTER). As particle integrator a *Velocity Verlet* has been chosen. This integrator works well. The particles are not yet implemented in terms of the relativistic laws. There is still a bug in the code as one can see in the generated movies. For particles close to the local cell interfaces unphysical electromagnetic waves are generated. On the other hand this unintentional radiation is propagating correctly.
Because of lack of time geometrically complex domains could not be studied. Likewise, ideas for parallelizing the whole program could not be developed. But an ab initio assumption can be made: IPPL is already fully parallelizable and the main functions of FEMSTER use BLAS routines. It should thus be possible to parallelize the code (e.g. using TRILINOS).

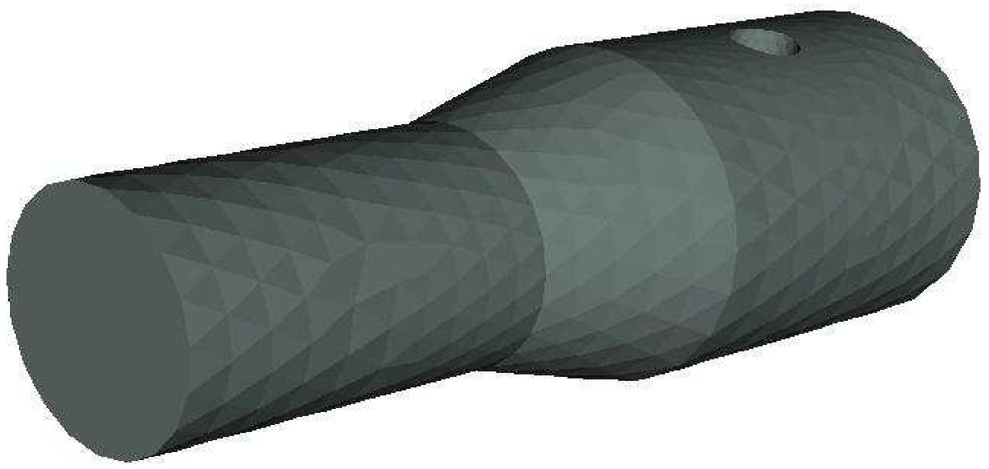


Figure 5.1: A tetraedral mesh of a possible geometrical domain Ω of a particle accelerators

5.2 Future work

This diploma work builds the foundation for a proceeding project (PhD thesis). In a few years at PSI a new particle accelerator is going to be built. For designing such a machine an FETD-particle tool is needed. On the way to a powerful program, many additional features have to be appended. Only a short list of what is to do reads:

- Analyzation of particle solutions
- Allow absorbing boundary conditions
- Involvement of tetrahedras
- Implement the hp-FEM
- Parallelization

By the way, questions could be answered for which I had not enough time:

- How does λ_{max} from equation (3.24) depend on the given mesh?
- What is the reason for the error wiggling as Δt gets very small?

6 Appendix

6.1 Modification of the saddle point problem

As we have seen in the Section 2.5, we need to solve the saddle point problem

$$\begin{pmatrix} \mathbf{M} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \tag{6.1}$$

Because of the 0 block the matrix from above is symmetric and indefinite. That means that the conjugate gradient algorithm can not be applied.

We define $\mathbf{C} := \mathbf{MK}$ and $\mathbf{S} := \mathbf{C}^T\mathbf{K} = \mathbf{K}^T\mathbf{MK}$ (for any symmetric, positive definite $\mathbf{M}$ and for an incidence matrix $\mathbf{K}$). Then the matrix from (6.1) can be expressed as

$$\begin{pmatrix} \mathbf{M} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{K} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \tag{6.2}$$

With the definition

$$\begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} := \begin{pmatrix} \mathbf{I} & \mathbf{K} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \tag{6.3}$$

equation (6.1) can be written as

$$\begin{pmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} - \mathbf{K}^T\mathbf{f} \end{pmatrix} \tag{6.4}$$

This system of equations is equivalent to the following two systems of equation plus a correction

$$\begin{aligned} \mathbf{Mv} &= \mathbf{f} && (6.5) \\ \mathbf{Sy} &= \mathbf{K}^T\mathbf{f} - \alpha^{-1}\mathbf{g} && (6.6) \end{aligned}$$

$$\mathbf{x} = \mathbf{v} - \mathbf{Ky}. \tag{6.7}$$

If $\mathbf{K}$ is the derivative matrix of the 0-form (i.e. the gradient matrix), it turns out (after integrating by parts) that $\mathbf{S}$ is the stiffness matrix of the 0-form (i.e. the Poisson matrix). The two new systems of equations are positive definite and can be solved by applying the CG method. The gain of efficiency is discussed in the next section.

6.2 Numerics of the modified saddle point problem

There were times when I didn't know about the saddle point modification. So I tried to solve equation (6.1) directly with the MINRES algorithm. As with the procedure of applying boundary conditions nonzero elements arise on the diagonal of the matrix (see figure 6.1), there is room for hope that the overall matrix is not indefinite anymore and that the CG method converges. For

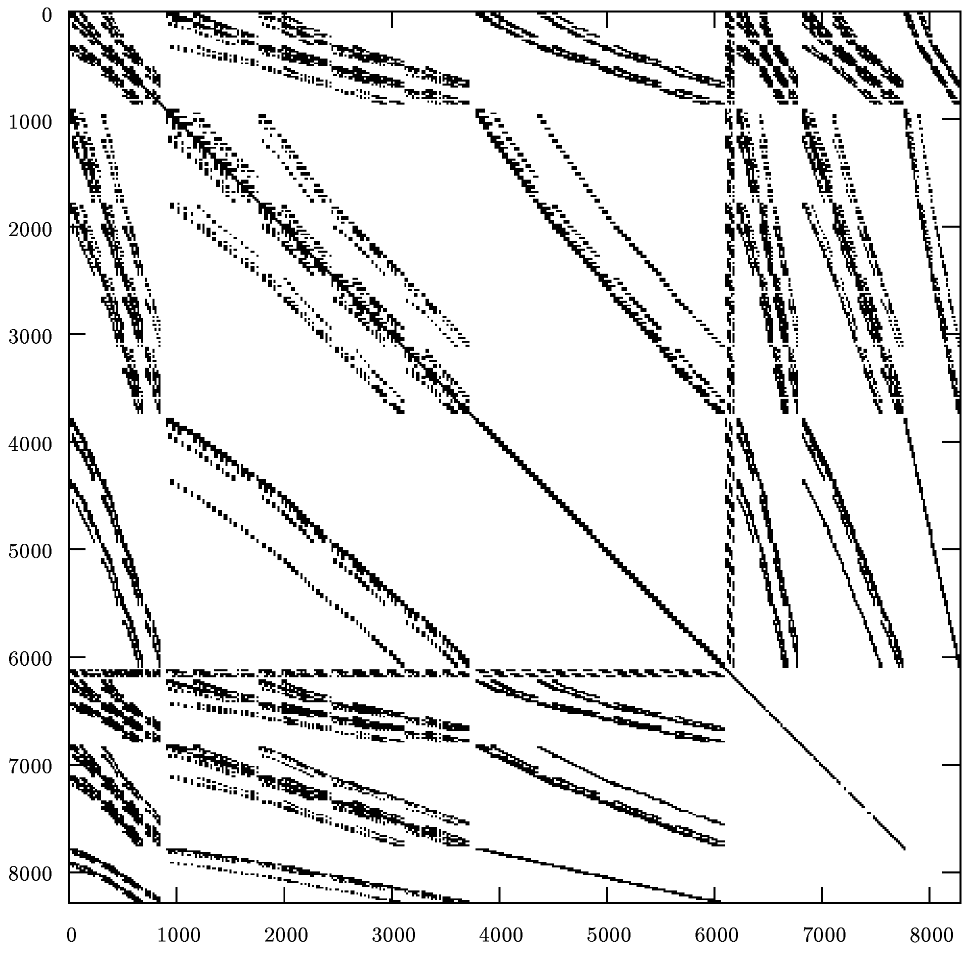
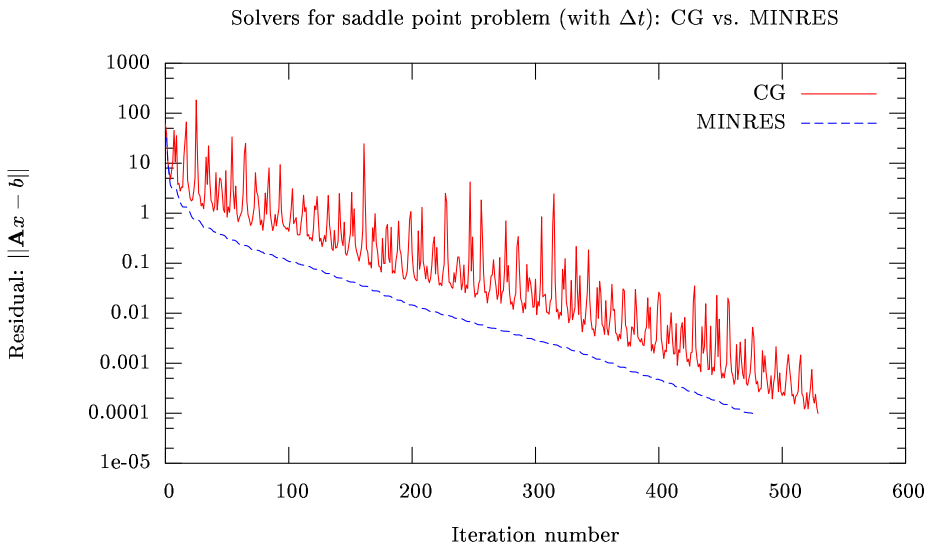
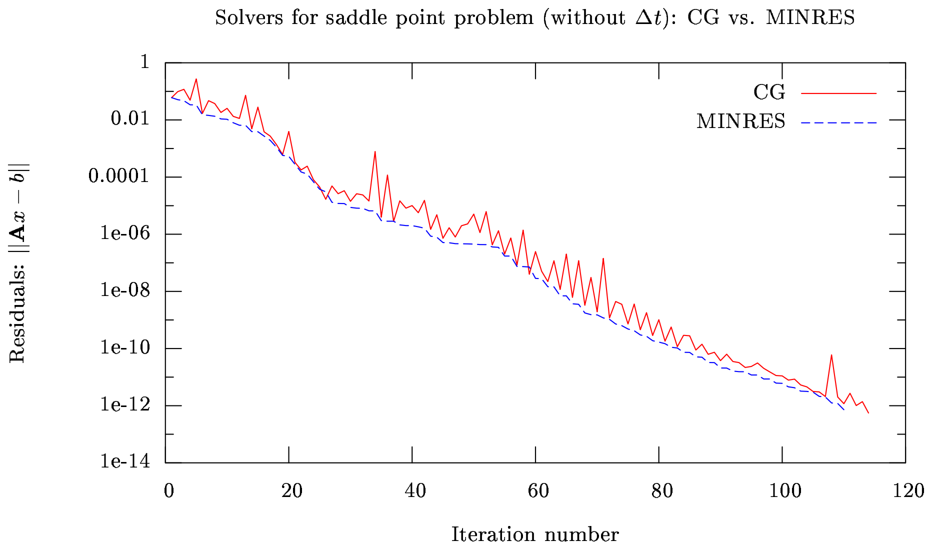


Figure 6.1: The global matrix of a hexaedral 64 cell grid of a cube with Dirichlet boundary conditions. The nonzero components in the lower right block appear as a result of applying Dirichlet boundary conditions to the matrix. Only 1.8 % of the components are nonzeros.

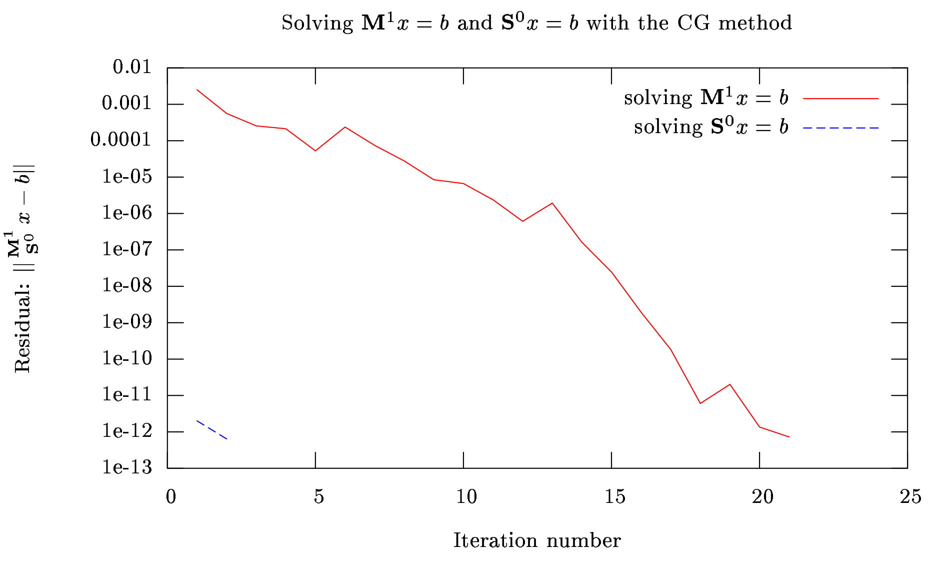
a random initial condition (64 cells,$p = 3$) I have analyzed the convergence behaviour with both the MINRES and the CG method. The error received with the MINRES converges smoother and faster than the CG error (see next figure). It is important to mention that here the components of the **MK** and $(\mathbf{MK})^T$ blocks are multiplied by Δt as in equation (2.35) (instead of joining Δt with $\tilde{\phi}$).



The next figure shows the convergence behaviour for the right hand side coming from a TM_{110} mode with the same system matrix (Figure 6.1). Here Δt is not included in the matrix. Now the convergence rate is much better. The error line of the MINRES defines a lower bound for the CG error.



Now we consider the convergence behaviour for the saddle point modification (2.31). Instead of one system of equations two have to be solved. But as can be seen by comparing the number of iterations the saddle point modification is 5 times more efficient! Solving the second system of equations ($\mathbf{S}_0\mathbf{x} = \mathbf{b}$) requires only two iterations for this special parameterization but is in general dependent on the problem size. It has to be mentioned that this grid is fairly basic and the simulated mode is well behaving.



6.3 The problem of assembling K_{01}

Applying the saddle point modification to (6.1) requirs that $\mathbf{C} = \mathbf{MK}$ in a global sense ($\mathbf{M}$ is the same matrix as the upper left block of the saddle point matrix). From an other point of view one could argue that the matrix $\mathbf{C}$ has to be assembled from local element matrices $\mathbf{C}^{\Sigma} = \mathbf{M}^{\Sigma}\mathbf{K}^{\Sigma}$ (where Σ symbolizes the element). For the assembling process we use the notation $\mathbf{C} = \mathcal{A}_{\Sigma}(C^{\Sigma})$. Incorporating both considerations yields the condition

$$\mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma} \cdot \mathbf{K}^{\Sigma}) = \mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma}) \cdot \mathcal{A}_{\Sigma}(\mathbf{K}^{\Sigma}). \tag{6.8}$$

If the assembling process of $\mathbf{K}$ were the same as the one of $\mathbf{M}$ - namely summing up the local element matrices - then it is easy to show that the claim (6.8) can not be satisfied! It took a long time until I recognized (by an aspicious hasard) that the assembling procedures of $\mathbf{K}$ and $\mathbf{M}$ differ. Many divergent runs had been performed until then and much desperation spread out.

The following compilation shows that if $\mathbf{K}$ is overwritten by the local matrices $\mathbf{K}^{\Sigma}$, claim (6.8) holds true:

- $\mathcal{M}_1()$ is the local to global mapping for 1-forms.
  $\mathcal{M}_1^{-1}()$ is the global to local mapping for 1-forms.
- $\mathcal{M}_0()$ is the local to global mapping for 0-forms.
  $\mathcal{M}_0^{-1}()$ is the global to local mapping for 0-forms.

$$\begin{array}{rrcll}
& \mathbf{M}_{m,n} & = & \mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma})_{m,n} & := \sum_{\Sigma} \mathbf{M}^{\Sigma}_{\mathcal{M}_1^{-1}(m),\mathcal{M}_1^{-1}(n)} \\
& \mathbf{K}_{m,n} & = & \mathcal{A}_{\Sigma}(\mathbf{K}^{\Sigma})_{m,n} & := \biguplus_{\Sigma} \mathbf{K}^{\Sigma}_{\mathcal{M}_1^{-1}(m),\mathcal{M}_0^{-1}(n)} \\
\Longrightarrow & \mathbf{C}^{\Sigma}_{i,j} & = & \mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma} \cdot \mathbf{K}^{\Sigma})_{i,j} & = \sum_k \mathbf{M}^{\Sigma}_{i,k} \cdot \mathbf{K}_{\mathcal{M}_1(k),\mathcal{M}_1(j)}
\end{array}$$

$$\begin{aligned}
\mathbf{C}_{m,n} = \mathcal{A}_{\Sigma}(C^{\Sigma})_{m,n} &= \sum_{\Sigma} \mathbf{C}^{\Sigma}_{\mathcal{M}_1^{-1}(m),\mathcal{M}_0^{-1}(n)} \\
&= \sum_{\Sigma}\sum_{k} \mathbf{M}^{\Sigma}_{\mathcal{M}_1^{-1}(m),k} \cdot \mathbf{K}_{\mathcal{M}_1(k),\mathcal{M}_0(\mathcal{M}_0^{-1}(n))} \\
&= \sum_{\Sigma}\sum_{k} \mathbf{M}^{\Sigma}_{\mathcal{M}_1^{-1}(m),k} \cdot \mathbf{K}_{\mathcal{M}_1(k),n} \\
&= \sum_{k}\sum_{\Sigma} \mathbf{M}^{\Sigma}_{\mathcal{M}_1^{-1}(m),k} \cdot \mathbf{K}_{\mathcal{M}_1(k),n} \\
&= \sum_{k} \mathbf{M}_{m,\mathcal{M}_1(k)} \cdot \mathbf{K}_{\mathcal{M}_1(k),n} \\
&= \sum_{\mathcal{K}=\mathcal{M}_1(k)} \mathbf{M}_{m,\mathcal{M}_1(\mathcal{M}_1^{-1}(\mathcal{K}))} \cdot \mathbf{K}_{\mathcal{M}_1(\mathcal{M}_1^{-1}(\mathcal{K})),n} \\
&= \sum_{\mathcal{K}} \mathbf{M}_{m,\mathcal{K}} \cdot \mathbf{K}_{\mathcal{K},n}
\end{aligned}$$

$$\Longrightarrow \qquad \mathbf{C} = \mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma} \cdot \mathbf{K}^{\Sigma}) = \mathcal{A}_{\Sigma}(\mathbf{M}^{\Sigma}) \cdot \mathcal{A}_{\Sigma}(\mathbf{K}^{\Sigma}) \tag{6.9}$$

Bibliography

[1] F. Assous, P. Degond, E. Heintze, P.A. Raviart, J. Segre, *On a Finite-Element Method for Solving the Three-Dimensional Maxwell Equations*, J. Comp. Phys. 109, 222-237 (1993)

[2] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numerica (2002), pp. 237-339

[3] FEMSTER homepage: `http://www.llnl.gov/casc/emsolve`

[4] P. Castillo et al., *Discrete Differential Forms: A Novel Methodology for Robust Computational Electromagnetics*, 2003, `http://www.doc.gov/bridge`

[5] J.C. Nedelec, *Mixed Finite Elements in* $\mathbb{R}^3$, 1980, Numer. Math. 35, 315-341

[6] V.I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer, 2000

[7] K. Königsberger, *Analysis 2*, Springer 1997, 2. Auflage

[8] R.N. Rieben, *A Novel High Order Time Domain Vector Finite Element Method for the Simulation of Electromagnetic Devices*, Diss. 2004, University of California, Davis, www.llnl.gov/CASC/emsolve/publications.html

[9] R.D. Graglia et al., *Higher Order Interpolatory Vector Bases for Computational Electromagnetics*, IEEE Trans. Ant., 45 no. 3 (1997)

[10] J.M. Jin, *The Finite Element Method in Electromagnetics*, second edition 2002, IEEE press, Wiley-Interscience

[11] P. Monk, *Finite Element Methods for Maxwell's Equations*, University Press, Oxford, 2003

[12] Ch.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, McGraw-Hill 1985

[13] J.D. Jackson, *Klassische Elektrodynamik*, 3. Auflage, De Gruyter 2002

[14] G. Lehner, *Elektrodynamische Feldtheorie*, 5. Auflage, Springer 2006

[15] W. Demtröder *Experimentalphysik 2, Elektrizität und Optik*, 2. Auflage, Springer 2004

[16] Ch. Schwab, *Numerical Methods for Differential Equations*, Lecture Notes, ETHZ, Wintersemester 2004/2005

[17] Th.M. Antonsen et al., *Advances in Modeling and Simulation of Vacuum Electronic Devices*, IEEE proceedings, 87 no. 5 (1999)

[18] D.R. Lynch, K.D. Paulsen, *Time-Domain Integration of the Maxwell Equations on Finite Elements*, IEEE trans. ant., 38 no. 12 (1990)

[19] G. Rodrigue, D. White, *A Vector Finite Element Time-Domain Method for solving Maxwell's Equations on unstructured Hexaedral Grids*, SIAM J. Sci. Comput., 23 no. 3 (2001) pp 683-706

[20] E. Sonnendruecker, *A Finite Element Formulation of the Darwin PIC Model for use on unstructured Grids*, J. Comp. Phys., 121, 281-297 (1995)

[21] C.-D. Munz, R. Schneider, U. Voss, *A Finite-Volume Method for the Maxwell Equations in the Time Domain*, SIAM J. Sci. Comp., 22 no. 2 (2000), pp. 449-475

[22] P. Arbenz, Z. Drmač, *On Positive Semidefinite Matrices with Known Null Space*, SIAM J. Matrix Anal. Appl., 24 no. 1 (2002), pp. 132-149

[23] M. Rozloznik, *Saddle point problems, iterative solution and preconditioning: A short Overview*, AMS: 65F10 (2004)

[24] C.C. Paige, M.A. Saunders, *Solution of Sparse Indefinite Systems of Linear Equations*, SIAM J. Numer. Anal., 12 no. 4 (1975)

[25] J.H. Bramble, J.E. Pasciak, A.T. Vassilev, *Analysis of the inexact Uzawa Algorithm for Saddle Point Problems*, SIAM J. Numer. Anal., 34 no. 3 (1997), pp. 1072-1092

[26] M.A. Heroux et al., *An Overview of the Trilinos project*, ACM Trans. Math. Soft., 31 no. 3 (2005), pp. 397-423.