Design and simulation of an invisibility cloak device in Julia

Rodney Voskamp

5-7-2020

Supervisors:
Dr. Cools
Dr. Adam

Delft University of Technology
Department of EEMCS

Abstract

Mankind has always adapted materials to fulfill a need. Recent developments in nano-science allows us to construct objects with varying permittivity and permeability. One possible structure is an invisibility cloak device. This device is put around an object. Any plane wave that enters the cloak should go around this object. The wave exits the cloak as a plane wave, rendering the object on the inside invisible. Since making such a device is quite difficult, it is useful to simulate it first. A relatively new programming language is Julia [1], [2]. Julia has a high performance and this makes it an interesting language to use for these simulations. In this work, the material properties of such a device was calculated. A simulating tactic was developed. This tactic was a hybrid version between the Method of Moments (MoM) and the Finite Elements Method (FEM). Both these methods and the hybrid method were implemented in BEAST.jl [3] and CompScienceMeshes [4] and were tested using simulations in EMwavesBEP.jl[5]. While the FEM and MoM gave great results, the hybrid algorithm had only a good result for the electric fields. The magnetic fields failed to show the wanted results. The previous determined cloak was simulated using the hybrid algorithm. The cloaking property was clearly visible and the wave exited as a plane wave. A far field should be calculated to verify this further.

Acknowledgements

In this bachelor thesis, I present the results of simulating an invisibility cloak device in Julia. This thesis could not have been written without the help of my supervisors.

I would like to thank Dr. Cools for his help. The work he had already done in BEAST.jl [3] and CompScienceMeshes.jl [4] proved to be invaluable. He was a great help in spotting mistakes in the scripts and was always available to ask for advice.

I want to thank Dr. A.J.L. Adam too. His ideas for the cloak and the literature he found were a big help in this work.

Rodney Voskamp, 5-7-2020

Publications from this Thesis

In this work, we simulate the electric field in an object using the FEM and a hybrid method that uses the FEM. The magnetic field can be reconstructed from the electric field using Faraday's law. In this work, the magnetic field could not be successfully simulated. In these methods, both field are simulated using curl-conforming Nedelec test functions. When we calculate the magnetic field in this way, the curl-conforming test functions become div-conforming test functions. This could explain the problems with the simulations of the magnetic field.
During the writing of this work, progress was made in this department. Using different test functions, the magnetic field is constructed using curl-conforming elements. This work was submitted to two conferences:

1. K. Cools, R. Voskamp, A. Adam, "Conforming Magnetic Field Computation in 3D-FEM Using Dual Basis Functions", *IEEE Antennas and Propagation Symposium 2020* (accepted but postponed due to COVID-19)

2. K. Cools, R. Voskamp, A. Adam, "Dual Spaces for 3D Finite Element Method", *URSI General Assembly 2020* (accepted but postponed due to COVID-19)

Table of contents

1 Introduction

During your life, you will use a lot of different materials with different properties. Some of these properties can only be achieved when they are manually constructed. Only recently, we have been able to construct materials with almost arbitrarily varying permittivity and permeability. This allows us to design materials with interesting properties like light concentrators and cylindrical and planar hyperlenses.[6]. In this work, we look at another property: cloaking an area from a plane wave. A cloak like this works by constructing a design around what we want to cloak. This design should let a plane wave enter a cloak, split this wave around the inner object. This split wave will connect on the other side of the design and should continue like there was no cloak in the first place. This will make the inner object and the cloak undetectable from outer viewers, practically making the object invisible. If we want to make such a design, it is useful to test it in a simulation. In this work, a simulation strategy will be constructed using different methods.
A good simulation requires the choice of a good programming language. An interesting upcoming programming language is Julia, [1], [2]. Julia is a young, free, high-level and high-performance programming language. Julia is fast due to its JIT-compiler and allow users to use an interactive command line. In this report, we will try to implement the simulation of an invisibility cloak in Julia. As stated, Julia is a young language and this means that there is no complete numerical electromagnetic package. We start with two existing packages: BEAST.jl [3] and CompScienceMeshes.jl [4]. We add new functions to these packages and then test them using simulations. At the end we should be able to simulate an invisibility cloak. All of these simulations can be found in EMwavesBEP.jl [5].
In chapter 2, we will give an overview of the theory. We start by deriving the curl-curl equation. After that, the derivation of the cloak properties using tensor transformations are given. This will be followed by a description of the used test functions. At last, the different simulation methods are given. Actual simulations using these methods can be found in chapter 3. We discuss the conclusions of the simulations in chapter 4. The appendixes at the end of the report give more information about the different elements of the program in use and the assembly of the test functions.

2 Background and state of the art

2.1 Derivation of the curl-curl equation

In this section, the curl-curl equation will be derived. This equation will be used in our simulations. In this report, we expose an object with light. Light is an electromagnetic wave, so to make an accurate simulation, the object has to agree with the physics rules of electromagnetics. The four Maxwell equations describe them. We will derive the curl-curl-equation by using the Maxwell equations:

$$\nabla \times \mathcal{E} = -\frac{\partial \mu \mathcal{H}}{\partial t}$$

$$\nabla \cdot \epsilon \mathcal{E} = \rho$$

$$\nabla \times \mathcal{H} = \frac{\partial \epsilon \mathcal{E}}{\partial t} + \mathcal{J}$$

$$\nabla \cdot \mu \mathcal{H} = 0$$

Here $\mathcal{E}$ and $\mathcal{H}$ are the electric and magnetic fields. The material properties that are connected to them are ϵ, the permittivity and μ, the permeability. The source terms are the electric charge density ρ and the current $\mathcal{J}$. We assumed that we used a linear material. We can write for a linear medium that:

$$\mathcal{D} = \epsilon \mathcal{E}$$

$$\mathcal{B} = \mu \mathcal{H}$$

We will now adept the Maxwell equations to study them in frequency domain. We assume that the fields oscillate only at frequency ω. This gives us a sinusoidal wave. We rewrite the fields as complex function with complex phasors $E(r)$ and $H(r)$ as followed:

$$\mathcal{E}(r,t) = Re[E(r)e^{j\omega t}]$$

$$\mathcal{H}(r,t) = Re[H(r)e^{j\omega t}]$$

We substitute these into the Maxwell equations to obtain:

$$\nabla \times E = -j\omega\mu H \tag{1}$$

$$\nabla \cdot \epsilon E = \rho \tag{2}$$

$$\nabla \times H = j\omega\epsilon E + J \tag{3}$$

$$\nabla \cdot \mu H = 0 \tag{4}$$

We can now derive the curl-curl-equation. We substitute H from formula 1 into formula 3.

$$\nabla \times (\frac{1}{-j\omega\mu} \nabla \times E) = j\omega\epsilon E + J$$

By setting $\mu = \mu_0\mu_r$, $\epsilon = \epsilon_0\epsilon_r$, $k_0 = \omega\sqrt{\mu_0\epsilon_0}$ and $Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}}$, we rewrite the equation as

$$\nabla \times (\frac{1}{\mu_r}\nabla \times E) - k_0^2\epsilon_r E = -jk_0 Z_0 J \tag{5}$$

This is the wave equation for light with a source. We simplify this equation by giving a general source term $f(r)$. We also write it with a general wavelength $\kappa = \omega\sqrt{\mu\epsilon}$. In vacuum, this wavelength is equal to $\frac{1}{c}$ with c being the speed of light.

$$\nabla \times \nabla \times E - \kappa^2 E = f(r) \tag{6}$$

In this work, we will use complex varying permeability and permittivity. To do this, we will use a tensorial ϵ and μ. The curl-curl equation can be derived in the same way as was done above. We obtain the same equation, with the difference that the ϵ and μ are tensors instead of scalars. This was shown in Koshiba [7].

2.2 Cloak properties and tensor transformation

In this section, we will derive the material properties of the cloak. These properties are given as tensors in cylindrical coordinates. To implement these tensors in our simulation, we will need to transform them to cartesian coordinate.

2.2.1 Transformation between cartesian and cylindrical coordinates

In this section, some important transformations are shown. A transformation from coordinate system A to coordinate system B is written by Jacobian matrix $J_{A,B}$. This goes as followed:

$$J_{A,B} = \frac{\partial(A_1, A_2, A_3)}{\partial(B_1, B_2, B_3)} = \begin{bmatrix} \frac{\partial A_1}{\partial B_1} & \frac{\partial A_1}{\partial B_2} & \frac{\partial A_1}{\partial B_3} \\ \frac{\partial A_2}{\partial B_1} & \frac{\partial A_2}{\partial B_2} & \frac{\partial A_2}{\partial B_3} \\ \frac{\partial A_3}{\partial B_1} & \frac{\partial A_3}{\partial B_2} & \frac{\partial A_3}{\partial B_3} \end{bmatrix} \tag{7}$$

Next, we define the transformation matrix. This transformation matrix shows how a tensor gets mapped to a different coordinate system. We start by giving the matrix. Afterwards, we show the relations:

$$T_{A,B} = \frac{J_{A,B}^T J_{A,B}}{det(J_{A,B})} \tag{8}$$

$$\epsilon' = \epsilon T_{A,B}^{-1} \quad \mu' = \mu T_{A,B}^{-1} \tag{9}$$

We can now transform from a cartesian (x,y,z) to a cylindrical (r,θ,z) system. If we plug this information into formula 7, we see that:

$$J_{x,r} = \begin{bmatrix} cos(\theta) & -rsin(\theta) & 0 \\ sin(\theta) & rcos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = R(\theta)D(1, r, 1) \tag{10}$$

Matrix D is a diagonal matrix and matrix R is a rotation matrix. This last matrix is given as:

$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The more important properties about this rotation matrix are:

$$det(R(\theta)) = 1 \tag{12}$$

$$R(\theta)^{-1} = R(\theta)^T = R(-\theta) \tag{13}$$

We described how we can transform from one system to the other, but we have to inverse this relation too. We can calculate this transformation from cylindrical to cartesian coordinates by using the Jacobian. However, it is quicker to take the inverse of Eq. 10. Since transforming back without doing anything should give us the identity matrix.

$$J_{r,x} = D(1, \frac{1}{r}, 1)R(-\theta) \tag{14}$$

2.2.2 Permittivity and permeability tensor in the cloak

In this chapter, we determine the material properties of the cloak. We do this by transforming our system from cartesian to cylindrical coordinates. We continue by reforming this system to get the cloaking property. At last, we transform the system back into cartesian coordinates. This was done in Nicolet et al.[8] The transformation into cylindrical coordinates was shown in the previous section. We now transform these coordinates (r,θ,z) into different cylindrical coordinates (r',θ',z'). We want to map a cylinder with radius R_2 into a cylinder with a hole of radius R_1. This can be achieved as followed:

$$\begin{aligned} r' &= \frac{R_2 - R_1}{R_2} r + R_1 \\ \theta' &= \theta \\ z' &= z \end{aligned} \tag{15}$$

We can quickly determine the Jacobian by using Eq. 7:

$$J_{r,r'} = D(\frac{R_2}{R_2 - R_1}, 1, 1) \tag{16}$$

At last, we transfer these system back into a new cartesian system (x',y',z'). This relation was shown in Eq. 14. We can now combine all the Jacobian to get a Jacobian for the whole procedure.

$$
\begin{aligned}
J_{x,x'} &= R(\theta)D(\frac{R_2}{R_2-R_1}, \frac{r}{r'}, 1)R(-\theta) \\
&= R(\theta)D(\frac{R_2}{R_2-R_1}, \frac{R_2-R_1}{R_2}\frac{(r-R_1)}{r'}, 1)R(-\theta)
\end{aligned} \tag{17}
$$

We now have all the tools to calculate the transformation matrix. Instead of giving this matrix, we immediately give the inverse of it $T^{-1}_{x,x'}$.

$$
T^{-1}_{x,x'} = R(\theta)D(\frac{r-R_1}{r}, \frac{r}{r-R_1}, (\frac{R_2-R_1}{R_2})^2\frac{r-R_1}{r})R(-\theta) \tag{18}
$$

$T^{-1}_{x,x'}$ give us the material properties in cylindrical coordinates (r',θ',z'). We start with a permittivity and permeability for air: $\epsilon = \mu = D(1,1,1)$. Using this and Eq. 9, we get an expression for the material properties in cylindrical coordinates. This means that we can implement material properties into code by changing the diagonal entries and then using Eqs. 9 and 18, we get the properties for the material in cartesian coordinates. This result is also given in Schurig et al. [9]. We will only show the inside of the cloak, so we can use the simpler material as described in Schurig [9] (Eq. 3):

$$
\begin{aligned}
\epsilon_x &= \left(\tfrac{R_2}{R_2-R_1}\right)^2 \\
\mu_r &= \left(\tfrac{r-R_1}{r}\right)^2 \\
\mu_\theta &= 1
\end{aligned} \tag{19}
$$

These are inserted in the correct place in the diagonal matrix like in formula 18. The unspecified diagonal entries are set as 0. Formula 9 gives the material properties that will be defined in the simulations.

2.3 Used trial and test functions

In this work, we calculate the electric field in an object. Depending on the simulation method, we either calculate the field inside the entire object, or we calculate it by using a solution on the boundary. Since there are an infinite amount of point on- and inside an object, we cannot calculate the electric field directly. We solve this by dividing the object into smaller polytopes like cubes and triangles. Each of these smaller elements consist of vertices edges and faces. All of them and the element together are called a mesh.
In the simulation techniques that will follow, our mesh will consist of triangles or tetrahedrons. On these geometries, a function will be described in such a way that all boundary conditions will be satisfied. Because we try to solve Maxwell equation, the basis functions are of a vectorial nature. The four used basis functions are described below.

2.3.1 Raviart-Thomas (2D)

A useful test function for the Method of Moments is the Raviart-Thomas (RT or Rao-Wilton-Glisson) test function. This test function lives on triangles. The RT we use here is its simplest form with only one basis function per edge. We make sure that there is a normal component only on this edge. A useful property of the RT is that this component is always 1. This makes the RT function divergence-conforming and guarantees continuity. We show RT by picking two adjacent triangles in the mesh. We now construct the function such that only the normal component on the shared edge is non-zero. This is done in the following figure:

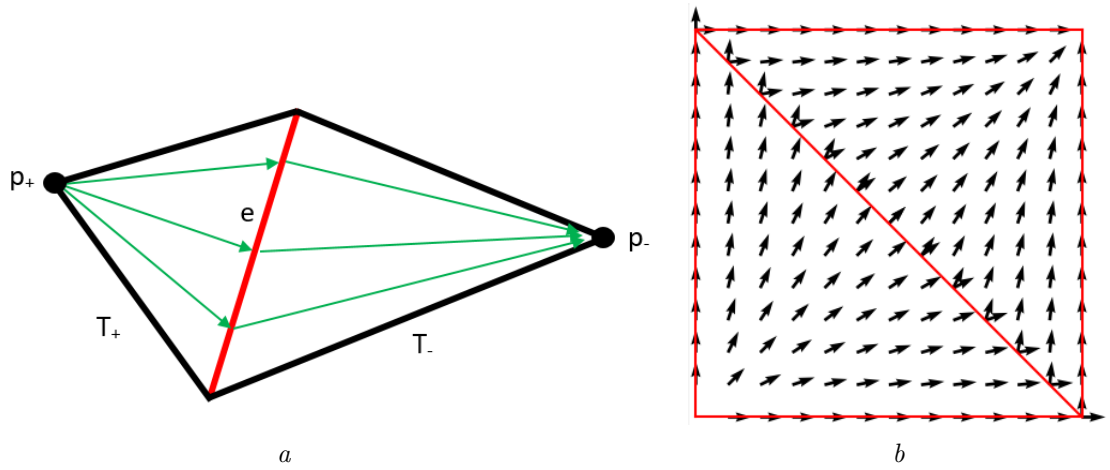


Figure 1: The RT test function shown schematically (a) and an example (b). The vectors in (b) are normalized.

Figure 1 shows the RT basis function. For a point r inside triangle T_+, the vector in r points away from p_+. Likewise, for a point r in triangle T_-, the vector in r points towards p_-. This ensures that there is no normal component on each edge, except for the shared edge e. Now using that A_+ and A_- are the areas of respectively T_+ and T_- and defining that l_e is the length of edge e, we get the following function $f(r)$ inside the two triangles:

$$f(r) = \begin{cases} \frac{l_e}{2A_+}(r - p_+) & r \in T_+ \\ -\frac{l_e}{2A_-}(r - p_-) & r \in T_- \\ 0 & otherwise \end{cases} \tag{20}$$

2.3.2 Raviart-Thomas (3D)

It is possible to construct a basis function in 3D in the same way as in the previous section. This test function is defined on a tetrahedron. The big differences are that there is now a unique test function per every face, instead of every edge. In Eq. 2.3.1, the areas A change into the volumes V and the edge e into a face F. in the next figure an example of this function is shown:

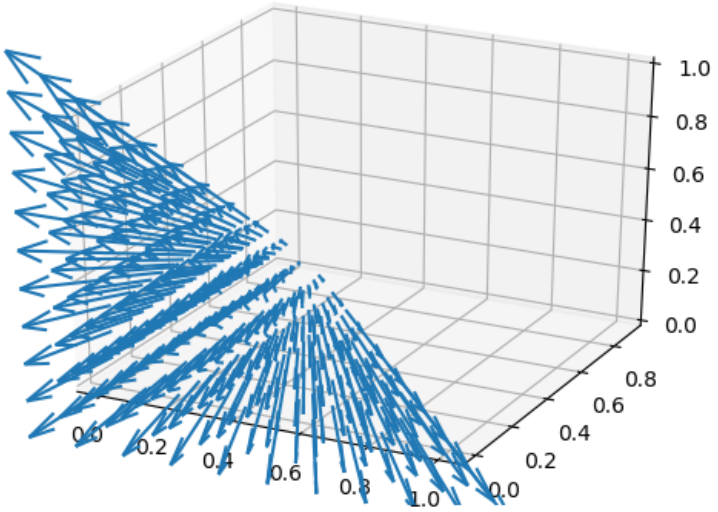


Figure 2: The RT basis function in 3D on a face.

As can been seen in Fig. 2, there is no normal component on all but one faces. If we take one of these faces, we will see that this gives us the RT_{2D} basis function. The RT_{3D} basis function are sometimes called the Nedelec divergence conforming basis function [10]. This name will not be used in this report, because we use another Nedelec basis function, as described in the next subsections.

2.3.3 Nedelec (2D)

This Nedelec basis functions are not used extensively in this report. It is only used to make one plot. However, one of the more important test functions has a lot in common with this one. Just like the RT case, we pick the easiest Nedelec functions. This means that we have only one basis function per edge. The Nedelec basis functions are curl conforming. This means that the tangential components are continuous. To achieve this, we make sure that the Nedelec function is tangential on the used edge only. An example is given below in Fig. 3.

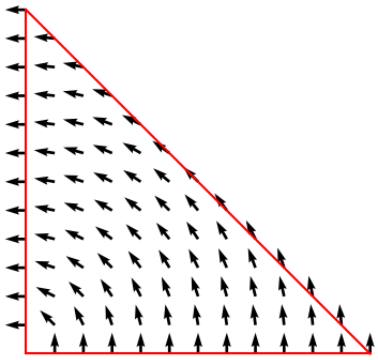


Figure 3: The Nedelec basis function on the edge that starts in the bottom right and ends in the top left. Each vector is normalized.

2.3.4 Nedelec (3D)

Just like the RT basis functions, we will now construct a 3D version of the Nedelec basis functions. If the 3D Raviart-Thomas was called the Nedelec divergence conforming basis function, this one is called the *Nedelec curl conforming basis* function. We will just use Nedelec. The 3D Nedelec test functions are still defined on its edges, so there will be 6 basis functions per tetrahedron. One of these is given in the next figure:

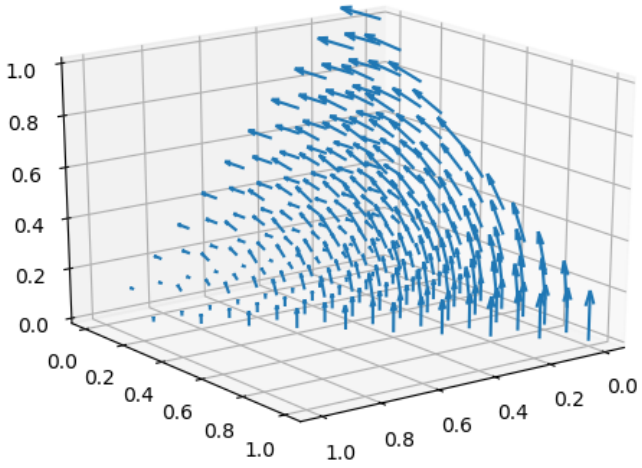


Figure 4: The Nedelec basis function on the edge that goes from the bottom right to the top.

The same link that was made between the RT_{2D} and the RT_{3D} basis functions can be made here. If we look at fig. 4, we see that the 2D Nedelec basis functions appear on the two faces that have the edge used in the 3D variant.

2.3.5 Relations between basis functions

We now give some relations between the 4 basis functions. This relations are given using the deRham diagram. The first two relations are between the Nedelec and the Raviart-Thomas basis functions. For both the 2D as the 3D variant the following holds: the curl of the Nedelec functions can be written as RT functions (see [11]- [12]). Besides that, the tangential trace of a Nedelec function is a RT function. These relations are summarised in the scheme that was given by [12]:

$$\begin{array}{ccccc} \mathcal{S}_1(\Omega_h) & \xrightarrow{\textbf{grad}} & \mathcal{ND}_1(\Omega_h) & \xrightarrow{\textbf{curl}} & \mathcal{RT}_0(\Omega_h) \\ \downarrow{\scriptstyle |\partial\Omega \text{ (point trace)}} & & \downarrow{\scriptstyle \cdot \times \boldsymbol{n}\,|\partial\Omega \text{ (tangential trace)}} & & \downarrow{\scriptstyle \cdot\cdot\,\boldsymbol{n}\,|\partial\Omega \text{ (normal trace)}} \\ \mathcal{S}_1(\Gamma_h) & \xrightarrow{\textbf{curl}_\Gamma} & \mathcal{RT}_0(\Gamma_h) & \xrightarrow{\text{div}_\Gamma} & \mathcal{Q}_0(\Gamma_h). \end{array}$$

2.4 Method of Moments

If we want to test if our design makes the inner part of the cloak invisible, it is useful to test it in a simulation. One simulation approach is based on the Method of Moments (MoM). This method is sometimes called the Boundary Element Method (BEM). An important advantage of the MoM is that it can simulate an infinite space. This is important to test our design, because we want that it is indistinguishable from a simulation without a design in the far-field range. The discretisation in this method are triangles over the surface of the cloak. We use triangles, because this geometry can accurately describe difficult surfaces. This chapter starts by giving a general approach of the MoM. After that, the expressions that follow from the Maxwell equations are stated. At last, the method of simulating a homogeneous object is given. The used expressions will be derived first and a schematic overview of the algorithm will be given afterwards.

2.4.1 General Method

In this section, a general approach of the MoM is given. We start by stating our integral equation as followed:

$$\mathcal{L}u = f$$

with $\mathcal{L}$ being the integral operator, u the wanted solution and f a source term. Integral equations can be solved using a Greens function G. The Greens function verifies:

$$\mathcal{L}(G(r, r')) = \delta(r - r')$$

where $\delta(r - r')$ is the Dirac-delta function. We note the following relation

$$\int \mathcal{L}(G(r, r')f(r')d\Gamma' = \int \delta(r - r')f(r')d\Gamma' = f(r).$$

This allows us to see that the following solution must be true

$$u = \int G(r - r')f(r')d\Gamma'.$$

The last formula shows that the Greens function gives the solution of the integral equation using the response of the source function. More importantly, the Greens function allows us to get the solution in the domain outside the object [13]. If we use the left-hand-side of equation 6 for $\mathcal{L}$, we get the following Greens function:

$$G(r, r') = \frac{e^{-j\kappa|r-r'|}}{|r - r'|} \tag{21}$$

We will now try to solve the integral equation after discretization of the volume. The Greens formula is not explicitly stated, but is hidden in the operator $\mathcal{L}$. We first assume that we can rewrite u as a linear combination of our test functions:

$$u = \sum_{j=1}^{N} c_j v_j$$

with v_n being a test function. This gives us the following equation:

$$\sum_{j=1}^{N} c_j \mathcal{L}(v_j) = f$$

We would like to solve a square matrix equation. To make this a square matrix, we multiply the previous equation by a weighting function w_k with $k = 1, 2, .., N$ and integrate over the domain. In this report, we make sure that w_k are the same functions as v_k. This gives us the following formula:

$$\sum_{j=1}^{N} c_j \int_\Gamma w_k \mathcal{L}(v_j)d\Gamma = \int_\Gamma w_k f d\Gamma \tag{22}$$

We can rewrite this as a matrix equation.

$$\begin{aligned} [A][c] &= [b] \\ A_{kj} &=< w_k, \mathcal{L}(v_j) >_\Gamma \\ b_k &=< w_k, f >_\Gamma \end{aligned} \tag{23}$$

Solving this matrix equation gives us the solution for c. We use this in a linear combination to get an approximation for u.

In our case, we have to use the Maxwell equations. From Eq. 6, we can derive two equations [14]). These equations are based on the incident fields (notated wth *inc*) and two operators. The operators depend on the Greens function and the κ from Eq. 6. These operators are stated first:

$$\mathcal{T}(X) = j\kappa \oiint [X(r')G(r,r') + \frac{1}{\kappa^2}\nabla' \cdot X(r')\nabla G(r,r')]dS'$$

$$\mathcal{K}(X) = \oiint X(r') \times \nabla G(r,r')dS'$$

In the two equations, we use the cross product of the normal with the electric and magnetic fields.

$$J_s(r) = \sqrt{\frac{\mu_0}{\epsilon_0}}\hat{n} \times H(r)$$

$$M_s(r) = E(r) \times \hat{n}$$

We can now state the equations. We will call Eq. 24 the Electric Field Integral Equation (EFIE). Similarly, Eq. 25 is the Magnetic Field Integral Equation (MFIE). For example, we can use these equations to simulate a homogeneous object. This is shown in the next subsection.

$$-\hat{n} \times E^{inc}(r) = \frac{1}{2}M_s - \hat{n} \times \mathcal{T}(J_s) + \hat{n} \times \mathcal{K}(M_s) \tag{24}$$

$$\hat{n} \times H^{inc}(r) = \frac{1}{2}J_s + \hat{n} \times \mathcal{T}(M_s) + \hat{n} \times \mathcal{K}(J_s) \tag{25}$$

2.4.2 Derivation of the PMCHWT equations

In this section we derive a method for simulating a homogeneous object using eqs. 24 and 25. If the object is homogeneous, we can write the inside of this object as followed [14]. The equations are analoguus to the EFIE and the MFIE. The differences are that the operators are defined on the inside. This means that κ becomes a $\kappa_i = \omega\sqrt{\mu_i\epsilon_i}$. Moreover, the Greens function G_i is defined on the inside of the material, which is just using $\kappa = \kappa_i$ in Eq. 21. The operators on the inside are given below.

$$\mathcal{T}_i(X) = j\kappa_i \oiint [X(r')G_i(r,r') + \frac{1}{\kappa_i^2}\nabla' \cdot X(r')\nabla G_i(r,r')]dS'$$

$$\mathcal{K}_i(X) = \oiint X(r') \times \nabla G_i(r,r')dS'$$

To obtain the inside EFIE Eq. 26 and MFIE Eq. 27, we have to add a material factor $\eta_i = \sqrt{\frac{\mu_i}{\epsilon_i}}$ too. The operators work using the cross product of the electric and magnetic field with the normal as was given in the previous section.

$$\frac{1}{2}M_s + \eta_i\hat{n} \times \mathcal{T}_i(J_s) - \hat{n} \times \mathcal{K}_i(M_s) = 0 \tag{26}$$

$$\frac{1}{2}\eta_i J_s - \hat{n} \times \mathcal{T}_i(M_s) - \eta_i\hat{n} \times \mathcal{K}_i(J_s) = 0 \tag{27}$$

We can quickly see that this is correct, because We combine the EFIE on the outside (Eq.24) with Eq. 26 and the MFIE Eq. 25 with Eq. 27. This results in:

$$\hat{n} \times [\mathcal{T}(J_s) + \eta_i \mathcal{T}_i(J_s)] - \hat{n} \times [\mathcal{K}(M_s) + \mathcal{K}_i(M_s)] = \hat{n} \times E^{inc}(r) \tag{28}$$

$$\hat{n} \times [\mathcal{T}(M_s) + \frac{1}{\eta_i}\mathcal{T}_i(M_s)] - \hat{n} \times [\mathcal{K}(J_s) + \mathcal{K}_i(J_s)] = \hat{n} \times H^{inc}(r) \tag{29}$$

These are called the PMCHWT-equations, named after the first people that found them: Poggio, Miller, Chang, Harrington, Wu and Tsai. We discretise these two equations with the Raviart-Thomas test functions. This gives us a large matrix equation. Solving this will give us the solutions of both the electrical and the magnetic fields. The proposed algorithm is shown schematically in the next figure.

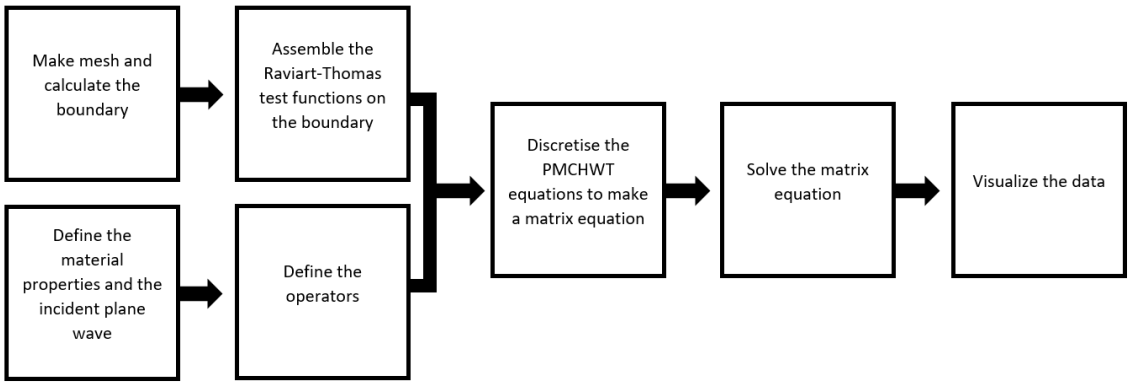


Figure 5: A schematic view of the PMCHWT algorithm

To give a quick overview of the algorithm in words: the two leftmost blocks in Fig. 5 define our problem. We define a mesh and since this method works on the boundary of the mesh, we calculate the boundary of the mesh. It is possible to make the boundary mesh directly. However, we will use the results obtained with this algorithm by comparing them to those obtained with another method. This method requires the use of a tetrahedral mesh. Adding this step assures us that we have the exact same mesh without doing any extra work. The other initial parameters consist of the ϵ ans μ of the object and the space around it. If we want to put our object in a plane wave, we have to add it too.
The second steps are using the initial parameters to prepare for the actual solving of the problem. We do this by assembling the Raviart-Thomas basis functions over each triangle. Information over this basis function can be found in chapter 2.3.1 and the information over the assembly in Appendix C.
The third step consist of evaluating the operators in each edge using Eq. 22. This value is stored in a matrix at the positions corresponding to the positions of the edges. The same is done with the plane wave.
For the last 2 steps in Fig. 5, all that is left is solving the matrix equation. Now using that we approximated the solution with a linear combination of test functions, we can visualize it.

2.5 Finite Elements Method

If we want our cloak to have the desired properties, we will have to make a cloak with a complex varying dielectric. This is hard to do with the Method of Moments, so we will use a different numerical approach. We use the finite elements method (FEM). The biggest advantage of the FEM is that it is able to simulate strongly varying material effects. The FEM makes it possible to simulate local effects, so we can "zoom in" if a certain part of our design needs a more accurate solution. In the next section, we will show the general idea of the FEM. We continue by using the Maxwell equations on this method. We end by giving an algorithm of the FEM.

2.5.1 General Method

In this section a general method of the FEM is shown. To do this, we start with a differential equation:

$$\mathcal{L}u = f$$

Here is f a source function and $\mathcal{L}$ a differential equation. The latter is different from the MoM. Like the MoM, we will now assume that u is a linear combination of test functions v_j.

$$u = \sum_{j=1}^{N} c_j v_j$$

We will continue to follow the procedure of the Method of Moments. We will now plug the linear combination in the differential equation. We integrate the equation over the domain and multiply both sides by test functions w_k. This gives us:

$$\sum_{j=1}^{N} c_j \int_\Omega w_k \mathcal{L}(v_j) d\Omega = \int_\Omega w_k f d\Omega \tag{30}$$

This can be written as a matrix equation in the same way as Eq. 23:

$$\begin{aligned} [A][c] &= [b] \\ A_{kj} &= < w_k, \mathcal{L}(v_j) >_\Omega \\ b_k &= < w_k, f >_\Omega \end{aligned} \tag{31}$$

There are a lot of similarities between the two methods. So what are the differences? First, the test functions are different. In the MoM, the test functions are in 2D, while the test function of the FEM are in 3D. The other differences are due to the different operators $\mathcal{L}$. In the MoM, $\mathcal{L}$ is an integral equation. This allows the use of the Greens function. The Greens function can solve our problem in an infinite domain, but matrix A (Eq. 23) becomes a full matrix. Meanwhile, the differential equation of the Finite Elements Method can only calculate the discretised space. A benefit is that the FEM gives a sparse matrix A (equation 31). There is only a contribution when the basis functions overlap. Sparse matrices can be solved by using sparse matrix solvers. We will now apply this method on the Maxwell equations:

2.5.2 FEM in the curl-curl equation

In this section we apply the theory from above. we start with the curl-curl equation from Eq. 6:

$$\nabla \times \nabla \times E - \kappa^2 E = f$$

Here f is a source term and the material properties are described in $\kappa = \omega\sqrt{\epsilon\mu}$. We now test this equation with the Nedelec test functions, described in chapter 2.3.4. We use the Nedelec functions because they are curl conforming and we need to calculate the curl of these test functions. Testing the curl-curl equation gives us the following:

$$< \tilde{E}, \nabla \times \nabla \times E >_\Omega - \kappa^2 < \tilde{E}, E >_\Omega = < \tilde{E}, f >_\Omega$$

We can now use the divergence theorem to obtain a new more symmetric system. A quick derivation is shown in Appendix B. In the FEM, we will only use $\hat{n} \times E = 0$ for the boundary. This way the boundary term disappears from the divergence theorem:

$$< \nabla \times \tilde{E}, \nabla \times E >_\Omega - \kappa^2 < \tilde{E}, E >_\Omega = < \tilde{E}, f >_\Omega$$

This allows us to solve the interior of an object. We will give the algorithm schematically:

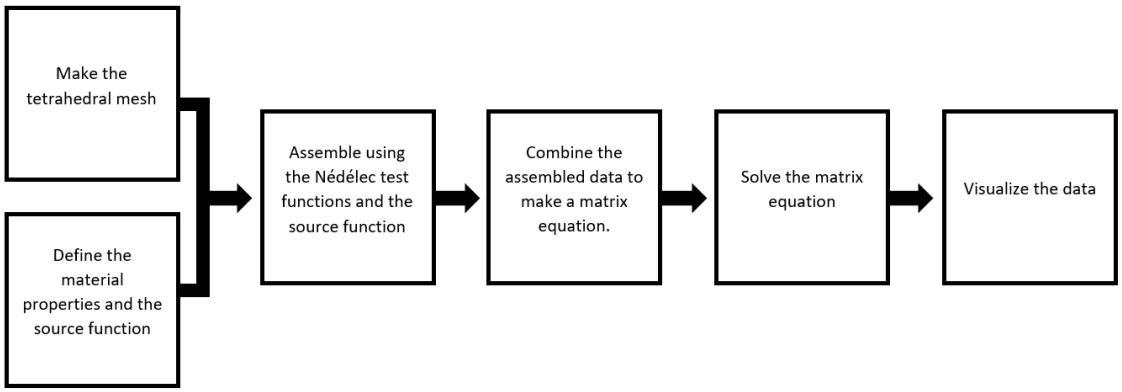


Figure 6: A schematic view of the FEM algorithm

We will now describe the algorithm from Fig. 6. The first step is the initialization of the starting mesh and its parameters. The object is not necessarily homogeneous, so a more difficult dielectric can appear. This is described in κ and is represented as a function instead of a constant.
The second step is to assemble the source function over each edge and to assemble the curl $< \nabla \times \tilde{E}, \nabla \times E >_\Omega$. The used basis functions are Nedelec basis functions. Notice that the $\nabla \times E$ are in fact Raviart-Thomas (3D) basis functions, see chapter 2.3.5. The other assembly procedure is a bit more difficult. This is because the possible more difficult dielectric adds another factor. This is represented as assembling $< \tilde{E}, \kappa^2 E >_\Omega$. The assembly works by storing the contribution of each (pair of) edges in a vector or matrix. The matrices are a sparse

matrices, because almost all contributions are 0. The descriptions of the basis functions can be found in chapter 2.3. More information of the assembly can be found in Appendix C.
The last three steps consist of making the matrix equation described above and solving it. Now all that is left is visualizing the results to control the correctness of the algorithm.

2.6 A hybrid FEM-MoM (FEBI) method

In this section, we combine the Method of Moments with the Finite Element Method. This allows us to see the far-field while still be able to simulate a complex dielectric. We start with MoM like expressions. These expressions show the relation of the electric and magnetic traces between the boundary radiate and the outer domain. We write this as a matrix product. This is described with the Calderon Operator [15]. This gives us the following representation on the outside of the cloak using the operators defined in the MoM:

$$\begin{bmatrix} E \times \hat{n} - E^{inc} \times \hat{n} \\ \hat{n} \times H - \hat{n} \times H^{inc} \end{bmatrix} = \begin{bmatrix} \mathcal{K} + \frac{1}{2} & -\eta\mathcal{T} \\ \frac{1}{\eta}\mathcal{T} & \mathcal{K} + \frac{1}{2} \end{bmatrix} \begin{bmatrix} E \times \hat{n} - E^{inc} \times \hat{n} \\ \hat{n} \times H - \hat{n} \times H^{inc} \end{bmatrix}$$

Solving for $\hat{n} \times H - \hat{n} \times H^{inc}$ in the EFIE gives

$$\hat{n} \times H - \hat{n} \times H^{inc} = [\frac{1}{\eta}\mathcal{T}^{-1}(\mathcal{K} - \frac{1}{2})][E \times \hat{n} - E^{inc} \times \hat{n}]$$

If we substitute this in the MFIE, we get:

$$\begin{aligned} \hat{n} \times H - \hat{n} \times H^{inc} &= [\frac{1}{\eta}\mathcal{T} + \frac{1}{\eta}(\mathcal{K} + \frac{1}{2})\mathcal{T}^{-1}(\mathcal{K} - \frac{1}{2})][E \times \hat{n} - E^{inc} \times \hat{n}] \\ &= \mathcal{S}[E \times \hat{n} - E^{inc} \times \hat{n}] \end{aligned} \tag{32}$$

Here we call $\mathcal{S}$ the Poincare-Steklov (PS) operator. We will use this result later. We will continue with the FEM part of the algorithm. We start with equation 5 and simplify it by making sure there is no source.

$$\nabla \times (\frac{1}{\mu}\nabla \times E) - \omega^2\epsilon E = 0$$

This gets discretized as followed, using that Ω is the whole domain.

$$< \tilde{E}, \nabla \times (\frac{1}{\mu}\nabla \times E) >_\Omega - < \tilde{E}, \omega^2\epsilon E >_\Omega = 0$$

Using the divergence theorem (see Appendix B for the exact formula):

$$\frac{1}{\mu} < \tilde{E}, \hat{n} \times \nabla \times E >_\Gamma + < \nabla \times \tilde{E}, \frac{1}{\mu}\nabla \times E >_\Omega - < \tilde{E}, \omega^2\epsilon E >_\Omega = 0$$

Using $\nabla \times E = -j\omega\mu H$, we get

$$-j\omega < \tilde{E}, \hat{n} \times H >_\Gamma - < \nabla \times \tilde{E}, \frac{1}{\mu}\nabla \times E >_\Omega - < \tilde{E}, \omega^2\epsilon E >_\Omega = 0$$

We can now substitute our earlier work into this equation, giving:

$$
\begin{aligned}
&- j\omega < \tilde{E}, S[E \times \hat{n}] >_\Gamma + < \nabla \times \tilde{E}, \frac{1}{\mu} \nabla \times E >_\Omega - < \tilde{E}, \omega^2 \epsilon E >_\Omega \\
&= -j\omega < \tilde{E}, S[E^{inc} \times \hat{n}] >_\Gamma + j\omega < \tilde{E}, \hat{n} \times H^{inc} >_\Gamma
\end{aligned} \tag{33}
$$

The above formula gives a matrix equation. Solving for $\tilde{E}$ allows us to approximate the real solution with a linear combination. Some problems that may arise are explained in the analysis of the algorithm. This algorithm is shown in the next figure.

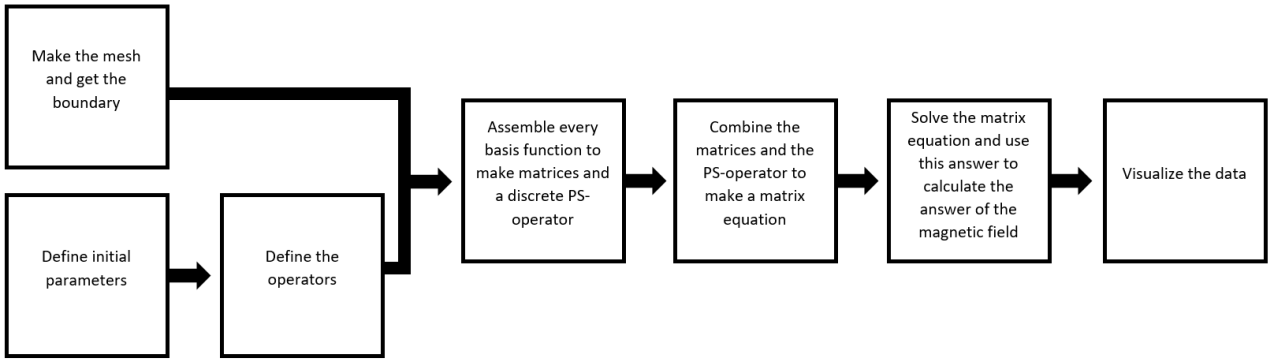


Figure 7: A schematic view of the FEBI algorithm

The first step is simply defining the mesh and the used initial parameters. Like in the FEM, the μ and ϵ of the dielectric can be complicated and are written as a function.
For the second step we define the operators $\mathcal{T}$ and $\mathcal{K}$ as in the MoM.
In the third step we assemble matrices and vectors. $< \tilde{E}, \omega^2 \epsilon E >$ is assembled with Nedelec basis functions. $< \nabla \times \tilde{E}, \frac{1}{\mu} \nabla \times \ E >$ gets assembled with RT_{3D}. This can be seen in the deRham diagram, chapter 2.3.5. The other matrices are made using the Poincare-Steklov operator. They are assembled with the tangential trace of the Nedelec basis functions. This makes sure that it is defined on the boundary. The obtained test functions are RT_{2D}.
In the fourth step we add the matrices into a matrix equation.
This matrix equation is solved in the fifth block. Here, we will also calculate the magnetic field. We do this in two ways: 1) We use the PS operator. Equation 32 gives this relation. 2) We use $\nabla \times E = -j\omega\mu H$ and change test functions. Remember that the answer we get from the matrix equation were the coefficients used in a linear combination of test function of the answer. We will now change the coefficients with a factor $-j\omega\mu$ and make the basis functions RT_{3D}. RT_{3D} was used because this is the basis function that corresponds to the curl of Nedelec basis functions.
At last, we visualize the data.

3 Methodology and Results

3.1 MoM: No contrast sphere (PWCHWT)

In this section we simulate a plane wave meeting a sphere. This sphere will have the same ϵ and μ as its environment. The solution in the sphere should give a plane wave. This makes this case easily verifiable. We do this by using the PMCHWT-equations and the algorithm described in Fig. 5. The source is a plane wave moving in the $+\hat{z}$-direction and is polarized in the $\hat{x}$-direction. The wave number of the plane wave is $\kappa = 5.0$. We plot the tangential traces for the norm of the electric and magnetic fields using the PMCHWT algorithm. These are found in the fig. 8

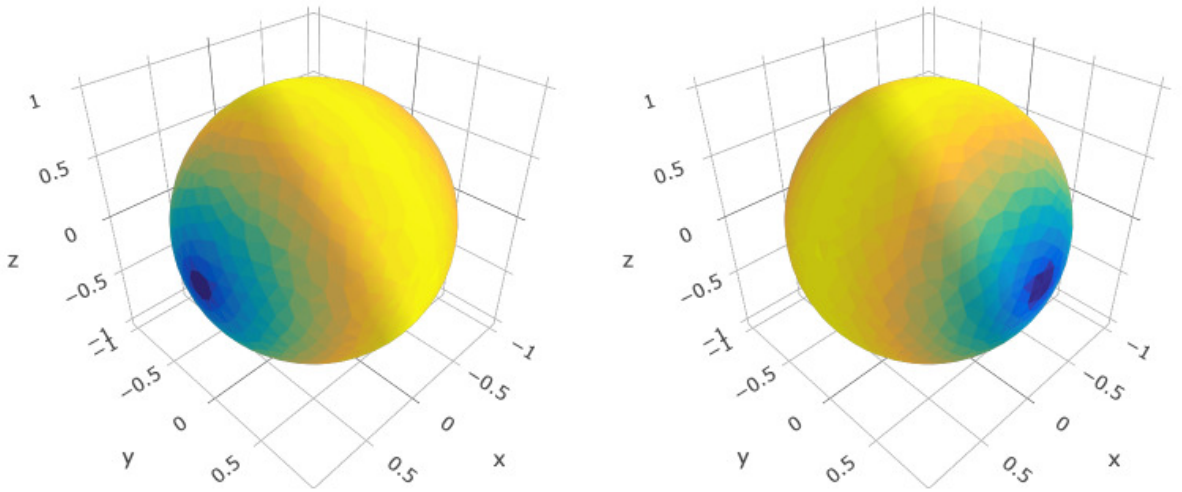
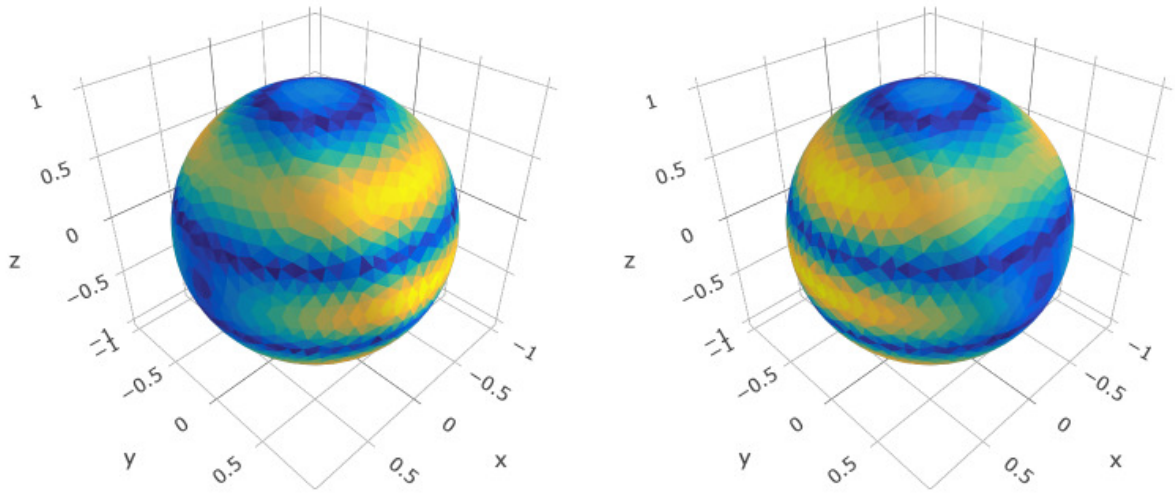


Figure 8: The solution at the surface for the magnitude of the scattered electric (a) and magnetic (b) fields for a sphere with a dielecric that is the same as its surroundings.

and we conclude that the simulations are correct because we get a continuous uniform field. On top of that, the cold spot present in the fig. 8a is in the direction of its polarization. The same can be said about the magnetic field. This solution does not look like a plane wave. That is because the amplitude is plotted in Fig.8. To show the plane wave, the real part of the electric and magnetic traces are plotted in Fig. 9



a *b*

Figure 9: The tangential electric (a) and magnetic (b) fields at the surface for a sphere made out of air. The real part are plotted to visualize the plane wave.

Fig. 9 shows the real part of the electric and magnetic fields. This should and does look like a plane wave. Fig. 9 also shows that the electric solution is polarized in the $\hat{x}$-direction and the magnetic field in the $\hat{y}$-direction. This is in agreement with what we expected. This suggest that the simulations work as intended.
To verify the correctness of other simulation methods, we will compare the results of that algorithm to the PMCHWT. Since Fig. 8 describes a simpler figure then Fig. 9, we will use the former.

3.2 FEM: Cavity in PEC

Stating the problem

In this section, we give a result of our FEM-script. We use the algorithm described in Fig. 6. In this example, we have a sphere made of a Perfect Electric Conductor (PEC). In the middle of this sphere, a source radiates in the $\hat{x}$-direction uniformly. This source is the strongest in the center of the sphere and is given by $f(r) = (1, 0, 0) \cdot e^{-\frac{r^2}{4}}$. The material in the sphere is such that $k^2 = \omega^2 \epsilon \mu = 9.0$. Because the sphere is a PEC, we know on the boundary that $\hat{n} \times E = 0$. This means that we can implement the boundary condition by stripping the boundary-edges from the mesh.

Expectations

We will now look at what the solution should look like. Obviously, because the boundary condition has to be satisfied, there is no tangential trace at the boundary. However, there must be a solution for the curl of the solution. To guess what this will look like, we note that we

expect a line where everything is pointed in the $\hat{x}$-direction. This is shown in Fig. 10. We remark that, due to symmetry, Fig. 10 would look the same if we used y instead of z.

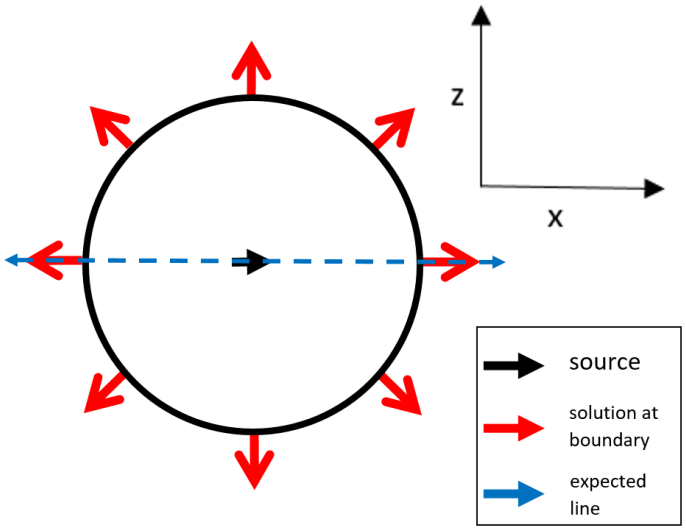


Figure 10: The PEC in the xz-plane. Along the line (x,0,0)', all vectors are expected to be in the x-direction

If we take the curl of the expected line, we get vectors that circle around this line in the yz-plane. Due to the symmetry in the problem, we expect a constant value for each point on the surface of the PEC in the yz-plane. The circles close to (-1,0,0)' and (1,0,0)' are small. Since these circles enclose less of the PEC, the curl is likely small. This means that we have cold spots around (-1,0,0)' and (1,0,0)'.

Results

In this section, the results are given. We start by plotting the expected line. We do did by plotting the values of the solution in the x,y and z direction independently. To view the expected line, we have to plot on it. So we plot them between (-1.5,0,0)' to (1.5,0,0)'. This is done in Fig. 11.

On the next page, in Fig. 11, the solution seems to be mostly continuous. More importantly, the solution is "polarized" in the $\hat{x}$-direction. This is what we expected. This suggests that the simulation is probably correct.

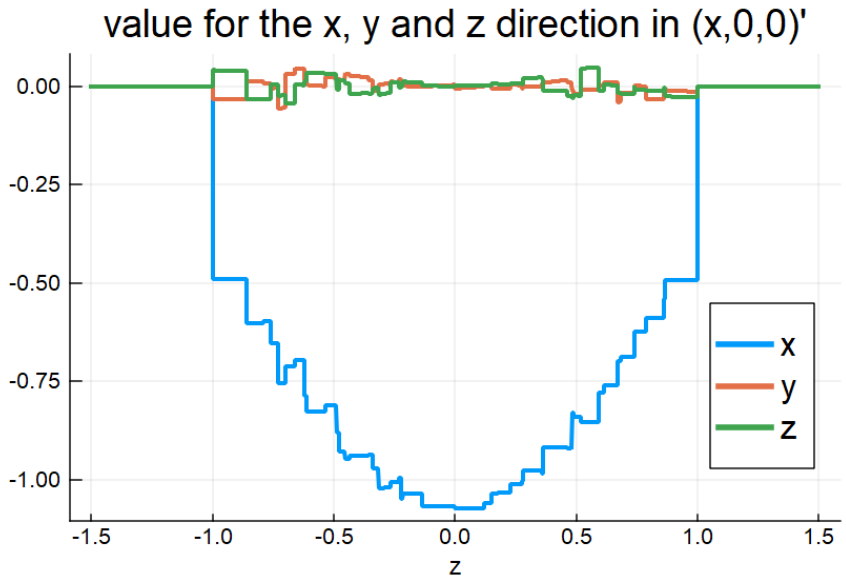


Figure 11: The electric field for a PEC sphere with a source in the $\hat{x}$-direction. The figure shows that the field is continuous and is aligned in the $\hat{x}$-direction.

It is however not very conclusive that we did everything correctly. This is why we try to show the solution on the boundary. However, because of the boundary condition, the tangential trace must be 0 everywhere. This could not readily done by our script. For this reason, the curl of the magnetic field on the surface is plotted in Fig. 12.

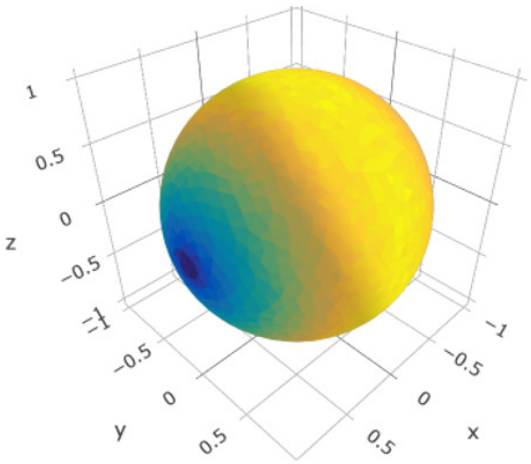
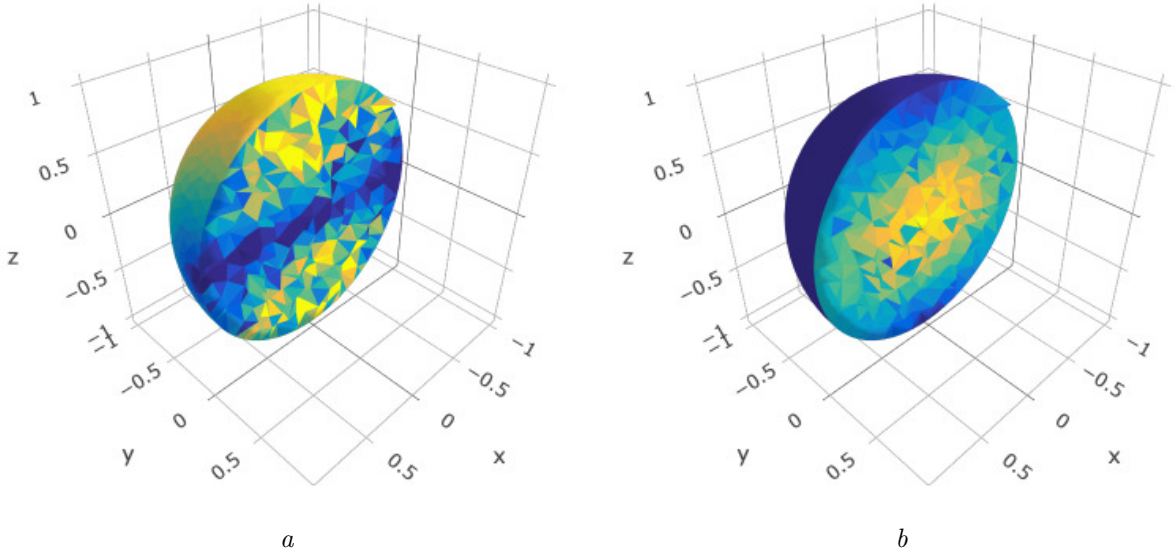


Figure 12: The curl of the electric field for a PEC sphere with a source in the x direction. The figure shows that the solution is continuous and that cold spots appear in the x direction.

Fig. 12 shows yet again that the solution is continuous. We have already seen that the solution

is aligned in the $\hat{x}$-direction. So the curl makes the electric field circle around this axis. Fig. 12 confirms this.
At last the solution on the inside will be shown. We do this by making an intersection along the y-axis. Fig. 13 will show an intersection of Fig. 12. More importantly, this intersection allows us to plot the same figure wthout taking the curl. So the tangential electric field traces are plotted in Fig. 13 too.



a *b*

Figure 13: The curl of the electric field (a) and the electric field (b). (a) is scaled like figure 12. Note that in (b) the traces on the boundary are zero everywhere. Only the real parts of the electric field are plotted.

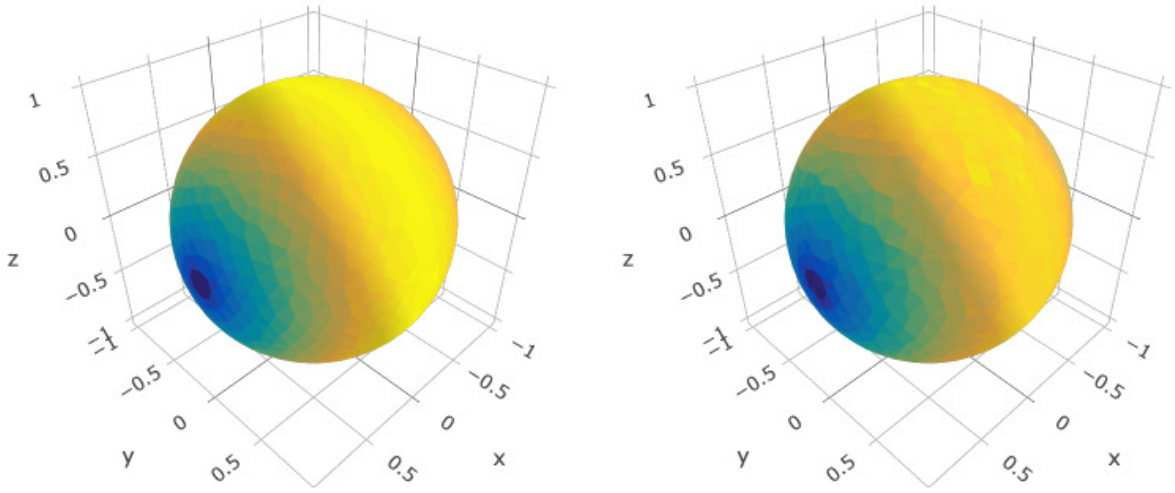
The results in Fig. 13 are as expected. In Fig. 13a, the line along the x-axis is clearly visible. Fig. 13b shows a convincing cavity. More importantly, it shows that there are no tangential traces of the electric field on the boundary. This is exactly the boundary condition $\hat{n} \times E = 0$. These results show that the implementation of the FEM was correct.

3.3 FEBI

In the following chapters, results using the FEBI algorithm will be given. For all of these simulations, we use the algorithm described in Fig. 7. We will start by simulating a sphere with the same properties as its surroundings (transparent). We will compare the results to the PMCHWT solution from chapter 3.1. After this no contrast case, we will simulate a Luneburg lens. The Luneburg lens has more complex material properties while still having well known solutions. This should verify the correctness again. At last, we will simulate an invisibility cloak.

3.3.1 Zero contrast sphere

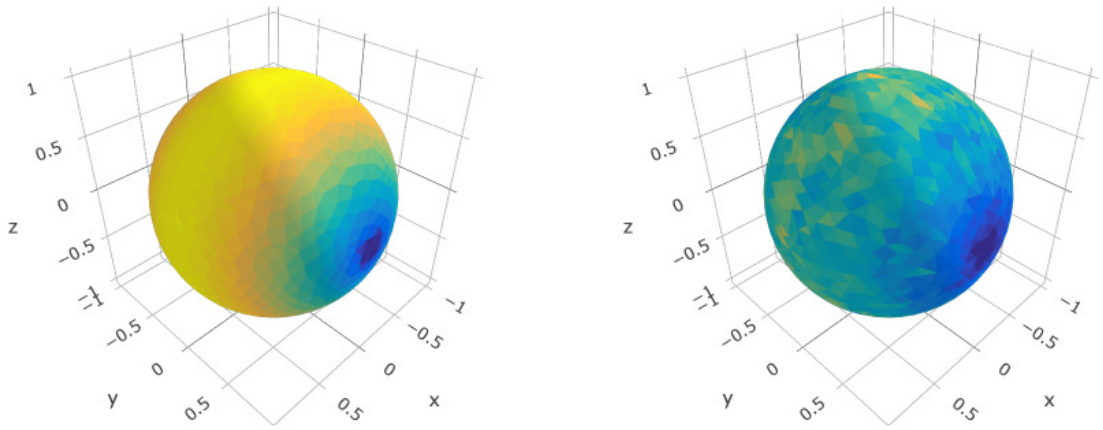
In this section, we look at the FEBI solution for a sphere made out of the same dielectric as its surroundings. The material was such that $\kappa = 5.0$. The source was a plane wave moving towards the $\hat{z}$ direction and was polarized in the $\hat{x}$ direction. This is the same problem as the air sphere from the Method of Moments (chapter 3.1). In Fig. 14 the PMCHWT solution together with the FEBI solution for electric fields will be shown. The FEBI solution is made by showing the tangential traces.



a *b*

Figure 14: The tangential electric traces at the surface using the PMCHWT (a) and the FEBI (b) method for a no contrast sphere. The two figures are almost identical.

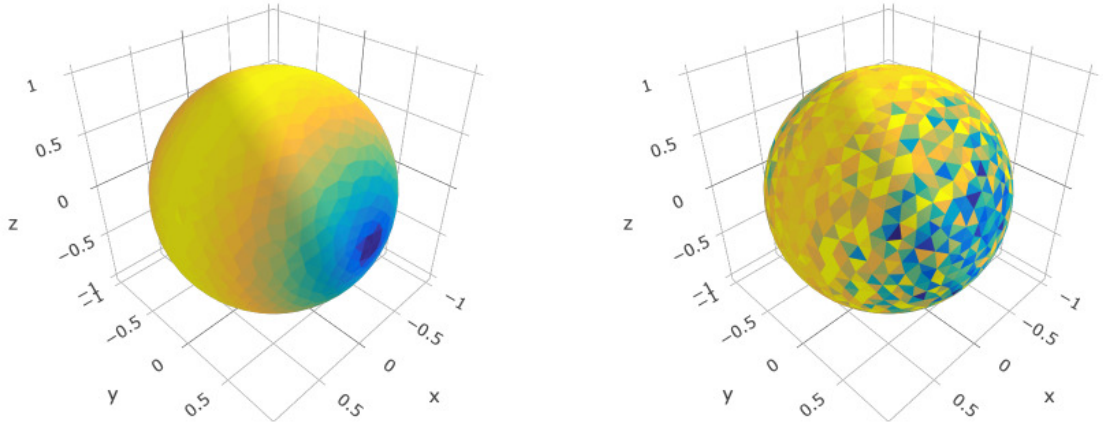
The two subfigures in Fig. 14 are almost identical. This suggest that the FEBI method was correctly implemented. However, the magnetic fields should be correct too. One way to calculate the FEBI solution for magnetic fields is to use the Poincare-Steklov operator, equation (32). Like Fig. 14, we will plot the magnetic tangential traces for both the PMCHWT and the FEBI method. This is done in Fig. 15



a *b*

Figure 15: The tangential magnetic field traces for the PMCHWT (a) and the FEBI (b) method for a sphere with the same properties as its environments.

The result of Fig. 15 is not completely correct. While it shows a correct aligning and a barely visible band around the y-axis, the mosaic-like structure and lower values in general makes it look completely different. We try another way to calculate the magnetic fields. The figures were made by having a tuple with both a prefactor and its space. In the electric FEBI solution, this was a RT_{2D} (the tangential trace of the ND_{3D}) space. Using Eq. 1, we can modify the prefactor and change the space into ND_{2D} (the tangential trace of the curl of ND_{3D}). The tangential magnetic traces are plotted using the PMCHWT and the above described method in Fig. 16.



a *b*

Figure 16: The solution at the surface of the magnetic fields for the PMCHWT (a) and the FEBI (b) method for a no contrast sphere. The higher values in (b) are capped by the highest value in (a).

The solution in Fig. 16 is not right either. While Fig. 15 values were too low, fig. 16 are too high. The mosaic structure is worse or is at least more obvious. The cold spots in Fig. 16 were reasonably plotted, but it is a mess in Fig. 15. However, the yellow band around the y-axis is now clearly visible, while it was not in Fig. 16. The lesser results could be due to the relatively weak computer I used in this project. It is possible that better/good result can be obtained with a better PC. When we made these figures, the maximum and minimum value on the boundary were printed. The FEBI solution of the electric fields showed good agreement with the PMCHWT solution. Both were approximately between 0.0 - 1.0. The PMCHWT magnetic field solution was between 0.0 - 1.0 too. This was not the case in the FEBI solution. Using the curl solution from Fig. 16 was between 0.0 - 1.3. However, if we use the Poincare-Steklov method from Fig. 15 the boundary was between 0.0 - 0.9.
At last, we show the solution on the inside of the sphere. We make an intersection along the yz-plane. Fig. 17 shows the electric field in the $\hat{x}$-direction on this plane.

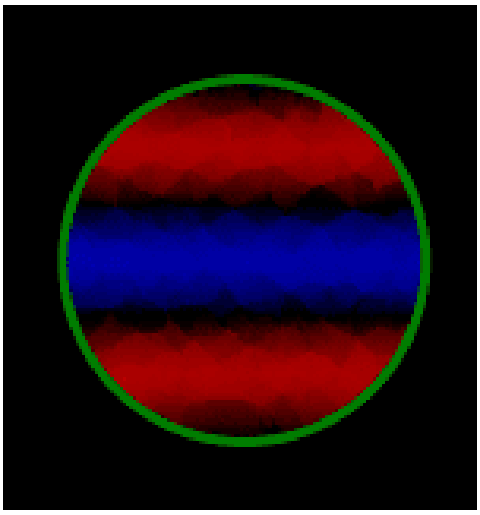


Figure 17: The $\hat{x}$-direction of the electric field on the inside of the sphere. The solution seems to be a plane wave.

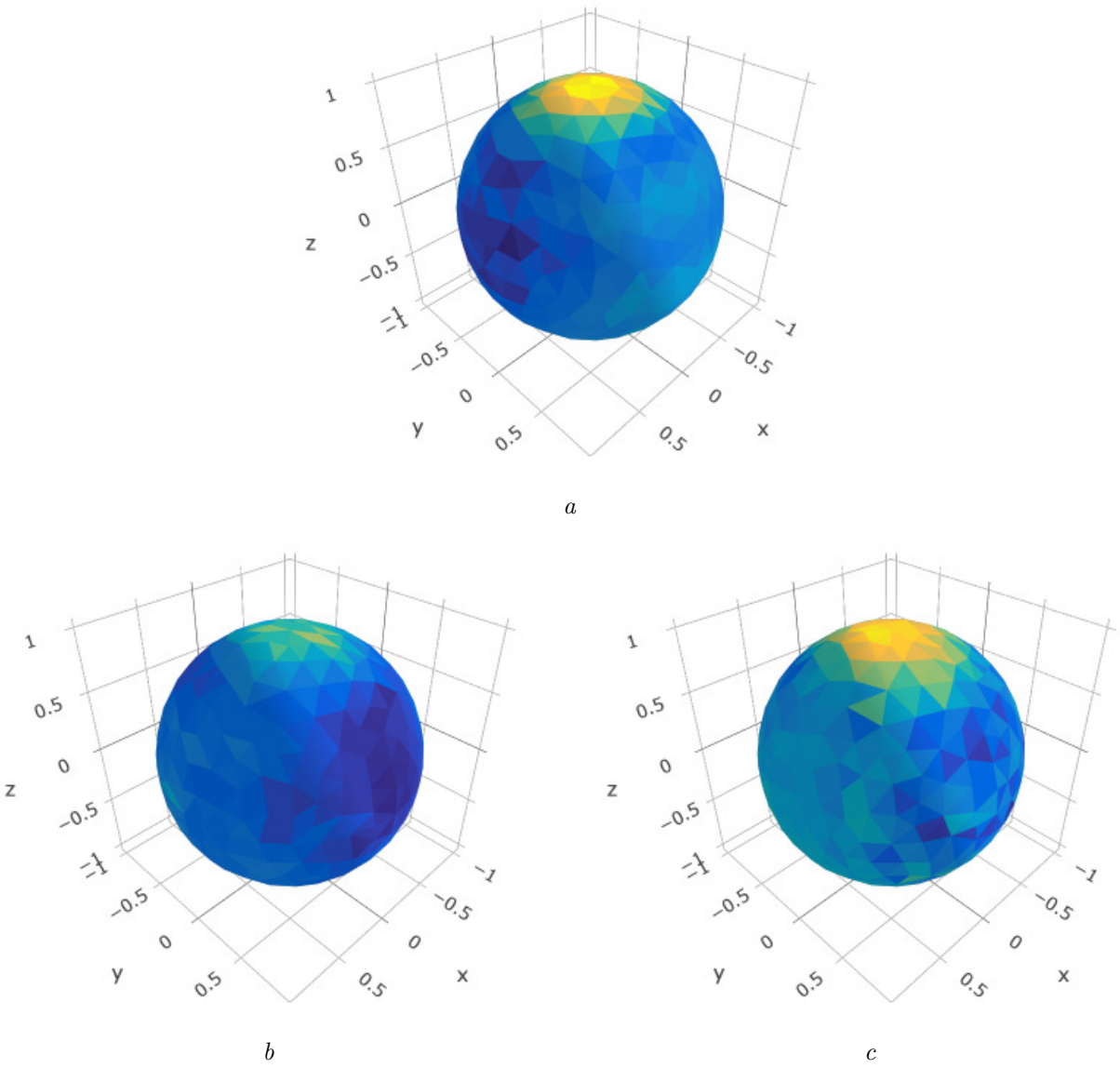
The solution on the inside looks good. A plane wave can be seen in Fig. 17. This is what we wanted.

3.3.2 Luneburg lens

In this section we give another example of the FEBI-code. One of the reasons why we made the hybrid algorithm was that we wanted simulate a more complex inner volume. We show this by simulating a luneburg lens [16]. A luneburg lens is a sphere that focus a plane wave at the opposite point on the sphere by changing the material properties radially. In this example we use the following formula for the lens:

$$\kappa = \omega\sqrt{\epsilon\mu} = \omega\sqrt{2 - (\frac{r}{R})^2} \tag{34}$$

In formula 34 is R the radius of the sphere. In our example we use a sphere with radius 1 and use a plane wave in the $+\hat{z}$ direction with a polarization in the $\hat{x}$ direction. The plane wave has a wavelength of $\kappa = 10.0$. The electric and magnetic traces on the boundary are shown in the following figure.



a

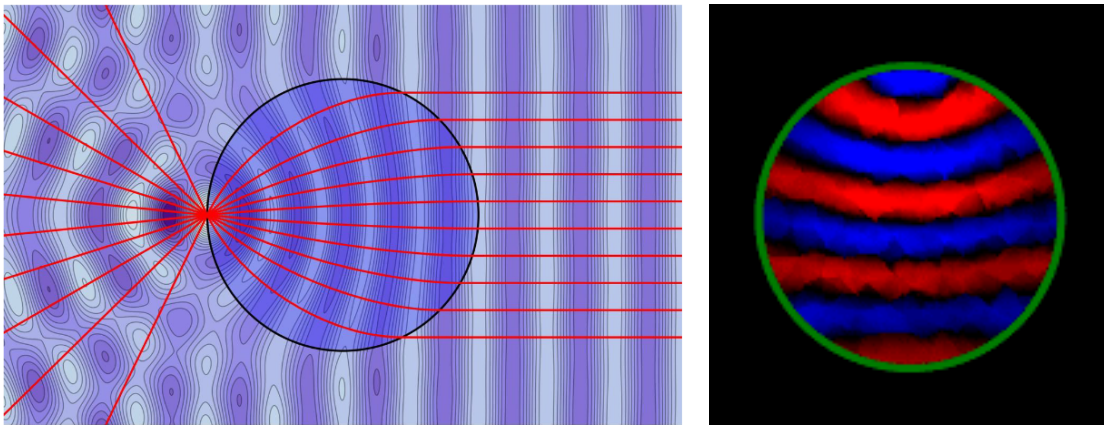
b

c

Figure 18: The solution for the luneburg lens on the surface for the electric fields (a) and the magnetic fields (b),(c). (b) was made using the Poincare operator while (c) was made using a the curl representation.

The results in figure 18 look decent. There is only a hot spot around (0,0,1)', which was were

the plane wave should be focused. The two ways to make magnetic fields are shown using the same scale as the electric fields. And just as in the no contrast sphere (chapter 3.3.1), figure 18b's maximum value is lower than the maximum in the electric field solution. However, figure 18c's maximum value is the same as in the electric field solution.
To see if we can confirm the correctness of the results, we look at the solution on the inside of the sphere. In Fig. 19, we show a reference figure from diFalco [16] and compare it to the electric solution in the $\hat{x}$-direction on an intersection along the yz-plane.



a *b*

Figure 19: A reference solution for the Luneburg lens from diFalco [16] (a) and the $\hat{x}$-direction of the electric field on the inside for the FEBI-code of the Luneburg lens (b). The plotted area is an intersection along the yz-plane. The two figures are in agreement.

As we can see in figure 19, the solution seems to be correct. The waves seem to bend around (0,0,1)' and this was where the focus point was designed to be. If we compare Fig. 19 to work done by diFalco, [16] we notice that the two results are in agreement. In both images, the waves bend towards the focus point.

3.3.3 Cloak

In this section, a simulation of an invisibility cloak device is made using the FEBI-code. An example of a design is given by Shurig [9]. We try to get the results of the simulation part of this paper in our own Julia-code. The values for ϵ and μ and the transformation to cartesian coordinates can be found in chapter 2.2.2. In this chapter and Schurig ([9]), the ϵ and μ are tensors. This means that we cannot use the κ^2 method, but we have to use the FEBI with the split ϵ and μ. The design is in cylinder-coordinates, so we use a cylinder mesh in this simulation. In this mesh, the cylinder-axis is along the x-axis. In this report, we do not put a hole in the cylinder for the cloaked area. Instead, we use a solid cylinder and change the material properties in this area to see if any change happen. This should not be the case. We put the mesh in a plane wave moving into the z-direction with the electric component being polarized into the

x-direction. First, we use a plane wave with $\kappa = 7$ and the cloak has a cloaked radius of $R_2 = 0.325$. These simulations are made for different materials in the cloaked area. We compare the solutions of these different materials. If the cloaking is successful enough, there should not be a big difference between the solutions. Fig. 20 shows the $\hat{x}$-direction of electric field of an intersection that shows the yz-plane.

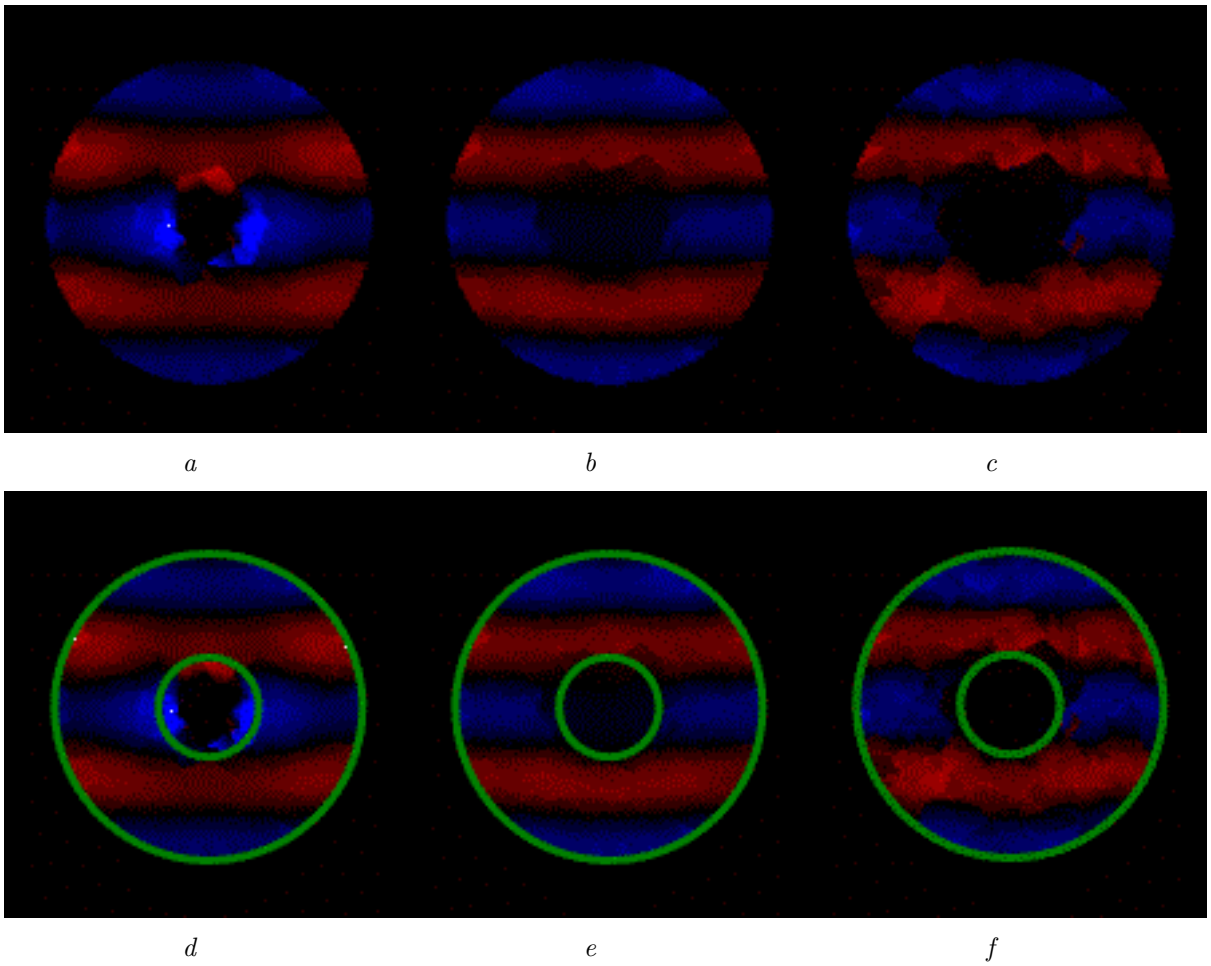
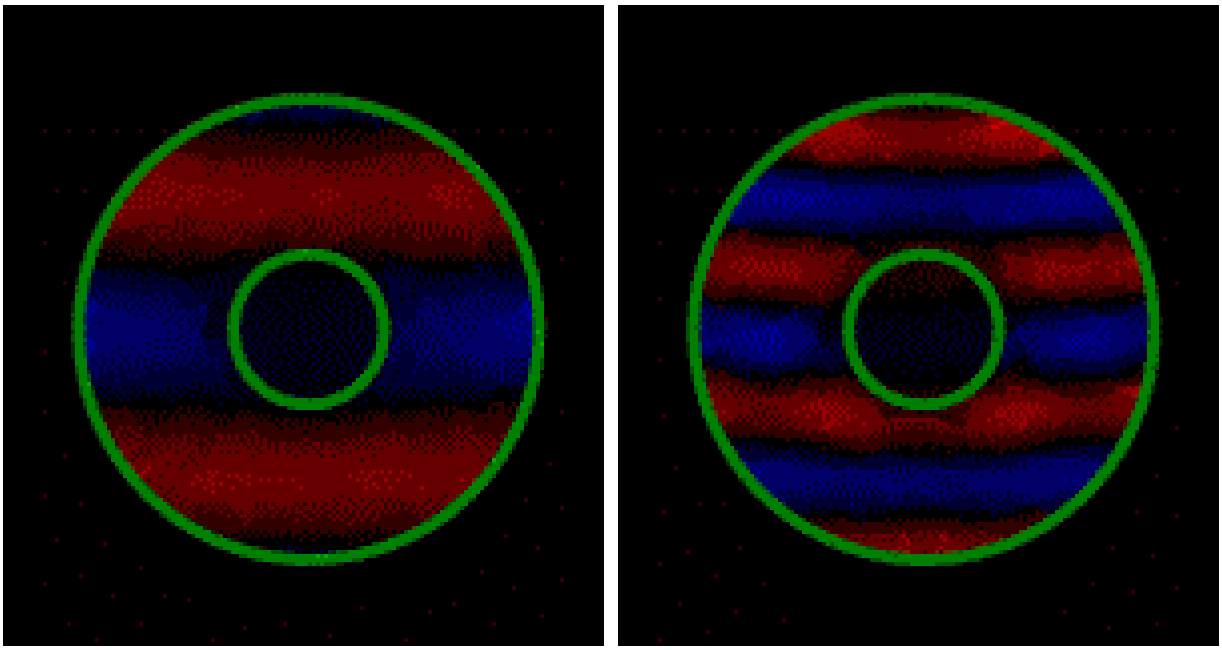


Figure 20: The electric field in the $\hat{x}-$direction in the yz-plane for the inside of the cloak. The bottom row shows an outline of the cloak. (a) and (d) have ϵ and μ equal to 0, (b) and (e) equal to 10 and the other equal to 100.

The results in figure 20 look promising. All subfigures show cloaking properties with an electric field of 0 within the cloaked area. Moreover, it leaves the cloak as a plane wave. Some small differences can be notices between the different inner areas. If ϵ and μ are 0, the cloaked area is not completely 0. The reason for this might be the discretization. In this script, the permittivity and permeability are calculated in the center of each tetrahedron. If this center is inside the cloak, while some parts are in the cloaked area, a little bit of the wave will be simulated there.

In the other figures, this does not seem to be much of a problem. Since these results look good, we will pick one value for the inner part and compare the results when we change some other parameters. First, we will change the wavelength of the plane wave.



a *b*

Figure 21: The electric field of the cloak under a plane wave with $\kappa = 5$ (a) and with $\kappa = 10$ (b)

As seen in figure 21, the cloak still works. We find no surprises when we use a different wavelength. The wave still splits around the inner area and leaves the cloak as a plane wave.

4 Discussion and Conclusion

In this work, the derivation and simulation of a cloak device were made. All the simulations were made in Julia [1], [2]. To do this, the Method of Moments was verified in the Julia-package BEAST.jl [3]. The 3D Raviart-Thomas and curl-conforming Nedelec test functions and the curl and traces of these functions were added to BEAST.jl. Tests for these funtions were written and added too. To use these functions in simulations, an assembly-routine for these functions was added. 3D meshes were created and could be read by BEAST.jl and CompScienceMeshes.jl [4]. In the Finite Elements Method, the algorithm can assemble using sparse matrices.
To verify the correctness of the implementations, some simulations were made. The PMCHWT (Method of Moments) and FEM algorithm agreed with our expectations. The hybrid or FEBI code was tested using an zero-contrast sphere, a Luneburg lens and an invisibility cloak. All of these algorithms can be found in EMwavesBEP.jl [5].
The main simulation was the simulation of an invisibility cloak. The results of this simulation were promising. It showed a wave that never reached the inner area and continued as a normal plane wave. However, no contributions of this wave in either the far or the near field were shown. An analysis like this should be made to confirm that the cloak is indistinguishable from its surroundings.
The results for the electric field were good. This could be seen in the simulation of the no contrast sphere that was almost exactly the same as the simulation using the PMCHWT equations. However, the magnetic fields were not perfect. In this work, 2 methods to calculate the magnetic field were used, but both of these showed a mosaic-like structure somewhere on the boundary of the solution. To make it worse, the extrema on this boundary were either too high or too low. However, if we decide to look at it less precisely, the general structure of the PMCHWT solution can be seen in the FEBI solution. This means that it was not completely wrong. A possible reason is that the computer that was used to make the simulations was not that great. This means that with a better PC, a finer mesh that approaches the solution better could be made. Another mistake could be that the magnetic field were calculated per triangle. This local approximation means that the total solution is not necessarily continuous.
This leads to two obvious further research. First, visualize the cloak solution in the far field to determine the correctness of the simulation. Secondly, compute the magnetic field correctly by using a finer mesh or by calculating the magnetic field in a different manner. This has lead to other work, see the "Publications from this Thesis" section. More research can be done by simulating a cloak spherically or by using a different material properties (using a different transformation). Object with other acoustic properties, like bending a wave in a block, can be simulated. The material properties can be determined with the same method we used to get the properties for the cloak.

References

[1] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://doi.org/10.1137/141000671.

[2] The Julia Project, https://julialang.org.

[3] K. Cools, *Beast*, https://github.com/krcools/BEAST.jl, 2020.

[4] K. Cools, *Compsciencemeshes*, https://github.com/krcools/CompScienceMeshes.jl, 2020.

[5] R. Voskamp, *Emwavesbep*, https://github.com/RodVoskamp/EMwavesBEP, 2020.

[6] V. M. Shalaev, "Transforming light," en, vol. 322, p. 4, 2008.

[7] M. Koshiba, K. Hayata, and M. Suzuki, "Finite-element formulation in terms of the electric-field vector for electromagnetic waveguide problems," *IEEE Trans. Microwave Theory Techn.*, vol. 33, no. 10, pp. 900–905, 1985.

[8] A. Nicolet, F. Zolla, Y. Ould Agha, and S. Guenneau, "Geometrical transformations and equivalent materials in computational electromagnetism," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 27, no. 4, S. Wiak, Ed., pp. 806–819, 2008.

[9] D. Schurig, J. J. Mock, B. J. Justice, S. A. Cummer, J. B. Pendry, A. F. Starr, and D. R. Smith, "Metamaterial Electromagnetic Cloak at Microwave Frequencies," *Science*, vol. 314, pp. 977–980, 2006.

[10] K.R. Hiremath, L. Zschiedrich, and F. Schmidt, "Numerical solution of nonlocal hydrodynamic drude model for arbitrary shaped nano-plasmonic structures using nedelec finite elements," *Journal of Computational Physics*, vol. 231, no. 17, pp. 5890–5896, 2012.

[11] J. E. Pasciak and P. S. Vassilevski, "Exact de Rham Sequences of Spaces Defined on Macro-Elements in Two and Three Spatial Dimensions," *SIAM Journal on Scientific Computing*, vol. 30, no. 5, pp. 2427–2446, 2008.

[12] R. Hiptmair and S. Mao, "Stable multilevel splittings of boundary edge element spaces," en, *BIT Numerical Mathematics*, vol. 52, no. 3, pp. 661–685, 2012.

[13] M. Gossye, D. van de Ginste, and H. Rogier, "Electromagnetic modeling of high magnetic contrast media using calderón preconditioning," *Computers Mathematics with Applications*, vol. 77, no. 6, pp. 1626–1638, 2019.

[14] A. Poggio and E. Miller, "Integral Equation Solutions of Three-dimensional Scattering Problems," in *Computer Techniques for Electromagnetics*, 1973.

[15] L. F. Knockaert and D. De Zutter, "ON THE COMPLEX SYMMETRY OF THE POINCAR'E-STEKLOV OPERATOR," *Progress In Electromagnetics Research B*, vol. 7, pp. 145–157, 2008.

[16] A. Di Falco, S. C. Kehr, and U. Leonhardt, "Luneburg lens in silicon photonics," *Optics Express*, vol. 19, 2011.

[17] Gihub, *Atom*, `https://atom.io`.

[18] C. Geuzaine and J.-F. Remacle, "Gmsh: A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities.," *International Journal for Numerical Methods in Engineering*, pp. 1309–1331, 2009.

[19] C. Geuzaine and J.-F. Remacle, *Gmsh*, `https://gmsh.info/`.

[20] I. Anjam and J. Valdman, "Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements," *Applied Mathematics and Computation*, vol. 267, pp. 252–263, arXiv: 1409.4618.

[21] A. Schneebeli, "An H(curl;Ω)-conforming FEM: Nédélec's elements of first type," 2003. [Online]. Available: `https://www.dealii.org/reports/nedelec/nedelec.pdf`.

Appendix A: Used tools

In this project, some knowledge of the following programs has to been learned.

Julia

The simulations are all done in Julia, [1], [2]. The simulations can be found at [5] and its supporting packages at [3], [4]. Julia is a free programming language, which was first released in 2012. Julia is an interesting language, because it has a good performance while still having familiar features from other programming languages. It has an interface like C++, add packages like Python and can evaluate a vector point-wise in functions like MATLAB. An other nice feature is the use of symbols in the code. For example α, Γ and $\cdot$ can be used as variables or functions.

Atom

Atom is a free code editor that can be used to write, for example, in Julia. Atom was made by GitHub and that's why it is not surprising that Atom can be used well with GitHub.Atom is open source and can be used for many different programming languages. Atom is available for most platforms [17].

GitHub

This project was the first time I used GitHub. GitHub is a hosting service for source code. GitHub allowed me to make my own version of the already existing Julia-package BEAST and made it possible to add my own code. This code can be pushed back to the original version. In this project, a branch was made so a stable version of the previous work could be preserved. The code and its supporting packages can be found on Github [5], [3], [4].

Gmsh

Like the other software on this page, Gmsh [18], [19] is free. Gmsh is a mesh generator written in C++. Both 2D and 3D meshes can be made. A mesh can be made by code. This code can be used to generate a mesh in a programming language like Julia. It is also possible to make a mesh in a 3D raster, which will generate the code from the previous description.

Appendix B: Curl curl relation

In this section, some mathematical statements are given. We can derive the following vector properties:

$$(A \times B) \cdot C = -(C \times B) \cdot A \tag{35}$$

$$A \cdot (\nabla \times B) = B \cdot (\nabla \times A) - \nabla \cdot (A \times B) \tag{36}$$

These can be easily proven by just plugging in vectors. We will also state the divergence theorem:

$$\int_{\Omega} (\nabla \cdot A) d\Omega = \int_{\Gamma} (A \cdot \hat{n}) d\Gamma \tag{37}$$

This, together with the described vector properties allows us to prove the following:

$$< A, \nabla \times \nabla \times B >_{\Omega} = < \nabla \times A, \nabla \times B >_{\Omega} + < A, \hat{n} \times \nabla \times B >_{\Gamma} \tag{38}$$

A quick derivation is shown here:

$$< A, \nabla \times \nabla \times B >_{\Omega} =$$

$$< A, \nabla \times B2 >_{\Omega} =$$

$$< \nabla \times A, B2 >_{\Omega} - \int_{\Omega} \nabla \cdot (A \times B2) d\Omega =$$

$$< \nabla \times A, B2 >_{\Omega} - \int_{\Gamma} (A \times B2) \cdot \hat{n} d\Gamma =$$

$$< \nabla \times A, B2 >_{\Omega} + \int_{\Gamma} (\hat{n} \times B2) \cdot A d\Gamma =$$

$$< \nabla \times A, \nabla \times B >_{\Omega} + \int_{\Gamma} (\hat{n} \times \nabla \times B) \cdot A d\Gamma =$$

$$< \nabla \times A, \nabla \times B >_{\Omega} + < A, \hat{n} \times \nabla \times B >_{\Gamma}$$

Appendix C: Assembly

Raviart-Thomas

In this section, the assembly of the RT test function is described. A description of these basis functions can be found in chapter 2.3.1 We use the Piola mapping. In 2D, this means changing a triangle into a triangle with edges at (0,0), (1,0), (0,1). In 3D, we can transform a tetrahedron into the unit tetrahedron. This tetrahedron exists between the points (0,0,0), (1,0,0), (0,1,0), (0,0,1). The Piola mapping looks as followed:

$$F_k(x) = B_k(x) + b_k \qquad \forall x \in K$$

such that

$$F_k(K) = \hat{K}$$

Where B_k is a 3x3 matrix, b_k a vector. K is the original tetrahedron and $\hat{K}$ the unit tetrahedron. In our code, we have to assign a numbering system over the edges. To make a triangle/tetrahedron, we use the simplex(a,b(,c),o) command. This returns a triangle with vertices at a,b,(c) and the origin (o). The numbering of the basis functions on the edges is based on the order of the vertices. The first basis function is on the edge/face that does not contain vertice a. The second function is on the edge that does not contain vertice b. The same holds for vertices c and o.
Because RT is divergence-conforming, we want the normal component on an edge to be 1 over this edge and 0 over the others:

$$\int_{e_i} \hat{n}_i \cdot b_j^{ref} ds = \delta_{ij}$$

Here b_J^{ref} is the RT function in the reference triangle/tetrahedron. Solving the above condition give us the reference RT functions. An explicit form is given in [20]. To get the RT functions and their divergence in a general triangle/tetrahedron, we have to map the reference solution back to the original triangle/tetrahedron by ([21]):

$$b_j(x) = det(B_k) B_k b_j^{ref}(F_k^{-1}(x))$$

$$\nabla \cdot b_j(x) = \frac{1}{det(B_k)} \nabla \cdot b_j^{ref}(F_k^{-1}(x))$$

Nedelec

A summary of the Nedelec basis functions can be found in chapter 2.3.3. The assembly of the Nedelec basis functions work just like the Raviart-Thomas. Like the RT case, we use a Piola-transform. We want that the tangential component of an edge is one 1 on this edge and is 0 over the other edges, because Nedelec is curl conforming. This gives

$$\int_{e_i} t_i \cdot b_j^{ref} ds = \delta_{ij}$$

These reference basis functions have to be mapped back to the original tetrahedron (see [21]):

$$b_j(x) = B_k^{-T} b_j^{ref}(F_k^{-1}(x))$$

In 2D, the curl is mapped as followed according to [21]

$$\nabla \times b_j(x) = \frac{1}{det(B_k)} \nabla \times b_j^{ref}(F_k^{-1}(x))$$

In 3D, some more work is required. The previous described numbering system does not work here, so we use the following numbering scheme in our Julia-implementation: If we start with simplex(1,2,3,4), we use the vertices that, once the two numbers are combined, give the lowest number. So the first test function is defined on the edge that goes from vertice 1 to vertice 2 (12). The second one goes from vertice 1 to vertice 3 (13). Notice that the edge from vertice 2 to vertice 2 (21) does not count, since this is the same as the first test function in reverse. The numbering can be seen in the following figure:

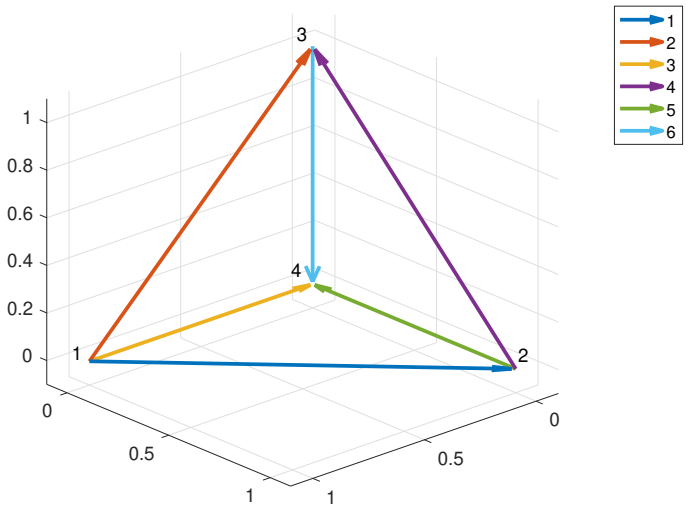


Figure 22: The order of appearance in the unit tetrahedron in the vertices (inserted numbering) and the order of the edges (in legend).

The second change is that the curl is mapped differently [20], [21]:

$$\nabla \times b_j(x) = \frac{1}{det(B_k)} B_k \nabla \times b_j^{ref}(F_k^{-1}(x))$$