



## July 24, 2020

### Leibniz Universität Hannover

Institut für Angewandte Mathematik

BACHELOR THESIS

# Geometric Multigrid Methods for Maxwell's Equations



Author: Julian ROTH (10012482) Advisor: Prof. Dr. Thomas WICK

# Abstract

Maxwell's equations have many applications in modern physics, but are inherently difficult to solve with standard methods. Therefore I will discuss a multilevel finite element method for Maxwell's equations in this work, since it is known that these methods achieve a great performance for elliptic partial differential equations. I will consider a geometric multigrid method for the Eddy Current problem, which is a special case of the time harmonic Maxwell's equations. Furthermore I show that using a multiplicative Schwarz smoother yields optimal cost complexity.

# Zusammenfassung

Die Maxwell-Gleichungen haben viele Anwendungen in der modernen Physik, sind jedoch grundsätzlich schwer mit den Standardverfahren zu lösen. Deswegen werde ich mich in dieser Arbeit mit einem Mehrgitterverfahren für die Maxwell-Gleichungen befassen, da diese Methoden eine gute Performance für elliptische partielle Differentialgleichungen liefern. Ich werde das Geometrische Mehrgitterverfahren auf das Eddy Current Problem anwenden, welches ein Sonderfall der zeitharmonischen Maxwell-Gleichungen ist. Zudem zeige ich, dass die Verwendung der mulitplikativen Schwarz Methode zu einer optimalen Kostenkomplexität führt.

# Contents

1.	Intr	oduction	1		
2.	. Mathematical Tools				
	2.1.	Domains	3		
	2.2.	Curl Operator	3		
	2.3.	Variational differential operators	4		
	2.4.	Function spaces	5		
		2.4.1. Continuous function spaces	5		
		2.4.2. Discretized function spaces	6		
	2.5.	Green's theorem	6		
	2.6.	Schwarz smoother	7		
3.	Tim	e-Harmonic Maxwell's equations	10		
	3.1.	Problem description	10		
	3.2.	De Rham complex	12		
	3.3.	Nédélec elements	13		
4.	. Multigrid methods				
	4.1.	Motivation	14		
	4.2.	Problem setup	15		
	4.3.	Two-grid algorithm	16		
	4.4.	Multigrid algorithm	17		
	4.5.	Grid transfer	19		
	4.6.	Cost complexity	20		
	4.7.	Convergence	22		
	4.8.	Application: Poisson equation	23		
	4.9.	Application: Maxwell's equations	26		

5.	Related research	27
	5.1. Chen [7]	27
	5.2. Zhong et al. $[25]$	29
	5.3. Arnold, Falk, Winther [2]	30
	5.4. Janssen and Kanschat [13]	31
6.	Numerical tests	32
	6.1. Eddy Current	32
	6.2. Time-Harmonic Maxwell	36
7.	Conclusion and Outlook	40
Α.	Convergence order	41
	A.1. Closed formula	41
	A.2. Linear regression	42

# 1. Introduction

Maxwell's equations are a set of partial differential equations (PDE) named after the 19th century physicist James Clerk Maxwell. His equations form the basis of electromagnetism and optics. Therefore these partial differential equations are a crucial part in the analysis and development of generators, motors and wireless communication. Moreover, in the field of optics Maxwell's equations describe complex optical components, like laser resonators or optical multiplexers [24]. Hence, it is of great interest for the scientific computing community to develop algorithms that can solve Maxwell's equations, since this enables the simulation of new electromagnetical and optical components.

In this work I will analyze a multilevel finite element method (FEM) scheme for solving the time-independent, i.e. time-harmonic, Maxwell's equations. In chapter 2, I will recapitulate the mathematical formulations of electromagnetism and introduce the multiplicative Schwarz smoother, a domain decomposition method. The problem description of the Time-Harmonic Maxwell's equations and the discretization of the corresponding function space with the Nédélec elements is then being discussed in chapter 3. In chapter 4, I present the theory of the geometric multigrid method (GMG), derive the cost complexity of this algorithm and point out how its convergence could be shown. Additionally, it is shown how this multilevel scheme can be applied to the Poisson equation and first numerical results are given. Before diving into my numerical experiments, in chapter 5 I summarize related research in which the multigrid method has been applied to various formulations of the Maxwell's equations. In the next chapter (chapter 6), the performance of my multigrid routine is tested on the Eddy Current problem and the Time-Harmonic Maxwell problem. Finally, I draw my conclusions in chapter 7 and give several ideas for future developments. In the appendix, two methods for the computation of the convergence order are introduced.

The code accompanying this thesis is available at https://github.com/mathmerizing/MultigridMaxwell.

# 2. Mathematical Tools

#### 2.1. Domains

Let us consider an open domain  $\Omega \subseteq \mathbb{R}^d$  with Lipschitz boundary  $\partial\Omega$ , where  $d \in \{2,3\}$  is the dimension. By **n** I denote the outer unit normal vector with respect to  $\partial\Omega$ .

## 2.2. Curl Operator

Let  $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ , which should always be identifiable from the context of its usage. Firstly, one needs to define the **cross product** of vectors  $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{K}^3$ :

$$oldsymbol{u} imes oldsymbol{v} = egin{pmatrix} u_1 \ u_2 \ u_3 \end{pmatrix} imes egin{pmatrix} v_1 \ v_2 \ v_3 \end{pmatrix} \coloneqq egin{pmatrix} u_2 v_3 - u_3 v_2 \ u_3 v_1 - u_1 v_3 \ u_1 v_2 - u_2 v_1 \end{pmatrix}.$$

A cross product can also be defined for vectors  $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{K}^2$ :

$$\boldsymbol{u} \times \boldsymbol{v} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} := u_1 v_2 - u_2 v_1.$$

The cross product is skew symmetric, i.e.  $\boldsymbol{u} \times \boldsymbol{v} = -\boldsymbol{v} \times \boldsymbol{u}$ . Furthermore,  $\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c}$  for  $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^3$ . With the help of this operator, one can define the **curl** of a vector  $\boldsymbol{u} \in \mathbb{K}^d$ :

$$\operatorname{curl} \boldsymbol{u} := \nabla \times \boldsymbol{u} = \begin{pmatrix} \partial_y u_3 - \partial_z u_2 \\ \partial_z u_1 - \partial_x u_3 \\ \partial_x u_2 - \partial_y u_1 \end{pmatrix} \quad \text{for } d = 3,$$

 $\operatorname{curl} \boldsymbol{u} := \nabla \times \boldsymbol{u} = \partial_x u_2 - \partial_y u_1 \quad \text{for } d = 2.$ 

### 2.3. Variational differential operators

In the previous section, the strong form of the curl operator was introduced. In a similar fashion, the strong forms of the gradient  $(\nabla f)$  and of the divergence  $(\operatorname{div} f := \nabla \cdot f)$  can be defined, as they are discussed in introductory calculus courses. However, in the following I will introduce the variational form of the before mentioned operators. They are a weaker version of their strong form, in the sense that they only require a smaller regularity of the function. If the strong form of the variational differential operator exists, both forms coincide.

Let  $C_c^{\infty}(\Omega)$  denote the space of infinitely differentiable functions with compact support. Let's call  $\boldsymbol{w} = \nabla u \in [L^2(\Omega)]^d$  the **variational gradient** of  $u \in L^2(\Omega)$ , if

$$\int_{\Omega} \boldsymbol{w} \cdot \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x} = -\int_{\Omega} u \operatorname{div} \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x} \quad \forall \boldsymbol{\phi} \in \left[C_c^{\infty}(\Omega)\right]^d.$$

Let's call  $w = \operatorname{div} \boldsymbol{u} \in L^2(\Omega)$  the variational divergence of  $\boldsymbol{u} \in [L^2(\Omega)]^d$ , if

$$\int_{\Omega} w\phi \, \mathrm{d}\boldsymbol{x} = -\int_{\Omega} \boldsymbol{u} \cdot \nabla \phi \, \mathrm{d}\boldsymbol{x} \quad \forall \phi \in C^{\infty}_{c}(\Omega).$$

Let's call  $\boldsymbol{w} = \operatorname{curl} \boldsymbol{u} \in [L^2(\Omega)]^{m_d}$  the variational curl of  $\boldsymbol{u} \in [L^2(\Omega)]^d$ , if

$$\int_{\Omega} \boldsymbol{w} \cdot \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega} \boldsymbol{u} \cdot \mathrm{curl} \, \boldsymbol{\phi} \, \mathrm{d}\boldsymbol{x} \quad \forall \boldsymbol{\phi} \in \left[C_c^{\infty}(\Omega)\right]^d$$

# 2.4. Function spaces

#### 2.4.1. Continuous function spaces

An important role in the analysis of PDEs is being played by the Sobolev spaces

$$\begin{split} H^{1}(\Omega) &:= \left\{ f \in L^{2}(\Omega) \mid \nabla f \in \left[ L^{2}(\Omega) \right]^{d} \right\}, \\ H^{1}_{0}(\Omega) &:= \left\{ f \in H^{1}(\Omega) \mid f_{\mid_{\partial\Omega}} \equiv 0 \right\}, \\ H^{\text{div}} &:= \left\{ \boldsymbol{f} \in \left[ L^{2}(\Omega) \right]^{d} \mid \text{div} \, \boldsymbol{f} \in L^{2}(\Omega) \right\}, \\ H^{\text{curl}} &:= \left\{ \boldsymbol{f} \in \left[ L^{2}(\Omega) \right]^{d} \mid \text{curl} \, \boldsymbol{f} \in \left[ L^{2}(\Omega) \right]^{m_{d}} \right\} \end{split}$$

where

$$m_d = \begin{cases} 1 & \text{for } d = 2\\ 3 & \text{for } d = 3 \end{cases}$$

and

$$L^{p}(\Omega) := \left\{ f: \Omega \to \mathbb{K} \mid f \text{ measurable}, \|f\|_{L^{p}(\Omega)} := \left( \int_{\Omega} |f(x)|^{p} \, \mathrm{d}x \right)^{\frac{1}{p}} \right\}.$$

Another important role is being played by inner products of function spaces. For this work in particular the  $L^2$  inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} 
angle \coloneqq \langle \boldsymbol{f}, \boldsymbol{g} 
angle_\Omega \coloneqq \int\limits_\Omega \boldsymbol{f} \cdot ar{oldsymbol{g}} \; \mathrm{d} oldsymbol{x}$$

is being used.

#### 2.4.2. Discretized function spaces

There is a vast variety of finite element spaces, but here I only want to mention the discretized function spaces from the De-Rham complex:

- $Q^{k+1}$ : continuous piecewise polynomials of degree k+1 [6, p. 64]
- $\mathcal{N}^k$ : Nédélec elements of degree k [18]
- $\mathcal{RT}^k$ : Raviart-Thomas elements of degree k [6, p. 142]
- $\mathcal{DGQ}^k$ : discontinuous piecewise polynomials of degree k [10, p. 191]

In the next chapters, I will mainly inspect the Nédélec elements, which are a cornerstone of the discretization of Maxwell's equations. For an in-depth analysis of the other finite element space, please refer to [2].

#### 2.5. Green's theorem

For  $\boldsymbol{f}, \boldsymbol{g} \in H^{\operatorname{curl}}(\Omega)$ , it holds that

$$\langle \operatorname{curl} oldsymbol{f}, oldsymbol{g} 
angle = \langle oldsymbol{f}, \operatorname{curl} oldsymbol{g} 
angle + \langle oldsymbol{n} imes oldsymbol{f}, oldsymbol{g} 
angle_{\partial\Omega}$$
 .

*Proof:* Integrating the identity  $\nabla \cdot (\boldsymbol{f} \times \boldsymbol{g}) = \operatorname{curl} \boldsymbol{f} \cdot \boldsymbol{g} - \boldsymbol{f} \cdot \operatorname{curl} \boldsymbol{g}$  over the domain  $\Omega$  yields

$$\int_{\Omega} \nabla \cdot (\boldsymbol{f} \times \boldsymbol{g}) \, \mathrm{d}x = \langle \operatorname{curl} \boldsymbol{f}, \boldsymbol{g} \rangle - \langle \boldsymbol{f}, \operatorname{curl} \boldsymbol{g} \rangle.$$

Now the left hand side can be rewritten with the help of Gauss' divergence theorem and the properties of the cross product

$$\int_{\Omega} \nabla \cdot (\boldsymbol{f} \times \boldsymbol{g}) \, \mathrm{d}x = \int_{\partial \Omega} (\boldsymbol{f} \times \boldsymbol{g}) \cdot \boldsymbol{n} \, \mathrm{d}s = \int_{\partial \Omega} (\boldsymbol{n} \times \boldsymbol{f}) \cdot \boldsymbol{g} \, \mathrm{d}s =: \langle \boldsymbol{n} \times \boldsymbol{f}, \boldsymbol{g} \rangle_{\partial \Omega}.$$

This is the integration by parts formula in  $H^{\text{curl}}(\Omega)$ , which will later be used to derive a variational form of the Eddy Current problem.

## 2.6. Schwarz smoother

To fully understand how the additive and the multiplicative Schwarz method work, it is beneficial to analyze how this method can be derived from the alternating Schwarz method, which has been developed by H.A. Schwarz in 1870 [21, 15].



Figure 2.1.: Overlapping domains

H.A. Schwarz was trying to answer the question how one can solve a PDE on two overlapping domains. As the name of the method suggests, this can be done in an iterative way, where one takes turn solving on the different domains. To comprehend this approach, let us take the Poisson problem as the model problem.

Example 2.6.1Alternating Schwarz for Poisson [4] Let  $\Omega := \Omega_1 \cup \Omega_2$ . Find  $u : \Omega \to \mathbb{R}$  such that  $-\Delta u = f \quad \text{in } \Omega_1 \cup \Omega_2,$ u = 0 on  $\partial \Omega$ . We then get the iterative scheme: # 1. Compute  $u^{(1,0)}$  as solution of  $-\Delta u = f$  in  $\Omega_1$ u=0 in  $\partial\Omega_1$ for k in range(1, N+1): # 2. Compute  $\boldsymbol{u}^{(2,k)}$  as solution of  $-\Delta u = f \qquad \qquad \text{in} \quad \Omega_2$  $u = u^{(1,k-1)} \quad \text{on} \quad \partial \Omega_2 \cap \Omega_1$ u = 0on  $\partial\Omega_2\cap\partial\Omega$ 10 # 3. Compute  $\boldsymbol{u}^{(1,k)}$  as solution of  $-\Delta u = f$ in  $\Omega_1$ 12  $u=u^{(2,k)}$ on  $\partial\Omega_1\cap\Omega_2$ 13 u = 0on  $\partial \Omega_1 \cap \partial \Omega$ 14

How can this idea be applied to Maxwell's equations? Let a decomposition of the function space  $V_l$  into subspaces  $(V_{l,k})_{k=1}^{n_l}$  be given, i.e.  $\sum_{k=1}^{n_l} V_{l,k} = V_l$ . This sum doesn't necessarily need to be direct. If it were, then it would be referred to as a non-overlapping Schwarz smoother. However for my problem I am using the overlapping Schwarz smoother [2]. Here the function spaces  $V_{l,k}$  are associated with the interior DoFs of a cell patch  $\Omega_{l,k}$  around a certain common vertex. Let  $R_{l,k}$  denote the restriction from  $V_l$  to  $V_{l,k}$ ,  $P_{l,k}$  denote the prolongation from  $V_{l,k}$ to  $V_l$  and  $A_{l,k} := R_{l,k}A_lP_{l,k}$  be the projection of the matrix  $A_l$  into the function space  $V_{l,k}$ . Then one step of the symmetric multiplicative Schwarz smoother with right hand side  $d_l$  and last iterate  $x^{i-1}$  reads:

Hence, we get that the error propagation matrix is given by

$$B := \left[ (I - P_1)(I - P_2) \cdots (I - P_{n_l}) \right] \left[ (I - P_{n_l})(I - P_{n_{l-1}}) \cdots (I - P_1) \right],$$

where  $P_k := P_{l,k}A_{l,k}^{-1}R_{l,k}A_l$ . This kind of smoother has been used in my thesis, since it yielded optimal convergence rates in the work of Janssen and Kanschat [13]. However, a big downside of this smoother is that it has to be evaluated sequentially. Thus often the simpler error propagation matrix  $\tilde{B} := I - (P_1 + P_2 + \cdots + P_{n_l})$ from the additive Schwarz smoother is being used, since the computations can be executed in parallel.

# 3. Time-Harmonic Maxwell's equations

This chapter gives a brief overview of the theory behind the Maxwell's equations. A detailed analysis of this topic is being presented in [16].

## 3.1. Problem description

I am dealing with the Time-Harmonic Maxwell's equations where the time dependence can be expressed by  $e^{ikt}$ .

DEFINITION 3.1.1Strong formFind  $\boldsymbol{u} \in H^{\operatorname{curl}}(\Omega)$  such thatin  $\Omega$ (1) $\operatorname{curl}\operatorname{curl}\boldsymbol{u} - \omega^2\boldsymbol{u} = \boldsymbol{f}$ in  $\Omega$ (1) $\boldsymbol{n} \times \boldsymbol{u} = \boldsymbol{0}$ on  $\partial\Omega$ .(2)

To be able to use the finite element method, one needs to multiply equation (1) with a test function  $\boldsymbol{v} \in X := \left\{ \boldsymbol{v} \in H^{\operatorname{curl}}(\Omega) \mid \boldsymbol{n} \times \boldsymbol{v}_{\mid_{\partial \Omega}} = \boldsymbol{0} \right\}$ . Applying Green's theorem, one then gets

$$\langle \operatorname{curl} \boldsymbol{u}, \operatorname{curl} \boldsymbol{v} \rangle + \langle \boldsymbol{n} \times (\operatorname{curl} \boldsymbol{u}), \boldsymbol{v} \rangle_{\partial \Omega} - \omega^2 \langle \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{f}, \boldsymbol{v} \rangle.$$

The boundary integral now vanishes, due to the properties of the test space and the properties of the cross product:

$$\langle \boldsymbol{n} \times (\operatorname{curl} \boldsymbol{u}), \boldsymbol{v} \rangle_{\partial \Omega} = - \langle \operatorname{curl} \boldsymbol{u}, \underbrace{\boldsymbol{n} \times \boldsymbol{v}}_{= 0} \rangle_{\partial \Omega} = 0.$$

Hence, the weak form is given by

DEFINITION 3.1.2 Variational form

Find  $\boldsymbol{u} \in X$  such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in X$$

where

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &:= \langle \operatorname{curl} \boldsymbol{u}, \operatorname{curl} \boldsymbol{v} \rangle - \omega^2 \langle \boldsymbol{u}, \boldsymbol{v} \rangle \,, \\ l(\boldsymbol{v}) &:= \langle \boldsymbol{f}, \boldsymbol{v} \rangle \,. \end{aligned}$$

It can be shown that the bilinear form is bounded and coercive [24]. With a sufficient regularity of f, the Lax-Milgram theorem then yields the existence and uniqueness of the solution of the variational formulation. Through the separation of the real and imaginery part of u and v, the variational form can be transformed into a linear equation system, which can be solved by a FEM approach.

DEFINITION 3.1.3 Linear equation system Find  $U = \begin{pmatrix} U_{\text{Re}} & U_{\text{Im}} \end{pmatrix}^T \in \mathbb{R}^{2n}$  such that  $\begin{pmatrix} A - \omega^2 M & 0 \\ 0 & A - \omega^2 M \end{pmatrix} \begin{pmatrix} U_{\text{Re}} \\ U_{\text{Im}} \end{pmatrix} = \begin{pmatrix} F_{\text{Re}} \\ F_{\text{Im}} \end{pmatrix}$ with  $A = (\langle \operatorname{curl} \boldsymbol{\phi}_i, \operatorname{curl} \boldsymbol{\phi}_j \rangle)_{i,j=1}^n,$   $M = (\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle)_{i,j=1}^n,$  $F_{\text{Re}} = (\langle \Re(\boldsymbol{f}), \boldsymbol{\phi}_j \rangle)_{j=1}^n,$ 

$$F_{\mathrm{Im}} = \left( \langle \Im(\boldsymbol{f}), \boldsymbol{\phi}_{\boldsymbol{j}} \rangle \right)_{j=1}^{n}$$

and  $\{\phi_i\}_{i=1}^n$  being a basis of  $\mathcal{N}^k$ .

In the following, I will work with the case where the wave frequency is 1, i.e.  $\omega = 1$ . Then I am dealing with the same problem statement as [13].

## 3.2. De Rham complex

In this section, I will restrict my analysis to d = 3, but the case d = 2 can be dealt with in a similar way (see [24]). The **De Rham complex** 

$$\mathbb{R} \xrightarrow{\mathrm{id}} H^1(\Omega) \xrightarrow{\nabla} H^{\mathrm{curl}}(\Omega) \xrightarrow{\mathrm{curl}} H^{\mathrm{div}}(\Omega) \xrightarrow{\mathrm{div}} L^2(\Omega) \xrightarrow{0} \{0\}$$

is an exact sequence of continuous function spaces, i.e. the range of each operator equals the kernel of the next operator. I now want to discretize these spaces in such a way that the exactness property is being preserved, since only then one has a conforming finite element method. The most widely known family of finite element spaces with this property is depicted below:

Since for Maxwell's equations I am dealing with subspaces of  $H^{\text{curl}}$ , I will now analyze their discretized finite element space, the Nédélec elements.

## 3.3. Nédélec elements

This family of finite elements is named after the French mathematician J.C. Nédélec, who was the first to find a discrete, conforming subspace of  $H^{\text{curl}}$  in the year 1980 [18]. The Nédélec elements of k-th order are defined as

$$\mathcal{N}^{k} := \left\{ \boldsymbol{v} \in X \mid \boldsymbol{v}_{\mid K}(\boldsymbol{x}) = \boldsymbol{a}_{\boldsymbol{K}}(\boldsymbol{x}) + \left(\boldsymbol{x} \times \boldsymbol{b}_{\boldsymbol{K}}(\boldsymbol{x})\right), \boldsymbol{a}_{\boldsymbol{K}}, \boldsymbol{b}_{\boldsymbol{K}} \in \left[P^{k}(K)\right]^{d} \forall K \in \mathbb{T} \right\}.$$

Sometimes these finite elements are also referred to as "Nédélec's edge elements" [12], since the degrees of the finite elements are located on the edges of the cells in the triangulation. This can be seen exceptionally well, when visualizing the DoFs of a  $\mathcal{N}^0$ -element on a tetrahedron:



Figure 3.1.: Degrees of freedom for  $\mathcal{N}^0$ 

In contrast to the  $Q^1$ -elements which are continuous between the neighboring elements, the Nédélec elements are only continuous in the tangential component of the edges. This is represented in the figure above by the blue arrows along the edges.

# 4. Multigrid methods

Multigrid methods are very successful, since they have proven to be very efficient for many real world problems. Thus there is also a lot of literature on this topic. The curious reader can find more detailed descriptions of the multigrid method in [6, 4].

#### 4.1. Motivation

There is a big array of numerical algorithms which can be deployed to solve a system of linear equations. Direct solvers, like Gaussian elimination, are a family of such algorithms, which can be quickly implemented. However, they are only suitable for relatively small problems, since they don't scale very well for large problems. However in practice mathematicians are dealing with large problems, where direct solvers are not feasible anymore and incomplete matrix factorizations are not robust enough. Hence iterative Krylov solvers like GMRES are employed. A crucial observation is that classical iterative methods damp out the large eigenvalues of the matrix. Building on this observation, in the 1960s Fedorenko and later on in 1985 Hackbusch [11] independently invented the multigrid method, which uses this smoothing property to quickly eliminate big and small eigenvalues of the system matrix, by solving the PDE on a hierarchy of grids. They could then prove that this method has a cost complexity of  $O(n_{\text{DoFs}})$  and thus has an optimal complexity on a classical computer, since each entry of the solution vector needs to be accessed at least once during computation. In the following, I will present the geometric multigrid for uniformly refined meshes. This concept can also be extended to adaptively refined meshes, as shown in [13].

## 4.2. Problem setup

Given some function space V and a bounded domain  $\Omega \subset \mathbb{R}^d$ , I am trying to find a solution  $\boldsymbol{u} \in V$  to the variational formulation

$$a(\boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V$$

where  $a: V \times V \to \mathbb{K}$  is a bilinear form and the linear form  $f: V \to \mathbb{K}$  is some right hand side.

Let a triangulation  $\mathbb{T}_0$  of the domain  $\Omega$  be given with a discrete function space  $V_0$ . The discrete variational form

$$a(\boldsymbol{u_0}, \boldsymbol{v_0}) = f(\boldsymbol{v_0})$$

can be rewritten as a linear equation system

$$A_0 \boldsymbol{x_0} = \boldsymbol{b_0}$$
 with  $A_0 \in \mathbb{R}^{n_0 \times n_0}, \ \boldsymbol{x_0}, \boldsymbol{b_0} \in \mathbb{R}^{n_0}$ .

Through a series of global refinements, a sequence of finer grids  $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_L$  can be created with the corresponding linear equation system formulations

 $A_l \boldsymbol{x_l} = \boldsymbol{b_l} \quad \text{with} \quad A_l \in \mathbb{R}^{n_l \times n_l}, \ \boldsymbol{x_l}, \boldsymbol{b_l} \in \mathbb{R}^{n_l}$ 

for  $1 \leq l \leq L$ . Furthermore, it is advisbale to work with conforming finite element spaces  $(V_l)_{l=0}^L$ , i.e.  $V_0 \subset V_1 \subset \cdots \subset V_L$ , since otherwise the grid transfer operations need to be modified.



Table 4.1.: Level grids

### 4.3. Two-grid algorithm

To understand the multigrid algorithm I start by looking at the case where we only have two grids  $\mathbb{T}_l$  and  $\mathbb{T}_{l+1}$ . The multigrid algorithm is then only a recursive application of the two grid version.

DEFINITION 4.3.1 Two-grid algorithm

Let  $A_h \boldsymbol{x_h} = \boldsymbol{b_h}$  and  $A_{2h} \boldsymbol{x_{2h}} = \boldsymbol{b_{2h}}$  with  $A_h \in \mathbb{R}^{n \times n}$ ,  $A_{2h} \in \mathbb{R}^{m \times m}$  and m < n denote the linear equation systems from the grids  $\mathbb{T}_{l+1}$  and  $\mathbb{T}_l$ . Let the k-th iterate  $\boldsymbol{x_h^k}$  on the finer grid be given.

1 def TGM  $(x_h^k)$ : 2 # 1. Apply  $\nu_1$  smoothing steps of an iterative method  $S_1$ . 3  $x_h^{k,1} = S_1^{\nu_1} x_h^k$  # PRE - SMOOTHING 4 5 # 2. Restrict defect to coarse grid. 6  $d_{2h} = I_h^{2h}(b_h - A_h x_h^{k,1})$  #  $I_h^{2h}$  := restriction operator 7 8 # 3. Coarse grid correction. 9  $x_h^{k,2} = x_h^{k,1} + I_{2h}^h(A_{2h}^{-1}d_{2h})$  #  $I_{2h}^h$  := prolongation operator 10 11 # 4. Apply  $\nu_2$  smoothing steps of an iterative method  $S_2$ . 12  $x_h^{k,3} = S_2^{\nu_2} x_h^{k,2}$  # POST - SMOOTHING 13 14 return  $x_h^{k+1} := x_h^{k,3}$ 

*Remark:* In most cases one wants the two-grid method to be a symmetric iteration. Therefore one needs  $\nu := \nu_1 = \nu_2$  and  $\mathbf{S} := \mathbf{S_1} = \mathbf{S_2}^*$  [7], e.g. choose  $\mathbf{S_1}$  as forward Gauss-Seidel and  $\mathbf{S_2}$  as backward Gauss-Seidel. Furthermore  $A_{2h}^{-1}\mathbf{d_{2h}}$  is not feasible to compute with a direct solver if  $A_{2h}$  is too large, which is often the case. Thus  $A_{2h}^{-1}\mathbf{d_{2h}}$  can be understood as solving the linear equation system and can be done for example by another two-grid method. This recursion then produces the multigrid algorithm.

## 4.4. Multigrid algorithm

DEFINITION 4.4.1 Multigrid algorithm

Let  $A_L x_L = b_L$  denote the problem on the finest grid and  $A_l x_l = b_l$  the problems on the coarser grids for  $0 \le l \le L - 1$ . Let  $\nu$  denote the number of pre- and post-smoothing steps. Let the k-th iterate  $x_l^k$  on the l-th level be given.

```
1 def MGM(l, x_l^k, b_l):
    # 1. Apply 
u smoothing steps of an iterative method S.
3
     x_l^{k,1}=S^
u x_l^k
                  # PRE - SMOOTHING
     # 2. Restrict defect to coarse grid.
    d_{l-1} = I_l^{l-1}(b_l - A_l x_l^{k,1}) # I_l^{l-1} := restriction operator
     # 3. Coarse grid solution.
9
     if 1 == 1:
     y_0 = A_0^{-1} d_0
                    # direct solver on coarsest grid
10
11
     else: # 1 > 1
12
     y_{l-1} = 0
      for i in range(\mu):
13
14
         y_{l-1} = MGM(l-1, y_{l-1}, d_{l-1})
     # 4. Coarse grid correction.
16
    m{x_l^{k,2}} = m{x_l^{k,1}} + I_{l-1}^l m{y_{l-1}} # I_{l-1}^l := prolongation operator
17
18
     # 5. Apply 
u smoothing steps of an iterative method S.
19
    x_l^{k,3} = S^
u x_l^{k,2} # POST - SMOOTHING
20
21
    \texttt{return} \ x_l^{k+1} \coloneqq x_l^{k,3}
```

*Remark:* The parameter  $\mu \in \mathbb{N}^+$  determines the cycle of the multigrid iteration. For  $\mu = 1$  we get the V-cycle





and for  $\mu = 2$  we get the W-cycle.



Figure 4.2.: W cycle

By slightly changing the algorithm we can also get the full multigrid scheme.



Figure 4.3.: Full Multigrid (F cycle)

The full multigrid scheme can be found in [4]. In the figures of these schemes, white circles stand for  $\nu$  steps of an iterative solver, black circles represent a direct solver, blue arrows illustrate a restriction and green arrows illustrate a prolongation.

To fully comprehend the geometric multigrid, I now need to clarify how the grid transfer operations, i.e. the restriction  $I_l^{l-1}$  and the prolongation  $I_{l-1}^l$ , work.

## 4.5. Grid transfer

As we have seen in the previous sections, the multigrid algorithm requires the ability to prolongate vectors from  $\mathbb{R}^{n_{l-1}}$  to  $\mathbb{R}^{n_l}$ . I will only show how the grid transfer operations work for conforming finite elements. For information on how to deal with non-conforming finite element spaces, please refer to [6]. Let  $\{\varphi_1^l, \ldots, \varphi_{n_l}^l\}$ and  $\{\varphi_1^{l-1}, \ldots, \varphi_{n_{l-1}}^{l-1}\}$  be some given bases of  $V^l$  and  $V^{l-1}$ . Due to the conformity of the finite element spaces,  $V^{l-1} \subset V^l$  holds and there exists a matrix  $I_l^{l-1} \in \mathbb{R}^{n_{l-1} \times n_l}$  such that

$$\begin{pmatrix} \varphi_1^{l-1} \\ \vdots \\ \varphi_{n_{l-1}}^{l-1} \end{pmatrix} = I_l^{l-1} \begin{pmatrix} \varphi_1^l \\ \vdots \\ \varphi_{n_l}^l \end{pmatrix}.$$

The matrix  $I_l^{l-1}$  is called the restriction matrix and its transpose  $I_{l-1}^l = (I_l^{l-1})^T$  is called the prolongation matrix. These matrices are dependent on the finite elements that are being used and on the way that the grids have been refined. They have only very few non-zero entries and thus are stored as sparse matrices. Furthermore, they also have a significant impact on the rate of convergence of the multigrid algorithm [6]. Additionally, for linear partial differential equations the identity

$$A_{l-1} = I_{l-1}^{l} A_{l} I_{l}^{l-1}$$

is fulfilled.



# 4.6. Cost complexity

This proof follows closely [4][Section 2.1.6]. It still remains to be shown that a multigrid iteration can be performed in  $O(n_L)$  operations.

Let  $h_l = \frac{1}{2}h_{l-1}$  be the mesh discretization parameter on the level l. Thus  $n_{l-1} \leq \frac{1}{2^d}n_l$  holds, where d is the spatial dimension of  $\Omega$ . Let  $W_l$  denote the cost of 1 iteration of the l-grid method. Looking back at the multigrid algorithm, one can see that for an upper bound to the cost  $W_l$  one needs:

- the cost for per-/ post-smoothing which is  $c_1 n_l$  per smoothing iteration
- the cost  $c_2 n_l$  to calculate the defect
- the cost  $c_3n_l$  for a grid transfer operation, i.e. restriction or prolongation
- the cost for the addition of the coarse grid solution  $n_l$
- the cost for the coarse grid solution which is  $c_4 n_0$  for l = 1and  $\mu W_{l-1}$  for l > 1

Combining the different components yields for l > 1:

$$W_{l} \leq \underbrace{\nu c_{1}n_{l}}_{1. \text{ pre-smoothing}} + \underbrace{c_{2}n_{l} + c_{3}n_{l}}_{2. \text{ defect restriction}} + \underbrace{\mu W_{l-1}}_{3. \text{ coarse grid solution}} \\ + \underbrace{n_{l} + c_{3}n_{l}}_{4. \text{ coarse grid correction}} + \underbrace{\nu c_{1}n_{l}}_{5. \text{ post-smoothing}} \\ = \underbrace{(2\nu c_{1} + c_{2} + 2c_{3} + 1)}_{=: C} n_{l} + \mu W_{l-1}$$

Applying this estimate recursively gives

$$W_{L} \leq Cn_{L} + \mu W_{L-1}$$

$$\leq Cn_{L} + \mu Cn_{L-1} + \mu^{2} W_{L-2}$$

$$\vdots$$

$$\leq C \sum_{i=0}^{L-1} \mu^{i} n_{L-i} + \mu^{L} c_{4} n_{0}$$

$$\stackrel{n_{l-1} \leq 2^{-d} n_{l}}{\leq} C \sum_{i=0}^{L-1} (\mu 2^{-d})^{i} n_{L} + (\mu 2^{-d})^{L} c_{4} n_{L}$$

$$\leq \max(C, c_{4}) n_{L} \sum_{i=0}^{L-1} (\mu 2^{-d})^{i}$$

$$\leq \max(C, c_4) \ n_L \frac{(\mu 2^{-d})^{L+1} - 1}{\mu 2^{-d} - 1} = O(n_L)$$

In the last step, I assumed that  $\mu 2^{-d} < 1$ , i.e.  $\mu < 2^{-d}$ . Hence for d = 2 one should use  $\mu \leq 3$  and for d = 3 one should use  $\mu \leq 7$ .

## 4.7. Convergence

The proofs for the convergence of the multigrid algorithm are quite lengthy. Hence, I only want to scratch the surface of what one needs to account for in such proofs. Firstly, the problem statement, i.e. the PDE, plays a major role. For example the Poisson problem and the Eddy Current problem need different smoothers in the multigrid method. For the Poisson problem simple point smoothers, like Jacobi or successive over relaxation (SOR) [19, pp. 285–290], are sufficient for the damping of the large eigenvalues of the system matrix. However, Maxwell's equations require the more sophisticated block smoothers, like Schwarz methods [2]. The convergence proofs heavily rely on the properties of the smoothers that are chosen and the number of smoothing steps  $\nu$  needs to be large enough. Additionally, the type of multigrid cycle has an impact on the analysis of the multigrid algorithm. When dealing with the Poisson problem, it is more straightforward to show the convergence of the W cycle than to work with the V cycle. In most cases, combining the smoothing property

$$\left\|\boldsymbol{S}^{\nu}\boldsymbol{u}_{\boldsymbol{l}}\right\|_{X} \leq c \frac{h_{l}^{-\beta}}{\nu} \left\|\boldsymbol{u}_{\boldsymbol{l}}\right\|_{Y}$$

with the approximation property

$$\left\|\boldsymbol{u}_{l}-\boldsymbol{u}_{l-1}\right\|_{Y}\leq Ch^{\beta}\left\|\boldsymbol{u}_{l}\right\|_{X},$$

where  $\|\cdot\|_X$  and  $\|\cdot\|_Y$  are suitable norms, yields the convergence of the multigrid algorithm [4]. Convergence proofs for the Poisson problem can be found in [6, 4].

For convergence of the multigrid algorithm in  $H^{\text{div}}$  and  $H^{\text{curl}}$  take a closer look

at [2]. Among other things, it has been shown there that the Schwarz smoothed mutligrid algorithm yields optimal complexity for

$$\rho^{2} \langle \mathbf{p}, \mathbf{q} \rangle + \kappa^{2} \langle \operatorname{curl} \mathbf{p}, \operatorname{curl} \mathbf{q} \rangle = \langle \mathbf{f}, \mathbf{q} \rangle,$$

where  $\kappa, \rho > 0$ . Here the condition number of the preconditioned system is bounded independently of h, the number of levels and  $\kappa$  and  $\rho$ . When working with other Time-Harmonic Maxwell problems, using a Schwarz smoother isn't always sufficient, which I experienced in my computations. Additionally one might need to apply special decompositions to the function space, which for example has been demonstrated in [7] for anisotropic finite elements.

#### 4.8. Application: Poisson equation

Together with Max Schröder, I implemented the multigrid method for two dimensional linear finite elements on triangular grids [20]. The domain  $\Omega := (-1, 1)^2 \setminus (0, 1)^2$  is an L-shape,  $\Gamma_D := (0, 1) \times \{0\} \cup \{0\} \times (0, 1)$  is the homogeneous Dirichlet boundary and  $\Gamma_N := \partial \Omega \setminus \Gamma_D$  is the homogeneous Neumann boundary. The weak form is given by:

DEFINITION 4.8.1 Variational form

Find  $u \in V := \{ u \in H^1(\Omega) \mid u = 0 \text{ on } \Gamma_D \}$  such that

 $a(u,v) = l(v) \quad \forall v \in V$ 

where  $a: V \times V \to \mathbb{R}$  is the bilinear form defined as

$$a(u,v) := \int_{\Omega} \nabla u \cdot \nabla v \, dx$$

and the right hand side  $l:V\rightarrow \mathbb{R}$  is a linear form defined as

$$l(v) := \int_{\Omega} f \cdot v \, dx.$$

Here, we use the right hand side function

$$f(x) := \begin{cases} -1 & \text{for } x \in (-1,0) \times (0,1) \\ 0 & \text{for } x \in (-1,0) \times (-1,0) \\ 1 & \text{for } x \in (0,1) \times (-1,0) \end{cases}$$

As a smoother, we use the  $\omega$ -Jacobi method

$$x^{k+1} = x^k + \omega \operatorname{diag}(A)^{-1} (b - Ax^k)$$

with  $\omega = 0.8$ . We created a sequence of globally refined grids.



Figure 4.5.: Level grids

These grids were refined by bisecting all edges of a triangle and drawing a new triangle out of the three new nodes.



Figure 4.6.: Refining a triangle

When refining, one needs to store the parents of the new nodes. The parents are

the two end nodes of the edge that has been bisected, e.g. node 1 and node 2 are the parents of node 4. To create the prolongation matrix  $I_{l-1}^l$ , we follow the rules:

```
if (i.th node already exists on level l-1):

(I_{l-1}^{l})_{i,i} = 1
selse:

# \text{ get the indices of the parents of the i.th node}
(I_{l-1}^{l})_{i,\text{parent}_{1}} = \frac{1}{2}
(I_{l-1}^{l})_{i,\text{parent}_{2}} = \frac{1}{2}
```

For this kind of prolongation matrix, the restriction matrix is given by

$$I_{l}^{l-1} = \frac{1}{4} \left( I_{l-1}^{l} \right)^{T}.$$

Running the simulation on an AMD Ryzen 7 2700X with 16 GB RAM, using a tolerance of  $10^{-12}$  for the defect and applying 2 smoothing steps, we then get

Refinements	DoFs	Multigrid iterations
1	21	14
2	65	14
3	225	14
4	833	14
5	3201	13
6	12545	13
7	49665	12
8	197633	12
9	788481	11

Figure 4.7.: Results of multigrid for Poisson with uniform refinement

These results show that for certain problems the multigrid method can yield a fast iterative solver which converges in a constant number of iteration steps.

# 4.9. Application: Maxwell's equations

In practice the multigrid method is usually used as a preconditioner for other iterative solvers like the preconditioned conjugate gradient (PCG) [19, pp. 324–325] or the generalized minimal residual (GMRES) [19, p. 327] method. This can also be seen in the literature described in the next chapter, where Schwarz smoothers are employed in multigrid preconditioners that efficiently solve the Eddy Current or the Time-Harmonic Maxwell problem.

# 5. Related research

In this chapter I will briefly discuss the previous results from other researchers who applied a multigrid approach to Maxwell's equations:

# 5.1. Chen [7]

In his PhD thesis, Chao Chen used the multigrid method for time-harmonic Eddy Current problems. He demonstrated how one can deal with anisotropic finite element meshes and proposed a plane smoother for the Nédélec elements. Chen created numerical methods for the magnetostatic and for the time-harmonic Eddy Current problem. Furthermore, he implemented a multigrid preconditioner for the CG method and demonstrated how the multigrid method can be used for nonlinear problems.

First he analyzed the magnetostatic problem in the "A-formulation".

DEFINITION 5.1.1 Strong form	n (Magnetostatic)
$\operatorname{curl}\left(  u \operatorname{curl} oldsymbol{A}  ight) = \operatorname{curl} oldsymbol{T}_{0}$	in $\Omega$
$oldsymbol{A} imesoldsymbol{n}=oldsymbol{0}$	on $\Gamma_B$
$ u \operatorname{curl} oldsymbol{A}  imes oldsymbol{n} = oldsymbol{T}_0  imes oldsymbol{n}$	on $\Gamma_H$

DEFINITION 5.1.2 Matrix form (Magnetostatic)  

$$A\boldsymbol{x} = \boldsymbol{b}$$
where  

$$A_{ij} := \nu \langle \operatorname{curl} \boldsymbol{N}_i, \operatorname{curl} \boldsymbol{N}_j \rangle,$$

$$b_j := \langle \boldsymbol{T}_0, \operatorname{curl} \boldsymbol{N}_j \rangle.$$

Then he continued with the time-harmonic Eddy Current problem in the "A,V-formulation".

Definition 5.1.3	Strong form (Time-Harmonic I	Eddy Current)
$\left( u \right) \left( u \right) $	$\dot{a}$ $a = \dot{a}$ $a = \nabla V = 0$	in O
$\operatorname{curr}(\nu\operatorname{curr}\mathbf{A})$ -	$-j\omega \delta \mathbf{A} + j\omega \delta \nabla V = 0$	
	$oldsymbol{A} imesoldsymbol{n}=oldsymbol{0}$	on $\Gamma_E$
	$ u \operatorname{curl} \boldsymbol{A}  imes \boldsymbol{n} = \boldsymbol{T}_0  imes \boldsymbol{n}$	on $\Gamma_{H_c}$

DEFINITION 5.1.4 Matrix form (Time-Harmonic Eddy Current)  $\begin{pmatrix} K & C^{T} \\ C & B \end{pmatrix} \begin{pmatrix} a \\ v \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$ where  $K_{ij} := \nu \langle \operatorname{curl} \mathbf{N}_{i}, \operatorname{curl} \mathbf{N}_{j} \rangle + j\omega\sigma \langle \mathbf{N}_{i}, \mathbf{N}_{j} \rangle,$   $C_{ij} := j\omega\sigma \langle \nabla \mathbf{N}_{i}, \mathbf{N}_{j} \rangle,$   $B_{ij} := j\omega\sigma \langle \nabla \mathbf{N}_{i}, \nabla \mathbf{N}_{j} \rangle,$   $f_{ij} := \langle \mathbf{T}_{0}, \operatorname{curl} \mathbf{N}_{j} \rangle.$ 

In these formulations  $N_i$  denotes the Nédélec edge elements. Overall, it has been shown that block smoothers are more suitable for anisotropic finite element meshes than point smoothers. Furthermore, Chen's plane smoother has been shown to be cheap and efficient for Maxwell's equations.

# 5.2. Zhong et al. [25]

In this paper Zhong et al. presented a two-grid approach for the time-harmonic Maxwell's equations and analyzed the error between the numerical and the analytical solution when varying parameters like  $\omega$  or the mesh sizes.

in  $\Omega$ 

on  $\Gamma$ 

DEFINITION 5.2.1 Strong form  $\operatorname{curl} \left( \mu^{-1} \operatorname{curl} \boldsymbol{E} \right) - \omega^2 \eta \boldsymbol{E} = \boldsymbol{F}$  $\boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{0}$ 

with  $\eta = \varepsilon + \frac{i\sigma}{\omega}$  and  $\mathbf{F} = i\omega \mathbf{J}$ .

DEFINITION 5.2.2 Variational form

 $a(\mathbf{E}, \boldsymbol{\psi}) = l(\boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_0^{\mathrm{curl}}(\Omega) \coloneqq \left\{ \boldsymbol{v} \in H^{\mathrm{curl}}(\Omega) \mid \mathbf{n} \times \boldsymbol{v}_{|_{\Gamma}} = 0 \right\}$ 

where

$$a(\mathbf{E}, \boldsymbol{\psi}) := \langle \operatorname{curl} \mathbf{F}, \operatorname{curl} \boldsymbol{\psi} \rangle - \langle \left( \omega^2 \varepsilon + i \omega \sigma \right) \mathbf{E}, \boldsymbol{\psi} \rangle,$$
$$l(\boldsymbol{\psi}) := \langle \mathbf{F}, \boldsymbol{\psi} \rangle.$$

The numerical experiments have been carried out on a cube  $\Omega := (0, 1)^3$  with  $\varepsilon = \mu = 1$  and  $\mathbf{F} \in H_0^{\text{curl}}(\Omega)^*$ , the dual space of  $H_0^{\text{curl}}(\Omega)$  w.r.t. the  $L^2$  inner product.  $\omega$  has been chosen out of  $\{1, 5, 10\}$ . Nédélec edge elements of the 0.th order have been used. The smoothers on the local meshes have been the minimum residual method (MinRes) [22] for indefinite and PCG for positive problems.

No iteration numbers or convergence rates have been mentioned in this publication. Instead error estimates have been used to prove the efficiency of the presented methods. However, the authors pointed out that for the Maxwell problem certain iterative two-grid methods don't converge.

## 5.3. Arnold, Falk, Winther [2]

In their paper from the year 2000 Arnold, Falk and Winther layed out the theory for the convergence of a multigrid method in  $H^{\text{div}}$  and  $H^{\text{curl}}$ . They further showed that domain decomposition methods, like the overlapping multiplicative Schwarz method, are suitable smoothers for these kind of problems.

DEFINITION 5.3.1 Strong form

 $\rho^2 \boldsymbol{u} + \kappa^2 \operatorname{curl} \operatorname{curl} \boldsymbol{u} = \boldsymbol{f}$ 

with natural boundary conditions.

```
DEFINITION 5.3.2 Variational form
```

```
a(\mathbf{p}, \mathbf{q}) = l(\mathbf{q})
```

where

$$\begin{split} a(\mathbf{p}, \mathbf{q}) &\coloneqq \rho^2 \langle \mathbf{p}, \mathbf{q} \rangle + \kappa^2 \langle \operatorname{curl} \mathbf{p}, \operatorname{curl} \mathbf{q} \rangle \,, \\ l(\mathbf{p}) &\coloneqq \langle \mathbf{f}, \mathbf{q} \rangle \,. \end{split}$$

They also point out that "some of the simplest and most frequently used smoothers for elliptic problems do not yield effective multigrid iterations [...] here" [2]. The authors proved that the V cycle algorithm is an efficient solver and preconditioner, if

- 1. a suitable finite element subspace of  $H^{\text{curl}}$  is used and
- 2. an appropriate smoother is being applied.

They propose the usage of

- 1.  $\mathcal{N}^k \subset H^{\operatorname{curl}}(\Omega)$ , the Nédélec edge elements, and
- 2. smoothers based on a decomposition based on vertex patches or a decomposition arising from the Helmholtz decomposition.

## 5.4. Janssen and Kanschat [13]

This paper extends the multigrid approach from Arnold, Falk and Winther [2] for  $H^{\text{curl}}$  to adaptively refined meshes. A framework for dealing with locally refined level meshes is discussed both theoretically and how it can be implemented in software libraries, like deal.II [3]. The authors used a symmetric, multiplicative Schwarz smoother inside their multigrid method.

DEFINITION 5.4.1 Variational form Find  $\mathbf{u} \in V$  such that  $a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \quad \forall \mathbf{v} \in V,$ 

where

$$a(\mathbf{u}, \mathbf{v}) := \langle \operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v} \rangle - \sigma \langle \mathbf{u}, \mathbf{v} \rangle,$$
$$l(\mathbf{v}) := \langle \mathbf{f}, \mathbf{v} \rangle,$$

V is a subspace of  $H^{\text{curl}}(\Omega)$  with suitable conditions on the tangential traces at the boundary and  $\sigma$  is less than the smallest nonzero eigenvalue of the Maxwell operator.

The numerical experiments have been done for  $\mathbb{R}^2$  and  $\mathbb{R}^3$  with  $\sigma = 1$ . On the whole problem GMRES with reorthogonalization is being used and a Schwarz smoother on the multigrid level problems. In  $\mathbb{R}^2$ , the eddy current problem is solved on an L-shape  $\Omega := (-1, 1)^2 \setminus (0, 1)^2$  with right hand side  $\boldsymbol{f} = (1, 1)^T$ . In  $\mathbb{R}^3$ , the eddy current problem is solved on a cube with one corner removed  $\Omega := (-1, 1)^3 \setminus (0, 1)^3$  with right hand side  $\boldsymbol{f} = (1, 1, 1)^T$ . In their numerical experiments, Janssen and Kanschat observed that

- the convergence rates are independent of the mesh size,
- the convergence rates are independent of the polynomial degree of  $\mathcal{N}^k$  and
- the method is robust w.r.t. local refinement.

# 6. Numerical tests

In the numerical tests the grids are being refined uniformly and 5 pre-/post smoothing steps are being made.

#### 6.1. Eddy Current

Trying to test the performance of my multigrid routine, I implemented the problem statement of Kanschat and Janssen, which I already described in depth in sections 3.1 and 5.4. Given that the authors didn't explicitly state their boundary conditions, I applied the constraint  $\mathbf{n} \times \mathbf{u} = \mathbf{0}$  on the boundary  $\partial \Omega$ . The computations are done on an AMD Ryzen 7 2700X with 16 GB RAM. Only the Schwarz smoother needs more RAM for more than 5 global refinements. There the computations are executed on a cluster with eight Intel Xeon Platinum 8268 and 200 GB RAM.

Although it is well known that simple smoothers, like Jacobi or Gauss-Seidel, don't have the required smoothing property for curl-elliptic problems [2, 13], I initially tested the multigrid algorithm with a SOR smoother [19, pp. 285–290]. This only serves as a lower performance bound for the multiplicative Schwarz smoother, which has been used later on. The results of the numerical experiments with the SOR smoother have been summarized in the table below:

Refinements	DoFs	GMRES Iter.	Condition number	Wall time [s]
2	3200	139	3570	1.33
3	12544	253	16299	9.49
4	49664	435	76669	126.12
5	197632	698	292518	893.67
6	788480	1167	$1.0946 \cdot 10^{6}$	13022.80

Figure 6.1.: Results of GMRES with GMG preconditioner and SOR smoother for the Eddy Current problem

It can clearly be seen that the number of iteration steps is not constant and the order of the wall time is significantly greater than 1. This is also shown in the corresponding plots.



Figure 6.2.: The iteration number of the SOR smoothed multigrid algorithm is in  $O(\text{DoFs}^{0.38})$  for the Eddy Current problem.



Figure 6.3.: The wall time of the SOR smoothed multigrid algorithm is in  $O(\text{DoFs}^{1.66})$  for the Eddy Current problem.

Since the SOR smoothed multigrid yields suboptimal results, I will now take a closer look at the multiplicative Schwarz smoother, which should yield better performance. These results are depicted in the following table.

Refinements	DoFs	GMRES Iter.	Condition number	Wall time [s]
2	3200	2	1.00423	1.73
3	12544	2	1.00497	8.43
4	49664	2	1.00537	38.28
5	197632	2	1.00559	162.24
6	788480	2	1.00570	675.22
7	3149824	2	1.00576	2776.22

Figure 6.4.: Results of GMRES with GMG preconditioner and Schwarz smoother for the Eddy Current problem

The geometric multigrid method is very appealing to researchers, since it promises that the number of operations needed to solve the linear system grows linearly with an increasing number of degrees of freedom. This is approximately the case in my experiments, since the iteration steps are constant and quadrupling the number of DoFs also quadruples the wall time as a rule of thumb. Additionally, the condition number of the preconditioned system is almost 1, which signifies that the preconditioner is a good approximation for the inverse of the system matrix. At first it might be surprising that GMRES only requires 2 steps to solve the linear equation system. However, I decided to use 5 smoothing steps which is quite high. For simpler PDEs, like the Poisson equation, 1 or 2 smoothing steps are sufficient (see section 4.8). More complex problems require more smoothing steps to ensure the convergence of the multigrid algorithm. Therefore, I am applying 5 smoothing steps. Consequently the number of GMRES iterations is very low.



Figure 6.5.: The wall time of the Schwarz smoothed multigrid algorithm is in  $O(\text{DoFs}^{1.07})$ .

In the figure below, the solution of my model problem has been plotted. The Lshape has been globally refined 7 times, which yields 3,149,824 degrees of freedom. No analytical solution to this problem is known a priori. In the plot the magnitude of the electric and magnetic field is shown. In the origin, which is a sharp edge of the domain, the intensity has a big spike. Hence the FEM solution is reasonable from a physics point of view, since "the electric field near a sharp point on a conductor is very high" [9]. This phenomenon is sometimes referred to as the electric point effect (German: elektrische Spitzenwirkung).



Figure 6.6.: Intensity of the solution of GMRES with GMG preconditioner and Schwarz smoother of the Eddy Current problem on a 7 times globally refined grid.

The code for the Eddy Current problem with all my parameters is available at https://github.com/mathmerizing/MultigridMaxwell.

# 6.2. Time-Harmonic Maxwell

The Eddy Current problem is just a special case of the Time-Harmonic Maxwell equations. Hence now I will analyze the more general formulation, which is a current research problem and can be used to assess the benefits and shortcomings of the multigrid method presented in Chapter 4.

DEFINITION 6.2.1 Strong form					
Find $\mathbf{u} \in H^{\operatorname{curl}}(\Omega)$ such that					
$\operatorname{curl}\left(\mu^{-1}\operatorname{curl}\mathbf{u}\right) - \varepsilon\omega^{2}\mathbf{u} = \mathbf{s}$	in $\Omega$	(1)			
$\mu^{-1} \operatorname{curl} \mathbf{u} \times \mathbf{n} - i\kappa\omega \left(\mathbf{n} \times \mathbf{u}\right) \times \mathbf{n} = \boldsymbol{u_1}$	on $\Gamma_1$	(2)			
$\mathbf{n}  imes \mathbf{u} = \mathbf{n}  imes oldsymbol{u}_2$	on $\Gamma_2$	(3)			
where $\varepsilon, \mu, \omega \in \mathbb{R}^+$ .					

Before I derived a weak form of this PDE, I first decided to achieve homogeneous boundary conditions on  $\Gamma_2$  by decomposing the solution variable into two parts

$$u = u_0 + u_2,$$

where  $u_0$  is my new solution and  $u_2$  fulfills the  $\Gamma_2$  - boundary condition. To make the further analysis more readable, I will stick to my physical problem, with  $\varepsilon = \kappa = 1$ ,  $\mathbf{s} = u_1 = 0$  and material parameter  $\mu^{-1} = v^2$ . The weak form has been derived in the same way as for the Eddy Current problem in section 3.1.

For the numerical simulations in  $\mathbb{R}^2$ , I used  $\Omega := (0, 1)^2$ . This domain is composed of the subdomains  $\Omega_1 := \left(\frac{3}{8}, \frac{5}{8}\right) \times (0, 1)$  and  $\Omega_0 := \Omega \setminus \Omega_1$ . One has to differentiate between these two regions, since different materials are being used there, resulting in different material parameters that are being accounted for in the assembly of the system matrix. The boundary of the domain is made up of  $\Gamma_2 := [0, 1] \times \{0\}$ and  $\Gamma_1 := \partial \Omega \setminus \Gamma_2$ . The entire setup of  $\Omega$  can be seen in the figure below.



Figure 6.7.: The domain  $\Omega$ 

On the subdomains of  $\Omega$ , the constants  $\omega$  (wave frequency) and v (material parameter) are being defined.  $\omega$  has been chosen to be 20 on the entire domain. v equals 1 on  $\Omega_0$  and  $\frac{1}{1.516}$  on  $\Omega_2$ . The system matrix reads:

$$\begin{pmatrix} A & B \\ -B & A \end{pmatrix} \begin{pmatrix} U_{\rm Re} \\ U_{\rm Im} \end{pmatrix} = \begin{pmatrix} F_{\rm Re} \\ F_{\rm Im} \end{pmatrix}$$

where

$$\begin{split} A &:= \left( v^2 \left\langle \operatorname{curl} \boldsymbol{\phi_i}, \operatorname{curl} \boldsymbol{\phi_j} \right\rangle - \omega^2 \left\langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \right\rangle \right)_{i,j=1}^n, \\ B &:= \left( \omega \left\langle \mathbf{n} \times \boldsymbol{\phi_i}, \mathbf{n} \times \boldsymbol{\phi_j} \right\rangle_{\Gamma_1} \right)_{i,j=1}^n, \\ F_{\text{Re}} &:= \left( -v^2 \left\langle \operatorname{curl} \Re(\boldsymbol{u_2}), \operatorname{curl} \boldsymbol{\phi_j} \right\rangle + \omega^2 \left\langle \Re(\boldsymbol{u_2}), \boldsymbol{\phi_j} \right\rangle \right)_{j=1}^n \\ &- \left( \omega \left\langle \mathbf{n} \times \Im(\boldsymbol{u_2}), \mathbf{n} \times \boldsymbol{\phi_j} \right\rangle_{\Gamma_1} \right)_{j=1}^n, \\ F_{\text{Im}} &:= \left( -v^2 \left\langle \operatorname{curl} \Im(\boldsymbol{u_2}), \operatorname{curl} \boldsymbol{\phi_j} \right\rangle + \omega^2 \left\langle \Im(\boldsymbol{u_2}), \boldsymbol{\phi_j} \right\rangle \right)_{j=1}^n \\ &+ \left( \omega \left\langle \mathbf{n} \times \Re(\boldsymbol{u_2}), \mathbf{n} \times \boldsymbol{\phi_j} \right\rangle_{\Gamma_1} \right)_{j=1}^n. \end{split}$$

This matrix is obviously not symmetrical, due to the different signs in front of the block matrix B. However, symmetry can be achieved, by multiplying the bottom blocks with -1, i.e.

$$\begin{pmatrix} A & B \\ B & -A \end{pmatrix} \begin{pmatrix} U_{\rm Re} \\ U_{\rm Im} \end{pmatrix} = \begin{pmatrix} F_{\rm Re} \\ -F_{\rm Im} \end{pmatrix}.$$

For the boundary condition on  $\Gamma_2$ , the function

$$\boldsymbol{u_2}: \Gamma_2 \to \mathbb{C}^2, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \exp\left(-\frac{(x-0.5)^2}{0.1}\right) \\ 0 \end{pmatrix} + i \begin{pmatrix} \exp\left(-\frac{(x-0.5)^2}{0.1}\right) \\ 0 \end{pmatrix}.$$

is used.

In my experiments the multigrid method from Chapter 4 diverges. Consequently, one needs a special decomposition of the function space for the multigrid algorithm to be efficient. In the literature, it has also been shown that multigrid requires special components for the indefinite Helmholtz problem, which are sophisticated and difficult to implement [8]. By the De Rham complex, the same argument also holds true for the Time-Harmonic Maxwell problem. Therefore I decided to stop my experiments on the Time-Harmonic Maxwell problem.

# 7. Conclusion and Outlook

In this thesis, the application of the geometric multigrid method to Maxwell's equations has been investigated. This work has been inspired by the paper of Janssen and Kanschat [13], in which a multiplicative Schwarz smoother has been used in the multigrid method to solve the Eddy Current problem. I reimplemented the approach of Janssen and Kanschat in the deal.II software library [3] and the results of my numerical tests indicate that the number of arithmetic operations of this technique grows linearly with the number of degrees of freedom. Hence, in terms of cost complexity (asymptotically) optimal results have been achieved with my C++ code. However, when working with the Time-Harmonic Maxwell problem, it has been shown that the multigrid method diverges and a suitable decomposition of the function space is required.

During this work several ideas for future development arose. Although the multiplicative Schwarz smoother is a powerful method, it has a few limitations. Firstly, it has to be applied sequentially and thus can't be used when working with parallel triangulations. Hence the multiplicative Schwarz smoother is not suitable for high performance computing. Furthermore, the inversion of the local matrices on the vertex patches has a high memory consumption, since the matrix inverses need to be stored in RAM. Therefore, a different smoother with similar properties could be used instead. In further research, it could be investigated whether other smoothers which rely on decompositions of the test function space could replace the Schwarz smoother in the multigrid method. Moreover, it would be interesting to examine how the approach of this work would need to be adapted to solve other problems in  $H^{curl}$ . Finally, it would be worthwile to use Janssen and Kanschat's methodology on adaptively refined meshes, e.g. by refining the grid with the dual weighted residual method.

# A. Convergence order

The goal of this thesis was to investigate whether GMG is a viable preconditioner for Maxwell's equations. For that purpose one needs to be able to assess the efficiency of a given algorithm. This can be done by determining the convergence order of a quantity of interest (QoI), e.g. the number of iteration steps or the wall time of the computation, plotted against the number of degrees of freedom.

Firstly, let's assume that we have a converging process  $P(\cdot)$ , which represents our quantity of interest. Furthermore, let P(N) converge to some limit P for  $N \to 0$ and  $P(N) - P = O(N^{\alpha})$ . Here  $\alpha \in \mathbb{R}_{>0}$  denotes the convergence order, which we are interested in. This assumption can be further simplified by P = 0, since for N = 0 we have 0 DoFs and thus our QoI, which is related to the computational complexity of the PDE, also equals 0. Overall, our assumption can therefore be written as

$$\exists \ c, \alpha \in \mathbb{R}_{>0}: \qquad P(N) = cN^{\alpha}.$$

## A.1. Closed formula

It has been shown in [23] that given three values P(N), P(N/2) and P(N/4), the convergence order can be estimated by:

$$\alpha = \frac{1}{\log(2)} \log \left( \frac{P(N) - P(N/2)}{P(N/2) - P(N/4)} \right)$$

This formula can also be generalized for some arbitrary factor  $k \in \mathbb{R}_{>0}$  and three values P(N), P(N/k) and  $P(N/k^2)$  to:

$$\alpha = \frac{1}{\log(k)} \log \left( \frac{P(N) - P(N/k)}{P(N/k) - P(N/k^2)} \right).$$

If one has more than three values, one can repeat this procedure for all three consecutive values and average over the different values of  $\alpha$  to get a better approximation of the convergence order.

## A.2. Linear regression

If we have more than three values or if the inputs to the converging process  $P(\cdot)$  are not scaled by a common factor  $k \in \mathbb{R}_{>0}$ , one might want to take a more sophisticated approach to find the convergence order. I assumed that

$$\exists c, \alpha \in \mathbb{R}_{>0}: \qquad P(N) = cN^{\alpha}.$$

Taking the logarithm on both sides transforms this into a linear function

$$\log P(N) = \log (cN^{\alpha}) = \log c + \log N^{\alpha} = \log c + \alpha \log N$$

if we first take the logarithm of the N and P(N) components of our data points. Now the values  $\log c$  and  $\alpha$  can be simply calculated by a least squares approach, trying to fit a linear function to  $\log P(N) = \log c + \alpha \log N$ . In this work, my convergence orders have been computed with this method, since it is more robust than the closed formula.

# Bibliography

- D. Arndt et al. "The deal.II Library, Version 9.1". In: Journal of Numerical Mathematics 27.4 (2019), pp. 203-213. DOI: 10.1515/jnma-2019-0064. URL: https://dealii.org/deal91-preprint.pdf.
- [2] D. N. Arnold, R. S. Falk, and R. Winther. "Multigrid in H (div) and H (curl)". In: *Numerische Mathematik* 85.2 (Apr. 2000), pp. 197–217. DOI: 10. 1007/p100005386. URL: https://doi.org/10.1007/p100005386.
- [3] W. Bangerth, R. Hartmann, and G. Kanschat. "deal.II a General Purpose Object Oriented Finite Element Library". In: ACM Trans. Math. Softw. 33.4 (2007), pp. 24/1–24/27.
- [4] S. Beuchler. Lecture notes in 'Multigrid and domain decomposition'. Apr. 2020.
- [5] D. Boffi. "A note on the deRham complex and a discrete compactness property". In: *Applied Mathematics Letters* 14.1 (Jan. 2001), pp. 33–38. DOI: 10.1016/s0893-9659(00)00108-7. URL: https://doi.org/10.1016/s0893-9659(00)00108-7.
- [6] D. Braess. Finite Elemente. Springer Berlin Heidelberg, 2013. DOI: 10.1007/ 978-3-642-34797-9. URL: https://doi.org/10.1007%2F978-3-642-34797-9.
- [7] C. Chen. "Geometric multigrid for eddy current problems". PhD thesis. 2012.
- [8] O. G. Ernst and M. J. Gander. "Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods". In: *Numerical Analysis of Multiscale Problems*. Ed. by I. G. Graham et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 325–363. ISBN: 978-3-642-22061-6. DOI: 10.1007/978-

3-642-22061-6\_10. URL: https://doi.org/10.1007/978-3-642-22061-6\_10.

- R. P. Feynman, R. B. Leighton, and M. Sands. The Feynman lectures on physics; New millennium ed. Originally published 1963-1965. New York, NY: Basic Books, 2010. URL: https://cds.cern.ch/record/1494701.
- C. Großmann and H.-G. Roos. "Numerische Verfahren für die diskretisierten Probleme". In: Numerik partieller Differentialgleichungen. Wiesbaden: Vieweg+Teubner Verlag, 1992, pp. 246–292. ISBN: 978-3-322-96752-7. DOI: 10. 1007/978-3-322-96752-7\_5. URL: https://doi.org/10.1007/978-3-322-96752-7\_5.
- [11] W. Hackbusch. Multi-Grid Methods and Applications. Springer Berlin Heidelberg, 1985. DOI: 10.1007/978-3-662-02427-0. URL: https://doi. org/10.1007/978-3-662-02427-0.
- [12] P. Jacobsson. "Nedelec elements for computational electromagnetics". In: 2007.
- [13] B. Janssen and G. Kanschat. "Adaptive Multilevel Methods with Local Smoothing for H<sup>1</sup>- and H<sup>curl</sup> -Conforming High Order Finite Element Methods". In: SIAM Journal on Scientific Computing 33.4 (2011), pp. 2095–2114.
- J. Li and Y. Huang. "Introduction to Finite Element Methods". In: Time-Domain Finite Element Methods for Maxwell's Equations in Metamaterials. Springer Berlin Heidelberg, Oct. 2012, pp. 19–51. DOI: 10.1007/978-3-642-33789-5\_2. URL: https://doi.org/10.1007/978-3-642-33789-5\_2.
- P. L. Lions. "On the Schwarz alternating method I". In: Proc. 1st Int. Symp. on Domain Decomposition Methods (1988). Ed. by R. Glowinski et al., pp. 1– 42.
- [16] P. Monk. Finite Element Methods for Maxwell's Equations. Oxford Science Publications, 2003.
- [17] J. C. Nédélec. "A new family of mixed finite elements in ℝ<sup>3</sup>". In: Numerische Mathematik 50.1 (Jan. 1986), pp. 57-81. DOI: 10.1007/bf01389668. URL: https://doi.org/10.1007/bf01389668.

- [18] J. C. Nédélec. "Mixed finite elements in ℝ<sup>3</sup>". In: Numerische Mathematik 35.3 (Sept. 1980), pp. 315-341. DOI: 10.1007/bf01396415. URL: https://doi.org/10.1007/bf01396415.
- [19] T. Richter and T. Wick. Einführung in die Numerische Mathematik Begriffe, Konzepte und zahlreiche Anwendungsbeispiele. Berlin Heidelberg New York: Springer-Verlag, 2017. ISBN: 978-3-662-54178-4.
- [20] J. Roth and M. Schröder. Multigrid's documentation. 2020. URL: https: //julianroth.org/documentation/multigrid/index.html (visited on 06/22/2020).
- [21] H.A. Schwarz. "Über einen Grenzübergang durch alternierendes Verfahren".
   In: Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich (May 1870), pp. 272–286.
- [22] H. A. van der Vorst. "GMRES and MINRES". In: Iterative Krylov Methods for Large Linear Systems. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003, pp. 65–94. DOI: 10.1017/CB09780511615115.007.
- T. Wick. Numerical Methods for Partial Differential Equations. 2020, pp. 229–240. URL: https://doi.org/10.15488/9248.
- [24] S. Zaglmayr. "High Order Finite Element Methods for Electromagnetic Field Computation." PhD thesis. July 2006.
- [25] L. Zhong et al. "Two-grid methods for time-harmonic Maxwell equations". In: Numerical Linear Algebra with Applications 20.1 (Feb. 2012), pp. 93-111.
   DOI: 10.1002/nla.1827. URL: https://doi.org/10.1002/nla.1827.

# Statement of Authorship

I hereby declare that I am the sole author of this bachelor thesis and that I have not used any sources other than those listed in the bibliography and identified as references. In particular, I declare that all parts of this work that have been taken from other sources have been marked as such. I further declare that I have not submitted this thesis to any other institution in order to obtain a degree.

Place, Date

Julian Roth

# Selbstständigkeitserklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine weiteren als die angegebenen Hilfsmittel verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe. Die Prüfungsleistung wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Julian Roth