A Space-Time Parallel Multigrid Method for the Transient Eddy-Current Equation

Master Thesis
to obtain the academic degree of Diplom-Ingenieur in the Master’s Program Industriemathematik
Abstract

In this thesis, we solve a model problem for the Eddy-current equation parallel in time and space. Throughout the thesis we discuss various aspects of our algorithm and gather knowledge about its application in more general cases. The simple version of this algorithm uses multigrid in time to solve any parabolic problem, while the more complex version uses multigrid in space and time.

Therefore solving our model problem, which approximates an induction furnace, with the simple version is very straightforward and works as expected. On the contrary new concepts are required for the space-time multigrid version to deal with more general problems. This leads us to a generalized coarsening rule, which we verify for the heat equation by numerical experiments.

Our first attempts to solve the model problem with the space-time multigrid fails. A detailed analysis involving local Fourier analysis and numerical experiments, identifies the kernel of the curl operator as the root of the problem. We proceed to theoretically derive an improved smoother to remedy this problem. Said smoother relies on an auxiliary nodal problem, which can be solved very efficiently. The concept for this smoother can be applied to several parabolic problems, where the spatial operator features a non trivial kernel, for example the time dependent grad-div equation.

Our main contributions are the generalized coarsening rule, which makes solving practical problems more feasible, and the hybrid smoother, which enables coarsening in space. Both of these concepts could also be beneficial for related algorithms.
Acknowledgments

Special thanks go to Martin Neumüller for his helpful supervision of this thesis and for teaching me to properly write scientific texts. I acknowledge the Johann Radon Institute for Computational and Applied Mathematics (RICAM) for letting me use the computer cluster RADON 1.

Further I want to thank Bianca Klammer for carefully proofreading this thesis.

Finally I want to thank my parents for supporting me all the time during my studies.

Martin Schwalsberger
Contents

1 Introduction ................................................. 1
   1.1 Eddy-currents & Time Parallel Methods .............. 1
   1.2 Thesis Structure .................................. 2

2 Model Problem ............................................. 4
   2.1 Setup .................................................. 4
   2.2 Maxwell equations .................................. 5
   2.3 Eddy-current Approximation ......................... 7

3 FEM for Maxwell ........................................... 10

4 Time-Parallel Method ....................................... 12
   4.1 Smoother ................................................. 12
   4.2 A Multigrid Method .................................... 13
      4.2.1 Space-Time Coarsening ......................... 14

5 Local Fourier Analysis ..................................... 18
   5.1 Methodology ........................................... 18
      5.1.1 LFA Introduction ................................. 18
      5.1.2 Smoother ........................................ 23
      5.1.3 Two Grid Cycle .................................. 24
   5.2 Intermediate Results .................................. 25
   5.3 A Hybrid Smoother .................................... 27
      5.3.1 Helmholtz Splitting .............................. 27
      5.3.2 Hybrid smoother ................................ 30
   5.4 Spatial Smoothing Explained ......................... 36

6 Numerical Experiments .................................... 38
   6.1 Implementation ....................................... 38
   6.2 Space-Time coarsening ................................ 39
      6.2.1 Heat-equation setting ............................ 39
      6.2.2 Eddy-current setting ............................ 41
   6.3 Scaling results ....................................... 44
   6.4 Visualized Results .................................... 45
Chapter 1
Introduction

1.1 Eddy-currents & Time Parallel Methods

Electromagnetic phenomena are widely used for technological applications, such as generators, actuators, electronics, antennas and even optics, which are used in devices for everyday use and industrial applications. In general there's a spectrum of applications ranging from high to low frequencies, where many (but not all) applications of low frequencies are dominated by diffusion effects, while the high frequency applications are dominated by wave effects.

All electromagnetic effects are governed by the Maxwell’s equations, which are a set of partial differential equations, which were originally formulated as equivalent integral equations. However many low frequency applications can be modeled with a simplified version of the Maxwell’s equation, the so called Eddy-current approximation. This equation is free of wave effects and simplifies numerical simulations.

Highly popular methods for the Eddy-current equation are based on time harmonic models, which are limited to static geometry and invariant material parameters. Methods for more complex models involving moving parts, saturation effects due to nonlinear materials and even coupling with other PDEs require time dependent methods.

For technical applications numerical simulation is the key to faster, cheaper and better development. As processing power along with theoretical knowledge has been steadily increasing over the last decades, more accurate, faster and more versatile simulations became possible and the integration of control theory or shape optimization made simulations even more useful.

One of the most important developments in computer technology is the increase in processing power through the parallel use of many processors at the same time, instead of single cores with exponentially increasing performance. Graphical processing units (GPU) combine the power of thousands of processors into a small device and
are therefore the most powerful hardware for parallel computations. On the flip side, development for GPU based programs is hard, the possible applications are limited and in unoptimized programs memory access times easily dominate computation times.

To stay competitive simulation software needs to adapt to these new developments and utilize the full potential of parallel programming. However, many traditional approaches to problems cannot be structured in a parallel way, so from the ground up, new algorithms are needed. In the particular case of time-dependent simulation, the program usually computes consequent states in time based on previous time states, thus preventing a parallel execution in time. In the last decades, this was usually fine, as spatial parallelization techniques were sufficient to utilize the available processors, but now and in the near future, this is changing.

In 1964 Nievergelt was the first to consider solving time-dependent problems parallel in time. Since then, many different approaches for time parallel solving were invented and improved, see [3]. The contribution on which this thesis is based was made by M. Neumüller in his doctoral thesis (supervised by O. Steinbach). This method was further backed up by theoretical analysis, which is a handy tool to experiment with variations of the method. Yet, it was designed for a simple heat equation test problem, so many generalizations need to be explored before this algorithm can be applied in practical simulations. Since then, some contributions to the method were made, expanding it to different domains. It was successfully applied to stochastic differential equations [13], optimal control problems [12], and unpublished results for the Stokes equation exist.

As this algorithm is best suited for parabolic problems, we will apply it to the Eddy-current equation. We will present the results of this application in Chapter 6 along with a local Fourier analysis analogously to Neumüllers doctoral thesis [11] in Chapter 5. Additionally, we expand the space-time multigrid aspect of the algorithm to problems more general than the original one-dimensional heat equation test case.

This thesis can be seen as one of many stepping stones from the initial concept of a space-time parallel method to a fully general application of these methods in practical simulations of all kinds.

1.2 Thesis Structure

This thesis is structured the following way:

- Chapter 2 introduces the Maxwell equations and consequentially the Eddy-current equation. We also present our model problem, which will serve as a benchmark throughout the thesis.
• In Chapter 3 we derive a variational formulation of the Eddy-current equation and apply finite element discretization in space and a Runge-Kutta method in time, which results in a solvable linear system.

• Chapter 4 presents the time parallel algorithm from [11] and explains open questions and possible solutions regarding more general settings. The generalizations proposed in this part are of main concern in the following chapters.

• In Chapter 5 a local Fourier analysis is performed for the algorithm applied to the two dimensional analogue of the Eddy-current equation. We proceed to develop an improved smoother for the Eddy-current equation.

• Chapter 6 first investigates some generalizations for the heat equation by numerical experiments followed by a confirmation of the results from Chapter 5 by several numerical experiments. Finally the results of the simulation are presented.

• In Chapter 7 we sum up all findings of this thesis and give an outlook on open problems and possible applications.
Chapter 2

Model Problem

2.1 Setup

In this thesis we simulate a simplified induction furnace. The reason for this choice is, that it does not feature moving parts (unlike electric engines) and can be modeled by a very simple geometry. A positive feature are the highly visible effects of the solution, namely intense surface fluxes on the iron core. Simply put, the induction furnace features the essence of the Eddy-current equation with a simple geometry.

Further we want to emphasize that a realistic representation of an induction furnace is in no way the aim of this theses, but to formulate a mathematical problem which contains interesting effects and relates to a real world application. Hence we draw this information from Wikipedia [18]:

*Real induction furnaces are composed of an insulating crucible holding the materials (preferably iron) to be melted. The crucible is surrounded by a water-cooled copper coil, through which a powerful alternating current flows. The fast alternation of the resulting magnetic field induces strong Eddy-currents in the iron, which cause Joule heating. Once the metal is melted steering of the melt occurs distributing heat.*

Our simplified model (Figure 2.1) is composed of a cylindrical iron part, that is surrounded by a ring shaped copper "coil". The remaining domain is vacuum. The crucible is ignored as it is assumed to have the same electric properties as vacuum. We will only perform the Eddy-current simulation, because coupling with the heat equation would be interesting and most likely feasible, but is out of the scope of this thesis.
2.2 Maxwell equations

This section introduces the Maxwell equations based on the book [14].

The Maxwell equations are the foundation of classic electromagnetism and deliver a complete macroscopic model for problems with static geometry. For simplicity we will restrict ourself to the case of piecewise constant, scalar material parameters. More general results for not extremely anisotropic tensor parameters are very similar. An example for an extremely anisotropic material would be graphite crystals, which mainly conduct currents in planes.

The physical quantities and material constants used in the Maxwell equations are given in Table 2.1.
CHAPTER 2. MODEL PROBLEM

<table>
<thead>
<tr>
<th></th>
<th>Physical quantities and constants of the Maxwell equations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Magnetic field $A/m$</td>
</tr>
<tr>
<td>$B$</td>
<td>Magnetic induction $Vs/m^2$</td>
</tr>
<tr>
<td>$E$</td>
<td>Electric field $V/m$</td>
</tr>
<tr>
<td>$D$</td>
<td>Electric induction $As/m^2$</td>
</tr>
<tr>
<td>$J$</td>
<td>Current density $A/m^2$</td>
</tr>
<tr>
<td>$J_i$</td>
<td>Applied current density $A/m^2$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Electric charge $As/m^3$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Magnetic permeability $Vs/Am$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Electric permittivity $As/Vm$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Electric conductivity $S/m$</td>
</tr>
</tbody>
</table>

The Maxwell equations are given by

\[
\begin{align*}
\text{curl } H &= \frac{\partial D}{\partial t} + J, \\
\text{div } B &= 0, \\
\text{curl } E &= -\frac{\partial B}{\partial t}, \\
\text{div } D &= \rho.
\end{align*}
\]

The constitutive laws are

\[
\begin{align*}
B &= \mu H + \mu_0 M, \\
D &= \epsilon E + P, \\
J &= \sigma E + J_i + \sigma (v \times B).
\end{align*}
\]

The gray variables $M, P, v$ for magnetization, polarization and velocity are always 0 in the context of this thesis.

**Remark 2.1** (Applied current density). The applied current density $J_i$ can be seen as a simplified representation for an external current source. To properly model an external current source, a geometrical representation of cables leading to the boundary of the domain with specific boundary conditions would be necessary. As this is over-complicating the problem, a coil with 100 windings and a current of 1 Ampere is simplified to a single ring with a cross section current of 100 Ampere through an applied current density.

From the Maxwell equations the E-field based formulation follows

\[
\epsilon \frac{\partial^2 E}{\partial t^2} + \sigma \frac{\partial E}{\partial t} + \text{curl } \mu^{-1} \text{curl } E = -\frac{\partial J_i}{\partial t},
\]
CHAPTER 2. MODEL PROBLEM

by the following derivation

\[ \text{curl} \ E = -\frac{\partial}{\partial t} \frac{B}{\mu H} \]
\[ \mu^{-1} \text{curl} \ E + \frac{\partial}{\partial t} H = 0 \]
\[ \text{curl} \mu^{-1} \text{curl} \ E + \frac{\partial}{\partial t} \text{curl} H = 0 \]
\[ \text{curl} \mu^{-1} \text{curl} E + \frac{\partial^2}{\partial t^2} \frac{D}{\varepsilon E} + \frac{\partial}{\partial t} \frac{J}{\sigma E + J_i} = 0. \]

Additionally in insulators the electric charge \( \rho \) vanishes leading to

\[ \text{div}(\varepsilon E) = \text{div} D = 0. \]

As mentioned in Remark 2.5 there are no insulating regions in our model problem, thus this equation can be ignored.

For the material parameters of the model problem we choose the following slightly rounded values based on Wikipedia [19]:

- \( \mu_{\text{iron}} = 0.0002, \sigma_{\text{iron}} = 10^7 \),
- \( \mu_{\text{copper}} = 10^{-6}, \sigma_{\text{copper}} = 6 \cdot 10^7 \),
- \( \mu_{\text{vacuum}} = 10^{-6}, \sigma_{\text{vacuum}} = \varepsilon = 1 \).

The underlying reasons for the choice \( \sigma_{\text{vacuum}} = 1 \) instead of 0 will be explained in Remark 2.5.

### 2.3 Eddy-current Approximation

However the E-field based formulation is impractical or infeasible for simulating most electrical machines due to:

- Wave effects introduced by the second time derivative propagate at the speed of light, especially in vacuum.
- As machines are typically sized less than a few meters, the wave length can exceed the domain for low frequencies.

**Claim 1** (Eddy-current approximation). Under certain conditions the term \( \frac{\partial D}{\partial t} \) can be ignored, leading to the simpler Eddy-current approximation

\[ \sigma \frac{\partial E}{\partial t} + \text{curl} \mu^{-1} \text{curl} E = -\frac{\partial J_i}{\partial t}. \]  (2.1)
Necessary conditions:
The conditions all come from a heuristic physical background and underwent mathematical inspection afterwards. For now let us denote with $L$ the diameter of the domain of interest and with $\omega$ the highest occurring frequency. For the conductivity $\sigma$ of the conducting parts the conditions are

$$\mu_0\varepsilon_0\omega^2L^2 \ll 1,$$

$$\frac{\varepsilon_0}{\sigma}\omega \ll 1.$$  \hfill (2.2)

Please note, that the first condition is solely imposed on $\omega$ and $L$, as $\mu_0, \varepsilon_0$ are the material parameters of vacuum. Furthermore $\frac{1}{\sqrt{\mu_0\varepsilon_0}} = c_0$ is the speed of light.

For the materials only the conductivity $\sigma$ matters. These conditions were taken from [14, ch. 2].

Additionally the following geometrical features must not occur:

- Thin gaps between conductors
- Pointy extensions of conducting regions

The physical explanation to these conditions is, that they would introduce capacitive effects, which are neglected by the approximation. A detailed investigation of these conditions can be found in [15].

**Remark 2.2.** Linear convergence for the difference of the Maxwell solution and the Eddy-current approximation of $E$ is proven for the cases $\varepsilon \to 0$ and $\omega \to 0$ in [14, ch. 2].

**Remark 2.3** (Examples). For $\sigma_{\text{copper}} = 5.7 \cdot 10^7 \text{S/m}$, $\sigma_{\text{iron}} = 10^7 \text{S/m}$ and a frequency of 50 Hz, the conditions above are certainly fulfilled

$$L \ll 10^6 \text{m},$$

$$\frac{\varepsilon_0}{\sigma_{\text{copper}}}\omega \approx 5 \cdot 10^{-17} \ll 1,$$

$$\frac{\varepsilon_0}{\sigma_{\text{iron}}}\omega \approx 2.7 \cdot 10^{-16} \ll 1.$$  \hfill (2.3)

**Remark 2.4** (Boundary conditions). The most basic boundary conditions assume that the domain is surrounded with a material that features either $\mu \approx \infty$ or $\sigma \approx \infty$, for example iron.

- From $\mu \approx \infty$ follows $E \times n = 0$ and consequently $H \cdot n = 0$ (if $H$ is not eliminated like here).
- From $\sigma \approx \infty$ follows $H \times n = 0$ and consequently $E \cdot n = 0$ and $\mu^{-1}\text{curl}(E)\times n = 0$ by the following derivation.
\[
\mu^{-1} \text{curl}(E) = \mu^{-1} \frac{\partial B}{\partial t} = \frac{\partial}{\partial t} \mu^{-1} \mu H,
\]
\[
\mu^{-1} \text{curl}(E) \times n = \frac{\partial}{\partial t}(H \times n) = 0.
\]

Many more possible boundary conditions and also formulations for infinite domains exist, but we will restrict ourselves to the two above.

Typical problems are modeled as a machine surrounded by an infinite vacuum. Usually an enclosing iron box cannot be assumed, but it is possible to restrict the problem to a domain, where the expected effects at the boundary of said domain are negligible. Consequentially one of the two presented homogeneous boundary conditions is used.

Remark 2.5 (Vacuum regularization). The case of a true insulator introduces many difficulties, as highlighted in [14], and the possible time-harmonic models for the problem can be complicated or not transferable to the time-dependent case. Therefore we choose to set the conductivity of vacuum to 1, which is large enough for the numerical solvers to be stable and yet small enough to be close to the original problem, as proven in [1, ch. 4].

Furthermore we will see that this setup will pose sufficiently interesting challenges, so inspecting time parallel differential algebraic problems could be viable for a whole separate project.
Chapter 3

FEM for Maxwell

For practical simulations we require finite dimensional problems, which can be achieved by a variational reformulation and discretization schemes both in time and space. We will now perform these steps in detail.

Given the Eddy-current equation (2.1) for our model problem it is a standard procedure to derive a semi-variational formulation. First we multiply the equation (2.1) with a test function $F \in H_0(\Omega, \text{curl}) := \{ F \in H(\Omega, \text{curl}) | F \times n = 0 \text{ on } \Gamma_D \}$ for all $t \in (0, T)$

$$\sigma \frac{\partial E}{\partial t} \cdot F + \text{curl}(\mu^{-1} \text{curl } E) \cdot F = -\frac{\partial J_i}{\partial t} \cdot F.$$

Then we integrate over the domain $\Omega$

$$\int_{\Omega} \sigma \frac{\partial E}{\partial t} \cdot F dx + \int_{\Omega} \text{curl}(\mu^{-1} \text{curl } E) \cdot F dx = -\int_{\Omega} \frac{\partial J_i}{\partial t} \cdot F dx.$$

Next we apply integration by parts for the curl operator and swap integration and differentiation

$$\frac{\partial}{\partial t} \int_{\Omega} \sigma E(t) \cdot F dx + \int_{\Omega} \mu^{-1} \text{curl } E(t) \cdot \text{curl } F dx +$$

$$\int_{\Gamma_D} \mu^{-1} \text{curl } E(t) \cdot (F \times n) ds = -\int_{\Gamma_N} (\mu^{-1} \text{curl } E(t) \times n) \cdot F ds = \int_{\Omega} \frac{\partial J_i}{\partial t} (t) \cdot F dx.$$

With these steps we come to the semi-variational formulation, which is the last stage before applying finite elements.

Semi-variational formulation
CHAPTER 3. FEM FOR MAXWELL

Find \( E \in C^1([0,T], H_0(\Omega, \text{curl})) \) such that for all \( t \in (0,T) \) and all \( F \in H_0(\Omega, \text{curl}) \)
\[
\frac{\partial}{\partial t} \int_\Omega \sigma E(t) \cdot Fdx + \int_\Omega \mu^{-1} \text{curl} E(t) \cdot \text{curl} Fdx = \int_\Omega \frac{\partial J_i}{\partial t}(t) \cdot Fdx,
\]
\[
\int_\Omega E(0) \cdot Fdx = \int_\Omega E_0 \cdot Fdx.
\]

**Remark 3.1 (Nedelec Elements).** The discretization of variational formulations involving a curl-curl bilinear form proved to be difficult for a long time. Attempts to apply \( H^1 \) conforming elements proved to fail at interfaces of two materials, where the tangential component \((E \times n)\) is always continuous, but not necessarily the normal component \((E \cdot n)\). As explained in [9] the lowest order elements needed to be based on the edges of the mesh. The construction ensured that the tangential component on a face was only influenced by its three adjacent edges, thus ensuring tangential continuity and allowing normal discontinuities.

The lowest order Nedelec-elements have 6 degrees of freedom per tetrahedron, namely the average value of the tangential component of the vector field on each edge \((\int_E \psi(x) \cdot t_E dx)\). The polynomial function space is defined as
\[
R_1 := \{ p \in P_3|x \cdot p = 0 \} \oplus \mathbb{R}^3.
\]

With the Nedelec-elements we can spatially discretize the semi-variational equation, which leads to an ordinary differential equation,
\[
M_h \frac{\partial E_h}{\partial t}(t) + K_h E_h(t) = f_h(t),
\]
\[
M_h E_h(0) = E_{0,h}.
\]

This equation can be further discretized in time by using an implicit Runge-Kutta scheme, in our case the implicit Euler scheme with an uniform step-width \( \tau = T/m \)
\[
-M_h E_h^k + (M_h + \tau K_h) E_h^{k+1} = \tau f_h^{k+1},
\]
\[
M_h E_h^0 = E_{0,h}.
\]

This can be seen as an sequential procedure, but also as \( m \) linear equations, which can form a global linear system
\[
\begin{pmatrix}
A_{\tau,h} & A_{\tau,h} \\
-M_h & A_{\tau,h} \\
\vdots & \ddots \\
-M_h & A_{\tau,h}
\end{pmatrix}
\begin{pmatrix}
E_h^1 \\
E_h^2 \\
\vdots \\
E_h^m
\end{pmatrix} =
\begin{pmatrix}
\tau f_h^{1} + E_{0,h} \\
\tau f_h^{2} \\
\vdots \\
\tau f_h^{m}
\end{pmatrix},
\]
with \( A_{\tau,h} = M_h + \tau K_h \). This global system matrix will be called \( L_{\tau,h} \) and we aim to replace the usual forward elimination by an iterative scheme. For further notation we will now write this equation as
\[
L_{\tau,h} x = f.
\]
Chapter 4

Time-Parallel Method

This chapter introduces a simple version of the algorithm presented in [4] for solving the big linear system (3.1) derived from the Eddy-current equation (2.1). The algorithm is based on the multigrid idea, which requires a smoothing procedure and a coarse grid correction. We start with the smoother.

4.1 Smoother

The first idea is to utilize a "Jacobi-preconditioned" Richardson scheme

\[ x^{k+1} = x^k + \omega D_{\tau,h}^{-1}(f - L_{\tau,h}x^k). \] (4.1)

Here \( D_{\tau,h} = \text{diag}\{A_{\tau,h}\}_{i=1}^m \) is the block diagonal of \( L_{\tau,h} \) consisting of \( A_{\tau,h} \) matrices.

If \( M_h \) can be assumed to be positive definite and \( K_h \) positive semidefinite, then this scheme will smooth the error in the time direction for the choice \( \omega = 0.5 \), see [4]. In the framework of "Local Fourier Analysis" this means that the smoother reduces error components with high time frequencies, see Chapter 5.2 for visualized examples.

Remark 4.1 (Parallelization). The presented smoother can be easily executed in parallel. Each processor will be responsible for some consecutive time steps. At the start of one iteration, each processor communicates its last time step to the next processor, which will be needed to compute \( L_{\tau,h}x^k \). All the other computations are completely independent and can be executed fully in parallel. This is illustrated in figure 4.1.

To expand on the idea, instead of single processors, we can also use groups of processors, which in turn will apply spatially parallel methods for the approximate inverse of \( A_{\tau,h} \) or the application of spatial matrices. To simplify the implementation we restrict the number of time steps and the number of processor-groups to a power of 2.
CHAPTER 4. TIME-PARALLEL METHOD

Figure 4.1: Conceptual visualization of the smoother. The first processor (more abstractly thread) will manage the first four time nodes. Each processor sends information from his last time node to the next one via a buffer. In the implementation the index of the solution vector is 0-based, so \( u_0 \) is not the initial value.

Remark 4.2 (Computational effort). One application of the smoother (4.1) would be as much computational work as sequentially solving the whole system. However it is sufficient to not apply \( A_{\tau,h}^{-1} \) with high precision \([4]\), because typically 1 to 3 PCG-iterations will be sufficient to maintain smoothing properties. For a comparison of runtimes see Chapter 6.3.

4.2 A Multigrid Method

Due to the smoothing properties of (4.1), we can add a coarse time-grid correction to achieve proper convergence rates. For this we need a hierarchical time mesh with several levels like in Figure 4.2.

![Example for a hierarchical time mesh](image)

Figure 4.2: Example for a hierarchical time mesh

The Time-Multigrid algorithm will apply the smoother at the beginning and the end. In the middle it will perform a recursive coarse grid correction. If the algorithm is called on the lowest time level, it will directly solve the very small problem instead.

Algorithm 1 (Time-Multigrid).

\[ MGCycle(x, f, \tau, h) \]

if \( \tau = T \) then
CHAPTER 4. TIME-PARALLEL METHOD

Direct Solving: \( x = L_{\tau,h}^{-1} f \)

\else
\nu_1 \text{ times Presmothing: } x = x + \omega D_{\tau,h}^{-1} (f - L_{\tau,h} x) \\
\text{Defect: } d_{\tau} = f - L_{\tau,h} x \\
\text{Restriction: } d_{2\tau} = R_{\tau,h} d_{\tau} \\
\text{Initialize: } w_{2\tau} = 0 \\
\text{MGCycle}(w_{2\tau}, d_{2\tau,h}, 2\tau, h) \\
\text{Prolongation: } w_{\tau} = P_{\tau,h} w_{2\tau} \\
\text{Correction: } x = x + w_{\tau} \\
\nu_2 \text{ times Postsmoothing: } x = x + \omega D_{\tau,h}^{-1} (f - L_{\tau,h} x) \n\end{if}

We can also use several coarse grid corrections instead of just one. For a detailed version see [11, Algorithm 4.2].

Remark 4.3 (Sequential parts). The presented multigrid algorithm can be executed mostly in parallel, however if we get to a level, which contains less time steps than processors (-groups), half or more of them will become idle, which is undesirable. Therefore it is important to speed up this less parallel section by reducing the problem size in space.

4.2.1 Space-Time Coarsening

This section will first inspect the concept of space-time coarsening in the most basic setting and generalize it later on.

One idea is to not only use a coarser time grid for the correction, but a coarser spatial mesh as well. For this to work it is necessary that the smoother also has smoothing properties in the spatial direction (see Chapter 5.2).

Results have been proven for the 1D heat equation

\[
\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = f. \tag{4.2}
\]

The smoother (4.1) indeed has spatial smoothing properties, if the following inequality is satisfied for the current level [11, ch. 4.3]

\[
\frac{\tau}{h^2} \geq \lambda_{\text{crit}}. \tag{4.3}
\]

This kind of condition will be called a "coarsening rule" within this work. The parameter \( \lambda_{\text{crit}} \) depends on the chosen Runge-Kutta scheme and is \( \sqrt{3}/2 \) for the implicit Euler scheme in this setting.

Remark 4.4 (Space-Time Coarsening Strategy). The values of the equation (4.3) are all known in advance and can be used to create a so called "Space-Time Coarsening
Strategy. With this strategy we know in advance which matrices to assemble and how to allocate memory for all levels. Therefore the check if space-coarsening should happen in algorithm 2 is already predetermined.

\[
\begin{array}{cccccccccc}
\text{space/time} & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
3 & X & X & & & & & & & \\
2 & & X & X & & & & & & & \\
1 & & & X & X & & & & & & & \\
0 & & & & & X & X & X & X & & \\
\end{array}
\]

Table 4.1: Example for a possible coarsening strategy over the levels in space and time. This strategy will execute two smoothers on the finest spatial mesh before using a coarser one.

We can now conclude the full Space-Time-Multigrid algorithm:

**Algorithm 2** (Space-Time-Multigrid).

\[
\text{MGCycle}(x, f, \tau, h)
\]

if $\tau = T$ then

Direct Solving: $x = L^{-1}_{\tau,h}f$

else

$\nu_1$ times Presmoothing: $x = x + \omega D^{-1}_{\tau,h}(f - L_{\tau,h}x)$

Defect: $d_{\tau,h} = f - L_{\tau,h}x$

if $\frac{\tau}{h^2} \geq \lambda_{\text{crit}}$ AND spatial coarsening possible then

$\tilde{h} = h_{\text{coarser}} \approx 2h$

else

$\tilde{h} = h$

end if

Restriction: $d_{2\tau,h} = R_{\tau,h}d_{\tau,h}$

Initialize: $w_{2\tau,h} = 0$

$\text{MGCycle}(w_{2\tau,h}, d_{2\tau,h}, 2\tau, \tilde{h})$

Prolongation: $w_{\tau,h} = P_{\tau,h}w_{2\tau,h}$

Correction: $x = x + w_{\tau,h}$

$\nu_2$ times Postsmoothing: $x = x + \omega D^{-1}_{\tau,h}(f - L_{\tau,h}x)$

end if

**Remark 4.5** (Transfer operators). The restriction can always be written as Kronecker-product of a time restriction and a spatial restriction:

$R_{\tau,h} = R_{\tau} \otimes R_h$.

If no spatial coarsening should happen $R_h$ is replaced by the identity matrix. Analogously the same holds for the prolongation.
There are several open questions for this coarsening rule, which we will try to address.

- Coefficients
- Higher dimensional meshes
- Non-uniform mesh size $h$
- Non-constant coefficients
- Different spatial differential operators

**Remark 4.6** (Constant coefficients). The case of constant coefficients

\[
\frac{\alpha}{\partial t} - \beta \frac{\partial^2 u}{\partial x^2} = f.
\]

can be easily reduced to the case \((4.2)\) by scaling of variables. As a consequence the coarsening rule changes to

\[
\frac{\tau \beta}{\alpha h^2} \geq \lambda_{\text{crit}}.
\]

The following extensions to the coarsening rule are experimental and chosen in the most reasonable way. Our approach will be tested by numerical experiments and analysis to see if they hold.

The problem of higher dimensional uniform meshes is solved by replacing the one dimensional length of an element by the diameter of the element. If the mesh is non-uniform or the coefficients are non-uniform, a global condition which relies on the most extreme values of the parameters could be used. Instead we decided to use a local condition over all elements

\[
\frac{\tau}{\max_{x \in E} \alpha(x) h_E^2} \geq \gamma \lambda_{\text{crit}}. \quad \forall E \in T_h(\Omega).
\]

This coarsening rule requires time independent coefficients, which should be constant or linear on each spatial element, so the extrema can be quickly determined. This is already sufficient for many applications.

**Remark 4.7** (Implementation). As our induction furnace model features a mesh separating the different materials the implementation of this rule was very easy. This is often implemented by a set of part-IDs $I$, separating the domain $\Omega$ into $\Omega_i | i \in I$. For now let $\ell_t$ and $\ell_x$ be the time or spatial level, where 0 stands for the coarsest mesh.

- Search maximum simplex diameter $h_{\text{max},i}$ per sub-domain $\Omega_i$ on the coarsest mesh
- Determine the domain with the smallest left hand side in \((4.4)\)

\[
\gamma = \min_{i \in I} \frac{\beta_i}{\alpha_i h_{\text{max},i}^2}.
\]
Use as a new criterion
\[ 2^{-t_{coarse}} 2^{-2\ell_x \gamma} \geq \lambda_{crit}. \]

This procedure is equivalent to (4.4) in 1D and 2D and with negligible impact on the coarsening strategy for 3D. If the diameters of refined simplexes is not approximately half the diameter of the coarse one, then this simplification should not be used. If cubic meshes are used, this simplification is always equivalent.

The reliability of the coarsening rule (4.4) will be tested in the following chapters. The "Local Fourier Analysis" chapter will determine differences between the heat equation and the Eddy-current equation. Meanwhile the "Numerical Experiments" chapter will address all open questions about the validity of the coarsening rule in a heuristic way.
Chapter 5

Local Fourier Analysis

This chapter will determine whether the generalized coarsening rule (4.4) works for the Eddy-current equation or not. A local Fourier analysis (short LFA) was already performed for this algorithm applied to the one dimensional heat equation [11]. The program used for that task was adapted to fit the setting of the two dimensional analogue to the curl operator and the results correspond with the numerical experiments.

The Fourier analysis for the parts concerning the rotated gradient is based on [2], for a detailed, comprehensible introduction to local Fourier analysis we recommend [16].

5.1 Methodology

5.1.1 LFA Introduction

In the local Fourier analysis we apply the time parallel scheme to a time periodic problem, meaning $u_0^j = u_j^0$. For introduction we will first present the local Fourier analysis for our time-parallel scheme applied to a simple test equation

\[
\kappa u(t) + \mu u'(t) = f(t),
\]

\[
u(-T) = u(T),
\]

where $t \in [-T, T]$.

When the implicit Euler scheme is applied and we denote $\alpha = \kappa \tau + \mu$, the global linear system (3.1) becomes

\[
\begin{pmatrix}
\alpha & -\mu \\
-\mu & \alpha \\
\vdots & \ddots & \ddots \\
-\mu & \alpha & \ddots & -\mu \\
\end{pmatrix}
\begin{pmatrix}
u^{m+1} \\
u^{m+2} \\
\vdots \\
u^n \\
\end{pmatrix}
= f,
\]
or in short

\[ L_\tau \mathbf{x} = \mathbf{f}. \]

The smoother reads

\[ \mathbf{x}^{k+1} = \mathbf{x}^k + \frac{\omega}{\alpha_\tau} I_{2m}(\mathbf{f} - L_\tau \mathbf{x}^k). \]

By introducing the exact solution \( \mathbf{x} \) and the error \( \epsilon^k := \mathbf{x}^k - \mathbf{x} \) we can reformulate the smoother.

\[ \mathbf{x}^{k+1} - \mathbf{x} = \mathbf{x}^k - \mathbf{x} + \frac{\omega}{\alpha_\tau} I_{2m}(L_\tau \mathbf{x} - L_\tau \mathbf{x}^k) \]

\[ \epsilon^{k+1} = (I_{2m} - \frac{\omega}{\alpha_\tau} L_\tau)\epsilon^k. \]

A necessary condition for the convergence of the smoother (and later the two grid cycle) is imposed on the spectral radius of the Smoothing matrix \( S_\tau \)

\[ \sigma(S_\tau) < 1. \]

The spectral radius is important for the speed of convergence, if the smoother was used on its own. The main idea of the local Fourier analysis is to provide a system of complex eigenvectors for this and other matrices. The calculation of the corresponding eigenvalues also becomes an easy task. With all possible eigenvalues available the convergence analysis immediately follows and even more insightful properties can be observed.

**Definition 5.1 (Fourier mode).** For a fixed frequency \( \theta \in (-\pi, \pi] \) we define the Fourier mode

\[ [\phi(\theta)]_l := e^{i\theta}, \quad \phi(\theta) \in \mathbb{C}^{2m}. \]

**Theorem 5.2.** For the Fourier frequencies \( \theta_k := \frac{2\pi k}{m} \) the set \( \{ \phi(\theta_k) | k = -m + 1, ..., m \} \) forms an orthogonal basis of \( \mathbb{C}^{2m} \).

**Proof:** [17, Theorem 7.3.1].

To calculate the eigenvalues of the smoother we will need the equality

\[ [\phi(\theta)]_{l-1} = e^{-i\theta}[\phi(\theta)]_l. \]

First let us apply the Matrix \( L_\tau \)

\[ L_\tau \phi(\theta) = \alpha \phi(\theta) - \mu \begin{pmatrix} 0 & 1 & & \phantom{0} \phantom{0} \\ 1 & 0 & & \phantom{1} \phantom{1} \\ & & \ddots & \phantom{1} \phantom{1} \\ \phantom{0} & \phantom{0} & \phantom{0} \phantom{0} & 1 \\ \phantom{0} & \phantom{0} & \phantom{0} \phantom{0} & \phantom{0} 0 \end{pmatrix} \phi(\theta). \]
As one can see the matrix shifts the entries of the vector forward, thus the result can be expressed by using the previous formula

\[ L_\tau \phi(\theta) = (\alpha_\tau - e^{-i\theta} \mu)\phi(\theta). \]

The application to the smoother is now pretty trivial

\[ S_\tau \phi(\theta) = (I_{2m} - \frac{\omega}{\alpha_\tau} L_\tau) \phi(\theta) = (1 - \omega - \omega e^{-i\theta} \mu)\phi(\theta). \]

The eigenvalues \( \hat{S}_\tau(\theta) \) and \( \hat{L}_\tau(\theta) \) for the smoother and \( L_\tau \) are called "Fourier symbols". In more general settings they are matrices, where they represent the action of the operator on an appropriate eigenspace. This can be observed, as we will now investigate the Fourier symbols of the prolongation, restriction and the two grid cycle.

First let us recall the definition of the coarse grid correction

\[ x^{k+1} = x^k - P_\tau L_\tau^{-1} R_\tau (f - L_\tau x^k). \]

When switching to the error formulation the procedure has the form

\[ \xi^{k+1} = (I - P_\tau L_\tau^{-1} R_\tau L_\tau)\xi^k. \]

One iteration of the full multigrid method can be written as

\[ \xi^{k+1} = S_\nu C_\tau S_\nu \xi^k. \]

We will no longer find eigenvalues for \( C_\tau \), however we will find Fourier symbols for two dimensional eigenspaces. First we must think about the Fourier modes on the coarser grid. As the number of time steps is halved and \( \tau \) is doubled the dimension of the Fourier modes and the number of Fourier frequencies is halved. From now on we will denote with \( \phi^f(\theta) \) a Fourier mode on the fine grid and with \( \phi^{f-1}(\theta) \) one on the coarse grid. Therefore we need to find out which spaces are mapped by the restriction and prolongation. The definition of the restriction matrix is given by

\[ R_\tau := \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}. \]
So what happens if we apply this matrix to a Fourier mode?

\[
[R_r \phi^\ell(\theta)]_n = \phi^\ell(\theta)]_{2n-1} + [\phi^\ell(\theta)]_{2n} = (1 + e^{-i\theta})[\phi^\ell(\theta)]_{2n} = (1 + e^{-i\theta}) e^{i(2n)\theta} = (1 + e^{-i\theta}) e^{in(2\theta)} = (1 + e^{-i\theta}) e^{i\theta} = (1 + e^{-i\theta}) e^{i\theta}.
\]

Here \( n = 1, 2, ..., m \), because the coarse mesh has \( m \) degrees of freedom.

We can conclude that the restriction maps Fourier modes of the fine grid to Fourier modes of the coarse grid. However, this mapping cannot be injective, and therefore we need to determine which modes get mapped to the same mode. This is easily answered by doubling all possible frequencies of the fine mesh and including \( 2\pi \) periodic behavior of the exponential function.

**Definition 5.3 (Space of Harmonics).** For any frequency \( \theta \) contained in the low frequencies \( (-\pi/2, \pi/2] \) we define the frequency shift as

\[
\gamma(\theta) := \begin{cases} 
\theta + \pi & \theta < 0 \\
\theta - \pi & \theta \geq 0,
\end{cases}
\]

and the space of harmonics as

\[
\mathcal{E}_\ell(\theta) := \text{span}\{\phi^\ell(\theta), \phi^\ell(\gamma(\theta))\}.
\]

**Remark 5.4.** This definition is particularly useful, because

\[
2\theta = 2\gamma(\theta)
\]

in a periodic sense. We want to highlight that these always form pairs of a low frequency and a high frequency. Thus \( \mathcal{E}_\ell(\theta) \) is the preimage of \( \phi^\ell-1(\theta) \) wrt. the restriction. If we choose \( (\phi^\ell(\theta), \phi^\ell(\gamma(\theta))) \) as a basis for \( \mathcal{E}_\ell(\theta) \), we get the basis representation of the restriction

\[
R_r(a\phi^\ell(\theta) + b\phi^\ell(\gamma(\theta))) = (1 + e^{-i\theta}) (1 + e^{-i\gamma(\theta)}) \begin{pmatrix} a \\ b \end{pmatrix} \phi^\ell-1(\theta),
\]

which leads to the definition of the symbol for the restriction \( \hat{R}_r(\theta) := 1 + e^{-i\theta} \).

The only missing key is the symbol for the prolongation. For this to work we need to represent the result of the prolongation as a linear combination in the space of harmonics. From this we will get the mapping property and the symbol for the prolongation. First let’s take a look at the action of the prolongation \( P_r \), which is the transposed of the restriction \( R_r \). For \( n = 1, 2, ..., m \)

\[
[P_r \phi^\ell-1(2\theta)]_{2n-1} = [\phi^\ell-1(2\theta)]_n = [\phi^\ell(\theta)]_{2n} = e^{i\theta}[\phi^\ell(\theta)]_{2n-1},
\]

\[
[P_r \phi^\ell-1(2\theta)]_{2n} = [\phi^\ell-1(2\theta)]_n = [\phi^\ell(\theta)]_{2n}.
\]
Therefore we need to find coefficients $a, b$, which satisfy

$$
\begin{align*}
[a\phi^\ell(\theta) + b \phi^\ell(\gamma(\theta))]_n &= e^{i\theta} [\phi^\ell(\theta)]_n \quad n \text{ odd}, \\
[a\phi^\ell(\theta) + b \phi^\ell(\gamma(\theta))]_n &= [\phi^\ell(\theta)]_n \quad n \text{ even}.
\end{align*}
$$

To advance, we need a relation between the two harmonic Fourier modes

$$
[\phi^\ell(\gamma(\theta))]_n = e^{in(\theta - \text{sign}(\theta)\pi)} = e^{in\theta} e^{in\pi} = \begin{cases} 
- [\phi^\ell(\theta)]_n & n \text{ odd} \\
[\phi^\ell(\theta)]_n & n \text{ even}.
\end{cases}
$$

With this knowledge the equation for the linear combination simplifies to

$$
\begin{align*}
a - b &= e^{i\theta}, \\
a + b &= 1,
\end{align*}
$$

with the solution

$$
\begin{align*}
a &= (1 + e^{i\theta})/2 =: \hat{P}_\tau(\theta), \\
b &= (1 - e^{i\theta})/2 = (1 + e^{i\gamma(\theta)})/2 = \hat{P}_\tau(\gamma(\theta)).
\end{align*}
$$

With these insights we can conclude that one multigrid iteration indeed maps errors from a space of harmonics $E_\ell(\theta)$ again into the same space. The symbol for the coarse grid correction now is

$$
\hat{C}_\tau(\theta) = I_2 - \begin{pmatrix} \hat{P}_\tau(\theta) \\
\hat{P}_\tau(\gamma(\theta)) \end{pmatrix} \begin{pmatrix} \hat{L}_2^{-1}(2\theta) & \hat{R}_\tau(\theta) & \hat{R}_\tau(\gamma(\theta)) \end{pmatrix} \begin{pmatrix} \hat{L}_\tau(\theta) & 0 \\
0 & \hat{L}_\tau(\gamma(\theta)) \end{pmatrix}. \tag{5.1}
$$

Finally the symbol for the full multigrid iteration is

$$
\hat{M}_\tau(\theta) = \begin{pmatrix} \hat{S}_\tau(\theta) & 0 \\
0 & \hat{S}_\tau(\gamma(\theta)) \end{pmatrix}^\nu \hat{C}_\tau(\theta) \begin{pmatrix} \hat{S}_\tau(\theta) & 0 \\
0 & \hat{S}_\tau(\gamma(\theta)) \end{pmatrix}^\nu. \tag{5.2}
$$

Now to calculate the spectral radius of the original multigrid method, we just need to calculate the spectral radius of the symbol for each low frequency of the mesh and take the maximum. As we are interested in the general behavior of the algorithm independent of the used mesh, it makes sense to view the symbol’s spectral radius as a continuous function of $\theta$ and search its supremum. Because this analysis is only local and ignores boundary conditions, the frequency 0 needs to be excluded. Additionally if the analyzed multigrid is not working properly, one can check if the used smoother is acting properly on the high frequencies, because coarse grid corrections generally require a smooth error.
5.1.2 Smoother

Remark 5.5 (Generalizations). Usually the basis comprised of the $\phi(\theta_k)$ is not sufficient to deal with the problem at hand (i.e., higher order Runge Kutta schemes, or vector valued variables like in [2], etc.) but for the sake of simplicity we used this for the introduction. Including additional (spatial) dimensions, or a higher order Runge Kutta scheme can be resolved by generalizing the vector to a tensor, where adding another frequency or polynomial degree increases the order of the tensor and the symbols can be combined with the Kronecker product. This is what we will do now, we will replace $\kappa$ with the symbol for the rotated gradient stiffness matrix and $\mu$ with the mass matrix symbol.

The equation discussed in the paper is the two dimensional curl-curl equation

$$\text{curl} \text{ curl } A + \sigma A = J.$$ 

For this analysis we assumed that the spatial problems are exactly solved, as approximate solving cannot perform better and usually isn’t an issue. The needed spatial symbols for $K_h, M_h, R_h, P_h$ were taken from [2].

In [2] the authors worked on an infinite (which is an alternative to periodic meshes) quadrilateral grid, with the degrees of freedom based on the edges, similar to the Nedelec elements. Because of the two directions more than one frequency is needed, denoted with $\theta_x$ and $\theta_y$ in this context. Further we will refer to the previous $\theta$ as $\theta_t$ from now on.

Simply put, the eigenspaces consist of a combination of two Fourier modes, where one exists on the vertical edges and the other one on the horizontal edges. This is also consistent with the vector valued problem. As a consequence the symbols for mass- and stiffness matrix will be 2x2 matrices given by

$$\hat{K}_h(\theta_x, \theta_y) := \begin{pmatrix} 4 \sin(\theta_y/2)^2 & -4 \sin(\theta_x/2) \sin(\theta_y/2) \\ -4 \sin(\theta_x/2) \sin(\theta_y/2) & 4 \sin(\theta_x/2)^2 \end{pmatrix},$$

$$\hat{M}_h(\theta_x, \theta_y) := \sigma h^2 \begin{pmatrix} 2 + \cos(\theta_y) & 0 \\ 0 & 2 + \cos(\theta_x) \end{pmatrix}.$$ 

Substituting $\mu$ for $\hat{M}_h$ and $\kappa$ for $\hat{K}_h$ leads to the symbols

$$\hat{L}_{r,h}(\theta_t, \theta_x, \theta_y) = \hat{A}_{r,h}(\theta_x, \theta_y) - e^{-i\theta_t} \hat{M}_h(\theta_x, \theta_y),$$

$$\hat{S}_{r,h}(\theta_t, \theta_x, \theta_y) = (1 - \omega) I_2 - \omega e^{-i\theta_t} A_{r,h}^{-1}(\theta_x, \theta_y) \hat{M}_h(\theta_x, \theta_y).$$

The definition of $\hat{A}_{r,h}(\theta_x, \theta_y) := \tau \hat{K}_h(\theta_x, \theta_y) + \hat{M}_h(\theta_x, \theta_y)$ is analogous to the definition of $\alpha_x$. With these results we can already analyze the behavior of the smoother for various frequencies.
5.1.3 Two Grid Cycle

Remark 5.6 (Low frequencies). The definition of the low frequencies for multidimensional meshes depends on the used coarse grid correction. If coarsening happens in a direction, then the corresponding frequency must lie within \((-\pi/2, \pi/2]\), otherwise there are no restrictions. For example if we only apply coarsening in time, then the low frequencies would be the middle part in a sandwich. As always the high frequencies are the complement of the low frequencies.

For a more elegant expression of the prolongation and restriction the authors of [2] decided, that the spatial frequencies must lie within \((-\pi/2, 3\pi/2]\) in contrast to our definition. This implies that their shift operator is defined as \(\gamma(\theta_{x/y}) := \theta_{x/y} + \pi\), but there is practically no difference due to the periodic behavior of the exponential function.

To start we will introduce the symbols for the multigrid with coarsening in time only. The results are pretty much the same as in (5.1) and (5.2) only with block matrices. The new prolongation and restriction are a Kronecker product of the old versions with a 2x2 Identity matrix, resulting in

\[
\begin{pmatrix}
\hat{P}_\tau(\theta_t) & 0 \\
0 & \hat{P}_\tau(\theta_t)
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
\hat{R}_\tau(\theta_t) & 0 & \hat{R}_\tau(\gamma(\theta_t)) & 0 \\
0 & \hat{R}_\tau(\theta_t) & 0 & \hat{R}_\tau(\gamma(\theta_t))
\end{pmatrix}.
\]

As we can see, the additional two dimensional spatial modes doubled the matrix dimensions to a 4x4 matrix. In the next step we will discuss the symbol for the multigrid with full space-time coarsening. Then we will have two dimensional spatial modes, two harmonic time frequencies and four harmonic spatial frequencies, resulting in a 16x16 symbol. Therefore it will be a bit tricky to handle this in the given space.

The spatial restriction for one frequency is given by

\[
\hat{R}_h(\theta_x, \theta_y) := \begin{pmatrix}
-2\cos(\theta_x)\cos(\theta_y)\operatorname{sign}(\theta_x - \pi/2) & 0 \\
0 & -2\cos(\theta_y)\cos(\theta_x)\operatorname{sign}(\theta_y - \pi/2)
\end{pmatrix}.
\]

The sign function determines whether it is a high or a low frequency.

The final restriction is now given by

\[
\begin{pmatrix}
\hat{R}_h(\theta_x, \theta_y) & \hat{R}_h(\theta_x, \gamma(\theta_y)) & \hat{R}_h(\gamma(\theta_x), \theta_y) & \hat{R}_h(\gamma(\theta_x), \gamma(\theta_y))
\end{pmatrix} \otimes
\begin{pmatrix}
\hat{R}_\tau(\theta_t) & \hat{R}_\tau(\gamma(\theta_t))
\end{pmatrix}.
\]

The prolongation works analogously \((\hat{P}_h = \hat{R}_h/4)\), the smoother and the operator on the fine mesh become block diagonal matrices which iterate over all possible combinations of shifted and not shifted frequencies in a hierarchy compatible to the presented restriction (first \(\theta_t\), then \(\theta_x\) and finally \(\theta_y\)). The inverse operator on the coarse grid is
just a 2x2 matrix with all frequencies doubled and appropriate \( \tau, h \) arguments.

With all symbols defined we are finally able to numerically analyze the behavior of our time parallel method for the rotated gradient equation in Mathematica \([7]\).

### 5.2 Intermediate Results

First we will take a look at the results for the smoother in Figure 5.1. For the coefficients we choose the values \( h = \sigma = \mu = 1 \), for \( \tau \) we also choose 1, if it is not varied. The result will always look the same for the Eddy-current equation, regardless of spatial frequencies or coefficients. The striking property of this smoother is the **lack of smoothing for high spatial frequencies**. In contrast the same smoother for the 1D heat equation achieves spatial smoothing for coefficients fulfilling a coarsening rule, as seen in Figure 5.2.

With this in mind, we cannot expect a coarse grid correction with a coarser spatial mesh to have any effect. This becomes apparent in the results for the two grid cycle. While the method with coarsening only in time (Figure 5.3) achieves convergence, the full coarsening method (Figure 5.4) does not.

**Conclusion:** The version with the pure time multigrid works as expected. But the current smoother cannot achieve smoothing of high spatial frequencies for any combi-
Figure 5.2: LFA of the smoother for the **1D heat equation** showing convergence factors. Spatial smoothing happens because the coefficients fulfill the coarsening rule.

Figure 5.3: LFA of the two-grid-cycle for the Eddy-current equation with coarsening only in time. A convergence rate of 0.5 is achieved.
CHAPTER 5. LOCAL FOURIER ANALYSIS

Figure 5.4: LFA of the two-grid-cycle for the Eddy-current equation with coarsening in time and space. No convergence, the coarse grid correction has no effect.

ation of $h$, $\tau$, $\mu$, $\sigma$. Therefore a subspace correction with full space-time coarsening has no effect. If an altered smoother could be found which restores the spatial smoothing properties under certain conditions, then spatial coarsening would be possible again.

5.3 A Hybrid Smoother

Given the knowledge, that the smoother fails to smooth high spatial frequencies and that consequentially spatial coarsening is impossible, we try to find the underlying reasons for the behavior of the smoother. Then we proceed to search for a modification, which fixes the discussed problems.

5.3.1 Helmholtz Splitting

In [2] an alternative to the basis representing the two directions of the grid is proposed, namely the Helmholtz splitting in Fourier Space. The discrete gradient operator $G$ maps nodal degrees of freedom onto the presented edge degrees of freedom, while $C$ is the discrete curl operator. Just like the differential operators ($\text{curl } \nabla = 0$) the discrete operators satisfy $C \circ G = 0$. The ranges of $G$ and $C^\top$ compose the whole space and all previously given symbols can be transformed with the matrix

$$H(\theta_x, \theta_y) := \begin{pmatrix} \sin(\theta_x/2) & \sin(\theta_y/2) \\ \sin(\theta_y/2) & -\sin(\theta_x/2) \end{pmatrix}.$$

The first coordinate now corresponds to the image of the gradient and is called nodal part, while the second coordinate is called solenoidal part. For example the trans-
Figure 5.5: Absolute value of the smoother mapping from the solenoidal part onto itself. $\theta_t = 0.05$, for high time frequencies the plot is at most 0.5 everywhere. $h = 1, \tau = 1$

The stiffness matrix symbol has the form

$$\hat{K}_{h,\psi} := \begin{pmatrix} 0 & 0 \\ 0 & 4(\sin(\theta_x/2)^2 + \sin(\theta_y/2)^2) \end{pmatrix}.$$

This representation is consistent with $C \circ G = 0$, as the nodal part is the kernel of $\hat{K}_{h,\psi}$.

We can now analyze the action of our smoother on the nodal and solenoidal parts, by plotting the absolute value of each part of its symbol. The nodal part does not influence the solenoidal part during smoothing, because the corresponding entry is always 0. Figure 5.6 explains, why the smoother shows no spatial smoothing properties at all, which we can now attribute to the nodal part.

Figure 5.5 shows us, that the solenoidal part indeed shows good spatial smoothing properties. As it could be expected the plot reacts to the coefficients according to the coarsening rule. This means that the "hill" in the plot expands as the left side of (4.4) gets smaller. The solenoidal part also influences the nodal part, but as seen in Figure 5.7 the effect is always comparably small.
Chapter 5. Local Fourier Analysis

Figure 5.6: Absolute value of the smoother mapping from the nodal part onto itself. Completely independent of $\theta_x$, $\theta_y$ and coefficients.

Figure 5.7: Absolute value of the smoother mapping from the solenoidal part onto the nodal part. This so called aliasing effect is bounded by 0.2.
5.3.2 Hybrid smoother

As now established the smoother fails to smooth the spatial component of the nodal part. However [2] presents the possibility of solving an auxiliary nodal system to influence the nodal part. The general method works as follows:

- Presmoothing
- Restriction \( f_n = G^\top(f - L_{r,h}x) \)
- Smoothing \( L_n x_n = f_n \)
- Prolongation \( x = x + Gx_n \)
- Postsmoothing

The auxiliary nodal system is usually chosen as \( L_n := (G^\top \otimes I)L_{r,h}(G \otimes I) \). So what is now \( L_n \) actually in our case? As elaborated in [2, 6.2.] the stiffness matrix vanishes and the mass matrix turns into a stiffness matrix for a scaled Poisson equation. Therefore our system is now

\[
L_n x_n = K_n \otimes \begin{pmatrix}
1 & 1 & \ldots & 1 \\
-1 & 1 & \ldots & 1 \\
    &   & \ddots &   \\
    &   &      & 1 \\
-1 & 1 & \ldots & 1
\end{pmatrix} x_n = f_n.
\]

This system can be not just smoothed, but even solved, by applying \( K_n^{-1} \) to the right side and directly solving the remaining equation

\[
I \otimes \begin{pmatrix}
1 & 1 & \ldots & 1 \\
-1 & 1 & \ldots & 1 \\
    &   & \ddots &   \\
    &   &      & 1 \\
-1 & 1 & \ldots & 1
\end{pmatrix} x_n = \begin{pmatrix}
K_n^{-1}f_1^n \\
K_n^{-1}f_2^n \\
\vdots \\
K_n^{-1}f_m^n
\end{pmatrix}.
\]

The solution to this equation is surprisingly simple

\[
x_i^n = \sum_{j=1}^i K_n^{-1}f_j^n.
\]  \hspace{1cm} (5.3)

**Remark 5.7 (Parallel Solving).** Solving the equation in parallel is not that easy. The application of the inverse can be easily executed in parallel, whereas summing up the right side can be trickier. However the sequential approach might even be fast enough given that solving spatial linear systems is not required in this stage. To accelerate the sequential method all processors could calculate the sum over their assigned time steps beforehand. When a processor receives the sum from the previous processor, it instantly adds the sum over its own time steps and sends the result to the next processor. Finally it can compute the final result for each time step.
Figure 5.8: Illustration of the parallel summing algorithm to solve the auxiliary problem.

If an even more parallel method is desired, a "divide and conquer"-scheme is possible. To be precise in the first step each processor calculates the sum (5.3) over its own time steps, assigning each time step an intermediate solution. At this point each processor needs the sum over all previous time steps for the final solution, which will be called local initial value here. In the next step each processor with an even number passes its local sum to the next odd processor, which adds this value to its initial value and to its total sum. This way the odd processor holds the sum over its own and the previous processors time steps. At this point we can coarsen the grid and repeat the same procedure. The only difference is that the communicated sums need to be added to all local initial values in a group. To finalize each processor adds its local initial value to its time steps. An illustration of this algorithm can be seen in Figure 5.8.

**Remark 5.8 (Time Periodic Case).** While this smoother can be easily implemented for the initial value problem, the auxiliary system is singular for the time-periodic boundary conditions. A solution can only exist, if the sum over the right hand side is 0. Fortunately this is true for both parts of the residuum. For the part produced by the solution it holds

\[(G^T \otimes I) L_{r,h} x = \left(\tau G^T K_h \otimes I + (G^T M_h) \otimes \begin{pmatrix} 1 & 1 & -1 \\ -1 & 1 & \ddots \\ \ddots & \ddots & \ddots \\ 1 & -1 & 1 \end{pmatrix}\right) x.
\]

The remaining matrix guarantees, that the sum over the result is a 0-vector.

The other condition is \(\sum_{i=1}^{m} G^T [f]_i = 0\), which corresponds to the continuous condition

\[0 = \int_0^T \text{div} \frac{J_i}{\partial t}(x,t) dt = \int_0^T \text{div} \sigma \frac{\partial E}{\partial t}(x,t) dt = \sigma \text{div} (E(x,T) - E(x,0))\]

This is guaranteed by the periodic boundary condition. Therefore we can expect, that the sum over the restricted residuum is 0, apart from some discretization errors. To achieve a unique solution a certain gauge might needed to be imposed on the solution of the auxiliary problem. Therefore this method is also viable for the time periodic case.
This is very good news, because by solving the auxiliary nodal system, we can eliminate the nodal error component apart from aliasing of the solenoidal part. As derived in [2] the symbol for the auxiliary smoother reads

$$I - G_{\psi} K_n^{-1} G_{\psi}^T K_{\psi} = \begin{pmatrix} 0 & * \\ 0 & 1 \end{pmatrix}.$$ 

The aliasing term $*$, is bounded by 0.3 and appears to have little impact on the overall performance. A LFA plot of the composition of the original smoother and the auxiliary "smoother" can be seen in Figure 5.9.

Different combinations of the original smoother and the auxiliary smoother can now be used for the two grid cycles. But the prolongation and restriction are not scaled in the Helmholtz basis, therefore $P_{\psi} L_{\tau,h,\psi} R_{\psi}$ has to be used instead of $L_{2\tau,2h,\psi}$.

In Figure 5.11 to 5.15 the maximal contraction can be seen as a function of $\tau$. As an abbreviation we refer to the auxiliary smoother as $A$ and to the standard smoother as $S$. AS for example denotes that the pre-smoother is an application of $S$ followed by $A$, which is in an order from right to left like matrix multiplication. For the post-smoother the order is always reversed.

To find the maximal spectral radius, 80 evaluations per low frequency domain were performed, resulting in a total of $80^3$ spectral radius evaluations of 16x16 or larger matrices. This was performed for 150 different values of $\tau$ for each multigrid method.
Figure 5.10: The convergence rate as a function of $\tau$ for the SA smoother. Low accuracy was used, because no significant differences to the AS smoother were detected.

All plots were evaluated with a lower accuracy beforehand to get an impression of the behavior.

In [11] higher order time integration schemes were used, which could be identified with a polynomial degree. Although not introduced in this thesis, we also present results for degrees higher than 0 (Euler scheme).

We can conclude that the actual convergence rate of coarsening in space and time is always the maximum of coarsening only in time and coarsening only in space. Therefore the value of $\tau$ where the two lines cross is the ideal $\lambda_{\text{crit}}$ for the method. As we can see in all cases $\lambda_{\text{crit}} = 1$ is a safe bet, because choosing a value slightly below the ideal $\lambda_{\text{crit}}$ can severely impact the convergence rate.

Another conclusion is, that smoothing at least two times with the standard smoother is always beneficial and squares the convergence factor of the used method. If only coarsening in time is used, then smoothing more than twice (SAS) is not useful, but for very high $\tau$ or coarsening only in space smoothing more often indeed pays off (Figure 5.13).
Figure 5.11: The convergence rate as a function of $\tau$ for the AS smoother. High accuracy was used.

Figure 5.12: The convergence rate as a function of $\tau$ for the SAS smoother. High accuracy was used.
Figure 5.13: The convergence rate as a function of $\tau$ for the SSASS smoother. High accuracy was used. We can see, that for high $\tau$ all methods achieve a convergence rate of $0.5^8$ because the solenoidal parts gets dampened by 0.5 for each standard smoother. This observation can be applied to all cases.

Figure 5.14: The convergence rate as a function of $\tau$ for the AS smoother with polynomial degree 1. High accuracy was used.
5.4 Spatial Smoothing Explained

Over the course of this chapter two questions might arise:

- Why does spatial smoothing happen?
- Why is there no spatial smoothing for the kernel of the operator?

In particular, we are interested in the case \( \theta_t = 0 \), because all other cases behave better than it. To answer these questions, we need some analytical results [11, p. 91] for the time dependent test equation

\[ u' + \lambda u = f. \]

For the implicit Euler scheme and real, positive \( \lambda \) the squared symbol of the smoother is equal to

\[ \hat{S}(\omega, \alpha, \theta_t)^2 = (1 - \omega)^2 + 2\omega(1 - \omega)\alpha \cos(\theta_t) + \alpha^2 \omega^2, \]

\[ \alpha := \frac{1}{1 + \lambda \tau}. \]

If we focus on the spatial smoothing, we need to set the relaxation parameter \( \omega = 0.5 \) and the frequency of interest is \( \theta_t = 0 \). Then the expression simplifies to

\[ \hat{S} = \left( \frac{1}{4}(1 + \alpha)^2 \right)^{1/2} = \frac{1 + \alpha}{2}. \]

The symbol of \( M_h^{-1}K_h \) can be inserted for \( \lambda \) to gain results of the spatial smoothing behavior, as done in Figure 5.16. It can be observed, that the bigger the eigenvalue of
the stiffness matrix, the better the corresponding eigenvector gets dampened, which happen to be the high frequencies. For the heat-equation the kernel of the stiffness matrix is the 0 frequency, which poses no problem for the coarse grid correction. However for the Eddy-current equation the kernel is present over all frequencies, and therefore they do not get dampened at all.

In Figure 5.16 we can also observe that $\tau$ (and all other coefficients in the coarsening rule) can directly influence the spatial smoothing behavior.

Figure 5.16: Contraction of the solenoidal part as a function of $\theta_x$ for various $\tau$. Parameters are $\theta_y = 0, \theta_t = 0, h = 1$. 
Chapter 6

Numerical Experiments

In this chapter we will present and interpret results from numerical experiments. First we will specify, how the program was implemented, then we will investigate, how the rules for space-time coarsening carry over to more general settings. Then we will present, how the program scales and performs compared to the usual approach of forward solving. At last pictures of the actual simulation are presented.

6.1 Implementation

To create a program which can execute the time-multigrid algorithm parallel in space and time, we need two key components:

A library that handles spatial issues in parallel, such as:

- Assemble $K_h$, $M_h$
- Assemble linear forms
- Solve spatial linear systems
- Multigrid mesh management
- Prolongation and restriction operators

The second component was a template like algorithm, which uses the aforementioned components to execute the time related tasks parallel in time.

The first part was handled by MFEM [8] in combination with HYPRE [6], while the second part was directly implemented by modifying and expanding preexisting code from [11]. A fully C++ template based algorithm would be desirable, but this is not feasible due to divergent definitions of vector operations and similar issues. Nonetheless the adaption was possible without disruptive changes to the algorithms code. Additionally GLVIS [5] was used to illustrate the results of the simulation as
seen in Chapter 6.4.

The final program is capable of executing the described algorithm including Space-Time coarsening. It is also parallel in time and space, meaning that the time steps are divided between several groups of processors, who work in parallel to solve spatial problems. Because the transfer-operators between meshes only worked for Nedelec elements of order 1, we were restrained to using only those.

6.2 Space-Time coarsening

To start off we will conduct experiments for the 3D-heat equation to test how well the expanded coarsening-rule works for several more general settings. The second part is concerned with the behavior of space-time coarsening for the Eddy-current equation using the standard smoother. The tests from the first part were repeated and additional ones were added. As the results of the local Fourier analysis suggest, we can expect coarse grid corrections with a coarser spatial mesh to be useless, unless a hybrid smoother was used.

All performed tests try to achieve a relative reduction of the residual to a certain tolerance, but if the number of iterations exceeds 50 (later 100) the test cancels, which will be denoted by a X. For all tests we used two pre- and post-smoothing iterations.

All tests vary $\lambda_{\text{crit}}$ to compare the behavior of all possible coarsening strategies. In the following tests the coarsening strategy will be the same for either each of the columns or the diagonals, which depends on the influence of the other modified parameter. In the presented tables parts may be empty, because the most aggressive strategy or no coarsening at all already happens for less extreme values of $\lambda_{\text{crit}}$, so these cases would be duplicates. Additionally this hints how the parameters influence the strategy.

The tests will be conducted either for the induction furnace case (Chapter 2.1) or a mesh consisting of two tetrahedron, which are uniformly refined once. For the two-tetrahedron test case we used different material parameters for each of the two coarse tetrahedrons.

To exclude unwanted effects, no spatial parallelization was applied, whereas the time parallelization has no impact on the algorithm and was used when needed.

6.2.1 Heat-equation setting

In this section we heuristically verify the proposed generalization of the coarsening rule (4.4). The heat equation is given by

$$\alpha(x) \frac{\partial u}{\partial t}(x, t) - \text{div}_x(\beta(x) \nabla_x u(x, t)) = 0.$$
CHAPTER 6. NUMERICAL EXPERIMENTS

Test case 1

For this case we set \( \alpha = \beta = 1 \) for one of the two coarse tetrahedron and \( \alpha = \beta = m \) on the other one. This way we ensure that the ratio \( \frac{\beta}{\alpha} \) is constant over all elements, so as a consequence the coarsening strategy is completely independent of the variable \( m \). The intention of this test is to show that the coarsening rule can be applied locally, whereas taking the global extrema for \( \alpha \) and \( \beta \) would result in less spatial coarsening.

\[
\begin{array}{cccccccccccc}
 m/\lambda_{\text{crit}} & 2^{-7} & 2^{-6} & 2^{-5} & 2^{-4} & 2^{-3} & 2^{-2} & 2^{-1} & 2^{0} & 2^{1} & 2^{2} & 2^{3} \\
10^{-9} & 27 & 26 & 25 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^{-6} & 28 & 26 & 24 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^{-3} & 27 & 27 & 24 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
0.01 & 28 & 28 & 24 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
0.1 & 28 & 28 & 24 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
1 & 30 & 30 & 27 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10 & 29 & 30 & 28 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
100 & 29 & 29 & 27 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^3 & 30 & 29 & 27 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^6 & 30 & 28 & 27 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^9 & 29 & 28 & 28 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
\end{array}
\]

Table 6.1: Iterations for varying \( m \) and \( \lambda_{\text{crit}} \). The coarsening behavior only depends on \( \lambda_{\text{crit}} \). Solver tolerance \( 10^{-5} \) was used

Test case 2

In this test case \( \alpha = 1 \) holds globally, while \( \beta = 1 \) on one coarse tetrahedron and \( \beta \) is varied on the other. This implies that if \( m \) is greater than 1, the coarsening strategy is unaffected.

\[
\begin{array}{cccccccccccc}
 \beta_{T2}/\lambda_{\text{crit}} & 2^{-7} & 2^{-6} & 2^{-5} & 2^{-4} & 2^{-3} & 2^{-2} & 2^{-1} & 2^{0} & 2^{1} & 2^{2} & 2^{3} \\
10^0 & 32 & 31 & 26 & 13 & 13 & 13 & 13 & 13 & 13 & 13 & 13 \\
10^1 & 27 & 26 & 23 & 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\
10^2 & 20 & 20 & 18 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10^3 & 15 & 15 & 13 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
10^4 & 9 & 9 & 8 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
10^5 & 7 & 7 & 7 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
10^6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
\end{array}
\]

Table 6.2: Iterations for varying \( \beta_{T2} \) and \( \lambda_{\text{crit}} \). The coarsening behavior only depends on \( \lambda_{\text{crit}} \). Solver tolerance \( 10^{-5} \) was used
As this test apparently implied that the coarsening strategy does not matter for extreme cases, we lowered the tolerance to compensate for the discrepancy between the residual and the actual error. Otherwise the part of the domain with high $\beta$ could dominate the residual and after vanishing quickly, we would not notice if the solution converged on the other domain at all.

To handle this issue, we lowered the tolerance to $10^{-10}$ and additionally we also took a look at smaller values for $\beta$.

<table>
<thead>
<tr>
<th>$\beta T_2 / \lambda$</th>
<th>$2^{-7}$</th>
<th>$2^{-6}$</th>
<th>$2^{-5}$</th>
<th>$2^{-4}$</th>
<th>$2^{-3}$</th>
<th>$2^{-2}$</th>
<th>$2^{-1}$</th>
<th>$2^{0}$</th>
<th>$2^{1}$</th>
<th>$2^{2}$</th>
<th>$2^{3}$</th>
<th>$2^{4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-7}$</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-6}$</td>
<td>28</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-5}$</td>
<td>39</td>
<td>24</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-4}$</td>
<td>x</td>
<td>32</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-3}$</td>
<td>x</td>
<td>42</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-2}$</td>
<td>x</td>
<td>x</td>
<td>25</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{-1}$</td>
<td>x</td>
<td>x</td>
<td>32</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^0$</td>
<td>x</td>
<td>x</td>
<td>49</td>
<td>23</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^1$</td>
<td>x</td>
<td>x</td>
<td>46</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^2$</td>
<td>49</td>
<td>x</td>
<td>42</td>
<td>20</td>
<td>19</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^3$</td>
<td>44</td>
<td>45</td>
<td>38</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>35</td>
<td>37</td>
<td>29</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^5$</td>
<td>28</td>
<td>28</td>
<td>24</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^6$</td>
<td>21</td>
<td>23</td>
<td>18</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.3: Iterations for varying $\beta T_2$ and $\lambda_{crit}$. The coarsening behavior only depends on $\lambda_{crit}$ in the lower half and is constant along diagonals in the upper half. Solver tolerance $10^{-10}$ was used.

The test delivers the expected results for the lower half, where the convergence is unsatisfying if $\lambda_{crit}$ is too small. However the upper half shows that more aggressive coarsening would be possible. At least the rule guarantees convergence.

The results clearly indicate, that the rule is correct and that the choice of $\lambda_{crit}$ may also depend on further factors like the spatial dimension and not only on the time discretization method. For example the Euler scheme in 1D requires $\lambda_{crit} = \sqrt{2}/3$ [11, Remark 4.3.16].

### 6.2.2 Eddy-current setting

The same tests as for the heat equation were repeated for the Eddy-current equation

$$\sigma(x) \frac{\partial E}{\partial t}(x,t) + \text{curl}_x(\mu^{-1}(x)\text{curl}_x E(x,t)) = 0.$$
The iterative solver for the space problems was configured to accurately solve the equation and on all levels no accuracy problems could be detected. For the local Fourier analysis an exact spatial solver was also assumed, so the results can be better compared and we can rule out unwanted effects. The tests were performed with the standard smoother (4.1) and not with the hybrid-smoother.

Test case induction furnace

The following test was conducted on the induction furnace model problem, with a multiplier $\eta$ applied to $\mu_{\text{iron}}^{-1}$. As the ratio $\sigma \mu$ for iron dominates in the coarsening rule, any modification to those parameters directly influences the chosen strategy.

<table>
<thead>
<tr>
<th>$\eta/\lambda$</th>
<th>$2^{-5}$</th>
<th>$2^{-4}$</th>
<th>$2^{-3}$</th>
<th>$2^{-2}$</th>
<th>$2^{-1}$</th>
<th>$2^0$</th>
<th>$2^1$</th>
<th>$2^2$</th>
<th>$2^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-5}$</td>
<td>18</td>
<td>13</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$2^{-4}$</td>
<td>31</td>
<td>17</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>$2^{-3}$</td>
<td>x</td>
<td>30</td>
<td>16</td>
<td>11</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$2^{-2}$</td>
<td>x</td>
<td>x</td>
<td>30</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$2^{-1}$</td>
<td>x</td>
<td>x</td>
<td>30</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$2^0$</td>
<td>x</td>
<td>x</td>
<td>30</td>
<td>16</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$2^1$</td>
<td>x</td>
<td>x</td>
<td>31</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>$2^2$</td>
<td>x</td>
<td>x</td>
<td>31</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>$2^3$</td>
<td>x</td>
<td>x</td>
<td>31</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 6.4: Iterations for varying permeability of the iron part and $\lambda_{\text{crit}}$. The coarsening behavior is constant along the diagonals. Solver tolerance $10^{-6}$ was used.

The test from Table 6.4 shows that the used coarsening strategy doesn’t matter. Besides that the problem becomes easier to solve with increased $\mu^{-1}$, the convergence is utterly unaffected by anything but the coarsening behavior. As a side note it is pretty curious that our unaltered model problem ($\eta = 1$) behaves as if the coarsening rule was alright. So without systematic testing wrong conclusions would be drawn. Testing is the future!

Test case 1&2

For comparison we performed the same tests as for the heat equation. This time the maximum iterations were raised to 100 and the tolerance is always $10^{-10}$. 
CHAPTER 6. NUMERICAL EXPERIMENTS

Table 6.5: Iterations for varying $m$ and $\lambda_{\text{crit}}$. The coarsening behavior only depends on $\lambda_{\text{crit}}$. Solver tolerance $10^{-10}$ was used.

The results shown in Table 6.5 are inconclusive. It appears as if a $\lambda_{\text{crit}}$ of about 2 to 4 is needed, which is suspiciously higher than in the heat-equation case.

Table 6.6: Iterations for varying $\mu^{-1}$ and $\lambda_{\text{crit}}$. The coarsening behavior only depends on $\lambda_{\text{crit}}$ for $\mu^{-1} \geq 1$, while in the upper half it is constant along the diagonal. Solver tolerance $10^{-10}$ was used.

The results of the second test in Table 6.6 are unambiguous. The behavior on the
upper half is utterly uninfluenced by $\lambda_{\text{crit}}$ depending solely on the chosen coarsening strategy. We can conclude that this test shows that our coarsening rule does not work for standard smoother applied to the Eddy-current equation.

**Test case 3**

For the next test we modified the equation in the following way:

$$\alpha(x) \frac{\partial E}{\partial t}(x,t) + \text{curl}_x(\beta(x)\text{curl}_x E(x,t)) + \gamma E = 0. \quad (6.1)$$

We were hoping to fix the problem by removing the kernel of the space operator, but the results show, that this manipulation only improves the performance of the smoother.

<table>
<thead>
<tr>
<th>$\gamma/\lambda_{\text{crit}}$</th>
<th>Smoother</th>
<th>$2^1$</th>
<th>$2^2$</th>
<th>$2^3$</th>
<th>$2^4$</th>
<th>$2^5$</th>
<th>$2^6$</th>
<th>$2^7$</th>
<th>$2^8$</th>
<th>$2^9$</th>
<th>$2^{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>19</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>0.5</td>
<td>29</td>
<td>29</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>0.2</td>
<td>57</td>
<td>57</td>
<td>22</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>0.1</td>
<td>x</td>
<td>x</td>
<td>37</td>
<td>18</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>0.05</td>
<td>x</td>
<td>x</td>
<td>67</td>
<td>31</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>0.02</td>
<td>x</td>
<td>x</td>
<td>100</td>
<td>56</td>
<td>31</td>
<td>18</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>0.01</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>65</td>
<td>39</td>
<td>24</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>0.005</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>70</td>
<td>43</td>
<td>28</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>0.002</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>72</td>
<td>46</td>
<td>31</td>
<td>22</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>74</td>
<td>49</td>
<td>33</td>
<td>25</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 6.7: Iterations for varying regularization. For comparison a column for the behavior of the pure smoother without coarse grid corrections was included. Solver tolerance $10^{-10}$ was used.

In summary the numerical results suggest, that for the standard smoother applied to the Eddy-current equation a coarse grid correction with coarsening in time and coarsening in space has no impact on convergence, whereas just coarsening in time always yields the desired convergence.

These observations confirm the results of the local Fourier analysis from Chapter 5.

### 6.3 Scaling results

In Table 6.8 we observe the scaling properties of the parallel algorithm without spatial coarsening. For comparison the run times of the sequential algorithm are present. The sequential algorithm could only achieve a residual reduction of $\approx 5 \times 10^{-6}$, therefore
CHAPTER 6. NUMERICAL EXPERIMENTS

Table 6.8: Run time in seconds varying numbers of processors in time and space for solving the induction furnace problem with 301388 DoF in space and 512 time steps. Tolerance was $10^{-6}$, the sequential run times were rescaled to the tolerance.

<table>
<thead>
<tr>
<th>Time/Space</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>3201</td>
<td>1636</td>
<td>857</td>
<td>452</td>
<td>247</td>
<td>187</td>
</tr>
<tr>
<td>1</td>
<td>58664</td>
<td>30542</td>
<td>15602</td>
<td>8100</td>
<td>5309</td>
<td>3124</td>
</tr>
<tr>
<td>2</td>
<td>29282</td>
<td>16076</td>
<td>7904</td>
<td>4583</td>
<td>2680</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16807</td>
<td>8420</td>
<td>4563</td>
<td>2323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9901</td>
<td>5423</td>
<td>2794</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>5872</td>
<td>2794</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>3119</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

we rescaled the results by $\log(10^{-6})/\log(Reduction)$.

The multigrid method still has unused potential through spatial coarsening and further optimizations discussed later. Even though we can see, that we can already beat the sequential approach by using 32 times more processors.

### 6.4 Visualized Results

For simulation of the model problem, we choose to apply an oscillating, evenly distributed current to the copper coil. The results as seen in Figure 6.1 and Figure 6.2 show intense surface fluxes on the boundary of the iron core and the field vanishing inside the iron core. These intense surface fluxes would cause the iron to eventually melt.

For the visualized simulation of the induction furnace model problem we used a sequential implicit Euler scheme with a time step size of 0.05s and a PCG-solver with a comparably low accuracy of $10^{-6}$ per time step. The original mesh was refined uniformly three times, which resulted in 2399960 degrees of freedom. The pictures were taken at $t = 1$s, where the E-field was at its maximum.
Figure 6.1: Cross section through the induction furnace. The regions of the copper coil and the iron core are outlined in the picture. Intense surface fluxes on the iron core are visible.
Figure 6.2: Fluxes on the surface of the geometry.
Chapter 7

Conclusion & Outlook

7.1 Conclusion

We successfully implemented a time parallel scheme for the Eddy-current equation, which has good convergence properties. We further demonstrated that combining code for spatial utilities with the code for the abstract time multigrid was somewhat frictionless and it could be simplified to a fully interface or template based framework. For example once the space-time multigrid was set up for the Eddy-current equation, changing to the heat equation only required few trivial changes.

We consequentially focused on rules for space-time coarsening and verified a versatile local coarsening rule for the heat equation. This coarsening rule failed when applied to the standard procedure for Eddy-current equation.

As a helpful tool we used the local Fourier analysis, which helped identify the issues with the smoother for the Eddy-current equation. The main advantages are its ease of use and the immediate visualization of how procedures affect the different frequencies. We further used the LFA to construct an auxiliary space smoother, which restored the possibility of a space-time multigrid according to the coarsening rule.

The resulting space- and time-parallel scheme without spatial coarsening can already beat the time-sequential approach with 32 times the processors and achieves a higher accuracy. With the inclusion of the hybrid smoother and tweaking a few parameters, we could expect even better performance.

7.2 Outlook

We conclude this thesis with a wide array of loose ends that can be picked up.

First of all, we need numerical experiments to implement and verify the proposed hybrid smoother.
The presented method relied heavily on exactly solving spatial problems. We have seen for the heat equation, that a few preconditioned CG iterations are sufficient most of the time. Also for the auxiliary problem only a scheme to smooth high spatial frequencies in the first step might be sufficient, because smoothing high spatial frequencies is the only task of the auxiliary smoother. Many preconditioners for the curl $A_\tau$ matrix also use auxiliary methods to smooth the nodal part, but this might be obsolete due to our own auxiliary smoother. Therefore a cheaper preconditioner possibly could be used.

A local Fourier analysis could be applied to several more cases to compute an appropriate $\lambda_{crit}$ for different scenarios like higher dimensional heat equations.

When this algorithm is applied to the equation

$$\frac{\partial u}{\partial t} - \nabla \text{div} u = f,$$

then we will face the same challenges with the non-trivial kernel of the div operator. The image of the curl operator will show no spatial coarsening and will require separate smoothing.

Applying the time-parallel method to differential algebraic equations might be interesting. For example if we set the conductivity of vacuum to 0 instead of 1 this would have been a DAE.

Finally the method itself needs to become more flexible in terms of the time mesh to compete with the sequential approach, which can utilize adaptive time steps. Due to the local nature of the coarsening rule, it might be possible to coarsen the spatial domains separated by a domain decomposition method selectively. This would result in a coarsening strategy for each domain, so the iron in the induction furnace model would not slow down the coarsening of the other domains.
Bibliography


[19] English Wikipedia links:
Eidesstattliche Erklärung

Ich, Martin Schwalsberger, erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.
Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Linz, September 2018

__________________________
Martin Schwalsberger
Curriculum Vitae

Name: Martin Schwalsberger

Nationality: Austria

Date of Birth: 16. August, 1993

Place of Birth: Kirchdorf, Austria

Education:

2000–2004  Volksschule Kematen/Krems  
(elementary school)

2004–2012  Stiftsgymnasium Kremsmünster  
(secondary school)

2013–2017  Bachelor-studies in Technical Mathematics,  
Johannes Kepler University Linz

2017–2018  Master-studies in Industrial Mathematics,  
Johannes Kepler University Linz

Special Activities:

June–July 2016  Internship at Engineering Software Steyr (ESS), Steyr