A Parallel and Adaptive Space-Time Method for Maxwell's Equations

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Mathematik des
Karlsruher Institut für Technologie (KIT)
genehmigte

DISSERTATION

von
Dipl.-Math. techn.
Stefan Matthias Findeisen
aus Calw

Tag der mündlichen Prüfung: 6. Juli 2016

Referent: Prof. Dr. Christian Wieners
Korreferent: Prof. Dr. Willy Dörfler

Für Brigitte, Matthias und Florian

Danksagungen

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter und Doktorand am Institut für Angewandte und Numerische Mathematik am Karlsruher Institut für Technologie in den Jahren 2012 bis 2016. Mein Forschungsvorhaben war Bestandteil des DFG Graduiertenkollegs 1294 „Analysis, Simulation und Design nanotechnologischer Prozesse" und des DFG Sonderforschungsbereichs 1173 „Wellenphänomene: Analysis und Numerik".

Während meiner Zeit als Doktorand wurde ich von vielen Menschen unterstützt. Bei allen möchte ich mich an dieser Stelle bedanken. Mein größter Dank geht an meine beiden Betreuer Prof. Dr. Willy Dörfler und Prof. Dr. Christian Wieners. Ohne Ihr großes Engagement wäre diese Arbeit wohl kaum zustande gekommen. Danke, dass Sie trotz stressiger und anspruchsvoller Zeiten als Dekane und DFG-Antragssteller immer Zeit für mich hatten.

Darüber hinaus möchte ich mich bei allen ehemaligen und aktuell tätigen Kollegen und Mitarbeitern des Instituts, des Graduiertenkollegs 1294 und des Sonderforschungsbereichs 1173 bedanken. Besonderer Dank geht dabei an Johannes Ernesti, Carlos Galeano, Pascal Kraft, Lars Machinek, Markus Maier, Marcel Mikl und Daniel Ziegler die mir mit kritischen Anmerkungen und Diskussionen zu meiner Arbeit behilflich waren.

Außerdem danke ich Dr. Uwe Köcher (Helmut Schmidt Universität, Hamburg) und Dr. Martin Neumüller (Johannes Kepler Universität, Linz) für anregende und interessante Diskussionen. Dank der Übernahme der Reisekosten durch das Graduiertenkolleg 1294 bzw. des Sonderforschungsbereichs 1173 war es ihnen möglich das KIT im Sommer 2014 bzw. Januar 2016 zu besuchen.

Zu guter Letzt danke ich meinen Eltern Brigitte und Matthias, sowie meinem Bruder Florian für ihre jahrelange Unterstützung und Ermutigung. Danke, dass ihr immer an mich geglaubt habt. Euch sei diese Arbeit gewidmet.

Stefan Findeisen
Karlsruhe, Sommer 2016

Contents

Page

Contents v

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Outline . . . 3
- 1.3 Prepublication . . . 5

2 Fundamentals 7
- 2.1 Notations . . . 7
- 2.2 Definitions . . . 8

3 Hyperbolic Models and Analytic Framework 11
- 3.1 General Hyperbolic Model . . . 11
- 3.2 Variational Setting . . . 12
- 3.3 Linear Transport Equation . . . 15
- 3.4 Electromagnetic Waves . . . 16
- 3.5 Boundary Conditions in Applications . . . 20

4 A Space-Time dPG Discretization 23
- 4.1 Galerkin Methods . . . 23
- 4.2 Finite Element Method . . . 24
  - 4.2.1 Conforming Finite Element Discretization . . . 29
  - 4.2.2 Non-Conforming Finite Element Discretization . . . 29
- 4.3 Discontinuous Galerkin Operators in Space . . . 30
- 4.4 Upwind Flux . . . 31
  - 4.4.1 Linear Transport Equation . . . 36
  - 4.4.2 Electromagnetic Waves . . . 37
- 4.5 DPG Approximation in Space-Time . . . 40
- 4.6 Nodal Space-Time Discretization . . . 49

5 Error Estimation and Adaptivity 55
- 5.1 Dual Weighted Residuals . . . 56
- 5.2 Error Representation . . . 56

5.2.1 Linear Error Functionals . . . 62
5.2.2 Quadratic Error Functionals . . . 62
5.3 Error Estimation . . . 63
5.3.1 Higher-Order Approximation . . . 65
5.3.2 Higher-Order Interpolation on Patches . . . 65
5.4 Adaptive Algorithm and Implementation . . . 67

6 Space-Time Multilevel Preconditioning 71
6.1 Introduction to Multigrid Methods . . . 71
6.2 Two-Grid Cycle . . . 72
6.2.1 Iterative Methods and Preconditioning . . . 73
6.2.2 Smoothers . . . 74
6.2.3 Mesh Transfer Operations . . . 76
6.2.4 Extension to the p-Adaptive Case . . . 83
6.2.5 Coarse Grid Correction . . . 85
6.2.6 Complete Two-Grid Method . . . 86
6.3 Block-Jacobi Smoothing in Time . . . 89
6.3.1 Continuous Petrov–Galerkin Method in Time . . . 90
6.3.2 Block-Jacobi Smoother for a Test Equation . . . 92
6.3.3 Possible Transfer to the Space-Time Setting . . . 95
6.4 Testing different Multilevel Settings . . . 96
6.4.1 Linear Transport Equation . . . 96
6.4.2 Maxwell's Equations in 2D . . . 101

7 Implementation 103
7.1 Generating a Space-Time Mesh . . . 103
7.2 Cell Distribution and Load Balancing . . . 104
7.3 Polynomial Adaptivity . . . 107

8 Numerical Experiments 109
8.1 Higher-Order Space-Time Methods . . . 109
8.1.1 Linear Transport Equation . . . 110
8.1.2 Maxwell's Equations in 2D . . . 113
8.2 Parallel Scalability . . . 116
8.3 Parallel and Adaptive Space-Time Computation . . . 119
8.3.1 Linear Transport Equation . . . 119
8.3.2 Electromagnetic TM Waves . . . 125

9 Summary, Conclusion and Outlook 131
9.1 Summary . . . 131
9.2 Conclusion and Outlook . . . 132

A Appendix **133**
A.1 Barycentric Coordinates on Tetrahedral Cells 133
A.2 Legendre Polynomials 134
A.3 Specifications of Computational Resources 137

Bibliography **146**

1 Introduction

1.1 Motivation

This work is dedicated to the efficient numerical simulation of hyperbolic first-order problems by using a space-time finite element approach. Throughout, the emphasis is on linear transport and electromagnetic wave problems, described by linear advection and Maxwell's equations, respectively. The presented ideas and techniques, however, can be applied and extended to various other hyperbolic problems, acoustic and elastic wave problems for instance. We focus on a holistic approach to solve Maxwell's equations numerically. This means that we consider the main occurring issues such as discretization, implementation and solving of the obtained linear systems as well as adaptivity. As a consequence, there will be no elaborated discussion of every topic up to all specific details. Instead we often focus on selected representative ideas and methods to emphasize the importance and applicability of a whole class of concepts. In the following literature reviews we give an overview of existing ideas and methods and their evolution. Moreover, we highlight their importance and influence with respect to space-time approaches. Afterwards a short outline of our work is given.

Space-Time (Petrov–) Galerkin Methods The basic idea of space-time methods for time-dependent problems is that the time direction can be understood as another spatial dimension. Hence a finite element discretization can be applied in space and time simultaneously. The simplest (tensor product) approach consists of using a standard finite element in space discretization equipped with a suitable (Petrov–) Galerkin method in time. This procedure is quite common and has been applied already to various problems, see, e.g., [Hul72], [AM89], [Joh93], [BGR10], [MS11], [HST12], [KB14], [ZW14]. The reason for the popularity of these methods is that, depending on the chosen method, they are similar to explicit or implicit Runge–Kutta schemes and can be applied slice-wise as a time stepping method. Moreover, due to their finite element origin it is possible to use standard techniques

to analyze them. Apart from that features, space-time discretizations offer the possibility to consider the entire space-time domain at once. Thus it is possible to apply hybridization techniques by introducing skeleton variables, see, e.g., [DG14] or [EDCM14]. Additionally, space-time discretizations can be used for problems with moving boundaries as well as space-time parallel computation and local space-time refinement. The last two features are highlighted in the following paragraphs and investigated in more detail within this work.

To avoid instabilities due to a violated CFL (Courant–Friedrichs–Lewy) condition we focus on an implicit continuous Petrov–Galerkin discretization in time, see, e.g., [Hul72], [AM89], [BGR10], [KB14]. For the spatial discretization a discontinuous Galerkin finite element method (DG-FEM) is used. Due to their flexibility and low effort in terms of implementation, compared to other methods, DG-FEM methods for hyperbolic equations experienced an enormous attention during the last decade. In case of DG-FEM for Maxwell's equations see for example [CFP06], [GSS07], [HW02], [HW08], [HPS+15], [DFW16].

Space-Time Parallelization Nowadays the use of spatial parallelism has become a standard procedure and is part of almost every sophisticated commercial or scientific finite element software library. However, the need of new parallel in space **and** time methods is the result of two circumstances. On one hand three dimensional time-dependent problems, which occur from modern applications and issues, result in high numerical complexity. On the other hand the recent developments in computer science indicate that increasing computational power is not gained by increasing clock speed of a single core anymore. However, the computational resources are increased by adding more and more independent arithmetical units on one chip. Hence using spatial parallelization only will lead to a bottleneck in the future due to prevailing communication times, when running the algorithms on thousands of processes. So far, only efficient parallel in space and time solvers seem to be able to overcome this challenge.

The first idea of parallelizing in time already appeared more than 50 years ago in [Nie64]. But it took several years until the concept of parallel in time methods was continued in the mid 1990s by the *parareal* algorithm [ARW95], [LMT01] [GV07], multigrid methods [HV95], [EM12], [FFK+14] and direct methods later on, e.g., [GG13]. The development of most of these methods was started under the condition, that it should be easy to incorporate them into existing time stepping codes. Furthermore, their application is mainly focused on parabolic problems. Due to increasing computational resources and the realization that hyperbolic problems have to be considered in space and time simultaneously, new parallel space-time methods have been recently introduced, e.g., [GHN03] or [Neu13]. An almost complete overview over the progress of parallel in time methods during the last 50 years can be found in [Gan15].

Adaptivity Another aspect of space-time finite element discretizations is the "build-in" possibility of using adaptivity in space and time. Adaptivity for time-dependent problems always requires to track and store the evolution of the solution and the mesh over the whole time domain. Furthermore, one also has to mind changing local polynomial degrees. All these issues have to be considered when implementing adaptivity within an existing time stepping scheme. However, these features are naturally inhered in a space-time discretization since it can be understood as a $D + 1$ problem.

Despite a straightforward implementation in space-time discretizations, the main reason for using adaptivity is the high reduction of computational costs which can be achieved. In many application it is not necessary to compute a solution with the same (high) accuracy over the whole space-time domain. Examples would be the resolution of a single wave front, where reflections are simultaneously neglected or special properties of the solution within certain subsets of the space-time domain. For this purpose *dual weighted residual* methods are used.

These methods build up on the foundations achieved by Aubin [Aub67] and Nitsche [Nit68], known as the "Aubin-Nitsche trick", where a duality argument is used in the a priori error analysis for finite element methods. Later on, duality was exploited for a posteriori error analysis for elliptic and parabolic problems in [EJ88] and [EJ91], respectively. As a further development the dual weighted residual method was finally introduced in [BR96] and successfully applied to various problems, such as eigenvalue and elliptic problems as well as time-dependent parabolic and hyperbolic problems, see, e.g., [AO00, Ch. 8],[EG04, Sec. 10.3],[Ver13, Sec. 1.11] for basic introductions and [BR03] for an overview of applications.

1.2 Outline

This work is structured as follows:

In the following chapter we define notations frequently used throughout this work. Furthermore, basic results and definitions on Sobolev and Hilbert spaces are repeated.

In the third chapter we introduce a first-order hyperbolic model equation and derive a corresponding variational setting. We prove existence and uniqueness for solutions of the model equation. Finally, we focus on two applications, the linear transport example and Maxwell's equations, and show that they fit into the predefined hyperbolic setting.

In Chapter 4 we discuss a space-time discontinuous Petrov–Galerkin method for first-order hyperbolic problems. Starting with an introduction to Galerkin and finite element methods, we derive discontinuous Galerkin approximations to the spatial operators occurring in the hyperbolic framework. In doing so, upwind fluxes

are constructed by solving a general Riemann problem. On this basis the upwind fluxes on cell interfaces are explicitly computed for the two model applications. The results are then combined with an implicit continuous Petrov–Galerkin method in time to achieve a space-time discretization. In case of tensor product discretizations we are able to prove existence and uniqueness of solutions within the discrete setting. In the end of the chapter the application of nodal finite elements is discussed.

The fifth chapter is dedicated to dual error estimation and adaptivity. We introduce the concept of dual weighted residual methods, where one is interested in minimizing the approximation error with respect to a given quantity, expressed as a functional. In general this functional can be chosen almost arbitrarily with respect to the underlying problem. We restrict ourselves to commonly used linear and quadratic functionals. Furthermore, we derive error representations to approximate the error locally with respect to a chosen functional. As a result a local error indicator is achieved and used for p-adaptive refinement. The single components are finally combined within a p-adaptive algorithm.

In Chapter 6 we deal with the solving of large linear systems occurring as a result of the space-time discretization introduced in Chapter 4. We give a detailed introduction to iterative splitting and multigrid methods and their application to space-time problems. Moreover, we discuss the extension to p-adaptive problems. The two grid cycle in time is studied for a test equation to derive an optimal setting for a block-Jacobi smoother. Finally, several tests are performed for the linear transport equation to determine a suitable configuration for a multigrid method. Afterwards the derived setting is investigated in case of Maxwell's equations.

The implementation of a parallel space-time method is discussed in Chapter 7. We highlight how space-time meshes can be generated within an existing finite element library and p-adaptive methods can be implemented. The occurring performance problems, due to different polynomial degrees, can be reduced by applying a simple load balancing algorithm.

Finally, we perform several numerical experiments in Chapter 8. We begin with an experimental convergence analysis of our space-time method to verify the expected convergence rates. These tests are done in space and time for different polynomial degrees and both, the linear transport example and the Maxwell case. Afterwards we consider the parallel scaling performance of our code. Due to the difficulties discussed in Chapter 6 we do not achieve a so called perfect weak scaling behavior. The strong scalability results, however, indicate that our code is parallelized well. In two final numerical tests all described methods are combined. Again we use the linear transport example (with known exact solutions) as a test example to prove the reliability of the error indicator derived in Chapter 5. In a more realistic test case for electromagnetic waves we highlight that the introduced parallel and adaptive space-time method is suitable to solve real application problems efficiently.

1.3 Prepublication

The main results of this work have been published in advance together with Prof. Dr. Willy Dörfler and Prof. Dr. Christian Wieners in “Space-Time Discontinuous Galerkin Discretizations for Linear First-Order Hyperbolic Evolution Systems”, see [DFW16]. Corresponding citations are given within this work.

2 Fundamentals

In this chapter we introduce basic notations and definitions which are used throughout this work. Additional notations and definitions are introduced wherever needed. To help the reader, we give a corresponding reference if notations reappear in another context.

2.1 Notations

In general we distinguish between scalar- and vector-valued quantities by using normal and boldface lowercase letters, respectively. Hence the i-th scalar component of a vector $\mathbf{v}$ is denoted as v_i. If results are valid for both, the scalar- and vector-valued case, we stick to the boldface notation. Matrices and operators are represented in capital letters. Correspondingly we use boldface capital letters for vectors of matrices or operators, e.g., $\mathbf{F} = (B_1, \dots, B_D)^\top$. Vectors and matrices used in a computational context, i.e., they can be stored in the memory of a computer, are equipped with an underscore. For better readability we refrain from using the boldface notation in this case. Moreover, we use the bracket notation to access single entries of the vectors or matrices, e.g., $(\mathbf{v})_i = v_i$, $(\underline{v})_i = \underline{v}_i$ or $(\underline{M})_{i,j}$. One special vector is the so called *unity vector* $\mathbf{i}_d \in \mathbb{R}^D$. Its entries are equal to zero, except the d-th component which is set to one, i.e., $(\mathbf{i}_d)_j = i_{d,j} = \delta_{d,j}$. Here, $\delta_{d,j}$ denotes the so called *Kronecker delta*. The *identity matrix* is denoted as $\underline{\mathrm{Id}}_N \in \mathbb{R}^{N,N}$.

In various parts of this work we will consider restrictions, jumps and averages of scalar- or vector-valued functions $\mathbf{v}$. In principal we stick to the notations introduced in [DPE12, Sec. 1.2.3]. Consider two disjoint open set $\Omega_\mathrm{L}, \Omega_\mathrm{R} \subset \mathbb{R}^D$, $D \in \mathbb{N}$ with a non-empty common *interface* $f = \partial\Omega_\mathrm{L} \bigcap \partial\Omega_\mathrm{R}$. Assume that $\mathbf{v}$ is smooth enough such that the *restriction* $\mathbf{v}_{\Omega_\mathrm{L}} = \mathbf{v}|_{\Omega_\mathrm{L}}$ to Ω_L can be defined up to the boundary $\partial\Omega_\mathrm{L}$ by extension (analogously for $\mathbf{v}_{\Omega_\mathrm{R}} = \mathbf{v}|_{\Omega_\mathrm{R}}$). The *jump* of $\mathbf{v}$ with respect to Ω_L along the interface is then denoted as

$$[\mathbf{v}]_{\Omega_\mathrm{L},f}(\mathbf{x}) = \mathbf{v}_{\Omega_\mathrm{R}}(\mathbf{x}) - \mathbf{v}_{\Omega_\mathrm{L}}(\mathbf{x}), \qquad \text{for almost all } \mathbf{x} \in f$$

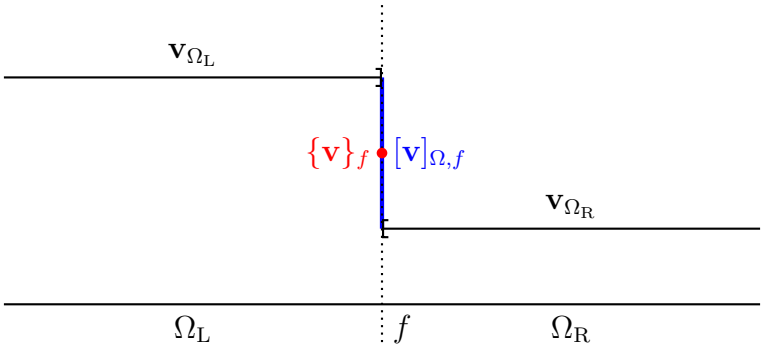

Figure 2.1: Illustration of the jump and average of $\mathbf{v}$ in one dimension.

and analogously the jump with respect to Ω_R is denoted as

$$[\mathbf{v}]_{\Omega_\mathrm{R},f}(\mathbf{x}) = \mathbf{v}_{\Omega_\mathrm{L}}(\mathbf{x}) - \mathbf{v}_{\Omega_\mathrm{R}}(\mathbf{x}), \qquad \text{for almost all } \mathbf{x} \in f.$$

The jumps are defined uniquely and it holds that $[\mathbf{v}]_{\Omega_\mathrm{L},f} = -[\mathbf{v}]_{\Omega_\mathrm{R},f}$. Note that the definitions of a jump are not consistent in the general literature. The *average* of $\mathbf{v}$ along the interface is given as

$$\{\mathbf{v}\}_f(\mathbf{x}) = \frac{1}{2}\Big(\mathbf{v}_{\Omega_\mathrm{L}}(\mathbf{x}) + \mathbf{v}_{\Omega_\mathrm{R}}(\mathbf{x})\Big), \qquad \text{for almost all } \mathbf{x} \in f.$$

See Figure 2.1 for an illustration of the jump and average of a function across an interface.

2.2 Definitions

In this section we briefly state basic norms and spaces used in this work. The following definitions can be found for example in [Bra13, Ch. II, §1], [DPE12, Sec. 1.1.3] or [Eva10, Sec. 5.2]. For an open set $\Omega \subseteq \mathbb{R}^D$ and $D \in \mathbb{N}$ we define the norms

$$\begin{aligned} \|v\|_{\mathrm{L}_p(\Omega)} &= \left(\int_\Omega |v|^p\right)^{\frac{1}{p}}, && \text{for all } v \in \mathrm{L}_p(\Omega), \\ \|v\|_{\mathrm{L}_\infty(\Omega)} &= \sup \operatorname{ess}_\Omega\{|v|\}, && \text{for all } v \in \mathrm{L}_\infty(\Omega), \end{aligned}$$

for $1 \leq p < \infty$. In particular, the Lebesgue space $\mathrm{L}_2(\Omega)$ equipped with the inner product

$$(v, w)_{0,\Omega} = \int_\Omega vw, \qquad \text{for all } v, w \in \mathrm{L}_2(\Omega)$$

is a Hilbert space.

Moreover, we denote the Sobolev spaces as $\mathrm{W}^{m,p}(\Omega)$ for $1 \leq p \leq \infty$ and $m \in \mathbb{N}_0$. For $p = 2$ the spaces

$$\mathrm{H}^m(\Omega) = \mathrm{W}^{m,2}(\Omega)$$

are again Hilbert spaces equipped with the inner product

$$(v, w)_{m,\Omega} = \sum_{|\alpha| \leq m} (\partial_\alpha v, \partial_\alpha w)_{0,\Omega}, \qquad \text{for all } v, w \in \mathrm{H}^m(\Omega).$$

Hence we achieve the corresponding norms

$$\|v\|_{m,\Omega} = \left(\sum_{|\alpha| \leq m} \|\partial_\alpha v\|^2_{\mathrm{L}_2(\Omega)} \right)^{\frac{1}{2}}$$

in multi-index notation. Note that $\mathrm{L}_2(\Omega) \equiv \mathrm{H}^0(\Omega)$ and hence $\|.\|_{\mathrm{L}_2(\Omega)} \equiv \|.\|_{0,\Omega}$. However, the Euclidean norm for vectors $\mathbf{v} \in \mathbb{R}^D$ is defined as

$$\|\mathbf{v}\|_2 = \sqrt{\mathbf{v} \cdot \mathbf{v}},$$

with respect to the standard dot-product.

In the following we introduce definitions stated in [Mon04, Ch. 3] to consider weak curl fields in $\mathrm{L}_2(\Omega)^3$. Therefore we additionally define the Sobolev space

$$\mathrm{H}(\mathrm{curl}, \Omega) = \{\mathbf{v} \in \mathrm{L}_2(\Omega)^3 \colon \nabla \times \mathbf{v} \in \mathrm{L}_2(\Omega)^3\}$$

with inner product

$$(\mathbf{v}, \mathbf{w})_{\mathrm{curl},\Omega} = (\mathbf{v}, \mathbf{w})_{0,\Omega} + (\nabla \times \mathbf{v}, \nabla \times \mathbf{w})_{0,\Omega}, \qquad \text{for all } \mathbf{v}, \mathbf{w} \in \mathrm{H}(\mathrm{curl}, \Omega)$$

and corresponding norm

$$\|\mathbf{v}\|_{\mathrm{H}(\mathrm{curl},\Omega)} = \left(\|\mathbf{v}\|^2_{0,\Omega} + \|\nabla \times \mathbf{v}\|^2_{0,\Omega} \right)^{\frac{1}{2}}, \qquad \text{for all } \mathbf{v} \in \mathrm{H}(\mathrm{curl}, \Omega).$$

To enforce homogeneous boundary conditions on $\partial\Omega$ we define the spaces

$$\mathrm{H}^m_0(\Omega) = \text{closure of } C^\infty_0(\Omega) \text{ in } \mathrm{H}^m(\Omega),$$
$$\mathrm{H}_0(\mathrm{curl}, \Omega) = \text{closure of } C^\infty_0(\Omega)^3 \text{ in } \mathrm{H}(\mathrm{curl}, \Omega),$$

where $C^\infty_0(\Omega)$ is the set of functions in $C^\infty(\Omega)$ which have compact support in Ω. Moreover, we can define a *dual pair* for a vector space X and its dual space X^* as

$$\langle v, w \rangle = w(v), \qquad \text{for all } v \in X, w \in X^*.$$

If we, for example, choose $X \equiv \mathrm{H}^m_0(\Omega)$ and $X^* \equiv \mathrm{H}^{-m}(\Omega)$ for $m \geq 1$, then the dual pair is a bilinear form and it yields

$$\langle v, w \rangle = (v, w)_{0,\Omega}, \qquad \text{for all } v \in \mathrm{H}^m_0(\Omega), w \in \mathrm{L}_2(\Omega).$$

Here $\mathrm{H}^{-m}(\Omega)$ is the closure of $\mathrm{L}_2(\Omega)$ with respect to the norm

$$\|w\|_{-m,\Omega} = \sup_{v \in \mathrm{H}^m_0(\Omega)} \frac{(v, w)_{0,\Omega}}{\|v\|_{m,\Omega}}$$

for $w \in \mathrm{L}_2(\Omega)$(cf. [Bra13, Ch. III, §3]).

3 Hyperbolic Model and Analytic Framework

In this chapter a general first-order hyperbolic model equation is introduced. We state a variational formulation of the hyperbolic problem and prove existence and uniqueness of solutions under certain conditions (cf. [DFW16]). For this purpose we apply the standard Babuška setting (cf. [Bra13]) and show inf-sup stability. Finally, we consider two different applications from the fields of advection phenomena and electromagnetic waves, which fit into the hyperbolic setting. Moreover, we show that these applications fulfill the corresponding requirements presumed in the derived existence and uniqueness result.

3.1 General Hyperbolic Model

Let $\Omega \subset \mathbb{R}^D$, $D \in \{1, 2, 3\}$, be a bounded Lipschitz domain, i.e., the boundary $\partial\Omega$ of Ω can be locally represented as a Lipschitz continuous function. Furthermore consider a time interval $(0, T)$ with final time $T > 0$. In this work we study first-order evolution equations of the following type

$$M\partial_t \mathbf{u}(t) + A\mathbf{u}(t) = \mathbf{f}(t), \qquad t \in (0, T). \tag{3.1}$$

Here $M \in \mathrm{L}_\infty(\Omega)^{J\times J}$ is a symmetric and uniformly positive definite matrix, i.e., there exists a constant $c > 0$ such that $(M\mathbf{v}, \mathbf{v})_{0,\Omega} \geq c\|\mathbf{v}\|^2_{0,\Omega} > 0$ for all $\mathbf{v} \in \mathbb{R}^J \backslash \{\mathbf{0}\}$. Additionally, $A\colon \mathcal{D}(A) \to H$ is a linear operator, which maps from its domain $\mathcal{D}(A)$ into a Hilbert space $H \subset \mathrm{L}_2(\Omega)^J$. H is equipped with the weighted inner product $(\mathbf{v}, \mathbf{w})_H = (M\mathbf{v}, \mathbf{w})_{0,\Omega} = \int_\Omega M\mathbf{v} \cdot \mathbf{w}$ and $\mathbf{f} \in \mathrm{L}_2(0, T; H)$ is a given right-hand side. To state the complete model, we have to provide a suitable initial condition $\mathbf{u}_0 \in \mathcal{D}(A)$ at time $t = 0$ and boundary conditions (b.c.) on $\partial\Omega$ for $t > 0$. Hence the full model reads as

$$\begin{aligned} M\partial_t \mathbf{u}(t) + A\mathbf{u}(t) &= \mathbf{f}(t), && t \in (0, T), \\ \mathbf{u}(0) &= \mathbf{u}_0, && \text{on } \Omega \times \{0\}, \\ \text{and } \mathbf{u} &\text{ fulfills b.c.} && \text{on } \partial\Omega \times (0, T). \end{aligned} \tag{3.2}$$

We restrict ourselves to the case of linear conservation laws (see, e.g., [Eva10, Sec. 11.1]), by claiming that the operator A can be written as

$$A\mathbf{v} = \nabla \cdot \mathbf{F}(\mathbf{v}) = \sum_{d=1}^{D} \partial_d(B_d\mathbf{v}) = \sum_{d=1}^{D} B_d(\partial_d\mathbf{v}) \in \mathrm{L}_2(\Omega)^J, \qquad \mathbf{v} \in \mathcal{D}(A), \tag{3.3}$$

where $\mathbf{F}(\mathbf{v}) = (B_1\mathbf{v}, \ldots, B_D\mathbf{v})^\top$ is called *flux function*, with symmetric matrices $B_d \in \mathrm{L}_\infty(\Omega)^{J\times J}$. Note that one can formally replace $\nabla = (\partial_1, \partial_2, \ldots, \partial_D)^\top$ by some arbitrary vector $\mathbf{n} = (n_1, \ldots, n_D)^\top \in \mathbb{R}^d$ and state the following lemma.

Lemma 3.1. *For every* $\mathbf{n} = (n_1, \ldots, n_D)^\top \in \mathbb{R}^D$, $\mathbf{n} \cdot \mathbf{F}$ *is diagonalizable with real eigenvalues.*

Proof. It holds that $\mathbf{n} \cdot \mathbf{F} = n_1B_1 + \ldots + n_DB_D$. Since B_d is symmetric the linear combination $\mathbf{n}\cdot\mathbf{F}$ is again symmetric and hence has real eigenvalues. Moreover, $\mathbf{n}\cdot\mathbf{F}$ is diagonalizable due to the finite dimensional spectral theorem, see, e.g., [Wer11, Thm. VII.1.13, Ex. a)]. □

Due to the following definition given in [Eva10, Sec. 7.3] and Lemma 3.1, (3.1) is a hyperbolic system.

Definition 3.1 (Hyperbolic)**.** A system of partial differential equations

$$M\partial_t\mathbf{v} + \sum_{d=1}^{D} B_d(\partial_d\mathbf{v}) = \mathbf{f}, \qquad \text{for } t \geq 0$$

is called *hyperbolic*, if $\mathbf{n} \cdot \mathbf{F}$ is diagonalizable for every $\mathbf{n} \in \mathbb{R}^D$ and the matrix M is symmetric and positive definite.

In the following we introduce a variational setting and consider a so called *weak* or *variational* formulation of problem (3.2). Correspondingly, (3.2) is denoted as *strong* form of the problem.

3.2 Variational Setting

Consider the abstract operator $L = M\partial_t + A$ on a space-time cylinder $Q = \Omega\times(0,T)$ and homogeneous initial conditions $\mathbf{u}_0 = \mathbf{0}$. The extension to inhomogeneous initial conditions is discussed later on. As in Section 2.2, we define the inner L_2-produc and corresponding norm in the space time domain, e.g.,

$$\begin{aligned} (\mathbf{v}, \mathbf{w})_{0,Q} &= \int_0^T \Big(\mathbf{v}(t), \mathbf{w}(t)\Big)_{0,\Omega} \,\mathrm{d}t, && \text{for all } \mathbf{v}, \mathbf{w} \in \mathrm{L}_2(Q)^J \\ \|\mathbf{v}\|_{0,Q} &= \sqrt{(\mathbf{v}, \mathbf{v})_{0,Q}}, && \text{for all } \mathbf{v} \in \mathrm{L}_2(Q)^J. \end{aligned}$$

The domain of the space-time operator L is given as

$$V = \mathcal{D}(L) = \overline{\left\{\mathbf{v} \in \mathrm{C}^1(0,T;\mathcal{D}(A))\colon \mathbf{v}(0) = \mathbf{0}\right\}},$$

where the closure is taken with respect to the weighted *graph norm*

$$\|\mathbf{v}\|_V^2 = (M\mathbf{v},\mathbf{v})_{0,Q} + (M^{-1}L\mathbf{v},L\mathbf{v})_{0,Q}. \tag{3.4}$$

This means that $\mathcal{D}(L)$ equipped with $\|.\|_V$ is complete and hence V is again a Hilbert space, see, e.g., [EG04, Sec. 5.2.1]. Since $\Omega\times\{0\}$ can be understood as part of the inflow boundary of the space-time cylinder Q, the quantity $\mathbf{v}(0)$ for $\mathbf{v}\in V$ is well-defined (cf. [EG04, Sec. 5.2.2] or [MSW95]). Moreover, we define

$$W = \overline{L(V)} \subseteq \mathrm{L}_2(0,T;H)$$

as the closure of the range with respect to the weighted norm $\|\mathbf{w}\|_W^2 = (M\mathbf{w},\mathbf{w})_{0,Q}$. Note that in terms of this definition, the graph norm (3.4) can be rewritten as $\|\mathbf{v}\|_V^2 = \|\mathbf{v}\|_W^2 + \|M^{-1}L\mathbf{v}\|_W^2$. The corresponding dual space to V is given as

$$V^* = \overline{\left\{\mathbf{v}^* \in \mathrm{C}^1(0,T;\mathcal{D}(A^*))\colon \mathbf{v}^*(T) = \mathbf{0}\right\}} \tag{3.5}$$

and equipped with a dual graph norm $\|\,.\,\|_{V^*}$, see, e.g., [SS98]. Here the initial condition transfers to a final condition and the evaluation of $\mathbf{v}^*\in V^*$at time $t=T$ is again justified by [EG04, Sec. 5.2.2] or [MSW95].

To derive a variational formulation of our problem, we multiply $L\mathbf{v}$ with an arbitrary test function $\mathbf{w}\in W$ and integrate over the space-time domain Q. This defines the bilinear form $b\colon V\times W\to\mathbb{R}$ with

$$b(\mathbf{v},\mathbf{w}) = (L\mathbf{v},\mathbf{w})_{0,Q}, \tag{3.6}$$

and we are able to establish the standard Babuška setting stated in the following lemma (cf. [EG04, Thm. 2.6 & Sec. 5.2.1] or [Bra13, Ch. III, §3]). The proofs base on techniques developed in [WW14].

Lemma 3.2 ([DFW16, Lem. 2.1])**.** *Assume that A is positive semi-definite, i.e., $(A\mathbf{v},\mathbf{v})_{0,\Omega}\geq 0$ for all $\mathbf{v}\in\mathcal{D}(A)$. Then, the bilinear form $b(\cdot,\cdot)$ from (3.6) is continuous and inf-sup stable in $V\times W$ with constant $\beta = 1/\sqrt{4T^2+1}$, i.e.,*

$$\sup_{\mathbf{w}\in W\setminus\{\mathbf{0}\}} \frac{b(\mathbf{v},\mathbf{w})}{\|\mathbf{w}\|_W} \geq \beta\,\|\mathbf{v}\|_V\,, \qquad \textit{for all } \mathbf{v}\in V\,.$$

Proof. First we show that b is bounded and hence continuous, by using Cauchy–Schwarz inequality. We obtain

$$\begin{aligned}
|b(\mathbf{v},\mathbf{w})|^2 &= (L\mathbf{v},\mathbf{w})_{0,Q}^2 = (MM^{-1}L\mathbf{v},\mathbf{w})_{0,Q}^2 \\
&\leq \|M^{-1}L\mathbf{v}\|_W^2\|\mathbf{w}\|_W^2 \leq (\|\mathbf{v}\|_W^2 + \|M^{-1}L\mathbf{v}\|_W^2)\|\mathbf{w}\|_W^2 \\
&= \|\mathbf{v}\|_V^2\|\mathbf{w}\|_W^2
\end{aligned}$$

and $|b(\mathbf{v},\mathbf{w})| \leq \|\mathbf{v}\|_V \|\mathbf{w}\|_W$. To prove the inf-sup condition we first note that for all $\mathbf{v} \in \mathrm{C}^1(0,T;\mathcal{D}(A))$ with $\mathbf{v}(0) = \mathbf{0}$ it holds that

$$\begin{aligned}
\|\mathbf{v}\|_W^2 &= \int_0^T \Big(M\mathbf{v}(t), \mathbf{v}(t)\Big)_{0,\Omega} \,\mathrm{d}t \\
&= \int_0^T \Big(\Big(M\mathbf{v}(t), \mathbf{v}(t)\Big)_{0,\Omega} - \Big(M\mathbf{v}(0), \mathbf{v}(0)\Big)_{0,\Omega}\Big) \,\mathrm{d}t \\
&= \int_0^T \int_0^t \partial_t\Big(M\mathbf{v}(s), \mathbf{v}(s)\Big)_{0,\Omega} \,\mathrm{d}s\,\mathrm{d}t = 2\int_0^T \int_0^t \Big(M\partial_t\mathbf{v}(s), \mathbf{v}(s)\Big)_{0,\Omega} \,\mathrm{d}s\,\mathrm{d}t \\
&\leq 2\int_0^T \int_0^t \Big(M\partial_t\mathbf{v}(s) + A\mathbf{v}(s), \mathbf{v}(s)\Big)_{0,\Omega} \,\mathrm{d}s\,\mathrm{d}t \\
&\leq 2\int_0^T \int_0^t \Big(M^{-1}L\mathbf{v}(s), L\mathbf{v}(s)\Big)_{0,\Omega}^{1/2} \Big(M\mathbf{v}(s), \mathbf{v}(s)\Big)_{0,\Omega}^{1/2} \,\mathrm{d}s\,\mathrm{d}t \\
&\leq 2T\,\|M^{-1}L\mathbf{v}\|_W \|\mathbf{v}\|_W\,.
\end{aligned}$$

This yields $\|\mathbf{v}\|_W \leq 2T\,\|M^{-1}L\mathbf{v}\|_W$ for $\mathbf{v} \in V$. Let $\mathbf{v} \in V \setminus \{\mathbf{0}\}$ and take $\mathbf{w} = M^{-1}L\mathbf{v} \in W \setminus \{\mathbf{0}\}$ as a special test function, then

$$\begin{aligned}
\sup_{\mathbf{w}\in W\setminus\{\mathbf{0}\}} \frac{b(\mathbf{v},\mathbf{w})}{\|\mathbf{w}\|_W} &\geq \frac{b(\mathbf{v}, M^{-1}L\mathbf{v})}{\|M^{-1}L\mathbf{v}\|_W} = \frac{(L\mathbf{v}, M^{-1}L\mathbf{v})_{0,\Omega}}{\|M^{-1}L\mathbf{v}\|_W} \\
&= \|M^{-1}L\mathbf{v}\|_W \geq \frac{1}{\sqrt{4T^2+1}}\,\|\mathbf{v}\|_V\,,
\end{aligned}$$

where the latter inequality follows from

$$\|\mathbf{v}\|_V^2 = \|\mathbf{v}\|_W^2 + \|M^{-1}L\mathbf{v}\|_W^2 \leq (4T^2+1)\|M^{-1}L\mathbf{v}\|_W^2.$$

□

The inf-sup stability ensures that the operator $L \in \mathcal{L}(V,W)$ is injective and that the range is closed. Thus, the operator is surjective by construction and the inverse L^{-1} is bounded in $\mathcal{L}(W,V)$. Due to [Bra13, Thm. III.3.6], this yields directly the following result.

Theorem 3.1 ([DFW16, Thm. 2.2])**.** *For given $\mathbf{f} \in \mathrm{L}_2(Q)^J$ there exists a unique solution $\mathbf{u} \in V$ of*

$$(L\mathbf{u}, \mathbf{w})_{0,Q} = (\mathbf{f}, \mathbf{w})_{0,Q}\,, \qquad \text{for all } \mathbf{w} \in W. \tag{3.7}$$

Moreover, $\mathbf{u}$ satisfies the a priori bound $\|\mathbf{u}\|_V \leq \sqrt{4T^2+1}\,\|M^{-1/2}\mathbf{f}\|_{0,Q}$.

Remark 3.1. The a priori bound in Theorem 3.1 is achieved by using the inf-sub condition

$$\begin{aligned}
\|\mathbf{u}\|_V &\leq \beta^{-1} \sup_{\mathbf{w}\in W\setminus\{\mathbf{0}\}} \frac{b(\mathbf{v},\mathbf{w})}{\|\mathbf{w}\|_W} = \beta^{-1} \sup_{\mathbf{w}\in W\setminus\{\mathbf{0}\}} \frac{(\mathbf{f},\mathbf{w})_{0,Q}}{\|\mathbf{w}\|_W} \\
&= \beta^{-1} \sup_{\mathbf{w}\in W\setminus\{\mathbf{0}\}} \frac{(MM^{-1}\mathbf{f},\mathbf{w})_{0,Q}}{\|\mathbf{w}\|_W} \leq \beta^{-1}\|M^{-1}\mathbf{f}\|_W \\
&= \beta^{-1}(M^{-1/2}\mathbf{f}, M^{-1/2}\mathbf{f})_{0,Q} = \beta^{-1}\|M^{-1/2}\mathbf{f}\|_{0,Q}.
\end{aligned}$$

Note that $M^{-1/2}$, given through the decomposition $M^{-1} = M^{-1/2}M^{-1/2}$, is well defined since M is symmetric positive definite.

The next lemma shows the connection of the weak and strong formulation of the problem.

Lemma 3.3 ([DPE12, Prop. 2.7])**.** *If* $\mathbf{u} \in V$ *solves* (3.7)*, then* $\mathbf{u}$ *solves the original problem* (3.2)

$$M\partial_t \mathbf{u} + A\mathbf{u} = \mathbf{f}$$

almost everywhere in Q*.*

Proof. Taking $\mathbf{w} \in C_0^\infty(0,T;C_0^\infty(\Omega))$ as a test function in (3.7) implies that

$$(L\mathbf{u} - \mathbf{f}, \mathbf{w})_{0,Q} = 0.$$

and the assertion follows from fact that $C_0^\infty(0,T;C_0^\infty(\Omega))$ is dense in $\mathrm{L}_2(0,T;H)$. □

Remark 3.2 (Inhomogeneous initial and boundary conditions)**.** Due to the definition of V, only homogeneous initial conditions have been assumed for every solution $\mathbf{u} \in V$ so far. One can extend the results to inhomogeneous initial values $\mathbf{u}_0 \neq \mathbf{0}$ by defining $\mathbf{u} = \mathbf{u}_{\mathrm{hom}} + \mathbf{u}_0$, where $\mathbf{u}_{\mathrm{hom}} \in V$. Hence from (3.7) one receives the following equivalent problem. Seek $\mathbf{u}_{\mathrm{hom}} \in V$ such that

$$b(\mathbf{u}_{\mathrm{hom}}, \mathbf{w}) = (\mathbf{f}, \mathbf{w})_{0,Q} - (A\mathbf{u}_0, \mathbf{w})_{0,Q}, \qquad \text{for all } \mathbf{w} \in W.$$

Similarly, the boundary conditions are given by the definition of $\mathcal{D}(A)$. Dirichlet boundary conditions can be imposed, as before for the initial condition, by modifying the right-hand side. Hence $\mathbf{u} = \mathbf{u}_{\mathrm{hom}} + \mathbf{u}_{\mathrm{bnd}}$ and again the problem results in seeking $\mathbf{u}_{\mathrm{hom}} \in V$ such that

$$b(\mathbf{u}_{\mathrm{hom}}, \mathbf{w}) = (\mathbf{f}, \mathbf{w})_{0,Q} - (L\mathbf{u}_{\mathrm{bnd}}, \mathbf{w})_{0,Q}, \qquad \text{for all } \mathbf{w} \in W.$$

To do this, a sufficiently smooth extension $\mathbf{u}_{\mathrm{bnd}}$ of the boundary data is needed (see, e.g., [EG04, Sec. 2.1.4], [Bra13, Ch. II, §2], [DPE12, Sec. 2.1.6]).

In the following sections we consider two examples, which fit into the introduced hyperbolic setting.

3.3 Linear Transport Equation

The linear transport of a concentration or quantity of a material through a fluid without any diffusion is modeled by the so called *advection* equation

$$\rho\partial_t u + \nabla \cdot (\mathbf{q}u) = f, \qquad \text{in } \Omega \times (0,T). \tag{3.8}$$

To solve this equation, one has to determine a scalar solution $u\colon \Omega \times (0,T) \to \mathbb{R}$ for a given initial condition u_0, right-hand side f and density distribution $\rho \in \mathrm{L}_\infty(\Omega)$, $\rho > 0$. The speed of the fluid is given by the vector field $\mathbf{q} \in \mathrm{W}^{1,\infty}(\Omega)^D$ which is assumed to be divergence free, i.e., $\nabla \cdot \mathbf{q} = 0$. Hence we can define the inflow and outflow boundary as

$$\begin{aligned}\Gamma_{\mathrm{in}} &= \overline{\{\mathbf{x} \in \partial\Omega\colon \mathbf{q}(\mathbf{x}) \cdot \mathbf{n}_\Omega(\mathbf{x}) < 0\}} \subseteq \partial\Omega,\\ \Gamma_{\mathrm{out}} &= \overline{\{\mathbf{x} \in \partial\Omega\colon \mathbf{q}(\mathbf{x}) \cdot \mathbf{n}_\Omega(\mathbf{x}) > 0\}} \subseteq \partial\Omega,\end{aligned}$$

respectively. Here $\mathbf{n}_\Omega$ is the outer unit normal vector on $\partial\Omega$. The full advection model with homogeneous inflow boundary condition then reads as

$$\begin{aligned}\rho\partial_t u + \nabla \cdot (\mathbf{q}u) &= f, && \text{in } \Omega \times (0,T),\\ u(0) &= u_0, && \text{on } \Omega \times \{0\},\\ u &= 0, && \text{on } \Gamma_{\mathrm{in}} \times (0,T).\end{aligned} \tag{3.9}$$

Remark 3.3. Let $\mathbf{F}(u) = \mathbf{q}u$ be the flux function, $Au = \nabla \cdot (\mathbf{q}u) = \mathbf{q}\nabla \cdot u$ with domain $\mathcal{D}(A) = \left\{u \in \mathrm{H}^1(\Omega)\colon u = 0 \text{ on } \Gamma_{\mathrm{in}}\right\}$, $H = \mathrm{L}_2(\Omega)$ and $Mu = \rho u$. Then equation (3.8) is a hyperbolic first-order evolution equation, due to Definition 3.1. Moreover, we achive that $J = 1$ and $B_d = q_d \in \mathrm{L}_\infty(\Omega)$. Note that for the adjoint operator A^* the roles of the inflow and outflow boundary are interchanged and hence $A^*u = -\nabla \cdot (\mathbf{q}u)$ with domain $\mathcal{D}(A^*) = \left\{u^* \in \mathrm{H}^1(\Omega)\colon u^* = 0 \text{ on } \Gamma_{\mathrm{out}}\right\}$.

Lemma 3.4. *The operator A is positive semi-definite and hence Theorem 3.1 holds true.*

Proof. For all $v \in \mathcal{D}(A)$ it holds that

$$\begin{aligned}(Av, v)_{0,\Omega} = (\nabla \cdot (\mathbf{q}v), v)_{0,\Omega} &= \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{q}\, v^2 \,\mathrm{d}\mathbf{a} - (\mathbf{q}v, \nabla v)_{0,\Omega}\\ &= \int_{\Gamma_{\mathrm{in}}} \mathbf{n} \cdot \mathbf{q}\, v^2 \,\mathrm{d}\mathbf{a} + \int_{\Gamma_{\mathrm{out}}} \mathbf{n} \cdot \mathbf{q}\, v^2 \,\mathrm{d}\mathbf{a} - (\nabla \cdot (\mathbf{q}v), v)_{0,\Omega}\\ &= \int_{\Gamma_{\mathrm{out}}} \mathbf{n} \cdot \mathbf{q}\, v^2 \,\mathrm{d}\mathbf{a} - (Av, v)_{0,\Omega}.\end{aligned}$$

Hence

$$(Av, v)_{0,\Omega} = \frac{1}{2} \int_{\Gamma_{\mathrm{out}}} \mathbf{n} \cdot \mathbf{q}\, v^2 \,\mathrm{d}\mathbf{a} \geq 0$$

since $\mathbf{n} \cdot \mathbf{q} \geq 0$ on Γ_{out}. □

3.4 Electromagnetic Waves

Electromagnetic waves are waves, which occur as a coupling of an electric and a magnetic field. Typical examples are radio and micro waves, (visible) light, X-rays and gamma rays, as well as thermal radiation. A very detailed introduction

to electrodynamics can be found for example in [Jac99]. In this work we restrict ourselves to a summary of the main points. We consider a spatial domain $\Omega \subset \mathbb{R}^3$ and a final time $T > 0$. Electromagnetic waves are then described by four vector fields $\mathbf{E}, \mathbf{D}, \mathbf{H}, \mathbf{B} \colon \Omega \times (0,T) \to \mathbb{R}^3$, which solve the so called *Maxwell's equations*

$$\partial_t \mathbf{D} - \nabla \times \mathbf{H} = -\mathbf{J} \qquad \text{(Ampère's circuital law)}, \tag{3.10}$$

$$\nabla \cdot \mathbf{D} = \rho \qquad \text{(Gauss's law)}, \tag{3.11}$$

$$\partial_t \mathbf{B} + \nabla \times \mathbf{E} = \mathbf{0} \qquad \text{(Faraday's law of induction)}, \tag{3.12}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{(Gauss's law for magnetism)}. \tag{3.13}$$

The data parameters $\mathbf{J} \colon \Omega \to \mathbb{R}^3$ and $\rho \colon \Omega \to \mathbb{R}$ are called *electrical current density* and *electric charge density*. The fields are called *electrical field* $\mathbf{E}$, *electric displacement field* $\mathbf{D}$, *magnetic field intensity* $\mathbf{H}$, *magnetic induction* $\mathbf{B}$, respectively. Furthermore, for a complete description a relation of the fields with matter is needed. It is given by the so called *constitutive relations* $\mathbf{D}(\mathbf{H}, \mathbf{E})$ and $\mathbf{B}(\mathbf{H}, \mathbf{E})$. In vacuum we have the linear relations

$$\mathbf{D}(\mathbf{x}, t) = \varepsilon_0 \mathbf{E}(\mathbf{x}, t) \quad \text{and} \quad \mathbf{B}(\mathbf{x}, t) = \mu_0 \mathbf{H}(\mathbf{x}, t)$$

with the *vacuum permittivity* ε_0 (*electric constant*) and the *vacuum permeability* μ_0 (*magnetic constant*). These constants fulfill the relation

$$\varepsilon_0 \mu_0 = \frac{1}{c_0^2},$$

where c_0 is the *speed of light in vacuum*. In matter the electric field $\mathbf{E}$ induces a dislocation of charges and hence a *polarization field* $\mathbf{P}$ occurs within the constitutive relation

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P}.$$

For linear media the polarization is given as

$$\mathbf{P}(\mathbf{E}) = \varepsilon_0 \chi_E \mathbf{E}.$$

We assume that the *electric susceptibility* χ_E is independent of the frequency of the electromagnetic waves. Furthermore, we assume that χ_E is real valued and scalar, i.e., the media is *isotropic*. For simplification we introduce the *relative permittivity* $\varepsilon_\mathrm{r} = 1 + \chi_E > 0$ and obtain the following linear constitutive relations

$$\mathbf{D} = \varepsilon \mathbf{E} = \varepsilon_0 \varepsilon_\mathrm{r} \mathbf{E} \quad \text{and} \quad \mathbf{B} = \mu \mathbf{H} = \mu_0 \mu_\mathrm{r} \mathbf{H},$$

where the relation for the magnetic fields $\mathbf{B}$ and $\mathbf{H}$ is deduced in an analogous way. Correspondingly, $\mu_\mathrm{r} > 0$ is called *relative permeability*. Note that the electric permittivity $\varepsilon \in \mathrm{L}_\infty(\Omega)$ and magnetic permeability $\mu \in \mathrm{L}_\infty(\Omega)$ may depend on Ω,

but not on time. Finally the general Maxwell's equations (3.10)–(3.13) reduce to the linear first-order system

$$\begin{aligned} \mu\partial_t \mathbf{H} + \nabla \times \mathbf{E} &= \mathbf{0}, \\ \varepsilon\partial_t \mathbf{E} - \nabla \times \mathbf{H} &= \mathbf{f}, \\ \nabla \cdot (\mu\mathbf{H}) &= 0, \\ \nabla \cdot (\varepsilon\mathbf{E}) &= \rho. \end{aligned} \tag{3.14}$$

Note that the divergence constraints require the *compatibility condition*

$$\partial_t \rho - \nabla \cdot \mathbf{f} = 0 \tag{3.15}$$

for the data $\mathbf{f} = -\mathbf{J}$ and ρ. We assume that ρ is independent of time and hence the compatibility condition reduces to $\nabla \cdot \mathbf{f} = 0$. Furthermore, the following Lemma shows that the divergence constraints are fulfilled for all times, if they are fulfilled for the initial condition.

Lemma 3.5 (Divergence conditions)**.** *We assume that the given data fulfill the compatibility condition* (3.15) *for* $\partial_t \rho = 0$*. Furthermore, assume that* $\mu\mathbf{H}$ *and* $\varepsilon\mathbf{E}$ *are smooth enough. If* $\nabla \cdot \Big(\mu\mathbf{H}(.,0)\Big) = 0$ *and* $\nabla \cdot \Big(\varepsilon\mathbf{E}(.,0)\Big) = \rho$*, then* $\nabla \cdot \Big(\mu\mathbf{H}(.,t)\Big) = 0$ *and* $\nabla \cdot \Big(\varepsilon\mathbf{E}(.,t)\Big) = \rho$*, for all* $t > 0$*.*

Proof. We prove the statement for $\nabla \cdot (\varepsilon\mathbf{E}) = \rho$. The other (magnetic) constraints can be shown analogously. It yields that

$$0 = \nabla \cdot \mathbf{f} = \nabla \cdot (\varepsilon\partial_t \mathbf{E} - \nabla \times \mathbf{H}) = \partial_t \Big(\nabla \cdot (\varepsilon\mathbf{E})\Big).$$

Hence $\nabla \cdot (\varepsilon\mathbf{E})$ is constant and since $\nabla \cdot \Big(\varepsilon\mathbf{E}(.,0)\Big) = \rho$ and $\nabla \cdot (\varepsilon\mathbf{E})$ is continuous in time, it holds that $\nabla \cdot (\varepsilon\mathbf{E}) = \rho$ for all $t \geq 0$. □

In the following two remarks, we give a short overview over further reductions of Maxwell's equations in special cases and applications.

Remark 3.4 (Wave equation)**.** The linear Maxwell's equations (3.14) can be further reduced to a second-order equation by eliminating $\mathbf{H}$ (or analogously $\mathbf{E}$), i.e.,

$$\varepsilon\partial_t^2 \mathbf{E} + \nabla \times \left(\frac{1}{\mu}\nabla \times \mathbf{E}\right) = \mathbf{f}. \tag{3.16}$$

In vacuum and with the absence of external charges and currents, it holds that $\varepsilon = \varepsilon_0$ and $\mu = \mu_0$ and $\rho = 0$ and $\mathbf{J} = \mathbf{0}$. Hence $0 = \nabla \cdot (\varepsilon_0\mathbf{E}) = \varepsilon_0 \nabla \cdot \mathbf{E}$ and with the representation of the Laplacian

$$-\Delta\mathbf{E} = \nabla(\nabla \cdot \mathbf{E}) - \Delta\mathbf{E} = \nabla \times (\nabla \times \mathbf{E})$$

we end up with the (second-order) *wave equation*

$$\partial_t^2 \mathbf{E} - c_0^2 \Delta\mathbf{E} = \mathbf{E}.$$

Remark 3.5 (Time-harmonic Maxwell's equations)**.** In case of a constant frequency ω, e.g., monochromatic laser light, we can assume that all fields are of the form $\mathrm{e}^{\mathrm{i}\omega t}\mathbf{E}$ and $\mathrm{e}^{\mathrm{i}\omega t}\mathbf{H}$, where $\mathbf{E}(\mathbf{x}) \in \mathbb{C}^3$ and $\mathbf{H}(\mathbf{x}) \in \mathbb{C}^3$ now only depend on space. This approach is known as *time-harmonic ansatz*. Again we assume that $\rho = 0$ and $\mathbf{J} = \mathbf{0}$. Hence the linear first-order Maxwell equations (3.14) can be written as eigenvalue problems

$$\nabla \times \left(\frac{1}{\varepsilon}\nabla \times \mathbf{H}\right) = \omega^2 \mu \mathbf{H} \quad \text{or} \quad \nabla \times \left(\frac{1}{\mu}\nabla \times \mathbf{E}\right) = \omega^2 \varepsilon \mathbf{E}$$

for $\mathbf{H}$ and $\mathbf{E}$, respectively. For example, these kind of Maxwell's equations occur when one wants to investigate band structures in photonic crystals or losses in photonic waveguide transitions (see, e.g., [DLP+11], [DF15]).

The next remark and lemma show how the linear first-order Maxwell system (3.14) can be fitted into the setting of first-order hyperbolic evolution equations (3.1).

Remark 3.6. Let $\mathbf{u} = (\mathbf{H}, \mathbf{E})^\top$, $H = \mathrm{L}_2(\Omega)^3 \times \mathrm{L}_2(\Omega)^3$, $M\mathbf{u} = (\mu\mathbf{H}, \varepsilon\mathbf{E})^\top$ and consider the operators $A\mathbf{u} = (\nabla \times \mathbf{E}, -\nabla \times \mathbf{H})^\top = -A^*\mathbf{u}$ in $\mathcal{D}(A) = \mathcal{D}(A^*) = \mathrm{H}(\mathrm{curl}, \Omega) \times \mathrm{H}_0(\mathrm{curl}, \Omega)$ to model a *perfect conducting boundary* $\mathbf{n} \times \mathbf{E} = \mathbf{0}$ on $\partial\Omega$. Hence, M and A can be written as

$$M = \begin{pmatrix} \mu & 0 \\ 0 & \varepsilon \end{pmatrix} \quad \text{and} \quad A = \begin{pmatrix} 0 & \nabla\times \\ -\nabla\times & 0 \end{pmatrix}.$$

Moreover, it holds that M is symmetric and positive definite and

$$B_d = \begin{pmatrix} 0 & C_d \\ -C_d & 0 \end{pmatrix} = \begin{pmatrix} 0 & -C_d^\top \\ C_d^\top & 0 \end{pmatrix} = \begin{pmatrix} 0 & C_d \\ -C_d & 0 \end{pmatrix}^\top = B_d^\top, \qquad \text{for } d = 1, \ldots, 3.$$

Here the $\mathbf{x}$-independent, anti-symmetric blocks $C_d \in \mathbb{R}^{3\times 3}$ are equivalent to the matrix representations of the cross product, i.e., $\mathbf{i}_d \times \mathbf{x} = C_d\mathbf{x} = -C_d^\top\mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^3$. We conclude that $J = 6$ and $B_d \in \mathrm{L}_\infty(\Omega)^{6\times 6}$ is symmetric. Thus (3.14) is a hyperbolic first-order evolution system of equations due to Definition 3.1.

Lemma 3.6. *The operator A is positive semi-definite and hence Theorem 3.1 holds true.*

Proof. Due to the perfect conducting boundary condition $\mathbf{n} \times \mathbf{E} = \mathbf{0}$, it holds for all $\mathbf{v} \in \mathcal{D}(A)$ that

$$\begin{aligned}(A\mathbf{v}, \mathbf{v})_{0,\Omega} &= \int_\Omega \nabla \times \mathbf{E} \cdot \mathbf{H} - \nabla \times \mathbf{H} \cdot \mathbf{E} \,\mathrm{d}\mathbf{x} \\ &= \int_\Omega \mathbf{E} \cdot \nabla \times \mathbf{H} - \nabla \times \mathbf{H} \cdot \mathbf{E} \,\mathrm{d}\mathbf{x} + \int_{\partial\Omega} \mathbf{n} \times \mathbf{E} \cdot \mathbf{H} \,\mathrm{d}\mathbf{a} = 0.\end{aligned}$$

□

For some applications it is not necessary to solve Maxwell's equations in a three dimensional spatial domain, since the problem can be reduced to one or two spatial dimensions. In case of polarized electromagnetic waves, one can derive 1D and 2D settings for Maxwell's equations (cf. [Jac99, Sec. 8.2]):

Transverse Electromagnetic (TEM) Waves

In this case let $\Omega \subset \mathbb{R}$ and the electric and the magnetic field vanish in x_3 direction. Hence $E_3 = H_3 = 0$ and the linear Maxwell's equations (3.14) reduce to two decoupled one-dimensional linear systems

$$\begin{aligned} \mu\partial_t H_2 + \partial_3 E_1 &= 0, \\ \varepsilon\partial_t E_1 + \partial_3 H_2 &= f_1 \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mu\partial_t H_1 - \partial_3 E_2 &= 0, \\ \varepsilon\partial_t E_2 - \partial_3 H_1 &= f_2, \end{aligned}$$

where $J = 2$, $\mathbf{u} = (H_2, E_1)^\top$ or $\mathbf{u} = (H_1, E_2)^\top$ and given data on the right-hand side $\mathbf{f} = (0, f_1)^\top$ or $\mathbf{f} = (0, f_2)^\top$, respectively.

Transverse Magnetic (TM) Waves

In this case let $\Omega \subset \mathbb{R}^2$ and the electric field $\mathbf{E}$ vanishes in x_1 and x_2 direction. Hence $H_3 = E_1 = E_2 = 0$, $J = 3$ and $\mathbf{u} = (H_1, H_2, E_3)^\top$. The two-dimensional TM case will be investigated numerically in Chapter 6 and 8.

Transverse Electric (TE) Waves

In this case let $\Omega \subset \mathbb{R}^2$ and the magnetic field $\mathbf{H}$ vanishes in x_1 and x_2 direction. Hence $H_1 = H_2 = E_3 = 0$, $J = 3$ and $\mathbf{u} = (H_3, E_1, E_2)^\top$.

3.5 Boundary Conditions in Applications

In Remark 3.2 we have seen, that inhomogeneous initial and boundary conditions can be applied by incorporating them into the right-hand side $\mathbf{f}$. For our analysis we assumed a given homogeneous boundary condition for simplicity and incorporated them into the domain of A. However, other boundary conditions are often needed in applications to model problems correctly. In the following we briefly introduce common boundary conditions for electromagnetic problems.

Conducting Boundary Conditions

So far we assumed a perfect electric conducting boundary condition

$$\mathbf{n} \times \mathbf{E} = \mathbf{0}, \qquad \text{on } \partial\Omega.$$

On the other hand, the magnetic counterpart of a perfect conducting boundary condition is given as

$$\mathbf{n} \times \mathbf{H} = \mathbf{0}, \qquad \text{on } \partial\Omega,$$

(cf. [Jac99, Sec. 5.8, Sec. 5.13]). These kind of boundary conditions correspond to reflecting boundary conditions, where the electric and magnetic fields are reflected back (with a phase shift) into Ω without any losses. Physically this means that the fields vanish inside a perfect conductor.

Unbounded Domains

If electromagnetic waves are not encapsulated in constructions with perfectly conducting boundaries, they usually travel up to infinity. Since we have to cut off the computational domain at some point, it is crucial to model the correct boundary conditions in this case. For time-harmonic waves (see Remark 3.5) one can impose the so called *Silver–Müller radiation conditions* and show that they model the decaying behavior of electromagnetic waves correctly (see, e.g., [AK04, Sec. 2.9]). In particular this means that waves cannot enter the computational domain from infinity.

Therefore, so called *perfectly matched layers* (PML) can be added to the computational domain in numerical simulations. Within these layers artificial material parameters are used to damp outgoing waves exponentially. Hence only very few layers of additional cells are needed around the computational domain to achieve a sufficiently large damping. Moreover one has to assure that the waves are not reflected at the interface between computational domain and the PML. For more details on perfectly matched layers in combination with discontinuous Galerkin methods and time-dependent problems see, e.g., [Sch15].

4 A Space-Time Discontinuous Petrov–Galerkin Discretization

In the previous chapter we have seen that for every given right-hand side $\mathbf{f} \in \mathrm{L}_2(Q)^J$ there exists a unique solution $\mathbf{u} \in V$ of the weak first-order evolution equation, such that for all $\mathbf{w} \in W$ equation (4.4) is fulfilled, i.e.,

$$(L\mathbf{u}, \mathbf{w})_{0,Q} = (M\partial_t \mathbf{u} + A\mathbf{u}, \mathbf{w})_{0,Q} = \langle \mathbf{f}, \mathbf{w} \rangle, \qquad \text{for all } \mathbf{w} \in W.$$

In the present chapter we want to derive a discrete version of this problem, to be able to compute approximations $\mathbf{u}_h$ to $\mathbf{u}$ later on. Therefore the spaces V and W are replaced by corresponding finite-dimensional spaces $V_h, W_h \subset \mathrm{L}_2(Q)$. This results in the idea of so called *Galerkin Methods*. A short introduction is given in the following section. Afterwards a special kind of Galerkin method, the so called *Finite Element Method* (FEM), is introduced. Finally, we derive a finite element space-time discretization of our hyperbolic problem (3.1). For this purpose we combine a discontinuous Galerkin (dG) method in space with a continuous Petrov–Galerkin (cPG) method in time. Moreover, we discuss an implementation, where a nodal space-time discretization is used.

4.1 Galerkin Methods

To approximate a solution $\mathbf{u}$ of a partial differential equation with a Galerkin method, we choose finite-dimensional vector spaces V_h, W_h and seek a solution $\mathbf{u}_h \in V_h$ such that

$$b_h(\mathbf{u}_h, \mathbf{w}_h) = \langle \mathbf{f}, \mathbf{w}_h \rangle \tag{4.1}$$

holds for all $\mathbf{w}_h \in W_h$. Here $b_h(.,.) = (L_h.,.)_{0,Q} \colon V_h \times W_h \to \mathbb{R}$ is the discrete bilinear form, which arises from the approximation $L_h = M_h\partial_t + A_h$ of the continuous operator L. Hence b_h is an approximation of the weak formulation

$$b(\mathbf{u}, \mathbf{w}) = \langle \mathbf{f}, \mathbf{w} \rangle, \qquad \text{for all } \mathbf{w} \in W$$

of the original problem (3.7). The space V_h is called *ansatz space* or *trial space*, whereas W_h is called *test space*. If both spaces do not coincide, i.e., $V_h \neq W_h$, the Galerkin method is usually called *Petrov–Galerkin method* or *non-standard Galerkin method* to emphasize this circumstance. A Galerkin method is called *conforming* if $V_h \subset V$ and $W_h \subset W$. Otherwise the method is called *non-conforming*. The term "Galerkin method" only indicates that formulation (4.1) is used to approximate a solution to the original problem. However, to achieve a suitable numerical method it is crucial to choose the finite-dimensional vector spaces V_h and W_h in a correct way. One possibility is the so called finite element method presented in the next section. For a more comprehensive introduction to Galerkin methods and their connections to FEM see, e.g., [Bra13, §4] or [EG04, Sec. 2.2].

4.2 Finite Element Method

To apply a finite element method, a decomposition of the underlying computational domain $\Omega \subset \mathbb{R}^D$, $D \in \mathbb{N}$, is needed. The following definitions build the foundations of suitable decompositions for finite element methods.

Definition 4.1 (Cells and reference cell)**.** A non-empty, open polyhedron $K \subset \Omega$ (e.g., an interval for $D = 1$, quadrilateral or triangle for $D = 2$, hexahedron or tetrahedron for $D = 3$) is called *(spatial) cell*. It is given as the interior of a convex hull which is characterized by the set of vertices $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_D$ (for tetrahedral cells) or $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{2^D-1}$ (for hexahedral cells). A subset $f \subset \partial K$ of the surface, which corresponds to one hyperplane is called *face*. The set of all faces of K is denoted as $\mathcal{F}_K$. The cell diameter is denoted as $h_K = \operatorname{diam}(K)$.
Moreover we define a so called *reference cell* $\widehat{K}$, not necessarily in Ω. Every "physical" cell $K \subset \Omega$ can be represented as the reference cell and a corresponding affine linear mapping $\phi_K \colon \widehat{K} \to K$. In particular ϕ_K is a C^1-diffeomorphism. Hence it is sufficient to define a finite element only once on a reference cell $\widehat{K}$ and use the linear mapping ϕ_K to map it to the "physical" elements. The mapping ϕ_K is usually given by so called *barycentric coordinates* (see Section A.1 for a short introduction on barycentric coordinates on tetrahedral cells).
Furthermore, we define the *Jacobian matrix* of ϕ_K as

$$F_K(\widehat{\mathbf{x}}) = \left(\partial_{\widehat{\mathbf{x}}_j} \phi_{K,i}(\widehat{\mathbf{x}})\right)_{i,j=1,\ldots,D}$$

and denote the corresponding determinant as $J_K(\widehat{\mathbf{x}}) = \det(F_K(\widehat{\mathbf{x}}))$.

Definition 4.2 (Meshes)**.** Assume that Ω has a polyhedral boundary, i.e., $\partial\Omega$ can be represented piecewise by C^1 functions (continuous and differentiable). Then we decompose Ω into a finite set $\mathcal{K}$ of non-overlapping cells $K \subset \Omega$ such that

$$\overline{\Omega} = \bigcup_{K \in \mathcal{K}} \overline{K}.$$

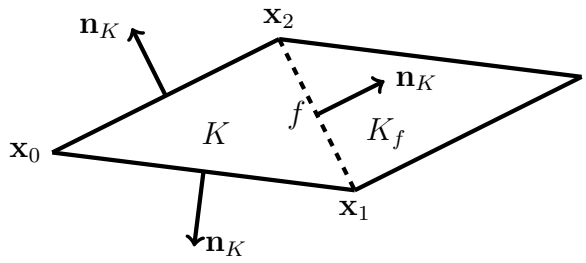

Figure 4.1: Illustration of an element K and a neighboring cell K_f in 2D.

The decomposition $\mathcal{K}$ is denoted as *(spatial) mesh* or *(spatial) grid*. A mesh decomposition is called *admissible* if every intersection of two cells $K_i, K_j \in \mathcal{K}$, $i \neq j$, fulfills exactly one of the following conditions:

a) $\partial K_i \cap \partial K_j = \emptyset$

b) $\partial K_i \cap \partial K_j \in \mathbb{R}^D$ is a vertex of K_i and K_j

c) $\partial K_i \cap \partial K_j \subset \mathbb{R}^D$ is a face of K_i and K_j

In particular it follows that no *hanging nodes* are allowed for admissible meshes.

The set of all faces of a mesh $\mathcal{K}$ is denoted as $\mathcal{F}_\mathcal{K}$. We distinguish between *interior* faces $f \not\subset \partial\Omega$ and *outer* faces $f \subset \partial\Omega$. For a cell K with interior face $f \in \mathcal{F}_K$ we denote the cell $K_f \in \mathcal{K}$ as neighboring cell if $f \in \mathcal{F}_K \cap \mathcal{F}_{K_f} \neq \emptyset$ is a common face. Furthermore let $\mathbf{n}_K$ denote the outer unit normal vector on ∂K (see Figure 4.1).

Definition 4.3 (Mesh size and shape regularity)**.** Let $\mathcal{K}$ be an admissible mesh. The maximal cell diameter of all cells

$$h = \max_{K \in \mathcal{K}} h_K$$

is called *mesh size*. Moreover, a family of meshes $\{\mathcal{K}_h\}_{h>0}$ is called *shape-regular* if there exists an upper bound $\sigma_0 > 0$ such that

$$\sigma_0 \geq \frac{h_K}{d_K}, \qquad \text{for all } h > 0 \text{ and } K \in \mathcal{K}_h.$$

Here d_K is the diameter of a ball with maximal volume, which can be fitted into $\overline{K}$.

The following definitions and remarks build the basis for the finite element discretization presented later on. A general introduction to finite element methods can be found for example in [Bra13] and [EG04]. We use the latter reference to give a small introduction to the main points of finite elements, see [EG04, Sec. 1.2]. For simplicity, we first consider the scalar case for $J = 1$. A vector valued extension for $J > 1$ can be achieved by applying finite elements component-wise. This case is discussed afterwards.

Definition 4.4 (Finite element)**.** In our setting a triplet $(K, H_{h,K}, \Sigma_K)$ is denoted as *finite element*, if it fulfills the following properties:

a) The element $K \in \mathcal{K}$ is a cell of an admissible mesh $\mathcal{K}$ (see Definitions 4.1 and 4.2).

b) $H_{h,K} \subset C(K)$ is a finite subspace of continuous functions. We use polynomial spaces $H_{h,K} = \mathbb{P}_{p_K}(K)$, where

$$\mathbb{P}_{p_K}(K) = \left\{ v_{h,K}(\mathbf{x}) = \sum_{\substack{0 \le i_1,\dots,i_D \le p_K, \\ i_1+\dots+i_D \le p_K}} \alpha_{i_1,\dots,i_D} x_1^{i_1} \dots x_D^{i_D}, \quad \mathbf{x} \in K \right\}$$

is the space of all polynomials with highest degree $p_K \ge 0$ and coefficients $\alpha_{i_1,\dots,i_D} \in \mathbb{R}$. The dimension of $H_{h,K}$ is denoted as $n_K = \dim(H_{h,K})$.

c) Σ_K is a set of n_K linear independent functionals on $H_{h,K}$. Every function (polynomial) $\psi \in H_{h,K}$ can be uniquely determined by using the values of these functionals $l_i \in \Sigma_K$, $1 \le i \le n_k$, i.e., the linear mapping

$$\psi \mapsto (l_1(\psi), l_2(\psi), \dots, l_{n_K}(\psi)) \in \mathbb{R}^{n_K}$$

is bijective and Σ_K is a basis of $\mathcal{L}(H_{h,K}, \mathbb{R})$. Σ_K is called the set of *degrees of freedom* (DoFs). As a consequence of the bijectivity we achieve that there exits a basis $\{\psi_j\}_{j=1,\dots,n_k}$ of $H_{h,K}$ such that

$$l_i(\psi_j) = \delta_{i,j}, \quad 1 \le i, j \le n_K.$$

The basis functions $\{\psi_1, \dots, \psi_{n_K}\}$ are called *shape functions*.

Definition 4.5 (Nodal finite elements)**.** Consider a finite element $(K, \mathbb{P}_{p_K}(K), \Sigma_K)$. Let $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n_K}\}$ be a set of points in K such that for all shape functions $\psi_j \in \mathbb{P}_{p_K}(K)$ it holds that

$$l_i(\psi_j) = \psi_j(\mathbf{x}_i), \quad 1 \le i, j \le n_K.$$

Then $(K, \mathbb{P}_{p_K}(K), \Sigma_K)$ is called *nodal finite element* (or *Lagrange finite element*). The associated points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_K}\}$ are called *nodes* of the finite element. In this context, the set of shape functions $\{\psi_j\}_{j=1,\dots,n_k}$ is denoted as *nodal basis* and we achieve that

$$\psi_j(\mathbf{x}_i) = \delta_{i,j}, \quad 1 \le i, j \le n_K. \tag{4.2}$$

Hence every polynomial $v_h \in \mathbb{P}_{p_K}(K)$ is uniquely defined by its corresponding coefficient vector $\underline{v} = (\underline{v}_1, \dots, \underline{v}_{n_K})^\top \in \mathbb{R}^{n_k}$ and can be represented as

$$v_h = \sum_{i=1}^{n_K} \underline{v}_i \psi_i.$$

The *local nodal interpolation operator* is defined as

$$I_K \colon \mathrm{C}^0(K) \to \mathbb{P}_{p_K}(K), \quad I_K v = \sum_{i=1}^{n_K} v(\mathbf{x}_i)\psi_i.$$

Definition 4.6 (Nodal shape function)**.** Let $K \in \mathcal{K}$ be a cell in $\mathbb{R}^D$ and choose a polynomial degree $p_K \geq 0$. We consider a set of nodes $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_K}\}$ where every element $\mathbf{x}_i$ can be represented by its barycentric coordinates

$$\boldsymbol{\lambda}_i = \begin{cases} \left(\frac{1}{D+1}, \ldots, \frac{1}{D+1}\right), & \text{for } p_K = 0, \\ \left(\frac{i_0}{p_K}, \ldots, \frac{i_D}{p_K}\right),\ 0 \leq i_0, \ldots, i_D \leq p_K,\ i_0 + \ldots + i_D = p_K, & \text{for } p_K > 0. \end{cases}$$

For an introduction to barycentric coordinates see Section A.1 in the appendix. The first three nodal sets are illustrated in Figure 4.2 for different dimensions $D = 1, 2, 3$ and $p_K = 0, 1, 2$. Moreover, let $(K, \mathbb{P}_{p_K}, \Sigma_K)$ be a nodal finite element with respect

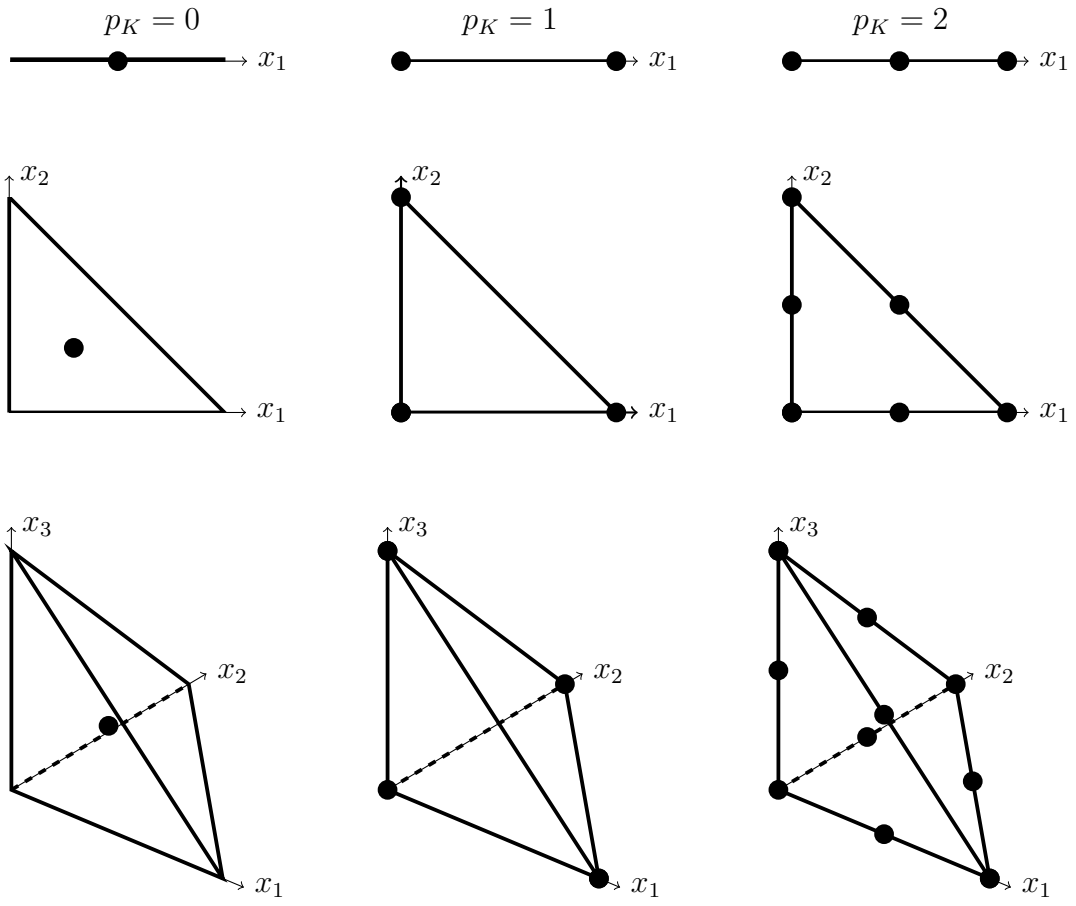

Figure 4.2: Illustration of one-, two- and three-dimensional nodal finite elements.

to the set of nodes with barycentric coordinates. Then the corresponding shape functions are given by Lagrange polynomials and the shape functions are called *nodal* or *Lagrange shape functions*, respectively (see Example A.1 in the appendix).

Remark 4.1 (Implementation)**.** As indicated above in Definition 4.1, we use a reference finite element element $(\widehat{K}, \mathbb{P}_p(\widehat{K}), \Sigma_{\widehat{K}})$ and the affine linear mapping ϕ_K to represent a "physical" finite element $(K, \mathbb{P}_p(K), \Sigma_K)$ for all $K \in \mathcal{K}$. Hence we achieve that a shape function $\psi \in \mathbb{P}_{p_K}(K)$ and its gradient can be represented as

$$\psi = \widehat{\psi} \circ \phi_K^{-1} \quad \text{and} \quad \nabla\psi = F_K^{-\top}\, \widehat{\nabla}\widehat{\psi} \circ \phi_K^{-1},$$

where $\widehat{\psi} \in \mathbb{P}_p(\widehat{K})$ is a reference shape function and F_K the Jacobian matrix.
Integrals are evaluated by numerical integration. Therefore Gaussian quadrature formulas $Q_{\widehat{K}}^{\text{Gauss}}$ with weights $\{\omega_i\}_{i=1,\dots,n_{\text{Gauss}}}$ and quadrature points $\{\widehat{\mathbf{q}}_i\}_{i=1,\dots,n_{\text{Gauss}}}$ on the reference cell are used. Thus, the integral over a polynomial $v_h \in \mathbb{P}_{p_K}(K)$ can be computed as

$$\int_K v_h(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\widehat{K}} \widehat{v}_h(\widehat{\mathbf{x}}) J_K(\widehat{\mathbf{x}})\,\mathrm{d}\widehat{\mathbf{x}} = Q_{\widehat{K}}^{\text{Gauss}}(\widehat{v}_h J_K) = \sum_{i=1}^{n_{\text{Gauss}}} \widehat{v}_h(\widehat{\mathbf{q}}_i)\,\omega_i J_K(\widehat{\mathbf{q}}_i),$$

where $\widehat{v}_h = v_h \circ \phi_K^{-1} \in \mathbb{P}_{p_K}(\widehat{K})$ and $J_K = \det(F_K)$. Note that (depending on the integrand v_h) the order of the quadrature formula has to be chosen sufficiently large.

Remark 4.2 (Vector valued case)**.** In case of vector valued problems, e.g., in the Maxwell case, $J > 1$ holds and thus $H_{h,K} = \mathbb{P}_{p_K}(K)^J$. Here we choose a finite element $(K, \mathbb{P}_{p_K}, \Sigma_K)$ for every component. Hence, every node $\mathbf{x}_i$ is assigned to J shape functions and we receive a vector valued version of property (4.2) for all shape functions $\boldsymbol{\psi}_j \in \mathbb{P}_{p_K}(K)^J$, i.e.,

$$\big(\boldsymbol{\psi}_j(\mathbf{x}_i)\big)_k = \delta_{i,j \bmod J}\ \delta_{k,\lceil j/J \rceil}, \quad 1 \le i \le \dim(\mathbb{P}_{p_K}),\ 1 \le j \le n_{p_K},\ 1 \le k \le J.$$

Remark 4.3 (Dimensions)**.** Let $\mathcal{K}$ be a decomposition of intervals, triangles or tetrahedrons, the dimension of the local polynomial space $\mathbb{P}_{p_K}(K)$, $K \in \mathcal{K}$, can be calculated as

$$\dim(\mathbb{P}_{p_K}(K)) = \binom{D + p_K}{p_K} = \begin{cases} p_K + 1, & \text{for } D = 1, \\ \frac{1}{2}(p_K+1)(p_K+2), & \text{for } D = 2, \\ \frac{1}{6}(p_K+1)(p_K+2)(p_K+3), & \text{for } D = 3. \end{cases}$$

Hence the dimension of $H_{h,K}$ is $n_K = \dim(\mathbb{P}_{p_K}(K))^J$.

Next we want to construct an ansatz space V_h. In principle one has the possibility to choose a conforming or non-conforming finite element discretization. This choice is crucial, since it will lead to two different finite element discretizations with different advantages and disadvantages. In the following section, we will briefly discuss the conforming case and resulting difficulties for Maxwell's equations. Afterwards we focus on the non-conforming case, which is finally used in this work. A detailed description of conforming finite elements for Maxwell's equations is given in [Hip02] or [Mon04]. For an introduction to general non-conforming finite element discretizations see [DPE12]. A description of non-conforming finite elements for Maxwell's equations can be found in [HW08].

4.2.1 Conforming Finite Element Discretization

We have seen that the ansatz space is given as $V = \mathcal{D}(A) = \mathrm{H}(\mathrm{curl}, \Omega) \times \mathrm{H}_0(\mathrm{curl}, \Omega)$ in the Maxwell case (cf. Remark 3.6). Taking standard continuous piecewise linear finite elements is not a reasonable choice, since the corresponding finite element space V_h is not dense in V for the limit case $h \to 0$. In this case, a finite element solution $\mathbf{u}_h$ will convergence towards a projection of the continuous solution into $\mathrm{H}^1(\Omega)^3 \times \mathrm{H}^1_0(\Omega)^3$ and not the continuous solution $\mathbf{u}$ itself. This results in approximations which are wrong in general, since the solution might be not regular enough, e.g., $\mathbf{u} \in \mathrm{H}^s(\Omega)^3 \times \mathrm{H}^s_0(\Omega)^3$ for $s < 1$ (cf. [Hip02, Thm. 6.3]). Moreover, this approach contains the danger of showing well convergence behavior and thus misleading trust in numerical results.

To circumvent this issue, one has to choose the correct H(curl)-conforming finite element space. Such an approach leads to the edge element discretization introduced by Nédélec [Néd80].

4.2.2 Non-Conforming Finite Element Discretization

Now we focus on the construction of a non-conforming finite element discretization by using so called *discontinuous Galerkin finite element methods* (DG-FEM). Depending on the point of view DG-FEM can be understood as a finite volume method. However, instead of using only constant functions to approximate solutions locally on one cell, the framework is extended by additionally allowing polynomials of higher degree $p \geq 0$. Thus for polynomial degree $p = 0$ a discontinuous Galerkin method coincides with a finite volume method. On the other hand DG-FEM can be seen as a finite element method, where we allow discontinuous ansatz and test spaces. Compared to conforming discretizations, non-conforming methods require more degrees of freedom on the same mesh. However, they are able to adapt to non-regular solutions more accurately. Due to that and the grater flexibility and easier implementation, we stick to a finite element framework for discontinuous Galerkin methods throughout this work.

The practical use of DG-FEM for first-order partial differential equations started almost 40 years ago by applying the method to a steady linear transport example (see [RH73]). In the following years the method was applied to time-dependent first-order problems by first using explicit Euler [CC89] and Runge–Kutta schemes later on, see, e.g., [CS91]. For problems with an additional diffusion term, *interior penalty* techniques for DG-FEM became very popular (see, e.g., [Bab73]). Meanwhile discontinuous Galerkin methods combined with Runge–Kutta methods are standard methods for solving partial differential equations as several textbooks indicate, see, e.g., [DPE12], [HW08]. On the other hand the development of new discontinuous Galerkin discretizations and their application is still a broad field

of open research. Examples are the investigation of local time stepping and p-adaptivity (cf. [DKT07]) or the progress in space-time Petrov–Galerkin methods (cf. [DG14], [EDCM14]).

4.3 Discontinuous Galerkin Operators in Space

We use the definitions and properties of finite elements, stated in the previous section, to construct a suitable non-conforming discontinuous Galerkin finite element discretization. We start by constructing a finite dimensional subspace $H_h \subset H \subset \mathrm{L}_2(\Omega)^J$ for the spatial discretization. Therefore we follow the approach stated in [HPS+15] and also used in [DFW16]. As introduced above, we select a polynomial degree $p_K \geq 0$ for every cell K and define the local spaces $H_{h,K} = \mathbb{P}_{p_K}(K)^J$. Taking all subspaces together leads to the finite, global discontinuous Galerkin space

$$H_h = \{\mathbf{v}_h \in \mathrm{L}_2(\Omega)^J \colon \mathbf{v}_h|_K \in H_{h,K} \text{ for all } K \in \mathcal{K}\}.$$

We want to define the discrete spatial operators M_h, A_h via their weak forms. Hence we define the discrete mass operator $M_h \in \mathcal{L}(H_h, H_h)$ by the *Galerkin approximation* of M as

$$(M_h \mathbf{v}_h, \mathbf{w}_h)_{0,\Omega} = (M\mathbf{v}_h, \mathbf{w}_h)_{0,\Omega} \tag{4.3}$$

for all $\mathbf{v}_h, \mathbf{w}_h \in H_h$. M_h is represented by a block-diagonal positive definite matrix. Since A contains spatial derivatives and the Galerkin space H_h is discontinuous, the Galerkin approximation, stated above, cannot be used to define the discrete operator A_h. Instead, we use the following approach. Multiply $A\mathbf{v} = \nabla \cdot \mathbf{F}(\mathbf{v})$ with a smooth test function $\boldsymbol{\phi}_K$, integrate by parts over one element $K \in \mathcal{K}$ and achieve for every smooth ansatz function $\mathbf{v}$ that

$$(A\mathbf{v}, \boldsymbol{\phi}_K)_{0,K} = -(\mathbf{F}(\mathbf{v}), \nabla\boldsymbol{\phi}_K)_{0,K} + \sum_{f \in \mathcal{F}_K} (\mathbf{n}_K \cdot \mathbf{F}(\mathbf{v}), \boldsymbol{\phi}_K)_{0,f}. \tag{4.4}$$

As shown in Lemma 3.1, $\mathbf{n}_K \cdot \mathbf{F}$ is a symmetric $J \times J$ matrix. We now extend this local approach to the whole spatial domain Ω. Therefore we couple two neighboring elements by introducing a new flux function.

Definition 4.7 (Numerical flux)**.** Let $f = \partial\Omega_\mathrm{L} \cap \partial\Omega_\mathrm{R}$ be a non-empty interface of two disjoint open sets $\Omega_L, \Omega_R \subset \mathbb{R}^D$ and $\mathbf{n}_f \in \mathbb{R}^D$ a unit normal vector to f. A function $\mathbf{n}_f \cdot \mathbf{F}_f^{\mathrm{num}} \colon \mathbb{R}^J \times \mathbb{R}^J \to \mathbb{R}^J$ across the interface f is called *numerical flux* to a flux $\mathbf{F}$, if it is consistent, i.e.,

$$\mathbf{n}_f \cdot \Big(\mathbf{F}_f^{\mathrm{num}}(\mathbf{z}, \mathbf{z}) - \mathbf{F}(\mathbf{z})\Big) = \mathbf{0}, \qquad \text{for all } \mathbf{z} \in \mathbb{R}^J$$

and $\mathbf{F}_f^{\mathrm{num}}$ is Lipschitz continuous, i.e., for all $\mathbf{v}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^J$ it holds that

$$\|\mathbf{n}_f \cdot \Big(\mathbf{F}_f^{\mathrm{num}}(\mathbf{v}, \mathbf{w}) - \mathbf{F}(\mathbf{z})\Big)\|_2 \leq C_{\mathrm{num}} \max\left\{\|\mathbf{v} - \mathbf{z}\|_2, \|\mathbf{w} - \mathbf{z}\|_2\right\},$$

with Lipschitz constant $C_{\mathrm{num}} \geq 0$.

Remark 4.4 (Numerical flux for DG methods)**.** In case of discontinuous Galerkin discretizations we consider fluxes across cell interfaces $f = \partial K \cap \partial K_f$ for two neighboring cells $K, K_f \in \mathcal{K}$ and use the notation

$$\mathbf{F}^{\text{num}}_{K,f}(\mathbf{v}_h) = \mathbf{F}^{\text{num}}_{K,f}(\mathbf{v}_{h,K}, \mathbf{v}_{h,K_f}), \qquad \text{for all } \mathbf{v}_h \in H_h.$$

Throughout this work we omit the indices K and f if the associated cell and face can be derived from the context. Due to consistency of a numerical flux it holds that

$$\mathbf{n}_K \cdot \Big(\mathbf{F}^{\text{num}}_{K,f}(\mathbf{v}) - \mathbf{F}(\mathbf{v})\Big) = \mathbf{0}, \qquad \text{for all } \mathbf{v} \in \mathcal{D}(A)$$

on all faces $f \in \mathcal{F}_K$. Furthermore, $\mathbf{n}_K \cdot (\mathbf{F}^{\text{num}}_{K,f}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,K}))$ depends only on the jump $[\mathbf{v}_h]_{K,f}$ and not on the absolute values, since the numerical flux is Lipschitz continuous.

With this definition we can define the discrete linear operator $A_h \in \mathcal{L}(H_h, H_h)$ locally as

$$(A_h \mathbf{v}_h, \boldsymbol{\phi}_{h,K})_{0,K} = -(\mathbf{F}(\mathbf{v}_{h,K}), \nabla \boldsymbol{\phi}_{h,K})_{0,K} + \sum_{f \in \mathcal{F}_K} (\mathbf{n}_K \cdot \mathbf{F}^{\text{num}}(\mathbf{v}_h), \boldsymbol{\phi}_{h,K})_{0,f}, \tag{4.5}$$

for any ansatz function $\mathbf{v}_h \in H_h$ and test function $\boldsymbol{\phi}_{h,K} \in H_{h,K}$. By again using integration by parts, one obtains

$$\begin{aligned}(A_h \mathbf{v}_h, &\boldsymbol{\phi}_{h,K})_{0,K} = \\ &(\nabla \cdot \mathbf{F}(\mathbf{v}_K), \boldsymbol{\phi}_{h,K})_{0,K} + \sum_{f \in \mathcal{F}_K} (\mathbf{n}_K \cdot (\mathbf{F}^{\text{num}}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,K})), \boldsymbol{\phi}_{h,K})_{0,f}.\end{aligned} \tag{4.6}$$

Moreover it yields that, for some $\mathbf{v} \in \mathcal{D}(A)$ and test function $\boldsymbol{\phi}_h \in H_h$

$$(A\mathbf{v}, \boldsymbol{\phi}_h)_{0,\Omega} = (A_h \mathbf{v}, \boldsymbol{\phi}_h)_{0,\Omega}, \tag{4.7}$$

due to the consistency of the numerical flux. Formulation (4.5) is usually known as the *weak form* of the operator A_h, whereas (4.6) is denoted as *strong form*. Mathematically both formulations are equivalent, but from a computational point of view differences can occur. We do not need any smoothness requirements for the test function in the strong form. Furthermore, the sum over the faces only depends on the jump of the ansatz function in the strong case. Thus we will use the strong formulation for numerical computation in this work. The construction of the numerical flux is crucial to achieve a stable numerical scheme and is discussed in the following section.

4.4 Upwind Flux

Not every arbitrary choice of a numerical flux will lead to a stable numerical scheme. For example, the so called *centered flux*, which is defined as the mean value of a

function on a face, is generally unstable for hyperbolic problems (cf. [LeV08, Sec. 3.5]). We follow the ideas given in [HPS+15] and [LeV08] and construct a numerical flux by investigating the behavior of so called *shock solutions*. These solutions are discontinuous along a given interface, but regular in the neighborhood of the interface.

To find weak shock solutions, we consider the weak formulation of the first-order evolution equation

$$(M\mathbf{u}, \partial_t \boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} + (\mathbf{F}(\mathbf{u}), \nabla\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} = 0, \tag{4.8}$$

which is obtained by multiplying (3.1) with a continuous test function $\boldsymbol{\phi}$ with compact support, integrating over the space-time domain $\mathbb{R}^D \times (0,T)$ and applying the integration by parts formula.

Definition 4.8 (Riemann problem)**.** Let $\mathbf{n} \in \mathbb{R}^D \setminus \{\mathbf{0}\}$ be an arbitrary unit vector. Then $\mathbb{R}^D = \overline{\Omega_\mathrm{L} \cup \Omega_\mathrm{R}}$ can be divided into two open subsets $\Omega_\mathrm{L} = \{\mathbf{x} \in \mathbb{R}^D \colon \mathbf{n}\cdot\mathbf{x} < 0\}$ and $\Omega_\mathrm{R} = \{\mathbf{x} \in \mathbb{R}^D \colon \mathbf{n}\cdot\mathbf{x} > 0\}$. One now seeks a weak solution $\mathbf{u}\colon \mathbb{R}^D \times (0,T) \to \mathbb{R}^J$ such that the weak first-oder evolution equation (4.8) is fulfilled for a discontinuous initial condition

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) = \begin{cases} \mathbf{u}_\mathrm{L}, & \text{for all } \mathbf{x} \in \Omega_\mathrm{L}, \\ \mathbf{u}_\mathrm{R}, & \text{for all } \mathbf{x} \in \Omega_\mathrm{R}, \end{cases}$$

with $\mathbf{u}_\mathrm{L}, \mathbf{u}_\mathrm{R} \in \mathbb{R}^J$ and constant $M = M_\mathrm{L}$ in Ω_L or $M = M_\mathrm{R}$ in Ω_R, respectively. This problem is called *Riemann problem* and the corresponding weak shock solution $\mathbf{u}$ is called *Riemann solution*.

To solve the Riemann problem we first look for separate solutions on Ω_L and Ω_R. From Lemma 3.1 we know that $B_\mathbf{n} = \mathbf{n}\cdot\mathbf{F}$ is diagonalizable with real eigenvalues. Hence let $[\lambda_{i,\mathrm{L}}, \mathbf{c}_{i,\mathrm{L}}] \in \mathbb{R}\times\mathbb{R}^J$, $i = 1,\dots,J$, be the M_L-orthogonal eigenpairs of the eigenvalue problem

$$B_\mathbf{n}\mathbf{c}_{i,\mathrm{L}} = \lambda_{i,\mathrm{L}} M_\mathrm{L}\mathbf{c}_{i,\mathrm{L}}, \quad \text{where } \mathbf{c}_{i,\mathrm{L}} \cdot M_\mathrm{L}\mathbf{c}_{j,\mathrm{L}} = 0 \quad (\text{for } i \neq j). \tag{4.9}$$

and analogously let $[\lambda_{i,\mathrm{R}}, \mathbf{c}_{i,\mathrm{R}}] \in \mathbb{R}\times\mathbb{R}^J$, $i = 1,\dots,J$, be the M_R-orthogonal eigenpairs of the eigenvalue problem

$$B_\mathbf{n}\mathbf{c}_{i,\mathrm{R}} = \lambda_{i,\mathrm{R}} M_\mathrm{R}\mathbf{c}_{i,\mathrm{R}}, \quad \text{where } \mathbf{c}_{i,\mathrm{R}} \cdot M_\mathrm{R}\mathbf{c}_{j,\mathrm{R}} = 0 \quad (\text{for } i \neq j). \tag{4.10}$$

Lemma 4.1 ([HPS+15, Sec. 3.1])**.** *A general Riemann solution on $\Omega_L \cup \Omega_R$ is given by*

$$\mathbf{u}(\mathbf{x},t) = \begin{cases} \mathbf{u}_L + \sum_{\mathbf{x}\cdot\mathbf{n} - \lambda_{i,L}t > 0} b_{i,L}\mathbf{c}_{i,L}, & \text{if } \mathbf{x} \in \Omega_L, \\ \mathbf{u}_R + \sum_{\mathbf{x}\cdot\mathbf{n} - \lambda_{i,R}t < 0} b_{i,R}\mathbf{c}_{i,R}, & \text{if } \mathbf{x} \in \Omega_R, \end{cases}$$

for $i = 1,\dots,J$ and arbitrary real coefficients $b_{i,L}$, $b_{i,R}$.

Proof. It is sufficient to show that

$$\widetilde{\mathbf{u}} = \begin{cases} \sum_{\mathbf{x}\cdot\mathbf{n}-\lambda_{i,\mathrm{L}}t>0} b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}, & \text{if } \mathbf{x}\in\Omega_{\mathrm{L}}, \\ \sum_{\mathbf{x}\cdot\mathbf{n}-\lambda_{i,\mathrm{R}}t<0} b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}, & \text{if } \mathbf{x}\in\Omega_{\mathrm{R}}, \end{cases}$$

is a solution of the Riemann problem

$$\begin{aligned} &(M\widetilde{\mathbf{u}}, \partial_t\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} + (\mathbf{F}(\widetilde{\mathbf{u}}), \nabla\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} \\ =& -(\nabla\cdot\mathbf{F}(\mathbf{u}_{\mathrm{L}}), \boldsymbol{\phi})_{0,\Omega_{\mathrm{L}}\times(0,T)} - (\nabla\cdot\mathbf{F}(\mathbf{u}_{\mathrm{R}}), \boldsymbol{\phi})_{0,\Omega_{\mathrm{R}}\times(0,T)} \end{aligned}$$

with homogeneous initial data, since this is an equivalent formulation to (4.8) with incorporated initial condition (cf. Remark 3.2). Since $\mathbf{u}_{\mathrm{L}}$ and $\mathbf{u}_{\mathrm{R}}$ are constant, the right-hand side vanishes and the Riemann problem reduces to

$$(M\widetilde{\mathbf{u}}, \partial_t\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} + (\mathbf{F}(\widetilde{\mathbf{u}}), \nabla\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} = 0.$$

We prove that $\widetilde{\mathbf{u}}$ solves this problem by splitting up the integrals. This results in

$$\begin{aligned} &(M\widetilde{\mathbf{u}}, \partial_t\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} + (\mathbf{F}(\widetilde{\mathbf{u}}), \nabla\boldsymbol{\phi})_{0,\mathbb{R}^D\times(0,T)} \\ = &\sum_{i=1}^{J}\Big\{(M_L b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}, \partial_t\boldsymbol{\phi})_{0,Q_{i,\mathrm{L}}} + (\mathbf{F}(b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}), \nabla\boldsymbol{\phi})_{0,Q_{i,\mathrm{L}}}\Big\} \\ &+\sum_{i=1}^{J}\Big\{(M_R b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}, \partial_t\boldsymbol{\phi})_{0,Q_{i,\mathrm{R}}} + (\mathbf{F}(b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}), \nabla\boldsymbol{\phi})_{0,Q_{i,\mathrm{R}}}\Big\}, \end{aligned}$$

where we integrate over $Q_{i,\mathrm{L}} = \{(\mathbf{x},t)\in\Omega_{\mathrm{L}}\times(0,T)\colon \mathbf{x}\cdot\mathbf{n}-\lambda_{i,\mathrm{L}}t>0\}$ and $Q_{i,\mathrm{R}} = \{(\mathbf{x},t)\in\Omega_{\mathrm{R}}\times(0,T)\colon \mathbf{x}\cdot\mathbf{n}-\lambda_{i,\mathrm{R}}t<0\}$, respectively. Using the integration by parts formula within the first sum leads to

$$\begin{aligned} &\sum_{i=1}^{J}\Big\{(M_L b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}, \partial_t\boldsymbol{\phi})_{0,Q_{i,\mathrm{L}}} + (\mathbf{F}(b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}), \nabla\boldsymbol{\phi})_{0,Q_{i,\mathrm{L}}}\Big\} \\ =&\sum_{i=1}^{J}(-\lambda_{i,\mathrm{L}}M_L b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}} + \mathbf{n}\cdot\mathbf{F}(b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}), \boldsymbol{\phi})_{0,\partial Q_{i,\mathrm{L}}} \\ =&\sum_{i=1}^{J}(-\lambda_{i,\mathrm{L}}M_L b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}} + B_{\mathbf{n}} b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}, \boldsymbol{\phi})_{0,\partial Q_{i,\mathrm{L}}} = 0. \end{aligned}$$

Here we used the definition of the eigenvalue problems (4.9) and that the volume term vanishes, since $b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}$ is constant. Applying the same techniques to $Q_{i,\mathrm{R}}$ and the second sum proves the assertion. □

Note that the Riemann solution is piecewise constant, as a direct consequence of this lemma. Furthermore, the number of intermediate steps is related to the characteristic wave speeds, i.e., the eigenvalues λ_i of $B_{\mathbf{n}}$. A typical situation is illustrated in Figure 4.3.

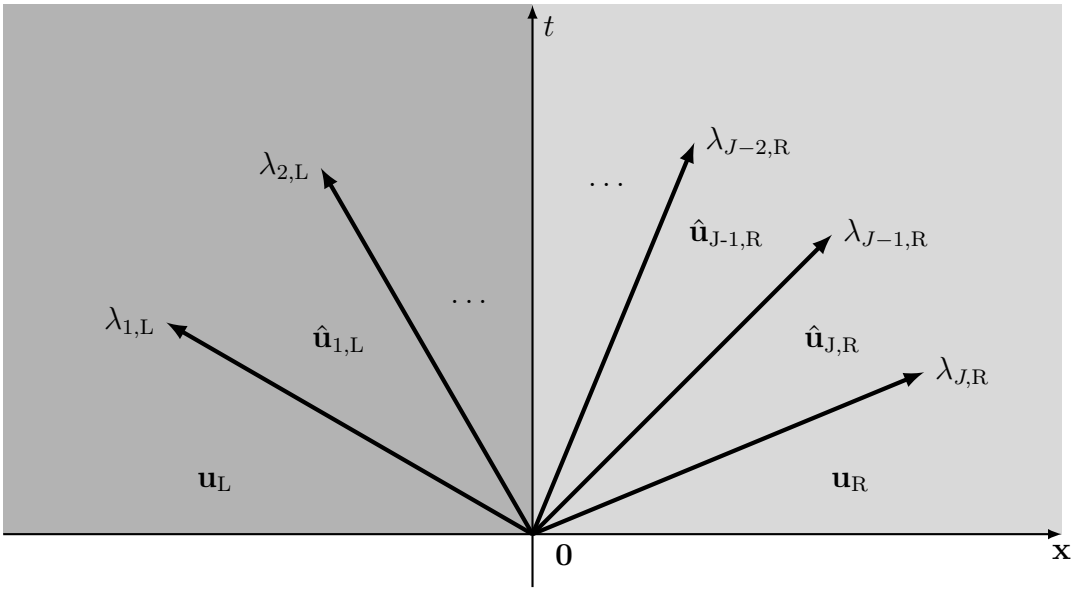

Figure 4.3: Riemann solution with intermediate states $\hat{\mathbf{u}}_{1,\mathrm{L}}, \hat{\mathbf{u}}_{2,\mathrm{L}}, \ldots, \hat{\mathbf{u}}_{J\text{-}1,\mathrm{R}}, \hat{\mathbf{u}}_{J,\mathrm{R}}$

To obtain a Riemann solution in $\mathbb{R}^D$, continuity of the flux across the interface $f_{\mathrm{L,R}} = \partial\Omega_{\mathrm{L}} \cap \partial\Omega_{\mathrm{R}} = \{\mathbf{x} \in \mathbb{R}^D \colon \mathbf{n} \cdot \mathbf{x} = 0\}$ is required, i.e.,

$$B_{\mathbf{n}}\left(\mathbf{u}_{\mathrm{L}} + \sum_{\lambda_{i,\mathrm{L}}<0} b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}\right) = B_{\mathbf{n}}\left(\mathbf{u}_{\mathrm{R}} + \sum_{\lambda_{i,\mathrm{R}}>0} b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}\right). \tag{4.11}$$

This equation is known as *Rankine–Hugoniot (jump-)condition*, see [LeV08, Sec. 3.6]. It can be used to determine the unknown coefficients $b_{i,\mathrm{L}}$ and $b_{i,\mathrm{R}}$ by solving

$$\begin{aligned}
\mathbf{c}_{j,\mathrm{R}} \cdot B_{\mathbf{n}}(\mathbf{u}_{\mathrm{R}} - \mathbf{u}_{\mathrm{L}}) = \mathbf{c}_{j,\mathrm{R}} \cdot B_{\mathbf{n}}[\mathbf{u}_0]_{\Omega_{\mathrm{L}},f_{\mathrm{L,R}}} &= \mathbf{c}_{j,\mathrm{R}} \cdot \sum_{\lambda_{i,\mathrm{L}}<0} B_{\mathbf{n}} b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}, \quad \text{for } \lambda_{j,\mathrm{R}} < 0, \\
\mathbf{c}_{j,\mathrm{L}} \cdot B_{\mathbf{n}}(\mathbf{u}_{\mathrm{L}} - \mathbf{u}_{\mathrm{R}}) = \mathbf{c}_{j,\mathrm{L}} \cdot B_{\mathbf{n}}[\mathbf{u}_0]_{\Omega_{\mathrm{R}},f_{\mathrm{L,R}}} &= \mathbf{c}_{j,\mathrm{L}} \cdot \sum_{\lambda_{i,\mathrm{R}}>0} B_{\mathbf{n}} b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}, \quad \text{for } \lambda_{j,\mathrm{L}} > 0.
\end{aligned}$$

We conclude that $b_{i,\mathrm{L}}$ and $b_{i,\mathrm{R}}$ only dependent on the jump $[\mathbf{u}_0]_{\Omega_{\mathrm{L}},f_{\mathrm{L,R}}}$. Moreover, one can define a numerical flux via the inflowing part of the flux through the interface $f_{\mathrm{L,R}}$ as

$$\begin{aligned}
\mathbf{n} \cdot \mathbf{F}^{\mathrm{num}}_{f_{\mathrm{L,R}}}(\mathbf{u}_0) =& \mathbf{n} \cdot \mathbf{F}^{\mathrm{num}}_{f_{\mathrm{L,R}}}(\mathbf{u}_{\mathrm{L}}, \mathbf{u}_{\mathrm{R}}) \\
=& B_{\mathbf{n}}\left(\mathbf{u}_{\mathrm{L}} + \sum_{\lambda_{i,\mathrm{L}}<0} b_{i,\mathrm{L}}\mathbf{c}_{i,\mathrm{L}}\right) = B_{\mathbf{n}}\left(\mathbf{u}_{\mathrm{R}} + \sum_{\lambda_{i,\mathrm{R}}>0} b_{i,\mathrm{R}}\mathbf{c}_{i,\mathrm{R}}\right).
\end{aligned} \tag{4.12}$$

This numerical flux is known as *upwind* flux, see, e.g. [HW08, Sec. 2.4] or [DPE12, Sec. 3.2.2.2].

Lemma 4.2. *If M can be replaced by a piece-wise constant scalar, e.g., in isotropic or homogeneous materials, the sums in (4.12) can be computed explicitly and the upwind flux can be written in a more elegant way, as*

$$\begin{aligned}\mathbf{n}\cdot\mathbf{F}^{\mathrm{num}}(\mathbf{u}_0) &= B_{\mathbf{n}}\mathbf{u}_L + \frac{1}{2}(B_{\mathbf{n}} - |B_{\mathbf{n}}|)[\mathbf{u}_0]_{\Omega_L,f_{L,R}},\\ &= B_{\mathbf{n}}\mathbf{u}_R + \frac{1}{2}(B_{\mathbf{n}} + |B_{\mathbf{n}}|)[\mathbf{u}_0]_{\Omega_R,f_{L,R}}.\end{aligned} \tag{4.13}$$

Here we use the component-wise absolute values $|\Lambda| = diag\big(|\lambda_1|,\dots,|\lambda_J|\big)$ of the diagonal eigenvalue matrix $\Lambda = diag\big(\lambda_1,\dots,\lambda_J\big)$ to define $|B_{\mathbf{n}}| = S|\Lambda|S^{-1}$, where $B_{\mathbf{n}} = S\Lambda S^{-1}$. Hence it holds that

$$\begin{aligned}\frac{1}{2}(B_{\mathbf{n}} - |B_{\mathbf{n}}|) &= S(\Lambda - |\Lambda|)S^{-1} = S\Lambda^- S^{-1},\\ \frac{1}{2}(B_{\mathbf{n}} + |B_{\mathbf{n}}|) &= S(\Lambda + |\Lambda|)S^{-1} = S\Lambda^+ S^{-1},\end{aligned} \tag{4.14}$$

where Λ^- and Λ^+ are negative semi-definite and positive semi-definite matrices, respectively, with $\Lambda = \Lambda^- + \Lambda^+$.

Proof. Since $M_{\mathrm{L}} > 0$ and $M_{\mathrm{R}} > 0$ are scalar, we achieve that $\lambda_i = M_{\mathrm{L}}\lambda_{i,\mathrm{L}} = M_{\mathrm{R}}\lambda_{i,\mathrm{R}}$ and $\mathbf{c}_i = \mathbf{c}_{i,\mathrm{L}} = \mathbf{c}_{i,\mathrm{L}}$ for $i = 1,\dots,J$. Due to the Rankine–Hugoniot (4.11) condition it yields that

$$\begin{aligned}\mathbf{c}_j\cdot B_{\mathbf{n}}[\mathbf{u}_0]_{\Omega_{\mathrm{L}},f_{\mathrm{L,R}}} &= \mathbf{c}_j\cdot B_{\mathbf{n}}\sum_{\lambda_i<0} b_{i,\mathrm{L}}\mathbf{c}_i, \qquad \text{for } \lambda_j < 0,\\ \mathbf{c}_j\cdot B_{\mathbf{n}}[\mathbf{u}_0]_{\Omega_{\mathrm{R}},f_{\mathrm{L,R}}} &= \mathbf{c}_j\cdot B_{\mathbf{n}}\sum_{\lambda_i>0} b_{i,\mathrm{R}}\mathbf{c}_i, \qquad \text{for } \lambda_j > 0\end{aligned}$$

and thus

$$\sum_{\lambda_i<0} b_{i,\mathrm{L}}\mathbf{c}_i = \begin{cases}[\mathbf{u}_0]_{\Omega_{\mathrm{L}},f_{\mathrm{L,R}}}, & \text{for } \lambda_j<0,\\ 0, & \text{otherwise,}\end{cases} \qquad \sum_{\lambda_i>0} b_{i,\mathrm{R}}\mathbf{c}_i = \begin{cases}[\mathbf{u}_0]_{\Omega_{\mathrm{R}},f_{\mathrm{L,R}}}, & \text{for } \lambda_j>0,\\ 0, & \text{otherwise.}\end{cases}$$

This representation is used in (4.12) together with notation (4.14) to obtain (4.13). □

Remark 4.5. In case of matrix-valued M, representations similar to (4.13) can be obtained by using the diagonalization $M^{-1}B_{\mathbf{n}} = S\Lambda S^{-1}$ (cf. [HW08, Sec. 2.4]).

Assume that M is a constant scalar within on cell $K \in \mathcal{K}$. Then we conclude from equation (4.13) that a stable upwind flux on every $f \in \mathcal{F}_K$ is given cell-wise as

$$\mathbf{n}_K\cdot\Big(\mathbf{F}^{\mathrm{num}}_{K,f}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,K})\Big) = \frac{1}{2}(\mathbf{n}_K\cdot\mathbf{B} - |\mathbf{n}_K\cdot\mathbf{B}|)[\mathbf{v}_h]_{K,f}$$

for all $\mathbf{v}_h \in H_h$ in case of the strong formulation (4.6), where $\mathbf{B} = (B_1,\dots,B_D)^\top$. This representation is used in the following lemma.

Lemma 4.3. *Let M be a piece-wise constant scalar within each cell $K \in \mathcal{K}$, then it holds that*

$$\sum_{K\in\mathcal{K}} (\mathbf{n}_K \cdot \mathbf{F}_K^{\text{num}}(\mathbf{v}_h), \mathbf{v})_{0,\partial K} = 0$$

for all $\mathbf{v}_h \in H_h$ and $\mathbf{v} \in \mathcal{D}(A) \cap \mathrm{H}^1(\Omega)^{\mathrm{J}}$.

Proof. To show this property we first conclude that

$$\begin{aligned} & \Big(\mathbf{n}_K \cdot \mathbf{B}\mathbf{v}_{h,K} + \mathbf{n}_{K_f} \cdot \mathbf{B}\mathbf{v}_{h,K_f}\Big) \cdot \mathbf{v} \\ = \; & \Big(\mathbf{n}_K \cdot \mathbf{B}\mathbf{v}_{h,K} - \mathbf{n}_K \cdot \mathbf{B}\mathbf{v}_{h,K_f}\Big) \cdot \mathbf{v} \\ = & -\Big(\mathbf{n}_K \cdot \mathbf{B}[\mathbf{v}_h]_{K,f}\Big) \cdot \mathbf{v} = -\Big(\mathbf{n}_{K_f} \cdot \mathbf{B}[\mathbf{v}_h]_{K_f,f}\Big) \cdot \mathbf{v} \end{aligned}$$

and hence

$$\begin{aligned} & \Big(\mathbf{n}_K \cdot \mathbf{B}\mathbf{v}_{h,K} + \mathbf{n}_{K_f} \cdot \mathbf{B}\mathbf{v}_{h,K_f}\Big) \cdot \mathbf{v} \\ = & -\frac{1}{2}\Big(\mathbf{n}_K \cdot \mathbf{B}[\mathbf{v}_h]_{K,f} + \mathbf{n}_{K_f} \cdot \mathbf{B}[\mathbf{v}_h]_{K_f,f}\Big) \cdot \mathbf{v}. \end{aligned}$$

We use the representation of the upwind flux given in Lemma 4.2 to verify that

$$\begin{aligned} & \sum_{K\in\mathcal{K}} (\mathbf{n}_K \cdot \mathbf{F}_K^{\text{num}}(\mathbf{v}_h), \mathbf{v})_{0,\partial K} \\ = & \sum_{K\in\mathcal{K}} (\mathbf{n}_K \cdot \mathbf{B}\mathbf{v}_{h,K}, \mathbf{v})_{0,\partial K} + \frac{1}{2}\Big((\mathbf{n}_K \cdot \mathbf{B} - |\mathbf{n}_K \cdot \mathbf{B}|)[\mathbf{v}_h]_{K,f}, \mathbf{v}\Big)_{0,\partial K} \\ = & \sum_{K\in\mathcal{K}} -\frac{1}{2}(\mathbf{n}_K \cdot \mathbf{B}[\mathbf{v}_h]_{K,f}, \mathbf{v})_{0,\partial K} + \frac{1}{2}\Big((\mathbf{n}_K \cdot \mathbf{B} - |\mathbf{n}_K \cdot \mathbf{B}|)[\mathbf{v}_h]_{K,f}, \mathbf{v}\Big)_{0,\partial K} \\ = & \sum_{K\in\mathcal{K}} -\frac{1}{2}(|\mathbf{n}_K \cdot \mathbf{B}|[\mathbf{v}_h]_{K,f}, \mathbf{v})_{0,\partial K} \\ = & -\frac{1}{2} \sum_{f\in\mathcal{F}_K\cap\partial\Omega} (|\mathbf{n}_K \cdot \mathbf{B}|[\mathbf{v}_h]_{K,f}, \mathbf{v})_{0,\partial K} = 0 \end{aligned}$$

due to homogeneous boundary conditions contained in $\mathcal{D}(A)$. □

For the examples given in Sections 3.3 and 3.4, i.e., for linear transport and electromagnetic waves, we compute the eigenpairs explicitly. Moreover, we compute the coefficients b_i and provide explicit expressions for the upwind flux. Additionally we discuss how homogeneous boundary data, corresponding to $\mathcal{D}(A)$, can be enforced.

4.4.1 Linear Transport Equation

We consider the Riemann problem in case of the linear transport example. It holds that $B_{\mathbf{n}} = \mathbf{n} \cdot \mathbf{q}$ is scalar and the eigenpairs are given as $[\lambda_{\mathrm{L}}, 1]$ and $[\lambda_{\mathrm{R}}, 1]$, with

eigenvalues $\lambda_{\mathrm{L}} = \rho_{\mathrm{L}}^{-1} B_{\mathbf{n}}$ and $\lambda_{\mathrm{R}} = \rho_{\mathrm{R}}^{-1} B_{\mathbf{n}}$, respectively, see (4.9) and (4.10). Due to the Rankine–Hugoniot condition (4.11) it yields for the coefficient b_{L} that

$$b_{\mathrm{L}} = \begin{cases} [u]_{\Omega_{\mathrm{L}}}, & \text{if } \lambda_{\mathrm{L}}, \lambda_{\mathrm{R}} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

Since $\rho_{\mathrm{L}}, \rho_{\mathrm{R}} > 0$ we conclude for the upwind flux

$$\begin{aligned} \mathbf{n} \cdot \mathbf{F}^{\mathrm{num}}(u) &= \begin{cases} B_{\mathbf{n}} u_{\mathrm{L}} + B_{\mathbf{n}} [u]_{\Omega_{\mathrm{L}}}, & \text{if } B_{\mathbf{n}} < 0, \\ B_{\mathbf{n}} u_{\mathrm{L}}, & \text{otherwise,} \end{cases} \\ &= (\mathbf{n} \cdot \mathbf{q}) u_{\mathrm{L}} + \frac{1}{2} (\mathbf{n} \cdot \mathbf{q} - |\mathbf{n} \cdot \mathbf{q}|) [u]_{\Omega_{\mathrm{L}}}. \end{aligned}$$

Thus we achieve for the flux term on the cell $K \in \mathcal{K}$

$$\mathbf{n}_K \cdot \left(\mathbf{F}_{K,f}^{\mathrm{num}}(v_h) - \mathbf{F}(v_{h,K}) \right) = \frac{1}{2} \left(\mathbf{n}_K \cdot \mathbf{q} - |\mathbf{n}_K \cdot \mathbf{q}| \right) [v_h]_{K,f},$$

for all $v_h \in H_h$. To simplify implementation, we want to use the same definition of the upwind flux on interior and exterior cell faces $f \in \mathcal{F}_K$. On exterior faces $f \subset \partial\Omega$ the value of v_{h,K_f} and hence the jump $[v_h]_{K,f}$ is not defined. The following idea consists of formally replacing $[v]_{K,f}$ in a way, such that the homogeneous conditions are correctly enforced on $f \subset \Gamma_{\mathrm{in}}$ and $f \subset \Gamma_{\mathrm{out}}$. We choose

$$[v_h]_{K,f} = \begin{cases} -2 v_{h,K}, & \text{for } f \subset \Gamma_{\mathrm{in}}, \\ 0, & \text{for } f \subset \Gamma_{\mathrm{out}}, \end{cases}$$

since we (virtually) require that $\{v_h\}_f = 0$ for $f \subset \Gamma_{\mathrm{in}}$, i.e., the average of v_h is zero on Γ_{in}.

4.4.2 Electromagnetic Waves

To compute the upwind flux for an electromagnetic wave problem explicitly, we proceed as stated in [HW02]. A simplification for transverse electromagnetic waves is given in [HW08, Ex. 2.6]. Assume that M_{L} and M_{R} are constant in Ω_{L} and Ω_{R} for the Riemann problem in the Maxwell case. It holds that $B_{\mathbf{n}} \mathbf{u} = \mathbf{n} \cdot \mathbf{F}(\mathbf{u}) = (\mathbf{n} \times \mathbf{E}, -\mathbf{n} \times \mathbf{H})^{\top}$. Due to (4.9) the eigenpairs $[\lambda_{i,\mathrm{L}}, \mathbf{c}_{i,\mathrm{L}}]$ in Ω_{L} are given as

$$\left[-c_{\mathrm{L}}, \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{L}}}\, \boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{L}}}\, \mathbf{n} \times \boldsymbol{\tau} \end{pmatrix} \right] \quad \text{and} \quad \left[-c_{\mathrm{L}}, \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{L}}}\, \mathbf{n} \times \boldsymbol{\tau} \\ -\sqrt{\mu_{\mathrm{L}}}\, \boldsymbol{\tau} \end{pmatrix} \right].$$

Correspondingly and due to (4.10) the eigenpairs $[\lambda_{i,\mathrm{R}}, \mathbf{c}_{i,\mathrm{R}}]$ in Ω_{R} are given as

$$\left[c_{\mathrm{R}}, \begin{pmatrix} -\sqrt{\varepsilon_{\mathrm{R}}}\, \boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{R}}}\, \mathbf{n} \times \boldsymbol{\tau} \end{pmatrix} \right] \quad \text{and} \quad \left[c_{\mathrm{R}}, \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{R}}}\, \mathbf{n} \times \boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{R}}}\, \boldsymbol{\tau} \end{pmatrix} \right].$$

Here the eigenvalues $c_{\mathrm{L}} = -\lambda_{1,L} = -\lambda_{2,L} = -(\varepsilon_{\mathrm{L}}\mu_{\mathrm{L}})^{-1/2}$, $c_{\mathrm{R}} = \lambda_{1,L} = \lambda_{2,L} = (\varepsilon_{\mathrm{R}}\mu_{\mathrm{R}})^{-1/2}$ correspond to the speed of light in the different materials Ω_{L}, Ω_{R}, respectively. Moreover, $\boldsymbol{\tau}$ denotes a unit tangent vector perpendicular to $\mathbf{n}$. Hence we achieve the Riemann solution

$$\mathbf{u}(\mathbf{x},t) = \begin{cases} \mathbf{u}_{\mathrm{L}}, & \text{for } \mathbf{x}\cdot\mathbf{n} + c_{\mathrm{L}}t < 0 \text{ and } \mathbf{x} \in \Omega_{\mathrm{L}}, \\ \hat{\mathbf{u}}_{\mathrm{L}}, & \text{for } \mathbf{x}\cdot\mathbf{n} + c_{\mathrm{L}}t > 0 \text{ and } \mathbf{x} \in \Omega_{\mathrm{L}}, \\ \hat{\mathbf{u}}_{\mathrm{R}}, & \text{for } \mathbf{x}\cdot\mathbf{n} - c_{\mathrm{R}}t < 0 \text{ and } \mathbf{x} \in \Omega_{\mathrm{R}}, \\ \mathbf{u}_{\mathrm{R}}, & \text{for } \mathbf{x}\cdot\mathbf{n} - c_{\mathrm{R}}t > 0 \text{ and } \mathbf{x} \in \Omega_{\mathrm{R}}, \end{cases}$$

with intermediate states

$$\begin{aligned} \hat{\mathbf{u}}_{\mathrm{L}} &= \mathbf{u}_{\mathrm{L}} + b_{1,\mathrm{L}} \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{L}}}\,\boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{L}}}\,\mathbf{n}\times\boldsymbol{\tau} \end{pmatrix} + b_{2,\mathrm{L}} \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{L}}}\,\mathbf{n}\times\boldsymbol{\tau} \\ -\sqrt{\mu_{\mathrm{L}}}\,\boldsymbol{\tau} \end{pmatrix} \\ \hat{\mathbf{u}}_{\mathrm{R}} &= \mathbf{u}_{\mathrm{R}} + b_{1,\mathrm{R}} \begin{pmatrix} -\sqrt{\varepsilon_{\mathrm{R}}}\,\boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{R}}}\,\mathbf{n}\times\boldsymbol{\tau} \end{pmatrix} + b_{2,\mathrm{R}} \begin{pmatrix} \sqrt{\varepsilon_{\mathrm{R}}}\,\mathbf{n}\times\boldsymbol{\tau} \\ \sqrt{\mu_{\mathrm{R}}}\,\boldsymbol{\tau} \end{pmatrix}. \end{aligned}$$

We use the Rankine–Hugoniot condition (4.11) at the interface $f_{\mathrm{L,R}}$ to determine the unknowns $b_{1,\mathrm{L}}, b_{2,\mathrm{L}}$. Thus for $[\mathbf{u}]_{\Omega_{\mathrm{L}}} = (\mathbf{H}_{\mathrm{R}} - \mathbf{H}_{\mathrm{L}}, \mathbf{E}_{\mathrm{R}} - \mathbf{E}_{\mathrm{L}})^{\top}$ it yields

$$\begin{aligned} b_{1,\mathrm{L}} \begin{pmatrix} -\sqrt{\mu_{\mathrm{L}}}\,\boldsymbol{\tau} \\ -\sqrt{\varepsilon_{\mathrm{L}}}\,\mathbf{n}\times\boldsymbol{\tau} \end{pmatrix} + b_{2,\mathrm{L}} \begin{pmatrix} -\sqrt{\mu_{\mathrm{L}}}\,\mathbf{n}\times\boldsymbol{\tau} \\ \sqrt{\varepsilon_{\mathrm{L}}}\,\boldsymbol{\tau} \end{pmatrix} \\ = B_{\mathbf{n}}[\mathbf{u}]_{\Omega_{\mathrm{L}}} + b_{1,\mathrm{R}} \begin{pmatrix} -\sqrt{\mu_{\mathrm{R}}}\,\boldsymbol{\tau} \\ \sqrt{\varepsilon_{\mathrm{R}}}\,\mathbf{n}\times\boldsymbol{\tau} \end{pmatrix} + b_{2,\mathrm{R}} \begin{pmatrix} \sqrt{\mu_{\mathrm{R}}}\,\mathbf{n}\times\boldsymbol{\tau} \\ \sqrt{\varepsilon_{\mathrm{R}}}\,\boldsymbol{\tau} \end{pmatrix}. \end{aligned}$$

Testing with a Basis of $\mathbb{R}^3$, e.g., $\{\mathbf{n}, \boldsymbol{\tau}, \mathbf{n}\times\boldsymbol{\tau}\}$, results in two independent systems of equations

$$\begin{aligned} \begin{pmatrix} -\sqrt{\mu_{\mathrm{L}}} & \sqrt{\mu_{\mathrm{R}}} \\ -\sqrt{\varepsilon_{\mathrm{L}}} & -\sqrt{\varepsilon_{\mathrm{R}}} \end{pmatrix} \begin{pmatrix} b_{1,\mathrm{L}} \\ b_{1,\mathrm{R}} \end{pmatrix} &= \begin{pmatrix} (\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau} \\ -(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau}) \end{pmatrix}, \\ \begin{pmatrix} -\sqrt{\mu_{\mathrm{L}}} & -\sqrt{\mu_{\mathrm{R}}} \\ \sqrt{\varepsilon_{\mathrm{L}}} & -\sqrt{\varepsilon_{\mathrm{R}}} \end{pmatrix} \begin{pmatrix} b_{2,\mathrm{L}} \\ b_{2,\mathrm{R}} \end{pmatrix} &= \begin{pmatrix} (\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau}) \\ -(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau} \end{pmatrix}. \end{aligned}$$

Thus we conclude that

$$\begin{aligned} b_{1,\mathrm{L}} &= -\frac{\sqrt{\varepsilon_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau} - \sqrt{\mu_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau})}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}} + \sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}, \\ b_{2,\mathrm{L}} &= -\frac{\sqrt{\varepsilon_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau}) + \sqrt{\mu_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau}}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}} + \sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}. \end{aligned}$$

By inserting this result into the general upwind flux definition (4.12) yields

$$
\begin{aligned}
&\mathbf{n}\cdot\mathbf{F}^{\mathrm{num}}(\mathbf{u})\\
&= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} - \frac{\sqrt{\varepsilon_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau} - \sqrt{\mu_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau})}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}}+\sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}\begin{pmatrix}-\sqrt{\mu_{\mathrm{L}}}\boldsymbol{\tau}\\ -\sqrt{\varepsilon_{\mathrm{L}}}\mathbf{n}\times\boldsymbol{\tau}\end{pmatrix}\\
&\qquad - \frac{\sqrt{\varepsilon_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\cdot(\mathbf{n}\times\boldsymbol{\tau}) + \sqrt{\mu_{\mathrm{R}}}(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\cdot\boldsymbol{\tau}}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}}+\sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}\begin{pmatrix}-\sqrt{\mu_{\mathrm{L}}}\mathbf{n}\times\boldsymbol{\tau}\\ -\sqrt{\varepsilon_{\mathrm{L}}}\boldsymbol{\tau}\end{pmatrix}\\
&= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} + \frac{\sqrt{\varepsilon_{\mathrm{R}}}}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}}+\sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}\begin{pmatrix}\sqrt{\mu_{\mathrm{L}}}\,\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}}\\ \sqrt{\varepsilon_{\mathrm{L}}}\,\mathbf{n}\times(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\end{pmatrix}\\
&\qquad + \frac{\sqrt{\mu_{\mathrm{R}}}}{\sqrt{\mu_{\mathrm{L}}\varepsilon_{\mathrm{R}}}+\sqrt{\varepsilon_{\mathrm{L}}\mu_{\mathrm{R}}}}\begin{pmatrix}\sqrt{\mu_{\mathrm{L}}}\,\mathbf{n}\times(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\\ -\sqrt{\varepsilon_{\mathrm{L}}}\,\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}}\end{pmatrix}\\
&= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} + \frac{\varepsilon_{\mathrm{R}}c_{\mathrm{R}}}{\varepsilon_{\mathrm{L}}c_{\mathrm{L}}+\varepsilon_{\mathrm{R}}c_{\mathrm{R}}}\begin{pmatrix}\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}}\\ \mathbf{0}\end{pmatrix} + \frac{1}{\mu_{\mathrm{L}}c_{\mathrm{L}}+\mu_{\mathrm{R}}c_{\mathrm{R}}}\begin{pmatrix}\mathbf{0}\\ \mathbf{n}\times(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\end{pmatrix}\\
&\qquad - \frac{\mu_{\mathrm{R}}c_{\mathrm{R}}}{\mu_{\mathrm{L}}c_{\mathrm{L}}+\mu_{\mathrm{R}}c_{\mathrm{R}}}\begin{pmatrix}\mathbf{0}\\ \mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}}\end{pmatrix} + \frac{1}{\varepsilon_{\mathrm{L}}c_{\mathrm{L}}+\varepsilon_{\mathrm{R}}c_{\mathrm{R}}}\begin{pmatrix}\mathbf{n}\times(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\\ \mathbf{0}\end{pmatrix},
\end{aligned}
$$

where we used the following vector identities for $\mathbf{x}\in\mathbb{R}^3$

$$
\begin{aligned}
\mathbf{n}\times\mathbf{x} &= \Big((\mathbf{n}\times\mathbf{x})\cdot\boldsymbol{\tau}\Big)\boldsymbol{\tau} + \Big((\mathbf{n}\times\mathbf{x})\cdot(\mathbf{n}\times\boldsymbol{\tau})\Big)(\mathbf{n}\times\boldsymbol{\tau}),\\
\mathbf{n}\times(\mathbf{n}\times\mathbf{x}) &= \Big((\mathbf{n}\times\mathbf{x})\cdot\boldsymbol{\tau}\Big)(\mathbf{n}\times\boldsymbol{\tau}) - \Big((\mathbf{n}\times\mathbf{x})\cdot(\mathbf{n}\times\boldsymbol{\tau})\Big)\boldsymbol{\tau}.
\end{aligned}
$$

For homogeneous materials (i.e., $\varepsilon=\varepsilon_{\mathrm{L}}=\varepsilon_{\mathrm{R}}$ and $\mu=\mu_{\mathrm{L}}=\mu_{\mathrm{R}}$) we achieve

$$
\begin{aligned}
\mathbf{n}\cdot\mathbf{F}^{\mathrm{num}}(\mathbf{u}) &= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} + \frac{1}{2}\begin{pmatrix}\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}}\\ -\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}}\end{pmatrix} + \frac{1}{2}\begin{pmatrix}\sqrt{\frac{\mu}{\varepsilon}}\mathbf{n}\times(\mathbf{n}\times[\mathbf{H}]_{\Omega_{\mathrm{L}}})\\ \sqrt{\frac{\varepsilon}{\mu}}\mathbf{n}\times(\mathbf{n}\times[\mathbf{E}]_{\Omega_{\mathrm{L}}})\end{pmatrix}\\
&= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} + \frac{1}{2}B_{\mathbf{n}}[\mathbf{u}]_{\Omega_{\mathrm{L}}} - \frac{1}{2}\begin{pmatrix}\sqrt{\frac{\mu}{\varepsilon}} & 0\\ 0 & \sqrt{\frac{\varepsilon}{\mu}}\end{pmatrix}B_{\mathbf{n}}B_{\mathbf{n}}[\mathbf{u}]_{\Omega_{\mathrm{L}}}\\
&= B_{\mathbf{n}}\mathbf{u}_{\mathrm{L}} + \frac{1}{2}B_{\mathbf{n}}[\mathbf{u}]_{\Omega_{\mathrm{L}}} - \frac{1}{2}\begin{pmatrix}\sqrt{\frac{\mu}{\varepsilon}} & 0\\ 0 & \sqrt{\frac{\varepsilon}{\mu}}\end{pmatrix}|B_{\mathbf{n}}|[\mathbf{u}]_{\Omega_{\mathrm{L}}}.
\end{aligned}
$$

Thus we state the flux term on the cell $K\in\mathcal{K}$ as

$$
\mathbf{n}_K\cdot\Big(\mathbf{F}^{\mathrm{num}}_{K,f}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,K})\Big) = \frac{1}{2}B_{\mathbf{n}}[\mathbf{v}_h]_{K,f} - \frac{1}{2}\begin{pmatrix}\sqrt{\frac{\mu}{\varepsilon}} & 0\\ 0 & \sqrt{\frac{\varepsilon}{\mu}}\end{pmatrix}|B_{\mathbf{n}}|[\mathbf{v}_h]_{K_f}
$$

for all $\mathbf{v}_h\in H_h$. The proof of Lemma (4.3) can be transferred analogously to show consistency of the upwind flux in this case. To enforced perfect electric conducting boundary conditions we proceed similar as in the linear transport case and choose

$$
\mathbf{n}_K\times[\mathbf{H}_h]_{K,f} = \mathbf{0} \qquad \text{and} \qquad \mathbf{n}_K\times[\mathbf{E}_h]_{K,f} = -2\mathbf{n}_K\times\mathbf{E}_{h,K}
$$

for $f\subset\partial\Omega$. This corresponds to the (virtual) requirement $\{\mathbf{E}_h\}_f = 0$ for $f\subset\partial\Omega$. A more detailed discussion on the upwind flux for Maxwell's equation and the application of boundary conditions can be found for example in [Sch15, Sec. 2.3.4, Sec. 2.3.5].

4.5 Discontinuous Petrov–Galerkin Approximation in Space-Time

Based on the discrete spatial operators M_h and A_h introduced in the previous section, we extend this discretization by a Petrov–Galerkin method with a continuous ansatz space and discontinuous test space in time, see for example [BR99]. This results in an implicit space-time setting by using a tensor product approach. Hence we avoid stability issues due to a violated CFL (Courant–Friedrichs–Lewy) condition which can occur when applying a discontinuous Galerkin method in time, see, e.g., [Joh93], [HW02, Sec. 4.8], [LeV08, Sec. 10.6].

The space-time cylinder Q is decomposed into a finite set $\mathcal{R}$ of non-overlapping space-time cells $R = K \times I$, where $K \subset \Omega$ is a spatial cell and $I = (t_-, t_+) \subset (0, T)$ a local time interval. If K is quadrilateral, R is a space-time parallelepiped and if K is a triangle, R becomes a space-time prism (cf. Figure 4.4). The decomposition is chosen such that $\overline{Q} = \bigcup_{R \in \mathcal{R}} \overline{R}$ holds (cf. Figure 4.5). As in the previous section, we

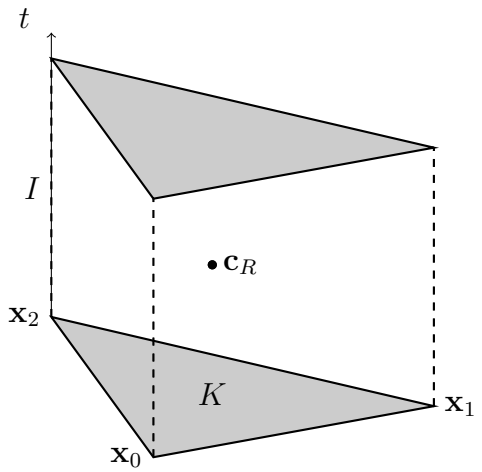

Figure 4.4: Illustration of a prismatic space-time cell R.

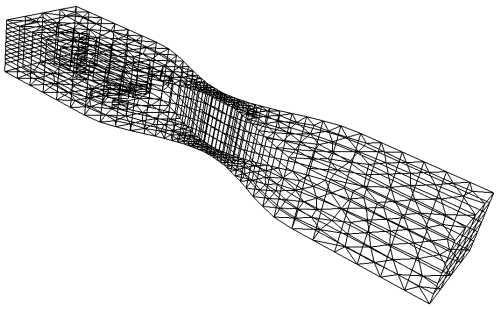

Figure 4.5: Illustration of an admissible space-time Mesh $\mathcal{R}$.

choose discrete local ansatz and test spaces $V_{h,R}, W_{h,R} \subset \mathrm{L}_2(R)^J$ with $W_{h,R} \subset \partial_t V_{h,R}$ for every cell R. The global ansatz and test spaces are defined as

$$V_h = \Big\{ \mathbf{v}_h \in \mathrm{H}^1(0,T;H) \colon \mathbf{v}_h(\mathbf{x},0) = \mathbf{0} \text{ for almost all } \mathbf{x} \in \Omega \\ \text{and } \mathbf{v}_{h,R} = \mathbf{v}_h|_R \in V_{h,R} \Big\},$$
$$W_h = \Big\{ \mathbf{w}_h \in \mathrm{L}_2(0,T;H) \colon \mathbf{w}_{h,R} = \mathbf{w}_h|_R \in W_{h,R} \Big\}.$$

These spaces are constructed such that functions in W_h are discontinuous in space and time, whereas functions in V_h are continuous almost everywhere in time, but

discontinuous in space. Hence this choice of spaces is a Petrov–Galerkin approach (cf. Section 4.1). To receive a unique discrete solution we require that $\dim(V_h) = \dim(W_h)$, which restricts the choice of the local spaces $V_{h,R}$. In this work we focus on a Petrov–Galerkin approach within a tensor product space-time discretization.

Definition 4.9 (Tensor product discretization)**.** We proceed as in the method of lines, where we choose a time series $0 = t_0 < t_1 < \ldots < t_N = T$ to discretize $(0, T)$. A *tensor product space-time discretization* then consists of local space-time cells $R = K \times (t_{n-1}, t_n)$ for $K \in \mathcal{K}$ and $n = 1, \ldots, N$, where $\mathcal{K}$ is a fixed spatial mesh $\mathcal{K}$. Hence the tensor product space-time discretization $\mathcal{R}$ is given as

$$\mathcal{R} = \bigcup_{K \in \mathcal{K}} \bigcup_{n=1}^{N} K \times (t_{n-1}, t_n).$$

In particular $\mathcal{R}$ is an admissible mesh and can be decomposed in time slices (cf. Section 4.6).

By using this discretization, we can chose the discrete space H_h as in section 4.3 independent in time. Furthermore we define $W_{h,R} = (\mathbb{P}_{p_R}(K) \times \mathbb{P}_0(t_{n-1}, t_n))^J$ to be constant in time on $R = K \times (t_{n-1}, t_n)$. This results in a piecewise linear approximation in time

$$\begin{aligned} V_h = \Big\{ \mathbf{v}_h \in & \mathrm{H}^1(0,T;H) : \\ & \mathbf{v}_h(\mathbf{x}, 0) = \mathbf{0}\,,\ \mathbf{v}_h(\mathbf{x}, t_n) \in H_h \text{ for almost all } \mathbf{x} \in \Omega, \\ & \mathbf{v}_h(\mathbf{x}, t) = \frac{t_n - t}{t_n - t_{n-1}} \mathbf{v}_h(\mathbf{x}, t_{n-1}) + \frac{t - t_{n-1}}{t_n - t_{n-1}} \mathbf{v}_h(\mathbf{x}, t_n) \\ & \text{for } t \in (t_{n-1}, t_n) \text{ and } n = 1, \ldots, N \Big\}. \end{aligned} \tag{4.15}$$

An illustration of a space-time solution $\mathbf{u}_h \in V_h$ is given in Figure 4.6. This idea can be extended to arbitrary high polynomial order in time. Thus, for every $R \in \mathcal{R}$, we select polynomial degrees $p_R \geq 0$ and $q_R > 0$ in space and time locally. The tensor product approach results in local test spaces $W_{h,R} = (\mathbb{P}_{p_R}(K) \times \mathbb{P}_{q_R - 1}(I))^J$ and local ansatz spaces

$$\begin{aligned} V_{h,R} = \Big\{ \mathbf{v}_{h,R} \in & \mathrm{L}_2(R)^J : \\ & \mathbf{v}_{h,R}(\mathbf{x}, t) = \frac{t_+ - t}{t_+ - t_-} \mathbf{v}_h(\mathbf{x}, t_-) + \frac{t - t_-}{t_+ - t_-} \mathbf{w}_{h,R}(\mathbf{x}, t)\,, \\ & \text{where } \mathbf{v}_h \in V_h|_{[0,t_-]} \text{ and } \mathbf{w}_{h,R} \in W_{h,R} \\ & \text{and } (\mathbf{x}, t) \in R = K \times (t_-, t_+) \Big\}. \end{aligned} \tag{4.16}$$

Hence V_h is continuous in time and $\mathbf{v}_{h,R} \in (\mathbb{P}_{p_R}(K) \times \mathbb{P}_{q_R}(I))^J$ for all $\mathbf{v}_{h,R} \in V_{h,R}$.

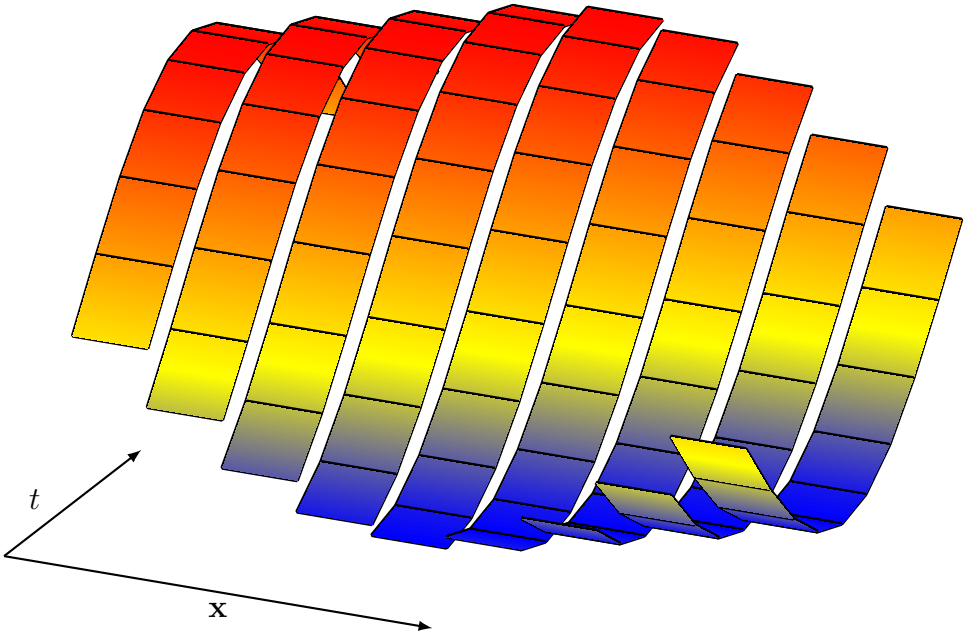

Figure 4.6: Illustration of a space-time solution $\mathbf{u}_h \in V_h$, which is discontinuous in space but continuous in time. In this example lowest polynomial degrees $(p, q) = (0, 1)$ are used.

The discontinuous Galerkin operator in space defined in (4.6) can be extended to the space-time setting, by defining $A_h \mathbf{v}_h \in W_h$ through

$$\begin{aligned}\left(A_h \mathbf{v}_h, \mathbf{w}_h\right)_{0,Q} = \sum_{R=K\times I\in\mathcal{R}} \Bigg(&\left(\nabla \cdot \mathbf{F}(\mathbf{v}_{h,R}), \mathbf{w}_{h,R}\right)_{0,R} \\ &+ \sum_{f\in\mathcal{F}_K} \left(\mathbf{n}_K \cdot (\mathbf{F}_K^{\text{num}}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,R})), \mathbf{w}_{h,R}\right)_{0,f\times I}\Bigg)\end{aligned} \tag{4.17}$$

for $\mathbf{v}_h \in V_h$ and $\mathbf{w}_h \in W_h$. Moreover, we define the discrete space-time operator $L_h = M_h \partial_t + A_h \in \mathcal{L}(V_h, W_h)$ and the discrete bilinear form $b_h(\cdot,\cdot) = (L_h \cdot, \cdot)_{0,Q}$. To show existence and uniqueness of our discrete Petrov–Galerkin scheme, we proceed similarly as in the continuous case (cf. Lemma 3.2) and extend the proof to the discrete spaces V_h and W_h. Therefore we first define a weighted L_2-projection into the test space W_h and discrete norms correspondingly to V_h and W_h. Furthermore, we prove auxiliary results in the following lemmas which are needed to show the inf-sub condition for our discrete bilinear form $b_h(.,.)$ afterwards.

Definition 4.10 (Discrete graph norm)**.** Let $\Pi_h \colon W \to W_h$ be the weighted L_2-projection given by

$$(\Pi_h M \mathbf{v}, \mathbf{w}_h)_{0,Q} = (M\mathbf{v}, \mathbf{w}_h)_{0,Q}, \qquad \text{for all } \mathbf{w}_h \in W_h.$$

Due to the definition of the discrete operator M_h (4.3) and the discrete discontinuous

Galerkin operator A_h (4.17) we conclude

$$M_h(W_h) = W_h, \quad \Pi_h M_h = M_h \Pi_h, \quad \Pi_h M_h^{-1} = M_h^{-1} \Pi_h, \quad \Pi_h A_h = A_h$$

and hence $\Pi_h L_h = L_h$. Furthermore, we define the discrete graph norm

$$\|\mathbf{v}_h\|_{V_h}^2 = \|\mathbf{v}_h\|_W^2 + \|M_h^{-1} L_h \mathbf{v}\|_W^2,$$

which is used for the following stability and convergence analysis.

Lemma 4.4. *For every $\phi \in \mathrm{L}_1(0,T)$ it holds that*

$$\int_0^T \int_0^t \phi(s) \,\mathrm{d}s \,\mathrm{d}t = \int_0^T d_T(t)\phi(t) \,\mathrm{d}t$$

with weight function $d_T(t) = T - t \geq 0$.

Proof. Define a characteristic function

$$\chi \colon (0,T) \times (0,T) \mapsto \{0,1\}, \chi(t,s) = \begin{cases} 1, & \text{if } s \leq t, \\ 0, & \text{otherwise.} \end{cases}$$

Hence with Fubini's theorem we conclude that

$$\begin{aligned} \int_0^T \int_0^t \phi(s) \,\mathrm{d}s \,\mathrm{d}t &= \int_0^T \int_0^T \chi(t,s)\phi(s) \,\mathrm{d}s \,\mathrm{d}t = \int_0^T \phi(s) \int_0^T \chi(t,s) \,\mathrm{d}t \,\mathrm{d}s \\ &= \int_0^T d_T(s)\phi(s) \,\mathrm{d}s. \end{aligned}$$

□

Lemma 4.5 ([DFW16, Lem. 6])**.** *We consider a tensor product space-time discretization with a fixed polynomial degree $q_I \in \mathbb{N}$ in time on every time interval $I = (t_{n-1}, t_n) \subset (0,T)$ and a fixed polynomial degree $p_K \in \mathbb{N}_0$ in space on every spacial cell $K \in \mathcal{K}$. Hence we get the local spaces $V_{h,R} = \mathbb{P}_{p_k}(K) \times \mathbb{P}_{q_I}(I)$ and $W_{h,R} = \mathbb{P}_{p_k}(K) \times \mathbb{P}_{q_I - 1}(I)$. In this case it holds for all $\mathbf{v}_h \in V_h$:*

a) $\Pi_h \partial_t \mathbf{v}_h = \partial_t \mathbf{v}_h$

b) $(M_h \partial_t \mathbf{v}_h, d_T \mathbf{v}_h)_{0,Q} \leq (M_h \partial_t \mathbf{v}_h, d_T \Pi_h \mathbf{v}_h)_{0,Q}$

c) $0 \leq (A_h \mathbf{v}_h, d_T \Pi_h \mathbf{v}_h)_{0,Q}$

Proof. Since we are only considering the tensor product case with fixed polynomial degrees in space and time, there exist local representations for $\mathbf{v}_h \in V_h$ and $\mathbf{w}_h \in W_h$ such that

$$\mathbf{v}_h(\mathbf{x},t) = \sum_{k=0}^{q_I} \boldsymbol{\psi}_{I,k}(\mathbf{x}) \lambda_{I,k}(t) \quad \text{and} \quad \mathbf{w}_h(\mathbf{x},t) = \sum_{k=0}^{q_I - 1} \boldsymbol{\phi}_{I,k}(\mathbf{x}) \lambda_{I,k}(t)$$

on a time slice $(\mathbf{x}, t) \in \Omega \times I$. Note that the coefficients $\boldsymbol{\psi}_{I,k}, \boldsymbol{\phi}_{I,k} \in H_h$ and $\lambda_{I,k} \in \mathbb{P}_k(I)$ only depend either on the spatial variable $\mathbf{x}$ or the temporal variable t. Furthermore, we can choose general orthonormal Legendre polynomial as a basis for the polynomial space $\mathbb{P}_k(I)$ (see Section A.2 in the appendix). We can now prove the assertions:

a) Taking the temporal derivative of the representation leads to

$$\partial_t \mathbf{v}_h(\mathbf{x}, t) = \sum_{k=0}^{q_I} \psi_{I,k}(\mathbf{x}) \partial_t \lambda_{I,k}(t),$$

where $\partial_t \lambda_{I,k} \in \mathbb{P}_{k-1}(I)$ for all $(\mathbf{x}, t) \in \Omega \times I$. Hence we obtain $\partial_t \mathbf{v}_h \in W_h$ and $\Pi_h \partial_t \mathbf{v}_h = \partial_t \mathbf{v}_h$.

b) It holds that

$$\begin{aligned}(d_T M_h \partial_t \mathbf{v}_h, \mathbf{v}_h - \Pi_h \mathbf{v}_h)_{0,Q} &= \sum_I \sum_{k=0}^{q_I} (M_h \psi_{I,k}, \psi_{I,q_I})_{0,\Omega} (d_T \partial_t \lambda_{I,k}, \lambda_{I,q_I})_{0,I} \\ &= \sum_I (M_h \psi_{I,q_I}, \psi_{I,q_I})_{0,\Omega} (d_T \partial_t \lambda_{I,q_I}, \lambda_{I,q_I})_{0,I} \\ &= -q_I \sum_I (M_h \psi_{I,q_I}, \psi_{I,q_I})_{0,\Omega} \leq 0.\end{aligned}$$

Here we used that $(d_T \partial_t \lambda_{I,k}, \lambda_{I,q_I})_{0,I} = 0$ for $k < q_I$ since the Legendre polynomials are orthogonal. In the last step we applied Lemma A.2 to conclude that

$$(d_T \partial_t \lambda_{I,q_I}, \lambda_{I,q_I})_{0,I} = -(t \partial_t \lambda_{I,q_I}, \lambda_{I,q_I})_{0,I} = -q_I.$$

c) In the tensor product case it holds that

$$\begin{aligned}&\Big(A_h \mathbf{v}_h, \mathbf{w}_h\Big)_{0,Q} \\ =&\sum_I \sum_K \Big(\Big(\nabla \cdot \mathbf{F}(\mathbf{v}_{h,R}), \mathbf{w}_{h,R} \Big)_{0,K \times I} \\ &\quad + \sum_{f \in \mathcal{F}_K} \Big(\mathbf{n}_K \cdot (\mathbf{F}_K^{\text{num}}(\mathbf{v}_h) - \mathbf{F}(\mathbf{v}_{h,R})), \mathbf{w}_{h,R} \Big)_{0,f \times I} \Big) \\ =&\sum_I \sum_K \sum_{k=0}^{q_I} \sum_{j=0}^{q_I - 1} \Big(\Big(\nabla \cdot \mathbf{F}(\boldsymbol{\psi}_{K \times I,k}), \boldsymbol{\phi}_{K \times I,j} \Big)_{0,K} \\ &\quad + \sum_{f \in \mathcal{F}_K} \Big(\mathbf{n}_K \cdot (\mathbf{F}_K^{\text{num}}(\boldsymbol{\psi}_{I,k}) - \mathbf{F}(\boldsymbol{\psi}_{K \times I,k})), \boldsymbol{\phi}_{K \times I,j} \Big)_{0,f} \Big) \Big(\lambda_{I,k}, \lambda_{I,j} \Big)_{0,I} \\ =&\Big(A_h \Pi_h \mathbf{v}_h, \mathbf{w}_h\Big)_{0,Q}\end{aligned}$$

and hence $A_h = A_h\Pi_h$. Finally we achieve

$$\begin{aligned}
&\Big(A_h\mathbf{v}_h, d_T\Pi_h\mathbf{v}_h\Big)_{0,Q}\\
=&\Big(A_h\Pi_h\mathbf{v}_h, d_T\Pi_h\mathbf{v}_h\Big)_{0,Q}\\
=&\sum_I\sum_K\sum_{k=0}^{q_I-1}\sum_{j=0}^{q_I-1}\Big(\Big(\nabla\cdot\mathbf{F}(\boldsymbol{\psi}_{K\times I,k}),\boldsymbol{\psi}_{K\times I,j}\Big)_{0,K}\\
&\qquad+\sum_{f\in\mathcal{F}_K}\Big(\mathbf{n}_K\cdot(\mathbf{F}_K^{\mathrm{num}}(\boldsymbol{\psi}_{I,k})-\mathbf{F}(\boldsymbol{\psi}_{K\times I,k})),\boldsymbol{\psi}_{K\times I,j}\Big)_{0,f}\Big)\Big(\lambda_{I,k},d_T\lambda_{I,j}\Big)_{0,I}\\
=&\sum_I\sum_{k=0}^{q_I-1}\sum_{j=0}^{q_I-1}\Big(A_h\boldsymbol{\psi}_{I,k},\boldsymbol{\psi}_{I,j}\Big)_{0,\Omega}(\lambda_{I,k},d_T\lambda_{I,j})_{0,I}\geq 0.
\end{aligned}$$

In the last step we used that the $q_I\times q_I$ matrices with entries $\Big(A_h\boldsymbol{\psi}_{I,k},\boldsymbol{\psi}_{I,j}\Big)_{0,\Omega}$ and $(\lambda_{I,k},d_T\lambda_{I,j})_{0,I}$ are always positive semi-definite.

□

Lemma 4.6 ([DFW16, Lem. 3])**.** *Assume that the assumptions in Lemma 4.5 are fulfilled. Then the bilinear form $b_h(\cdot,\cdot)$ is bounded and inf-sup stable in $V_h\times W_h$ with $\beta = 1/\sqrt{1+4T^2}$, i.e.,*

$$\sup_{\mathbf{w}_h\in W_h}\frac{b_h(\mathbf{v}_h,\mathbf{w}_h)}{\|\mathbf{w}_h\|_W}\geq\beta\,\|\mathbf{v}_h\|_{V_h}\,,\qquad \mathbf{v}_h\in V_h\,.$$

Proof. We transfer the proof of Lemma 3.2 to the discrete setting and conclude that

$$\begin{aligned}
\|\mathbf{v}_h\|_W^2&=\int_0^T\Big(M_h\mathbf{v}_h(t),\mathbf{v}_h(t)\Big)_{0,\Omega}\,\mathrm{d}t\\
&=\int_0^T\Big(\Big(M_h\mathbf{v}_h(t),\mathbf{v}_h(t)\Big)_{0,\Omega}-\Big(M_h\mathbf{v}_h(0),\mathbf{v}_h(0)\Big)_{0,\Omega}\Big)\,\mathrm{d}t\\
&=\int_0^T\int_0^t\partial_t\Big(M_h\mathbf{v}_h(s),\mathbf{v}_h(s)\Big)_{0,\Omega}\,\mathrm{d}s\,\mathrm{d}t\\
&=2\int_0^T\int_0^t\Big(M_h\partial_t\mathbf{v}_h(s),\mathbf{v}_h(s)\Big)_{0,\Omega}\,\mathrm{d}s\,\mathrm{d}t=2\Big(M_h\partial_t\mathbf{v}_h,d_T\mathbf{v}_h\Big)_{0,Q}\\
&\leq 2\Big(L_h\mathbf{v}_h,d_T\Pi_h\mathbf{v}_h\Big)_{0,Q}\leq 2T\,\|M_h^{-1}L_h\mathbf{v}_h\|_W\|\mathbf{v}_h\|_W\,.
\end{aligned}$$

Hence we achieve $\|\mathbf{v}_h\|_W\leq 2T\,\|M_h^{-1}L_h\mathbf{v}_h\|_W$. Applying this result to the discrete norm $\|.\|_{V_h}$ we get that $\|\mathbf{v}_h\|_{V_h}\leq\sqrt{1+4T^2}\,\|M_h^{-1}L_h\mathbf{v}_h\|_W$. Finally, we obtain the discrete inf-sup stability

$$\begin{aligned}
\sup_{\mathbf{w}_h\in W_h\setminus\{\mathbf{0}\}}\frac{b_h(\mathbf{v}_h,\mathbf{w}_h)}{\|\mathbf{w}_h\|_W}&=\sup_{\mathbf{w}_h\in W_h\setminus\{\mathbf{0}\}}\frac{(MM_h^{-1}L_h\mathbf{v}_h,\mathbf{w}_h)_{0,Q}}{\|\mathbf{w}_h\|_W}\\
&\geq\|M_h^{-1}L_h\mathbf{v}_h\|_W\geq\frac{1}{\sqrt{1+4T^2}}\|\mathbf{v}_h\|_{V_h}\,,
\end{aligned}$$

where we used the Galerkin approximation of M, i.e., Equation (4.3), and inserted a special choice of $\mathbf{w}_h\in W_h$, namely $\mathbf{w}_h = M_h^{-1}L_h\mathbf{v}_h$. □

As we have seen in Theorem 3.1, this shows existence of a unique solution of our Petrov–Galerkin scheme. Hence we can state the following theorem.

Theorem 4.1 ([DFW16, Thm. 4])**.** *Assume that the assumptions in Lemma 4.6 are fulfilled. For given* $\mathbf{f} \in \mathrm{L}_2(Q)^J$ *there exists a unique solution* $\mathbf{u}_h \in V_h$ *of*

$$(L_h \mathbf{u}_h, \mathbf{w}_h)_{0,Q} = (\mathbf{f}, \mathbf{w}_h)_{0,Q}\,, \qquad \mathbf{w}_h \in W_h\,, \tag{4.18}$$

satisfying the a priori bound $\|\mathbf{u}_h\|_{V_h} \leq \sqrt{4T^2+1}\|M_h^{-1}\Pi_h \mathbf{f}\|_W$.

Lemma 4.7 (Galerkin orthogonality)**.** *Let* $\mathbf{u} \in V$ *be the exact solution of problem* (3.1) *and let* $\mathbf{u}_h \in V_h$ *be the discrete solution of problem* (4.1)*. Then the* Galerkin orthogonality

$$b_h(\mathbf{u} - \mathbf{u}_h, \mathbf{w}_h) = 0 \tag{4.19}$$

holds for all $\mathbf{w}_h \in W_h$.

Proof. The Galerkin approximation (4.3) and the consistency of the discontinuous Galerkin method (4.7) yields that

$$(M\partial_t \mathbf{u}, \mathbf{w}_h)_{0,Q} = (M_h \partial_t \mathbf{u}, \mathbf{w}_h)_{0,Q} \quad \text{and} \quad (A\mathbf{u}, \mathbf{w}_h)_{0,Q} = (A_h \mathbf{u}, \mathbf{w}_h)_{0,Q}.$$

Hence we conclude that

$$\begin{aligned} b_h(\mathbf{u}, \mathbf{w}_h) &= (M_h \partial_t \mathbf{u} + A_h \mathbf{u}, \mathbf{w}_h)_{0,Q} = (M\partial_t \mathbf{u} + A\mathbf{u}, \mathbf{w}_h)_{0,Q} \\ &= b(\mathbf{u}, \mathbf{w}_h) = (\mathbf{f}, \mathbf{w}_h)_{0,Q} = b_h(\mathbf{u}_h, \mathbf{w}_h). \end{aligned}$$

□

We will now analyze the convergence with respect to the discrete norm $\|\cdot\|_{V_h}$. Due to the consistency of the numerical flux, it holds that $(A_h \mathbf{v}, \mathbf{w}_h)_{0,Q} = (\nabla \cdot \mathbf{F}(\mathbf{v}), \mathbf{w}_h)_{0,Q}$ for $\mathbf{v} \in V$. Thus, we obtain that $A_h \mathbf{v} = \Pi_h \nabla \cdot \mathbf{F}(\mathbf{v})$ and conclude that A_h and correspondingly $\|\cdot\|_{V_h}$ can be evaluated in $V + V_h$. Moreover, $b_h(\cdot,\cdot)$ is continuous with respect to this extension.

Theorem 4.2 ([DFW16, Thm. 5])**.** *Let* $\mathbf{u} \in V$ *be the solution of* (3.7) *and* $\mathbf{u}_h \in V_h$ *its approximation solving* (4.18)*. Then it hold that*

$$\|\mathbf{u} - \mathbf{u}_h\|_{V_h} \leq (1 + \beta^{-1}) \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{V_h}\,.$$

If in addition the solution is sufficiently smooth, we obtain the a priori error estimate

$$\|\mathbf{u} - \mathbf{u}_h\|_{V_h} \leq C\Big(\triangle t^q + \triangle \mathbf{x}^p\Big)\Big(\|\partial_t^{q+1}\mathbf{u}\|_{0,Q} + \|\mathrm{D}^{p+1}\mathbf{u}\|_{0,Q}\Big)$$

for $\triangle t, \triangle \mathbf{x}$ *and* $p, q \geq 1$*, where* $\mathrm{D}^{\mathrm{p+1}}$ *is the* $p+1$*-th spatial derivative. Moreover,* $\triangle t \geq t_+ - t_-$*,* $\triangle \mathbf{x} \geq \operatorname{diam}(K)$ *and* $p \leq p_R$ *and* $q \leq q_R$ *for all* $R = K \times (t_-, t_+)$.

Proof. With Galerkin orthogonality (4.19) we achieve that for all $\mathbf{v}_h \in V_h$ and $\mathbf{w}_h \in W_h$

$$b_h(\mathbf{v}_h - \mathbf{u}_h, \mathbf{w}_h) = b_h(\mathbf{v}_h - \mathbf{u}, \mathbf{w}_h) \leq \|\mathbf{v}_h - \mathbf{u}\|_{V_h} \|\mathbf{w}_h\|_W$$

and thus

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{V_h} &\leq \|\mathbf{u} - \mathbf{v}_h\|_{V_h} + \|\mathbf{v}_h - \mathbf{u}_h\|_{V_h} \\ &\leq \|\mathbf{u} - \mathbf{v}_h\|_{V_h} + \beta^{-1} \sup_{\mathbf{w}_h \in W_h \setminus \{\mathbf{0}\}} \frac{b_h(\mathbf{v}_h - \mathbf{u}_h, \mathbf{w}_h)}{\|\mathbf{w}_h\|_W} \\ &\leq (1 + \beta^{-1}) \|\mathbf{u} - \mathbf{v}_h\|_{V_h} . \end{aligned}$$

Now we assume that the solution is regular satisfying $\mathbf{u} \in \mathrm{H}^{q+1}(0, T; \mathrm{L}_2(\Omega)^J) \cap \mathrm{L}_2(0, T; \mathrm{H}^{p+1}(\Omega)^J)$. By consistency we have that $A_h \mathbf{v}_h = \Pi_h A \mathbf{v}_h$ for all $\mathbf{v}_h \in V_h \cap \mathrm{H}^1(Q)^J$ and the following error estimate yields

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{V_h} &\leq (1 + \beta^{-1}) \inf_{\mathbf{v}_h \in V_h \cap \mathrm{H}^1(Q)^J} \|\mathbf{u} - \mathbf{v}_h\|_{V_h} \\ &\leq C \Big(\|\partial_t (\mathbf{u} - \mathcal{C}_h \mathbf{u})\|_{0,Q} + \|\mathrm{D}(\mathbf{u} - \mathcal{C}_h \mathbf{u})\|_{0,Q} \Big) . \end{aligned}$$

Here we used a suitable Clément-type (patch-wise) interpolation operator $\mathcal{C}_h \colon V \to V_h \cap \mathrm{H}^1(Q)^J$. To do so, we have to assume that the meshes $\{\mathcal{R}_h\}$ are shape-regular and admissible, see Definitions 4.2 and 4.3, and that Q is covered by every mesh $\mathcal{R}_h$ exactly, see for example [EG04, Sec. 1.6.1] or [Bra13, Ch. II, §6]. Thus we obtain a bound depending on $\triangle t^q$ in time and $\triangle \mathbf{x}^p$ in space. □

Example 4.1 (Implicit midpoint rule)**.** In this example, we verify that for lowest polynomial degrees in time, the tensor product Petrov–Galerkin discretization is equivalent to a discontinuous Galerkin method in space, equipped with the implicit midpoint rule in time.
Therefore we choose constant polynomial degrees $p_R \equiv p$ and $q \equiv 1$ for all $R \in \mathcal{R}$ on a tensor product discretization. Due to (4.15), it yields for all $\mathbf{v}_h \in V_h$ that

$$\begin{aligned} \mathbf{v}_h(\,.\,, t) &= \frac{t_n - t}{\triangle t} \mathbf{v}_{h,n-1}(\,.\,) + \frac{t - t_{n-1}}{\triangle t} \mathbf{v}_{h,n}(\,.\,), \\ \partial_t \mathbf{v}_h(\,.\,, t) &= \frac{1}{\triangle t} \Big(\mathbf{v}_{h,n}(\,.\,) - \mathbf{v}_{h,n-1}(\,.\,) \Big), \end{aligned} \tag{4.20}$$

for $t \in I = (t_{n-1}, t_n)$ and $\triangle t = t_n - t_{n-1}$. The ansatz and test spaces are illustrated in Figure 4.7. Restricting the weak form of the problem to only one time interval I results in the following subproblem. Find $\mathbf{u}_h \in V_h$, such that

$$\begin{aligned} (M_h \partial_t \mathbf{u}_h + A_h \mathbf{u}_h, \mathbf{w}_h)_{0,\Omega \times I} &= (\mathbf{f}, \mathbf{w}_h)_{0,\Omega \times I}, \quad \text{for all } \mathbf{w}_h \in W_h, \\ \mathbf{u}_h(\,.\,, t_{n-1}) &= \mathbf{u}_{h,n-1}(\,.\,), \qquad \text{on } \Omega \times \{t_{n-1}\}. \end{aligned}$$

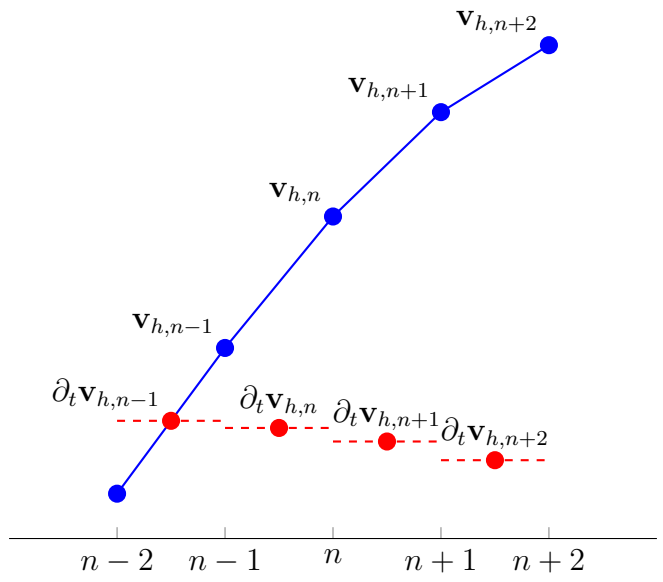

Figure 4.7: Illustration of the implicit midpoint rule, where $\mathbf{v}_h \in V_h$ and $\partial_t \mathbf{v}_h \in W_h$.

Since $\mathbf{u}_h$ is linear in time, we can replace the temporal integral by a suitable quadrature formula, e.g., the midpoint rule (cf. [HB09, Ch. 36]). Moreover, we choose the special test function $\mathbf{w}_h \equiv 1$. Hence we conclude

$$(M_h\partial_t + A_h)\,\mathbf{u}_h(\,.\,, \frac{t_n + t_{n-1}}{2}) = \mathbf{f}_{n-\frac{1}{2}}(\,.\,),$$

where $\mathbf{f}_{n-\frac{1}{2}}(\,.\,) = \mathbf{f}(\,.\,, \frac{t_n+t_{n-1}}{2})$. We now use the representation (4.20) of $\mathbf{u}_h$ and $\partial_t\mathbf{u}_h$ to recover the implicit midpoint rule

$$\begin{aligned} & \frac{1}{\triangle t}(M_h + \frac{1}{2}A_h)\,(\mathbf{v}_{h,n}(\,.\,) - \mathbf{v}_{h,n-1}(\,.\,)) = \mathbf{f}_{n-\frac{1}{2}}(\,.\,) \\ \iff \quad & (M_h + \frac{1}{2}\triangle t A_h)\mathbf{v}_{h,n}(\,.\,) = (M_h - \frac{1}{2}\triangle t A_h)\mathbf{v}_{h,n-1}(\,.\,) + \triangle t\mathbf{f}_{n-\frac{1}{2}}(\,.\,). \end{aligned}$$

This scheme, applied to the wave equation, is investigated by Bangerth et al. [BGR10] in detail.

Depending on the ansatz and test space, (Petrov–) Galerkin approaches in time can be traced back to a Runge–Kutta time stepping scheme. See for example [AM89], [Hul72] for continuous Petrov–Galerkin or [Neu13], [ZW14] for discontinuous Galerkin (dG) schemes. In the final section of this chapter we introduce a nodal discretization for our discontinuous Petrov–Galerkin space-time discretization and discuss the possibility of solving the resulting linear system using either a time stepping scheme or a space-time solver.

4.6 Nodal Space-Time Discretization

In this section we apply the finite element method from Section 4.2.2 to the discontinuous Petrov–Galerkin discretization. This means that we provide the tools to discretize the local spaces $V_{h,R}$ and $W_{h,R}$ and derive a linear system of equations which can be solved numerically.

We suppose that $\mathcal{K}$ is a spatial tetrahedral mesh, then the occurring space-time cells $R = K \times I$ for $K \in \mathcal{K}$ and $I = (t_-, t_+) \subset (0, T)$ are prism-shaped (cf. Figure 4.4 and 4.5). To achieve a space-time nodal discretization we define prismatic coordinates corresponding to Definition 4.6. Therefore we extend the definition given in [EG04, Sec. 1.2.5] such that different polynomial degrees in space and time are allowed. Again only the scalar case $J = 1$ is considered. As before, the following techniques can be applied component-wise in vector valued case for $J > 1$ (cf. Remark 4.2).

Definition 4.11 (Prismatic coordinates)**.** Let $R \in \mathcal{R}$ be a prism-shaped space-time cell and consider polynomial degrees $p_R, q_R \geq 0$ in space and time, respectively. Similar to Definition 4.6, the numbers $\lambda_0, \lambda_2, \ldots, \lambda_D, \lambda_{D+1}, \lambda_{D+2}$ are called *prismatic coordinates* and are given as

$$(\lambda_0, \ldots, \lambda_D) = \begin{cases} \left(\frac{1}{D+1}, \ldots, \frac{1}{D+1}\right), & \text{for } p_R = 0, \\ \left(\frac{i_0}{p_R}, \ldots, \frac{i_D}{p_R}\right),\ 0 \leq i_0, \ldots, i_D \leq p_R,\ i_0 + \ldots + i_D = p_R, & \text{for } p_R > 0, \end{cases}$$

and

$$(\lambda_{D+1}, \lambda_{D+2}) = \begin{cases} \left(\frac{1}{2}, \frac{1}{2}\right), & \text{for } q_R = 0, \\ \left(\frac{i_{D+1}}{q_R}, \frac{i_{D+2}}{q_R}\right),\ 0 \leq i_{D+1}, i_{D+2} \leq q_R,\ i_{D+1} + i_{D+2} = q_R, & \text{for } q_R > 0. \end{cases}$$

Hence every node $(\mathbf{x}_i, t_i) \in \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_{n_R}, t_{n_R})\}$ can be represented as

$$(\mathbf{x}_i, t_i) = (\lambda_0 \mathbf{y}_0 + \ldots + \lambda_D \mathbf{y}_D, \lambda_{D+1} t_- + \lambda_{D+2} t_+),$$

where $\{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_D\}$ is the set of vertices of the spatial cell K.

Lemma 4.8. *Let $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_{n_R}, t_{n_R})\}$ be the set of nodes associated to $R \in \mathcal{R}$ and $\mathbb{P}_{p_R,q_R}(R) = \mathbb{P}_{p_R}(K) \times \mathbb{P}_{q_R}(I)$. Moreover, let $\Sigma_R = \{l_1, \ldots, l_{n_R}\}$ be the set of degrees of freedom such that*

$$l_i(\psi_j) = \psi_j(\mathbf{x}_i, t_i), \quad \textit{for all } 1 \leq i, j \leq n_R \textit{ and } \psi_j \in \mathbb{P}_{p_R,q_R}(R),$$

then $(R, \mathbb{P}_{p_R,q_R}(R), \Sigma_R)$ is a nodal finite element with shape functions ψ_j. Furthermore, the local nodal interpolation operator in space-time is given as

$$I_R \colon \mathrm{C}^0(R) \to \mathbb{P}_{p_R,q_R}(R), \qquad I_R v = \sum_{i=1}^{n_R} v(\mathbf{x}_i, t_i)\psi_i. \tag{4.21}$$

Proof. Verify Definitions 4.5 and 4.6. □

Since we deal with different polynomial degrees on every cell, we use the nodal interpolation operator given in equation (4.21) to interpolate functions between different polynomial spaces. We consider two different polynomial spaces $\mathbb{P}_{p,q}(R)$ and $\mathbb{P}_{p',q'}(R)$ with $p, q, p', q' \geq 0$, on the same space-time cell R. The interpolation of $v'_h \in \mathbb{P}_{p',q'}(R)$ on $\mathbb{P}_{p,q}(R)$ is then given as

$$v_h = I_R v'_h = \sum_{i=1}^{n_R} v'_h(\mathbf{x}_i, t_i)\psi_i,$$

where I_R is the local nodal interpolation operator given in (4.21) and $\{\psi_i\}_{i=1,\dots,n_R}$ the corresponding nodal basis on $\mathbb{P}_{p,q}(R)$. Hence the coefficients of v_h are given as $\underline{v}_i = v'_h(\mathbf{x}_i, t_i)$ for $i = 1, \dots, n_R$. Since v'_h is a polynomial, it can be represented as

$$v' = \sum_{i=1}^{n'_R} \underline{v}'_j \psi'_j$$

by using the coefficient vector $\underline{v}' = \left(\underline{v}'_1, \dots, \underline{v}'_{n_{R'}}\right)^\top$ and the corresponding nodal basis $\{\psi'_i\}_{i=1,\dots,n_{R'}}$ of $\mathbb{P}_{p',q'}(R)$. The coefficient vector $\underline{v} = (\underline{v}_1, \dots, \underline{v}_{n_R})^\top$ of v_h can then be computed as

$$\underline{v} = \underline{I}^{p,q}_{p',q'} \underline{v}' = \begin{pmatrix} \psi'_0(\mathbf{x}_0, t_0) & \dots & \psi'_{n_{R'}}(\mathbf{x}_0, t_0) \\ \vdots & & \vdots \\ \psi'_0(\mathbf{x}_{n_R}, t_{n_R}) & \dots & \psi'_{n_{R'}}(\mathbf{x}_{n_R}, t_{n_R}) \end{pmatrix} \begin{pmatrix} \underline{v}'_1 \\ \vdots \\ \underline{v}'_{n_{R'}} \end{pmatrix}. \tag{4.22}$$

The $n_R \times {n_R}'$-matrix $\underline{I}^{p,q}_{p',q'}$ is denoted as *interpolation matrix*. The following remark collects some useful properties of the space-time finite element $(R, \mathbb{P}_{p_R,q_R}(R), \Sigma_R)$ defined in Lemma 4.8.

Remark 4.6. Due to the prismatic structure of $R = K \times I \in \mathcal{R}$, the Lagrange shape functions decouple in space and time. Hence it holds for every Lagrange shape function $\psi_i \in \mathbb{P}_{p_R,q_R}(R)$ of the finite element $(R, \mathbb{P}_{p_R,q_R}(R), \Sigma_R)$ that

$$\psi_i(\mathbf{x}, t) = \phi_i(\mathbf{x})\lambda_i(t), \qquad \text{for all } 1 \leq i \leq n_R,$$

where ϕ_i and λ_i are Lagrange shape functions of the spatial and temporal nodal finite elements $(K, \mathbb{P}_{p_K}(K), \Sigma_K)$ and $(I, \mathbb{P}_{q_I}(I), \Sigma_I)$, respectively. As a direct consequence, the dimension n_R of $\mathbb{P}_{p_R,q_R}(R)$ can be computed as

$$\begin{aligned} n_R = \dim(\mathbb{P}_{p_R,q_R}(R)) &= \dim(\mathbb{P}_{p_R}(K)) \dim(\mathbb{P}_{q_R}(I)) \\ &= \begin{cases} (p_R+1)(q_R+1), & \text{for } D = 1, \\ \frac{1}{2}(p_R+1)(p_R+2)(q_R+1), & \text{for } D = 2, \\ \frac{1}{6}(p_R+1)(p_R+2)(p_R+3)(q_R+1), & \text{for } D = 3, \end{cases} \end{aligned}$$

(cf. Remark 4.3).

To simplify notation we again consider a tensor product space-time mesh $\mathcal{R} = \bigcup_{n=1}^{N} \mathcal{R}^n$ with time slices $\mathcal{R}^n = \bigcup_{K\in\mathcal{K}} K \times (t_{n-1}, t_n)$, where $\mathcal{K}$ is an underlying fixed spatial mesh (cf. Definition 4.9). In every space-time cell $R = K \times (t_{n-1}, t_n) \in R$ we choose polynomial degrees $p_R \geq 0$, $q_R > 0$ in space and time, respectively. Furthermore we use the nodal finite element $\left(R, \mathbb{P}^J_{p_R,q_R}(R), \Sigma_R\right)$ introduce in Lemma 4.8 and the set of nodal shape functions $\{\boldsymbol{\psi}^n_{R,j}\}_{j=1,\dots,\dim W_{h,R}}$ as a basis of the local test spaces $W_{h,R}$. On every time slice $\mathcal{R}^n$ we define the subspace $W^n_h = \operatorname{span}\{\bigcup_{R\in\mathcal{R}^n} \bigcup_j^{\dim W_{h,R}} \boldsymbol{\psi}^n_{R,j}\} \subset W_h$.

According to (4.16), $\mathbf{v}_h \in V_h$ is represented by

$$\mathbf{v}_h(\mathbf{x}, t) = \frac{t_n - t}{t_n - t_{n-1}} \mathbf{w}^{n-1}_h(\mathbf{x}, t_{n-1}) + \frac{t - t_{n-1}}{t_n - t_{n-1}} \mathbf{w}^n_h(\mathbf{x}, t),$$

for all $(\mathbf{x}, t) \in \Omega \times (t_{n-1}, t_n)$, where $\mathbf{w}^0_h = \mathbf{0}$ and $\mathbf{w}^n_h \in W^n_h$ and $n = 1, \dots, N$. The corresponding coefficient vector on $\mathcal{R}$ is denoted by $\underline{w} = (\underline{w}^1, \dots, \underline{w}^N)^\top$, where $\underline{w}^n \in \mathbb{R}^{\dim W^n_h}$ is the coefficient vector of the representation

$$\mathbf{w}^n_h = \sum_{R\in\mathcal{R}^n} \sum_{j=1}^{\dim W_{h,R}} \underline{w}^n_{R,j} \boldsymbol{\varphi}^n_{R,j}$$

on a time slice $\mathcal{R}^n$. Here $\{\boldsymbol{\varphi}^n_{R,j}\}_{j=1,\dots,\dim W_{h,R}}$ is again a basis of $W_{h,R}$. In the most simple case one can choose the nodal shape functions as a basis, i.e., $\boldsymbol{\psi}^n_{R,j} = \boldsymbol{\varphi}^n_{R,j}$. However other basis functions are possible, too. For example, one can choose $\{\boldsymbol{\varphi}^n_{R,j}\}_{j=1,\dots,\dim W_{h,R}}$ such that the functions $\boldsymbol{\phi}^n_{R,j} \in V_{h,R}$ defined as

$$\boldsymbol{\phi}^n_{R,j}(\,.\,, t) = \frac{t_n - t}{t_n - t_{n-1}} \boldsymbol{\varphi}^{n-1}_{R',j}(\,.\,, t_{n-1}) + \frac{t - t_{n-1}}{t_n - t_{n-1}} \boldsymbol{\varphi}^n_{R,j}(\,.\,, t), \qquad \text{for all } t \in (t_{n-1}, t_n),$$

build a basis $\{\boldsymbol{\phi}^n_{R,j}\}_{j=1,\dots,\dim V_{h,R}}$ of nodal shape functions on the local ansatz space $V_{h,R}$. Here $R' = K \times (t_{n-2}, t_{n-1}) \in \mathcal{R}^{n-1}$ denotes the preceding space-time cell corresponding to $R = K \times (t_{n-1}, t_n) \in \mathcal{R}^n$ and $K \in \mathcal{K}$. Hence both spaces, V_h and W_h, can be represented by using a nodal structure. Moreover, $\mathbf{v}_h \in V_h$ can be directly expressed as

$$\mathbf{v}_h(\mathbf{x}, t) = \sum_{R\in\mathcal{R}^n} \sum_{j=1}^{\dim V_{h,R}} \underline{v}^n_{R,j} \boldsymbol{\phi}^n_{R,j}(\mathbf{x}, t),$$

for all $(\mathbf{x}, t) \in \Omega \times (t_{n-1}, t_n)$, where the continuity condition is incorporated into the coefficient vector $\underline{v} = (\underline{v}^1, \dots, \underline{v}^N)^\top$.

Finally, we test the discrete space-time system given in equation (4.18) with basis functions $\boldsymbol{\psi}^n_{R,j}$ of $W_{h,R}$ and achieve

$$(L_h \mathbf{u}_h, \boldsymbol{\psi}^n_{R,j})_{0,Q} = (\mathbf{f}, \boldsymbol{\psi}^n_{R,j})_{0,Q},$$

for all $n = 1, \dots, N$ and $j = 1, \dots, \dim W_{h,R}$ and $R \in \mathcal{R}$. Together with a nodal representation of $\mathbf{u}_h \in V_h$ this results in a linear system of equations with the matrix representation

$$\underline{L}\underline{u} = \underline{f}, \tag{4.23}$$

where $\underline{u} \in \mathbb{R}^{\dim V_h}$ is the coefficient vector of $\mathbf{u}_h$ and $\underline{L}$ a block-matrix

$$\underline{L} = \begin{pmatrix} \underline{D}^1 & & & \\ \underline{C}^1 & \underline{D}^2 & & \\ & \ddots & \ddots & \\ & & \underline{C}^{N-1} & \underline{D}^N \end{pmatrix}.$$

The block-entries on the diagonal are given as

$$\underline{D}^n_{R',k,R,j} = \int_{t_{n-1}}^{t_n} \Big(L_h\big(\frac{t - t_{n-1}}{t_n - t_{n-1}} \boldsymbol{\varphi}^n_{R,j}(\cdot, t)\big), \boldsymbol{\psi}^n_{R',k}(\cdot, t) \Big)_{0,\Omega} \,\mathrm{d}t, \quad \text{for } n = 1, \dots, N,$$

where $R, R' \in \mathcal{R}^n$ are cells within a common slice. The block-entries on the lower sub-diagonal are given as

$$\underline{C}^{n-1}_{R',k,R,j} = \int_{t_{n-1}}^{t_n} \Big(L_h\big(\frac{t_n - t}{t_n - t_{n-1}} \boldsymbol{\varphi}^{n-1}_{R,j}(\cdot, t_{n-1})\big), \boldsymbol{\psi}^n_{R',k}(\cdot, t) \Big)_{0,\Omega} \,\mathrm{d}t, \quad \text{for } n = 2, \dots, N,$$

where $R \in \mathcal{R}^{n-1}$ and $R' \in \mathcal{R}^n$ are cells within two batched slices. The right-hand side is computed by $\underline{f} = (\underline{f}^1, \dots, \underline{f}^N)$ with $\underline{f}^n_{j,R} = (\mathbf{f}, \boldsymbol{\psi}^n_{R,j})_R$.

Note that the system has a lower triangular structure. Thus it can be solved iteratively by using a block-Gauss–Seidel method

$$\underline{D}^1\underline{u}^1 = \underline{f}^1, \quad \underline{D}^2\underline{u}^2 = \underline{f}^2 - \underline{C}^1\underline{u}^1, \quad \dots, \quad \underline{D}^N\underline{u}^N = \underline{f}^N - \underline{C}^{N-1}\underline{u}^{N-1}, \tag{4.24}$$

where each block corresponds to a time slice $\mathcal{R}^n$, $n = 1, \dots, N$. This requires that $\underline{D}^n$ can be inverted efficiently and hence corresponds to an implicit time integration method. Due to this property, Petrov–Galerkin methods are often used to derive and investigate new time stepping schemes (see for example [KB14] or [MS11]). With respect to parallelization, only a distribution of the problem to various processes in space is possible in this case (cf. Figure 4.8).

In this work, we stick to the full space-time block-system to develop a multigrid preconditioner with a fully distributed space-time mesh (cf. Figure 4.9). This results in a parallel in space and time preconditioner for iterative solvers. Moreover this setting allows more flexibility in the application of space-time adaptivity as described in the following chapter.

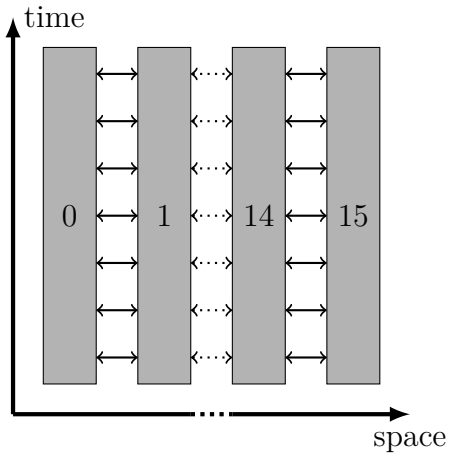

Figure 4.8: Spatial distribution of space-time cells to 16 processes and required communication (arrows).

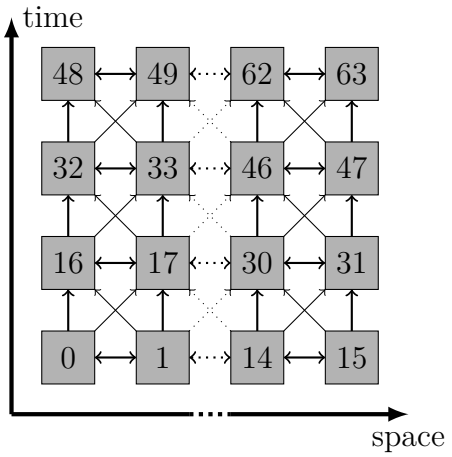

Figure 4.9: Space-time distribution of space-time cells to 64 processes and required communication (arrows).

5 Error Estimation and Adaptivity

In the previous chapter we have seen that space-time finite element methods lead to very large linear systems which have to be solved. Especially in case of real applications, where highly accurate solutions are needed, space-time methods can easily reach the range of hundreds of billion degrees of freedom. However, in most applications it is not necessary to use the same fine mesh size or high-order polynomials on every cell. Either because the solution is already "nice" in some parts of the domain and can be computed accurate enough with lower order polynomials or a coarser mesh size, or one is not even interested in these parts of the solution. Hence we want to develop a strategy which adapts the computed solution to the problem automatically. These kind of algorithms are called *adaptive*. In context of finite element methods one usually considers two different adaptive strategies to reduce the error of the computed solution. The first one consists of local adjustments of the mesh by refining or coarsening cells and is called *h-adaptivity*. The second one is called *p-adaptivity* and adjusts the polynomial degree of the discretization locally on every cell. Combining both methods results in the so called *hp-adaptivity* which can be used to achieve an exponential decrease of the error, see, e.g., [Dem07] and [DKP+08]. In many cases the crucial point concerning adaptive methods is not the refinement itself, but the development of suitable *error estimators* or *indicators* which estimate the errors locally on every cell. Here one usually uses representations of the error, a posteriori error bounds or heuristic arguments to compute approximations to the local errors.

In this work we focus on p-adaptivity and applications where one is only interested in small parts or certain properties of the solution. For this purpose so called *dual weighted residuals* (DWR) or *goal oriented* methods are used. These kind of methods, introduced in [BR96], base on duality techniques which are used to achieve a posteriori error representations. In this chapter we first give a short introduction to DWR methods and derive suitable error representations. On this basis we state a local error indicator to be able to estimate the error locally on every cell. Finally, we assemble an adaptive strategy and state the corresponding adaptive algorithm.

General introductions to adaptive finite element methods can be found for example in [BS08], [Ver96], [EG04, Ch. 10] and [Ver13].

5.1 Dual Weighted Residuals

The idea of DWR based methods is to model the *quantity of interest* in terms of an error functional $\mathcal{J}\colon \mathrm{L}_2(Q)^J \to \mathbb{R}$. For example one can use linear functionals to model point or mean errors (cf. Section 5.2.1) or quadratic functionals to model energy errors (cf. Section 5.2.2). The goal is to minimize the error

$$\triangle\mathcal{J}(\mathbf{u}, \mathbf{u}_h) = \mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_h)$$

with respect to $\mathcal{J}$, where $\mathbf{u}_h \in V_h$ is computed as cheap as possible. Therefore we have to find an error representation of $\triangle\mathcal{J}$ which can be used to derive a computable quantity $\mathcal{J}_h$ such that

$$|\triangle\mathcal{J}| \approx |\triangle\mathcal{J}_h| \leq \eta_{\mathcal{R}} = \sum_{R\in\mathcal{R}} \eta_R,$$

where $\eta_R \in \mathbb{R}$ are quantities correlating with the local errors on $R \in \mathcal{R}$.

In the following section we derive error representations for our discontinuous Petrov–Galerkin space-time discretization. In [BR01] and [BR03, Ch. 6] the general case of conforming finite elements is discussed in detail. Moreover, a continuous Petrov–Galerkin space-time discretization for the wave equation equipped with standard continuous finite elements in space is considered in [BGR10]. Thus we extend the ideas given in [BR03, Ch. 6] and [BGR10] to the case of DG-FEM in space using upwind fluxes.

5.2 Error Representation

In this section we want to derive an error representation for $\triangle\mathcal{J}(\mathbf{u}, \mathbf{u}_h) = \mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_h)$ and a given general (nonlinear) error functional $\mathcal{J}\colon \mathrm{L}_2(Q)^J \to \mathbb{R}$. Since the exact solution is unknown, the error representation of $\triangle\mathcal{J}$ should be independent of $\mathbf{u}$. We will see that this can be achieved by introducing an *adjoint* problem and its corresponding solution.

Definition 5.1 (Adjoint operator)**.** Let $L\colon V \to V^* \subseteq W$ be the bounded, linear space-time operator defined in Section 3.2. Due to the *Riesz representation theorem* we can express the dual (or *adjoint*) Hilbert space of V, defined in (3.5), equivalently as

$$\begin{aligned} V^* = \Big\{ \mathbf{w} \in W\colon \ &\text{there exists a unique } \mathbf{z} \in V \text{ such that} \\ &(L\mathbf{v}, \mathbf{w})_{0,Q} = (\mathbf{v}, \mathbf{z})_{0,Q}, \ \text{for all } \mathbf{v} \in V \Big\} \end{aligned} \tag{5.1}$$

(cf. [Wer11, Thm. V.3.6]). Moreover, the operator $L^*\colon V^* \to V$, which satisfies

$$(L\mathbf{v}, \mathbf{w})_{0,Q} = (\mathbf{v}, L^*\mathbf{w})_{0,Q} \tag{5.2}$$

for all $\mathbf{v} \in V$ and $\mathbf{w} \in V^*$, is called *adjoint* or *dual* operator with respect to L. Hence we conclude that $\mathbf{z} = L^*\mathbf{w}$, where $\mathbf{z} \in V$ is given as in (5.1).

Note that the homogeneous initial conditions incorporated in V are transferred to homogeneous final conditions in V^*, i.e., $\mathbf{w}(T) = \mathbf{0}$ for all $\mathbf{w} \in V^*$.

Lemma 5.1. *(Linear hyperbolic case) For hyperbolic evolution equations introduced in Section 3.2, the adjoint operator of L is given as*

$$L^* = -M\partial_t + A^*,$$

where A^* is the adjoint spatial operator of A. Moreover, it yields that

$$A^* = -A \qquad \text{on } \mathcal{D}(A) \cap \mathcal{D}(A^*).$$

Proof. By applying the integration by parts formula in time we derive from the previous definition that

$$\begin{aligned}(\mathbf{v}, L^*\mathbf{v}^*)_{0,Q} &= (L\mathbf{v}, \mathbf{v}^*)_{0,Q} = (M\partial_t\mathbf{v} + A\mathbf{v}, \mathbf{v}^*)_{0,Q} \\ &= (\mathbf{v}, -M\partial_t\mathbf{v}^* + A^*\mathbf{v}^*)_{0,Q} + (M\mathbf{v}(T), \mathbf{v}^*(T))_{0,\Omega} - (M\mathbf{v}(0), \mathbf{v}^*(0))_{0,\Omega} \\ &= \left(\mathbf{v}, (-M\partial_t + A^*)\mathbf{v}^*\right)_{0,Q}\end{aligned}$$

holds for all $\mathbf{v} \in V$ and $\mathbf{v}^* \in V^*$. Here we used that V and V^* contain homogeneous initial or final conditions, respectively. In the same way one can prove that

$$\begin{aligned}(\mathbf{v}, A^*\mathbf{v}^*)_{0,\Omega} &= (A\mathbf{v}, \mathbf{v}^*)_{0,\Omega} = (\nabla \cdot \mathbf{F}(\mathbf{v}), \mathbf{v}^*)_{0,\Omega} \\ &= -(\mathbf{v}, \nabla \cdot \mathbf{F}(\mathbf{v}^*))_{0,\Omega} + (\mathbf{n}_{\partial\Omega} \cdot \mathbf{F}(\mathbf{v}), \mathbf{v}^*)_{0,\partial\Omega} \\ &= -(\mathbf{v}, \nabla \cdot \mathbf{F}(\mathbf{v}^*))_{0,\Omega} = -(\mathbf{v}, A\mathbf{v}^*)_{0,\Omega}\end{aligned}$$

for $\mathbf{v} \in \mathcal{D}(A)$ and $\mathbf{v}^* \in \mathcal{D}(A^*)$, respectively, since we incorporated homogeneous boundary conditions on the inflow boundary (linear transport equation) or perfectly conducting boundary conditions (Maxwell's equations), respectively (cf. Remarks 3.3 and 3.6). □

As introduced in [BR01] and [BGR10], we are now able to derive an adjoint formulation of our original problem. Let $\mathbf{u} \in V$ and consider the optimization problem

$$\mathcal{J}(\mathbf{u}) = \min! \quad \text{under the constraint that } \mathbf{u} \text{ solves (4.18).}$$

Since we have shown in Theorem 3.1 that a unique solution exists, this problem is trivial. However to determine a solution of the optimization problem, we can introduce the Lagrange functional $\mathcal{E}\colon V \times V^* \to \mathbb{R}$ such that

$$\begin{aligned}\mathcal{E}(\mathbf{v}, \mathbf{v}^*) =& \mathcal{J}(\mathbf{v}) + (\mathbf{f}, \mathbf{v}^*)_{0,Q} - (L\mathbf{v}, \mathbf{v}^*)_{0,Q} \\ =& \mathcal{J}(\mathbf{v}) + (\mathbf{f}, \mathbf{v}^*)_{0,Q} - (\mathbf{v}, L^*\mathbf{v}^*)_{0,Q}.\end{aligned}$$

Seeking for a stationary point $\{\mathbf{u}, \mathbf{u}^*\} \in V \times V^*$ leads to the requirement that

$$\mathcal{E}'(\mathbf{u}, \mathbf{u}^*; \mathbf{v}, \mathbf{v}^*) = \begin{pmatrix} \mathcal{J}'(\mathbf{u}; \mathbf{v}) - (\mathbf{v}, L^*\mathbf{u}^*)_{0,Q} \\ (\mathbf{f}, \mathbf{v}^*)_{0,Q} - (L\mathbf{u}, \mathbf{v}^*)_{0,Q} \end{pmatrix} = \mathbf{0}, \tag{5.3}$$

for all $\mathbf{v} \in V, \mathbf{v}^* \in V^*$. Since $V^* \subseteq W$ the second condition is fulfilled for the solution $\mathbf{u} \in V$ of the original *primal problem*

$$(L\mathbf{u}, \mathbf{w})_{0,Q} = (\mathbf{f}, \mathbf{w})_{0,Q}, \qquad \text{for all } \mathbf{w} \in W.$$

Moreover, the first condition of (5.3) motivates the following definition of an additional problem:

Definition 5.2 (Dual problem)**.** For a given *primal solution* $\mathbf{u} \in V$ of (4.18), seek a solution $\mathbf{u}^* \in V^*$ such that

$$(L^*\mathbf{u}^*, \mathbf{w})_{0,Q} = \mathcal{J}'(\mathbf{u}; \mathbf{w}), \qquad \text{for all } \mathbf{w} \in W. \tag{5.4}$$

This problem is denoted as *dual problem* and correspondingly $\mathbf{u}^* \in V^*$ is called *dual solution*. Note that the primal problem is independent of $\mathbf{u}^*$, whereas the dual problem depends on $\mathbf{u}$ for nonlinear $\mathcal{J}$. We conclude that the primal and dual solution $\mathbf{u}$ and $\mathbf{u}^*$ are stationary points of the Lagrange functional $\mathcal{E}$.

Assumption 5.1. Throughout this work we assume that the dual solution $\mathbf{u}^* \in V^*$ of problem (5.4) is sufficiently smooth, i.e., $\mathbf{u}^*(., t)|_f \in \mathrm{L}_2(f)^J$ for all faces $f \in \mathcal{F}_\mathcal{R}$ and almost all $t \in (0, T)$.

Definition 5.3 (Discrete dual solution)**.** Correspondingly to the primal case let $\mathbf{u}_h^* \in W_h$ be an approximation of the dual solution which solves

$$b_h(\mathbf{v}_h, \mathbf{u}_h^*) = (L_h\mathbf{v}_h, \mathbf{u}_h^*)_{0,Q} = \mathcal{J}'(\mathbf{u}_h; \mathbf{v}_h), \qquad \text{for all } \mathbf{v}_h \in V_h. \tag{5.5}$$

Furthermore, we assume that

$$(L^*\mathbf{u}^*, \mathbf{u}_h)_{0,Q} = \mathcal{J}'(\mathbf{u}; \mathbf{u}_h). \tag{5.6}$$

holds for the discrete solution $\mathbf{u}_h \in V_h$ (cf. [BGR10, Cor. 4.1]). Note that this definition is only justified by numerical experiments. Within a rigorous analysis one has to define a corresponding discrete dual operator L_h^* and a discretization with ansatz and test spaces V_h^* and W_h^*, respectively. Finally, one has to prove that $\mathbf{u}_h^* \in V_h^*$ obtained by solving

$$(L_h^*\mathbf{u}_h^*, \mathbf{w}_h^*)_{0,Q} = \mathcal{J}'(\mathbf{u}_h; \mathbf{w}_h^*), \qquad \text{for all } \mathbf{w}_h^* \in W_h^*,$$

is indeed an approximation to the dual solution $\mathbf{u}^* \in V^*$ and that (5.6) is satisfied.

Definition 5.4 (Residuals and errors)**.** We define

$$\rho(\mathbf{u}_h, \mathbf{w}) = (\mathbf{f}, \mathbf{w})_{0,Q} - b_h(\mathbf{u}_h, \mathbf{w}), \tag{5.7}$$

$$\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{v}) = \mathcal{J}'(\mathbf{u}_h; \mathbf{v}) - b_h(\mathbf{v}, \mathbf{u}_h^*) \tag{5.8}$$

for all $\mathbf{v} \in V$ and $\mathbf{w} \in W$ and denote ρ as *primal residual* and ρ^* as *dual residual*. Note that in the general cases the dual residual (5.8) reads as

$$\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{v}) = \mathcal{J}'(\mathbf{u}_h; \mathbf{v}) - (L_h^* \mathbf{u}_h^*, \mathbf{v})_{0,Q}, \qquad \text{for all } \mathbf{v} \in W,$$

and the subsequent results can be transferred analogously (cf. [BR03, Ch. 6]). Moreover, we define the *primal error* $\mathbf{e}$ and the *dual error* $\mathbf{e}^*$ as

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_h \qquad \text{and} \qquad \mathbf{e}^* = \mathbf{u}^* - \mathbf{u}_h^*.$$

In the following lemma we derive an error representations for $\triangle\mathcal{J}$ depending only on the primal residual ρ and terms of higher-order.

Lemma 5.2 (cf. [BR03, Prop. 6.6])**.** *The error representation*

$$\triangle\mathcal{J} = \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \mathcal{T}_2$$

holds for all $\mathbf{w}_h \in W_h$ with second-order remainder term

$$\mathcal{T}_2(\mathbf{u}, \mathbf{u}_h) = -\int_0^1 s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s$$

with respect to $\mathcal{J}$.

Proof. First we apply Taylor's theorem to $\mathcal{J}$ and $\mathcal{J}'$, respectively, and achieve

$$\triangle\mathcal{J} = \mathcal{J}(\mathbf{u}) - \mathcal{J}(\mathbf{u}_h) = \mathcal{J}'(\mathbf{u}_h; \mathbf{e}) + \int_0^1 (1-s)\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s,$$

$$\mathcal{J}'(\mathbf{u}; \mathbf{e}) = \mathcal{J}'(\mathbf{u}_h; \mathbf{e}) + \int_0^1 \mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s.$$

Using the second equation within the first one leads to

$$\triangle\mathcal{J} = \mathcal{J}'(\mathbf{u}; \mathbf{e}) - \int_0^1 s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s = \mathcal{J}'(\mathbf{u}; \mathbf{e}) + \mathcal{T}_2(\mathbf{u}, \mathbf{u}_h).$$

Moreover, due to consistency of the upwind flux (cf. Lemma 4.3), it holds that

$$\begin{aligned}(\mathbf{u}_h, L^*\mathbf{u}^*)_{0,Q} &= \sum_{R \in \mathcal{R}} \Big((L\mathbf{u}_h, \mathbf{u}^*)_{0,R} - (\mathbf{u}_h, \mathbf{n}_R \cdot \mathbf{F}(\mathbf{u}^*))_{0,\partial R} \Big) \\ &= \sum_{R = K \times I \in \mathcal{R}} \Big((L\mathbf{u}_h, \mathbf{u}^*)_{0,R} \\ &\qquad + (\mathbf{n}_K \cdot (\mathbf{F}^{\mathrm{num}}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_h)), \mathbf{u}^*)_{0,\partial K \times I} \Big) \\ &= b_h(\mathbf{u}_h, \mathbf{u}^*).\end{aligned} \tag{5.9}$$

By combining the previous results and using (5.2) and (5.6), we conclude that

$$\begin{aligned}\mathcal{J}'(\mathbf{u};\mathbf{e}) &= (\mathbf{e}, L^*\mathbf{u}^*)_{0,Q} = (\mathbf{u}, L^*\mathbf{u}^*)_{0,Q} - (\mathbf{u}_h, L^*\mathbf{u}^*)_{0,Q} \\ &= (L\mathbf{u}, \mathbf{u}^*)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^*) = (\mathbf{f}, \mathbf{u}^*)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^*) \\ &= (\mathbf{f}, \mathbf{u}^* - \mathbf{w}_h)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) \\ &= \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h).\end{aligned}$$

Note that Galerkin orthogonality (4.19) was used in the last step. □

On the other hand we can achieve an even better error representation if we take the dual residual ρ^* into account. The result is given in the following lemma.

Lemma 5.3 (cf. [BR03, Prop. 6.2])**.** *The error representation*

$$\triangle\mathcal{J} = \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{v}_h) + \mathcal{T}_3$$

holds true for all $\mathbf{v}_h \in V_h$, $\mathbf{w}_h \in W_h$, *with third-order remainder term*

$$\mathcal{T}_3 = \frac{1}{2}\int_0^1 s(s-1)\mathcal{J}'''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e}, \mathbf{e})\,\mathrm{d}s$$

with respect to $\mathcal{J}$.

Proof. We take the result form Lemma 5.2 and write

$$\begin{aligned}\triangle\mathcal{J} &= \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) - \int_0^1 s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s \\ &= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\left\{\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) - \int_0^1 2s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s\right\} \\ &= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\left\{(\mathbf{f}, \mathbf{u}^*)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^*) - \int_0^1 2s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s\right\}.\end{aligned}$$

Now we take a closer look at the integral term. Integration by parts leads to

$$\begin{aligned}&\int_0^1 2s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s = \\ &\qquad \mathcal{J}''(\mathbf{u}_h; \mathbf{e}, \mathbf{e}) - \int_0^1 (s^2 - 1)\mathcal{J}'''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e}, \mathbf{e})\,\mathrm{d}s.\end{aligned}$$

From Taylor's theorem we achieve again that

$$\mathcal{J}'(\mathbf{u};\mathbf{e}) = \mathcal{J}'(\mathbf{u}_h;\mathbf{e}) + \mathcal{J}''(\mathbf{u}_h;\mathbf{e},\mathbf{e}) + \int_0^1 (1-s)\mathcal{J}'''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e}, \mathbf{e})\,\mathrm{d}s.$$

Combining both equations results in

$$\begin{aligned}&\int_0^1 2s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s = \\ &\qquad \mathcal{J}'(\mathbf{u};\mathbf{e}) - \mathcal{J}'(\mathbf{u}_h;\mathbf{e}) - \int_0^1 s(s-1)\mathcal{J}'''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e}, \mathbf{e})\,\mathrm{d}s.\end{aligned}$$

Hence, by applying (5.5), (5.9) and proceeding as in the proof of Lemma 5.2, we obtain the error representation

$$\begin{aligned}
\triangle\mathcal{J} &= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{(\mathbf{f}, \mathbf{u}^*)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^*) - \mathcal{J}'(\mathbf{u}; \mathbf{e}) + \mathcal{J}'(\mathbf{u}_h; \mathbf{e})\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{(\mathbf{f}, \mathbf{u}^*)_{0,Q} - (L_h\mathbf{u}_h, \mathbf{u}^*)_{0,Q} - (\mathbf{e}, L^*\mathbf{u}^*)_{0,Q} \\
&\qquad\qquad + \mathcal{J}'(\mathbf{u}_h; \mathbf{u}) - \mathcal{J}'(\mathbf{u}_h; \mathbf{u}_h)\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{(L\mathbf{u}, \mathbf{u}^*)_{0,Q} - (\mathbf{u}, L^*\mathbf{u}^*)_{0,Q} \\
&\qquad\qquad + \mathcal{J}'(\mathbf{u}_h; \mathbf{u}) - (L_h\mathbf{u}_h, \mathbf{u}_h^*)_{0,Q}\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{\mathcal{J}'(\mathbf{u}_h; \mathbf{u}) - (L_h\mathbf{u}_h; \mathbf{u}_h^*)_{0,Q}\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{\mathcal{J}'(\mathbf{u}_h; \mathbf{u}) - b_h(\mathbf{u}_h; \mathbf{u}_h^*)\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\Big\{\mathcal{J}'(\mathbf{u}_h; \mathbf{u}) - b_h(\mathbf{u}; \mathbf{u}_h^*)\Big\} + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u}) + \mathcal{T}_3 \\
&= \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{v}_h) + \mathcal{T}_3,
\end{aligned}$$

where Galerkin orthogonality (4.19) was used again. □

Combining Lemma 5.2 and Lemma 5.3 leads to the following relationship between the primal and the dual residual.

Lemma 5.4 (cf. [BR03, Prop. 6.6])**.** *The relationship*

$$\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{v}_h) = \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \mathcal{T}_2'$$

holds true for all $\mathbf{v}_h \in V_h, \mathbf{w}_h \in W_h$*, with second-order term*

$$\mathcal{T}_2' = -\int_0^1 \mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s.$$

Proof. By using Lemma 5.2 in Lemma 5.3 and Taylor's theorem we conclude that

$$\begin{aligned}
\mathcal{T}_2' = 2(\mathcal{T}_2 - \mathcal{T}_3) &= \int_0^1 \Big(- 2s\mathcal{J}''(\mathbf{u}_h; \mathbf{e}, \mathbf{e}) - s(s-1)\mathcal{J}'''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e}, \mathbf{e})\Big)\,\mathrm{d}s \\
&= \mathcal{J}'(\mathbf{u}_h, \mathbf{e}) - \mathcal{J}'(\mathbf{u}, \mathbf{e}) = -\int_0^1 \mathcal{J}''(\mathbf{u}_h; \mathbf{e}, \mathbf{e})\,\mathrm{d}s.
\end{aligned}$$

□

This approach allows a large variety of different error functionals. In the end of this section, we focus on linear and quadratic error functionals in particular:

5.2.1 Linear Error Functionals

In the most simple case, the error functional $\mathcal{J}\colon \mathrm{L}_2(Q)^J \to \mathbb{R}$ is linear. Hence we can write it as

$$\mathcal{J}(\mathbf{u}) = (\mathbf{j}, \mathbf{u})_{0,S} \quad \text{and} \quad \mathcal{J}'(\mathbf{u}; \mathbf{v}) = (\mathbf{j}, \mathbf{v})_{0,S}, \quad \text{for all } \mathbf{w} \in \mathrm{L}_2(Q)^J,$$

where $\mathbf{j}\colon S \to \mathbb{R}^J$ is a density function, e.g., a space-time measure or approximations to Dirac functionals and $S \subseteq Q$ the so called *region of interest*. Note that if one wants to measure a point-value (derivative) error at some point $(\mathbf{x}, t) \in Q$, regularization is needed, e.g.,

$$\mathcal{J}(\mathbf{u}) = \frac{1}{|B_\varepsilon(\mathbf{x}, t)|}(\boldsymbol{\chi}_{B_\varepsilon(\mathbf{x},t)}, \mathbf{u})_{0,Q} \approx \mathbf{u}(\mathbf{x}, t),$$

where $\boldsymbol{\chi}_{B_\varepsilon(\mathbf{x},t)}$ is a characteristic function on a ball $B_\varepsilon(\mathbf{x}, t) \subset Q$ around $(\mathbf{x}, t)$ with radius $\varepsilon > 0$ sufficiently small, see [BR03, Ex. 3.3 or Ex. 3.4].

In case of linear error functionals $\mathcal{J}''(\mathbf{u}; \mathbf{w}, \mathbf{v}) = 0$ holds true. Thus both error representations in Lemma 5.2 and Lemma 5.3 are exact and the primal and dual residuals coincide (cf. Lemma 5.4). Since $\mathcal{J}'(\mathbf{u}; \mathbf{v})$ is independent of $\mathbf{u}$, the dual solution becomes independent of the primal one. We conclude that

$$\begin{aligned}
\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) &= (\mathbf{f}, \mathbf{u}^* - \mathbf{w}_h)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) = b_h(\mathbf{e}, \mathbf{u}^* - \mathbf{w}_h) \\
&= b_h(\mathbf{e}, \mathbf{u}^*) = (\mathbf{j}, \mathbf{e})_{0,S}, \\
\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{v}_h) &= (\mathbf{j}, \mathbf{u} - \mathbf{v}_h)_{0,S} - b_h(\mathbf{u} - \mathbf{v}_h, \mathbf{u}_h^*) = b_h(\mathbf{u}, \mathbf{u}^*) - b_h(\mathbf{u}, \mathbf{u}_h^*) \\
&= b_h(\mathbf{u}, \mathbf{e}^*) = (\mathbf{f}, \mathbf{e}^*)_{0,Q},
\end{aligned}$$

and hence $\triangle\mathcal{J} = (\mathbf{j}, \mathbf{e})_{0,S} = (\mathbf{f}, \mathbf{e}^*)_{0,Q}$.

5.2.2 Quadratic Error Functionals

Quadratic error functionals are quite common since they are used to model the energy of the solution in a certain region of interest, see [BR03, Ex. 3.2]. For example one can take the special energy error functional

$$\mathcal{J}(\mathbf{u}) = \frac{1}{2}(M\mathbf{u}, \mathbf{u})_{0,S}, \tag{5.10}$$

with

$$\mathcal{J}'(\mathbf{u}; \mathbf{v}) = (M\mathbf{u}, \mathbf{v})_{0,S} \quad \text{and} \quad \mathcal{J}''(\mathbf{u}; \mathbf{w}, \mathbf{v}) = (M\mathbf{v}, \mathbf{w})_{0,S},$$

for all $\mathbf{v}, \mathbf{w} \in \mathrm{L}_2(Q)^J$. For quadratic error functionals the error representation in Lemma 5.3 is exact, i.e., $\mathcal{T}_3 = 0$ and we conclude that $\triangle\mathcal{J}$ can be represented as

$$\begin{aligned}
&\triangle\mathcal{J} = \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\mathcal{J}''(\mathbf{u}; \mathbf{e}, \mathbf{e}) = \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \mathcal{J}(\mathbf{e}) \\
\text{or} \qquad &\triangle\mathcal{J} = \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{w}_h) + \frac{1}{2}\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{v}_h).
\end{aligned}$$

5.3 Error Estimation

In this section we use the previous results to derive an approximation $\triangle\mathcal{J}_h$ to $\triangle\mathcal{J}$ and computable quantities $\eta_R \in \mathbb{R}$ for every cell $R \in \mathcal{R}$ such that

$$|\triangle\mathcal{J}| \approx |\triangle\mathcal{J}_h| \leq \eta_{\mathcal{R}} = \sum_{R\in\mathcal{R}} \eta_R.$$

The quantities η_R are denoted as (local) *error estimates* or *error indicators*. The development of a suitable and reliable error estimator for the presented error representations is a crucial point. To do so, we have to evaluate the residuals ρ or ρ^* defined in (5.7) or (5.8) and get

$$\triangle\mathcal{J} = \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{u}_h^*) + \mathcal{T}_2, \tag{5.11}$$

$$\triangle\mathcal{J} = \frac{1}{2}\rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{u}_h^*) + \frac{1}{2}\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{u}_h) + \mathcal{T}_3, \tag{5.12}$$

where we used the discrete solutions $\mathbf{u}_h$ and $\mathbf{u}_h^*$ as test functions. In both cases we have to compute a discrete dual solution $\mathbf{u}_h^*$, which has the same effort as computing the discrete primal solution $\mathbf{u}_h$. Since $\mathcal{T}_3$ is third-order, it seems that (5.12) is more accurate than (5.11), where $\mathcal{T}_2$ is a second-order remainder term. But in most applications it is already sufficient to compute (5.11) and use the relationship

$$\mathcal{T}_2' = \rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \mathbf{u} - \mathbf{u}_h) - \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{u}_h^*)$$

from Lemma 5.4 to validate that the second-order remainder $\mathcal{T}_2 = \frac{1}{2}\mathcal{T}_2' + \mathcal{T}_3$ is small. This can be done a posteriori by verifying that

$$|\mathcal{T}_2| \approx |\mathcal{T}_2'| \approx |\rho^*(\mathbf{u}_h; \mathbf{u}_h^*, \widetilde{\mathbf{u}} - \mathbf{u}_h) - \rho(\mathbf{u}_h, \widetilde{\mathbf{u}}^* - \mathbf{u}_h^*)| \leq \texttt{tol}$$

for a given tolerance `tol` and higher-order approximations $\widetilde{\mathbf{u}} \approx \mathbf{u}$ and $\widetilde{\mathbf{u}}^* \approx \mathbf{u}^*$. This strategy is proposed for example in [BR03, Rem. 6.7].

Since $\mathcal{T}_2$ can be approximated and the fact that (5.11) is not depending on $\mathbf{u}$, we will focus on this second-order accurate error representation. Moreover, we only use the energy error functional (5.10) in our numerical experiments, where

$$\mathcal{T}_2 = -\int_0^1 s\mathcal{J}''(\mathbf{u}_h + s\mathbf{e}; \mathbf{e}, \mathbf{e})\,\mathrm{d}s = -\frac{1}{2}(M\mathbf{e}, \mathbf{e})_{0,S}.$$

We now develop a (local) error estimation by using (5.11). Hence it holds that

$$\begin{aligned}
\triangle\mathcal{J} &= \rho(\mathbf{u}_h, \mathbf{u}^* - \mathbf{u}_h^*) + \mathcal{T}_2 = (\mathbf{f}, \mathbf{u}^* - \mathbf{u}_h^*)_{0,Q} - b_h(\mathbf{u}_h, \mathbf{u}^* - \mathbf{u}_h^*) + \mathcal{T}_2 \\
&= (\mathbf{f}, \mathbf{u}^* - \mathbf{u}_h^*)_{0,Q} - \sum_{R=K\times I\in\mathcal{R}} \Big((M\partial_t \mathbf{u}_h + \nabla\cdot\mathbf{F}(\mathbf{u}_h), \mathbf{u}^* - \mathbf{u}_h^*)_{0,R} \\
&\qquad + (\mathbf{n}_K \cdot (\mathbf{F}^{\mathrm{num}}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_h)), \mathbf{u}^* - \mathbf{u}_h^*)_{0,\partial K\times I} \Big) + \mathcal{T}_2 \\
&= \sum_{R\in\mathcal{R}} \Big((\mathbf{f} - M\partial_t \mathbf{u}_h - \nabla\cdot\mathbf{F}(\mathbf{u}_h), \mathbf{u}^* - \mathbf{u}_h^*)_{0,R} \\
&\qquad - (\mathbf{n}_K \cdot (\mathbf{F}^{\mathrm{num}}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_h)), \mathbf{u}^* - \mathbf{u}_h^*)_{0,\partial K\times I} \Big) + \mathcal{T}_2.
\end{aligned}$$

Using triangle and Cauchy-Schwarz inequality leads to

$$\begin{aligned}|\triangle\mathcal{J}| &\leq \sum_{R\in\mathcal{R}} \Big(\|\mathbf{f} - M\partial_t \mathbf{u}_h - \nabla\cdot\mathbf{F}(\mathbf{u}_h)\|_{0,R} \|\mathbf{u}^* - \mathbf{u}_h^*\|_{0,R} \\ &\qquad + \|\mathbf{n}_K \cdot (\mathbf{F}^{\text{num}}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_h))\|_{0,\partial K\times I} \|\mathbf{u}^* - \mathbf{u}_h^*\|_{0,\partial K\times I} \Big) + \mathcal{T}_2 \qquad (5.13) \\ &= \sum_{R\in\mathcal{R}} \boldsymbol{\rho}_R(\mathbf{u}_h)\cdot\boldsymbol{\rho}_R^*(\mathbf{u}_h^*) + \mathcal{T}_2 \leq \sum_{R\in\mathcal{R}} \|\boldsymbol{\rho}_R(\mathbf{u}_h)\|_2 \|\boldsymbol{\rho}_R^*(\mathbf{u}^*,\mathbf{u}_h^*)\|_2 + \mathcal{T}_2,\end{aligned}$$

where $\|\,.\,\|_2$ is the Euclidean vector norm and $\boldsymbol{\rho}_R, \boldsymbol{\rho}_R^* \in \mathbb{R}^2$ are assembled as

$$\begin{aligned}\boldsymbol{\rho}_R(\mathbf{u}_h) &= \begin{pmatrix} \|\mathbf{f} - M\partial_t\mathbf{u}_h - \nabla\cdot\mathbf{F}(\mathbf{u}_h)\|_{0,R} \\ \|\mathbf{n}_K\cdot(\mathbf{F}^{\text{num}}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_h))\|_{0,\partial K\times I} \end{pmatrix} \\ \boldsymbol{\rho}_R^*(\mathbf{u}^*,\mathbf{u}_h^*) &= \begin{pmatrix} \|\mathbf{u}^* - \mathbf{u}_h^*\|_{0,R} \\ \|\mathbf{u}^* - \mathbf{u}_h^*\|_{0,\partial K\times I} \end{pmatrix} = \begin{pmatrix} \|\mathbf{e}^*\|_{0,R} \\ \|\mathbf{e}^*\|_{0,\partial K\times I} \end{pmatrix}.\end{aligned}$$

If the second-order term $\mathcal{T}_2$ is neglected, we achieved an error estimate, where the local residual error $\|\boldsymbol{\rho}_R(\mathbf{u}_h)\|_2$ is weighted with the error of the dual solution $\|\boldsymbol{\rho}_R^*(\mathbf{u}^*,\mathbf{u}_h^*)\|_2$. However, in applications the dependency of the weights $\boldsymbol{\rho}_R^*$ on the error of the dual solution $\mathbf{e}^*$ is a crucial point. In general this value cannot be computed and a good approximation $\mathbf{e}^* \approx \mathbf{e}_{h,\mathrm{r}}^* = \mathbf{u}_{\mathrm{r}}^* - \mathbf{u}_h^*$ is needed to define a local error indicator

$$\eta_R(\mathbf{u}_h,\mathbf{u}_h^*) = \|\boldsymbol{\rho}_R(\mathbf{u}_h)\|_2 \|\boldsymbol{\rho}_R^*(\mathbf{u}_{\mathrm{r}}^*,\mathbf{u}_h^*)\|_2, \qquad \text{for all } R\in\mathcal{R}. \qquad (5.14)$$

To investigate the quality of approximations $\mathbf{u}_{\mathrm{r}}^*$ with respect to $\triangle\mathcal{J}$ one can proceed as follows. We again consider the error representation (5.11) from Lemma 5.2 and apply an approach which is illustrated in [BR03, Sec. 5.2 ii)]. This yields

$$\begin{aligned}\triangle\mathcal{J} &= \rho(\mathbf{u}_h;\mathbf{u}^* - \mathbf{u}_h^*) + \mathcal{T}_2 \\ &= \rho(\mathbf{u}_h;\mathbf{u}^* - \mathbf{u}_{\mathrm{r}}^*) + \rho(\mathbf{u}_h;\mathbf{u}_{\mathrm{r}}^* - \mathbf{u}_h^*) + \mathcal{T}_2 \qquad (5.15) \\ &= \rho(\mathbf{u}_h;\mathbf{u}^* - \mathbf{u}_{\mathrm{r}}^*) + \triangle\mathcal{J}_h + \mathcal{T}_2,\end{aligned}$$

where $\triangle\mathcal{J}_h = \rho(\mathbf{u}_h;\mathbf{u}_{\mathrm{r}}^* - \mathbf{u}_h^*)$ is the computable approximated error. To validate that $\mathbf{u}_{H,\mathrm{r}}^*$ is a suitable approximation to $\mathbf{u}^*$ one has to ensure that the first term is sufficiently small, i.e.,

$$|\rho(\mathbf{u}_h;\mathbf{u}^* - \mathbf{u}_{\mathrm{r}}^*)| \approx \mathcal{T}_2 \ll |\triangle\mathcal{J}_h|.$$

However, this is not a trivial task. Even in case of an elliptic Poisson problem with linear error functional stated on an "optimized" mesh, one has to assume restrictive mesh conditions and smoothness properties for $\mathbf{u}$ and $\mathbf{u}^*$ (cf. [BR03, Sec. 5.2 ii)]).

In the following we discuss two commonly used strategies for achieving a suitable approximation $\mathbf{u}_{\mathrm{r}}^* \approx \mathbf{u}^*$, see, e.g., [BR01], [BR03, Sec. 4.1], [NVV09] or [BGR10].

5.3.1 Higher-Order Approximation

In this approach one computes an approximated dual solution $\mathbf{u}_h^* \in W_h^{p+1,q+1}$ in an ansatz space $W_h^{p+1,q+1}$ with uniformly increased polynomial degrees $p, q \geq 0$. Hence, the dual error can be approximated as $\mathbf{e}_{h,\mathrm{r}}^* = \mathbf{u}_h^* - I_h \mathbf{u}_h^*$, where $I_h \colon W_h^{p+1,q+1} \to W_h^{p,q} \equiv W_h$ is the nodal interpolation operator which is defined locally in (4.21) on every $R \in \mathcal{R}$. Numerical experiments for an elliptic test case indicate this approach leads to very good approximations (cf. [BR03, Tab. 4.1]). However, the computation of $\mathbf{u}_h^*$ becomes very costly and even surpasses the computational effort which is needed to compute the discrete primal solution $\mathbf{u}_h$. Hence this approach is not suitable for most real applications, but it motivates the a strategy presented in the following.

5.3.2 Higher-Order Interpolation on Patches

Since the previous approach leads to good results, we introduce an approximation based on higher-order interpolation techniques. Therefore, we use a discrete dual solution which is computed on the same discretization as $\mathbf{u}_h$. Hence the computation of both solutions $\mathbf{u}_h$ and $\mathbf{u}_h^*$ requires the same computational effort. Moreover the system matrix of the primal problem can be reused as we will see in the next section.

Due to their flexibility and simplicity in terms of implementation, so called (patch-wise) *recovery* estimators are commonly in case of residual based a posteriori error estimation, see, e.g., [AO00, Ch. 4] or [Ver13, Sec. 1.9]. There, one uses a recovery operator to approximate the gradient of the solution from a computed discrete solution. In the following we use a similar technique to approximate $\mathbf{u}^*$ from $\mathbf{u}_h^*$. To achieve a higher-order approximation we first coarse the space-time mesh $\mathcal{R}_h$ once in space and time. This results in a mesh $\mathcal{R}_H$, where every cell covers $2^D \times 2$ space-time cells of $\mathcal{R}_h$, i.e., for every cell $R_\mathrm{c} \in \mathcal{R}_H$ there exists a set of cells $\{R_1, \dots, R_{2^{D+1}}\} \subset \mathcal{R}_h$ such that

$$\overline{R}_\mathrm{c} = \bigcup_{m=1}^{2^{D+1}} \overline{R}_m.$$

In this context we denote the coarse cells $R_\mathrm{c} \in \mathcal{R}_H$ as space-time *patches*. If multigrid techniques are used to solve the resulting linear finite element systems, the required data structures are already available and can be reused, see Section 6.2.3. On the coarse mesh we can choose a discretization W_H with higher polynomial degrees, e.g., $(p_\mathrm{c}, q_\mathrm{c}) = (p+2, q+2)$ (cf. [BR03, Rem. 4.4]), where $p = \max_{m=1,\dots,2^D} p_m$ and $q = \max_{m=1,\dots,2^D} q_m$ are the highest polynomial degrees on the cells $R_1, \dots, R_{2^{D+1}}$. Analogously to Section 4.6 we define a nodal interpolation $I_{h,\mathrm{r}}^{(2)} \colon W_h \to W_H$ locally

on every patch R_c as

$$I^{(2)}_{R_\mathrm{c},\mathrm{r}}\colon \operatorname{span}\Big\{ \bigcup_{m=1}^{2^{D+1}} W_{h,R_m} \Big\} \to W_{h,R_\mathrm{c}} \subset W_H, \qquad I^{(2)}_{R_\mathrm{c},\mathrm{r}} \mathbf{w}_h = \sum_{i=1}^{\dim W_{h,R_\mathrm{c}}} \{\mathbf{w}_h(\mathbf{x}^\mathrm{c}_i, t^\mathrm{c}_i)\}_i \boldsymbol{\psi}_{i,R_\mathrm{c}}$$

which maps the lower order discrete dual solution solution $\mathbf{u}^*_h \in W_h$ on the fine mesh $\mathcal{R}_h$ to a higher-order approximation $\mathbf{u}^*_{H,\mathrm{r}} = I^{(2)}_{h,\mathrm{r}} \mathbf{u}^*_h \in W_H$ on the coarse mesh $\mathcal{R}_H$. Here $\{.\}_i$ denotes the averages on the coarse nodal points $(\mathbf{x}^\mathrm{c}_i, t^\mathrm{c}_i)$, for $i = 1, \dots, \dim W_{h,R^\mathrm{c}}$. An example of the patch-wise polynomial interpolation in time is given in Figures 5.1 and 5.2, whereas an illustration for $D = 1$ of a higher-order patch-wise interpolation in space-time is shown in Figure 5.3.

To investigate the quality of the approximation $\mathbf{u}^*_{H,\mathrm{r}} \approx \mathbf{u}^*$, we transfer and verify the conditions postulated in [AO00, Sec. 4.2] for suitable gradient recovery operators to our case:

Lemma 5.5. *Let $\mathbf{u}^*_h \in W_h$ be a solution of the discrete dual problem* (5.5)*. The higher-order patch-wise interpolation operator $I^{(2)}_{h,\mathrm{r}}$ provides suitable approximations $\mathbf{u}^*_{H,\mathrm{r}} = I^{(2)}_{h,\mathrm{r}} \mathbf{u}^*_h \in W_H$ to the exact solution $\mathbf{u}^* \in V^*$ in the sense of [AO00, Sec. 4.2]. In particular $I^{(2)}_{h,\mathrm{r}}$ fulfills the following conditions:*

a) consistency: $I^{(2)}_{R_\mathrm{c},\mathrm{r}}(I_h \mathbf{w}_H) = \mathbf{w}_H$ *for all polynomials* $\mathbf{w}_H \in \mathbb{P}_{p,q}(R_\mathrm{c})^J$,

b) localization: $\mathbf{u}^*_{H,\mathrm{r}}|_{R_\mathrm{c}}$ *only depends on* $\mathbf{u}^*_h|_{R_\mathrm{c}}$ *for every patch* $R_\mathrm{c} \in \mathcal{R}_H$,

c) boundedness: $\|I^{(2)}_{h,\mathrm{r}} \mathbf{u}^*_h\|_{\mathrm{L}_\infty(R_\mathrm{c})} \le c \|\mathbf{u}^*_h\|_{\mathrm{L}_\infty(R_\mathrm{c})}$ *for all* $R_\mathrm{c} \in \mathcal{R}_H$, $\mathbf{u}^*_h \in W_h$ *and* $c > 0$ *independent of* h.

Here $I_h\colon W_H \to W_h$ is again the nodal space-time interpolation operator which is defined locally in (4.21).

Proof. By definition $I^{(2)}_{h,\mathrm{r}}$ and I_h interpolate polynomials of degree (p, q) exactly. Thus $I_h \mathbf{w}_h \in \mathbb{P}_{p,q}(R_\mathrm{c})^J$ is again a polynomial and it yields

$$I^{(2)}_{R_\mathrm{c},\mathrm{r}}(I_h \mathbf{w}_H) = I^{(2)}_{R_\mathrm{c},\mathrm{r}}(\mathbf{w}_H) = \mathbf{w}_H.$$

Condition b) is fulfilled by the patch-wise definition of $I^{(2)}_{h,\mathrm{r}}$. Note that in principle larger patches are also possible. To prove condition c), we use the definition of $I^{(2)}_{h,\mathrm{r}}$ to conclude that

$$\begin{aligned} \|I^{(2)}_{h,\mathrm{r}} \mathbf{u}^*_h\|_{\mathrm{L}_\infty(R_\mathrm{c})} &= \Big\| \sum_{i=1}^{\dim W_{h,R_\mathrm{c}}} \{\mathbf{u}^*_h(\mathbf{x}^\mathrm{c}_i, t^\mathrm{c}_i)\}_i \boldsymbol{\psi}_{i,R_\mathrm{c}} \Big\|_{\mathrm{L}_\infty(R_\mathrm{c})} \\ &\le \sum_{i=1}^{\dim W_{h,R_\mathrm{c}}} \|\boldsymbol{\psi}_{i,R_\mathrm{c}}\|_{\mathrm{L}_\infty(R_\mathrm{c})} \|\mathbf{u}^*_h\|_{\mathrm{L}_\infty(R_\mathrm{c})} \le c \|\mathbf{u}^*_h\|_{\mathrm{L}_\infty(R_\mathrm{c})}. \end{aligned}$$

Here c depends only on the polynomial degrees, since the range of the shape functions $\boldsymbol{\psi}_{i,R_\mathrm{c}} = \widehat{\boldsymbol{\psi}}_i \circ \boldsymbol{\phi}^{-1}_{R_\mathrm{c}}$ is independent of R_c (cf. Remark 4.1). □

However, the quality of recovery techniques strongly depends on the underlying discretization and can only be justified for sufficiently small mesh sizes, see, e.g., [Car05]. In adaptive algorithms this contains the danger that the presented strategy will lead to wrong results if the initial discretization is chosen not accurate enough (cf. [AO00, Sec. 4.7] or [NVV09]). This issue becomes relevant in the following section, where suitable stopping criteria are discussed.

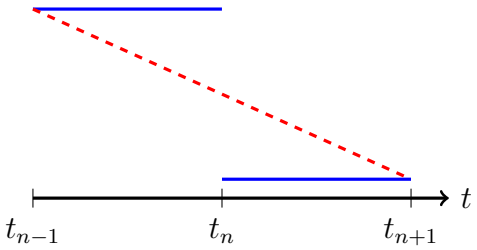

Figure 5.1: Example of higher-order patch-wise recovery (dashed line) for given data (solid line) and $q = 0$.

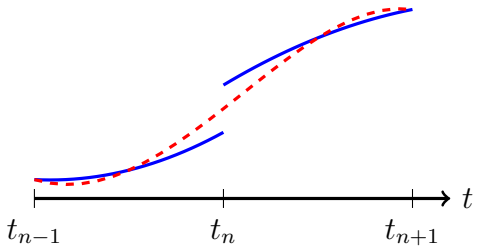

Figure 5.2: Example of higher-order patch-wise recovery (dashed line) for given data (solid line) and $q = 2$.

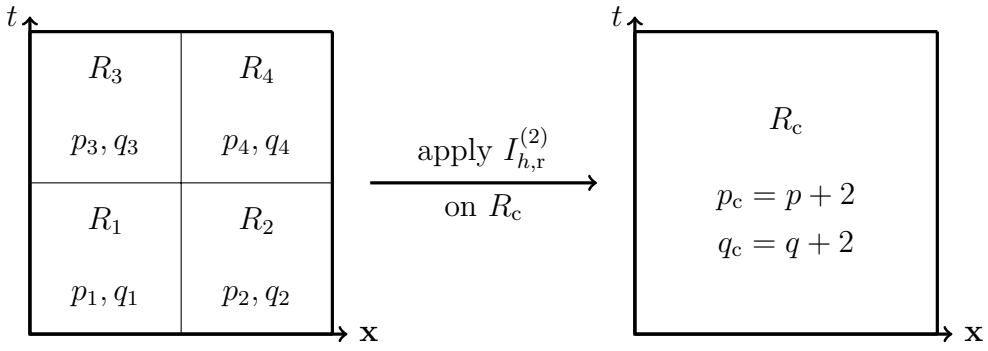

Figure 5.3: Illustration of the higher-order patch-wise recovery on a 2×2 space-time patch for $D = 1$.

5.4 Adaptive Algorithm and Implementation

By now, we have all necessary ingredients at hand to implement the adaptive algorithm. We start with a fixed mesh $\mathcal{R}$ and a discretization V_h, W_h which uses lowest order polynomial degrees in space and time, i.e., $(p_R, q_R) = (p, q) = (1, 1)$, for all space-time cells $R \in \mathcal{R}$. Afterwards we solve the resulting linear system $\underline{L}\underline{u} = \underline{f}$ to achieve a discrete solution $\mathbf{u}_h \in V_h$ (cf. Section 4.6). To do this efficiently, a multilevel strategy is discussed in the next chapter. Furthermore, we have to compute an approximation $\mathbf{u}_h^* \in W_h$ to the dual solution. Due to the properties of the discrete adjoint problem, introduced in Definition 5.3, we use the transpose finite

element matrix $\underline{L}^\top$ to archive an approximation of L^*. This leads to the dual linear problem $\underline{L}^*\underline{u}^* = \underline{f}^*$, where $\underline{L}^*$ is a block-matrix

$$\underline{L}^* = \underline{L}^\top = \begin{bmatrix} \underline{D}^{1\top} & \underline{C}^{1\top} & & & \\ & \underline{D}^{2\top} & \underline{C}^{2\top} & & \\ & & \ddots & \ddots & \\ & & & \underline{D}^{N-1\top} & \underline{C}^{N-1\top} \\ & & & & \underline{D}^{N\top} \end{bmatrix}.$$

The local matrices $\underline{D}^n$ and $\underline{C}^n$, $n = 1, \ldots, N$, correspond to the local matrices introduced in Section 4.6. For a linear error functional with density $\mathbf{j}$ (cf. Section 5.2.1), the right-hand side is computed by $\underline{f}^* = (\underline{f}^{*,1}, \ldots, \underline{f}^{*,N})$ with $\underline{f}^{*,n}_{j,R} = (\mathbf{j}, \boldsymbol{\phi}^n_{R,j})_{R\cap S}$, where $\left\{\boldsymbol{\phi}^n_{R,j}\right\}_{j=1,\ldots,\dim V_{h,R}}$ is a nodal basis of $V_{h,R}$. Note that this is in contrast to the primal case, where we test with a nodal basis of $W_{h,R}$. For the nonlinear error functionals, e.g., the quadratic functional given in Section 5.2.2, the right-hand side depends on $\mathbf{u}_h$ and hence the components of $\underline{f}^*$ are given as $\underline{f}^{*,n}_{j,R} = (\mathbf{u}_h, \boldsymbol{\phi}^n_{R,j})_{R\cap S}$. Thus $\mathbf{u}^*_h \in W_h$ solves the discrete dual problem (5.5) stated in Definition 5.3.

The resulting dual system has an upper triangular structure and can be again solved iteratively by a block-Gauss–Seidel method as stated before in (4.24). However, this time the method is performed backwards from N to 1, i.e.,

$$\underline{D}^{N\top}\underline{u}^{*,N} = \underline{f}^{*,N}, \quad \underline{D}^{N-1\top}\underline{u}^{*,N-1} = \underline{f}^{*,N-1} - \underline{C}^{N-1\top}\underline{u}^{*,N},$$
$$\ldots, \quad \underline{D}^{2\top}\underline{u}^{*,2} = \underline{f}^{*,2} - \underline{C}^{2\top}\underline{u}^{*,3}, \quad \underline{D}^{1\top}\underline{u}^{*,1} = \underline{f}^{*,1} - \underline{C}^{1\top}\underline{u}^{*,2}.$$

In a second step we perform the higher-order interpolation $\mathbf{u}^*_{H,\mathrm{r}} = I^{(2)}_{h,\mathrm{r}}\mathbf{u}^*_h$ as introduced in Section 5.3.2. By using the approximation $\mathbf{u}^* \approx \mathbf{u}^*_{H,\mathrm{r}}$, we receive the approximated error $\mathbf{e}^*_{h,\mathrm{r}}$. Hence equation (5.14) can now be evaluated on every cell $R \in \mathcal{R}$ to obtain the estimated local error η_R. One achieves a new discretization by increasing the polynomial degree on cells with highest estimated error η_R. For this purpose we apply a *maximum marking* strategy, see, e.g., [Dör96], and introduce a parameter $\vartheta \in [0, 1]$. The polynomial degree in space and time on a cell $R \in \mathcal{R}$ is increased if the condition

$$\eta_R > \vartheta \max_{\widetilde{R}\in\mathcal{R}} \eta_{\widetilde{R}}$$

is fulfilled. For $\vartheta = 0$ one receives uniform refinement, whereas $\vartheta = 1$ results in no refinement at all. Choosing the parameter ϑ is not a trivial task since it highly depends on the given problem. If it is chosen too big, the solution might not be accurate enough, whereas the effort becomes to large if ϑ is chosen too small. To reduce these effects, we additionally increase the polynomial degrees on all neighboring cells R_f of a refined cell $R \in \mathcal{R}$, too. This procedure increases the robustness of the adaptive strategy with respect to the parameter ϑ. Finally, we

use the new discretization $\widetilde{V}_h$, $\widetilde{W}_h$ to compute a new discrete solution $\mathbf{u}_h \in \widetilde{V}_h$. This procedure can be repeated several times until the total estimated error is small enough or a maximal polynomial degree is reached. Due to the construction of $\eta_{\mathcal{R}} = \sum_{R\in\mathcal{R}} \eta_R$, cancellations between elements are neglected and the error $|\triangle\mathcal{J}|$ might be overestimated too much. Thus, computing $\triangle\mathcal{J}_h$ and applying

$$|\triangle\mathcal{J}_h| \leq \texttt{tol}$$

as a stopping criterion for a given tolerance `tol` is more appropriate (cf. Section 5.3.2). At this point it is worth to emphasize that, due to (5.15), $\triangle\mathcal{J}_h \approx \triangle\mathcal{J}$ holds only true if the neglected errors are small. In [NVV09] a safeguarded version of DWR methods is presented by considering higher-order terms to achieve a reliable error estimation on coarse meshes. This procedure prevents the algorithm from underestimating the true error and terminating too early. Moreover, the results asymptotically coincide with the original DWR method. The complete p-adaptive strategy is summarized in Algorithm 5.1.

Algorithm 5.1 p-adaptive strategy

$\mathcal{R} = \textsc{Generate\_Mesh}()$

Generate initial polynomial distribution $\underline{p}_{\mathcal{R}}, \underline{q}_{\mathcal{R}}$*:*

for $R \in \mathcal{R}$ **do**

  $\underline{p}_{\mathcal{R}}(R) = 1$

  $\underline{q}_{\mathcal{R}}(R) = 1$

Solve primal problem with initial discretization:

$(\underline{L}, \underline{f}) = \textsc{Assemble}(\underline{p}_{\mathcal{R}}, \underline{q}_{\mathcal{R}})$

$\underline{u} = \textsc{Solve}(\underline{L}, \underline{f})$

while $|\triangle \mathcal{J}_h| >$ `tol` and $\max_{R \in \mathcal{R}} \underline{p}(R) < p_{\max}$ and $\max_{R \in \mathcal{R}} \underline{q}(R) < q_{\max}$ **do**

  $\underline{f}^* = \textsc{Assemble\_Dual\_Rhs}(\underline{p}_{\mathcal{R}}, \underline{q}_{\mathcal{R}})$

  $\underline{u}^* = \textsc{Solve}(\underline{L}^\top, \underline{f}^*)$

  $\underline{u}^*_{\mathrm{r}} = \textsc{Approximate\_Dual\_Solution}(\underline{u}^*)$

  $\underline{e}^*_{\mathrm{r}} = \underline{u}^* - \underline{u}^*_{\mathrm{r}}$

  for $R \in \mathcal{R}$ **do**

    $\eta_R = \textsc{Estimate\_Error}(R, \underline{u}, \underline{e}^*_{\mathrm{r}})$

  $\eta_{\max} = \max_{R \in \mathcal{R}} \eta_R$

  for $R \in \mathcal{R}$ **do**

    if $\eta_R > \vartheta \eta_{\max}$ or $\eta_{R_f} > \vartheta \eta_{\max}$ for one $f \in \mathcal{F}_K$ **then**

      $\underline{p}(R) = \underline{p}(R) + 1$

      $\underline{q}(R) = \underline{q}(R) + 1$

  Redistribute mesh on processes for better load balance:

  $\textsc{Redistribute\_Mesh}(\mathcal{R}, p_{\mathcal{R}}, q_{\mathcal{R}})$

  Solve primal problem with new polynomial discretization:

  $(\underline{L}, \underline{f}) = \textsc{Assemble}(\underline{p}_{\mathcal{R}}, \underline{q}_{\mathcal{R}})$

  $\underline{u} = \textsc{Solve}(\underline{L}, \underline{f})$

6 Space-Time Multilevel Preconditioning

In this chapter we investigate how the linear system of equations (4.23) can be solved efficiently in a parallel framework for a large number of unknowns $N \in \mathbb{N}$. We focus on so called *multigrid methods* or *multigrid strategies* which were first stated in the 1960s in [Fed64] and [Bak66]. For elliptic and (space-time) parabolic problems, multigrid methods can lead to optimal effort $\mathcal{O}(N)$, see, e.g. [Hac03, Ch. 6, Ch. 7], [Hac85] and [Neu13]. Thus they are a suitable choice for large problems with many unknowns. However, there exists no standard procedure for applying multigrid methods efficiently to arbitrary problems as the following statement by Hackbusch indicates:

> *"The previous characterisation does not mean that there is a fixed multi-grid algorithm applying to all boundary value problems. There is rather a multi-grid technique fixing only the framework of the algorithm. The efficiency of the multi-grid algorithm depends on the adjustment of its components to the problem in question."* (from [Hac03, Sec. 1.1])

In general, (space-time) multigrid methods for hyperbolic problems are still an open field of research (cf. [DFW16]). A detailed introduction to multigrid methods and their application as preconditioners can be found for example in [Hac03], [Saa03, Ch. 13] or [Mei11, Ch. 4.2].

6.1 Introduction to Multigrid Methods

The idea of multigrid methods is to derive a divide and conquer strategy with optimal convergence rate $\mathcal{O}(1)$ or at least near to optimal convergence rate $\mathcal{O}(\log(N))$. Thereby the fact that a problem can be discretized on different grids (meshes) is exploited. Since these grids are often derived from a very coarse initial mesh by using different levels of refinement the term *multilevel methods* is common, too. On every mesh one can perform several steps of a well known iterative splitting method, e.g., Jacobi or Gauss–Seidel. These methods usually damp highly oscillating parts in the error quite well, whereas smooth parts are damped arbitrarily bad as the

mesh size goes to zero. This motivates a strategy where the spectrum of the error is divided into a highly oscillating and a smooth part. Now one can adjust the standard splitting method in a way that the highly oscillating parts of the error are damped independently of the mesh size. This is done by choosing a suitable damping parameter in the iterative method. Since this smooths the error function, the standard iterative splitting methods are denoted as *smoothers* in this context. To damp the error in the smooth part, it is transferred to a coarser mesh (smaller level). On this coarse mesh the discrete solution is computed exactly, e.g., by a direct method. The information gained by the solution on the coarse mesh is again transferred to the fine mesh to correct the smooth parts of the error.

This approach is denoted as *two-grid cycle*. Generally speaking, a multigrid method is a two-grid cycle, where the solver of the coarse grid problem is replaced recursively by one (*V-cycle*) or two (*W-cycle*) two-grid cycles until a very coarse level is reached, see [Hac03, Sec. 2.5]. On the coarsest level the discrete problems can be solved by standard methods with almost no computational costs compared to the problem on the initial discretization. Thus a multigrid method can be analyzed by investigating the underlying two-grid cycles. Since our problem is stated in space and time simultaneously, we distinguish between two-grid cycles in space and time. This means that the fine mesh is a uniform refinement of the coarse mesh in space or in time, respectively. In the following the general two-grid cycle is introduced. Afterwards we apply this method to a time- dependent test equation to determine an optimal damping factor for the Jacobi smoother in time. Finally, the two-grid cycle is applied to the full space-time problem to determine a suitable space-time multilevel strategy.

6.2 Two-Grid Cycle

To apply multigrid methods to a general linear space-time system

$$\underline{L}\underline{u} = \underline{b}, \tag{6.1}$$

for $\underline{L} \in \mathbb{R}^{N\times N}$ and $\underline{u}, \underline{b} \in \mathbb{R}^N$, a hierarchy in space and time is needed. Therefore, we introduce a coarse space-time mesh denoted as $\mathcal{R}_{0,0}$. Furthermore let $\mathcal{R}_{l,k}$ be the mesh obtained by $l = 1, \dots, l_{\max}$ uniform refinements in space and $k = 1, \dots, k_{\max}$ refinements in time. In a tensor product structure $\mathcal{R} = \mathcal{K} \times \mathcal{I}$ the refinements in space and time result either in refining the spatial mesh K or the time discretization $\mathcal{I}$. This yields the splitting

$$\mathcal{R}_{l,k} = \mathcal{K}_l \times \mathcal{I}_k.$$

Thus a linear system

$$\underline{L}_{l,k}\underline{u}_{l,k} = \underline{b}_{l,k}$$

can be computed on each level (l, k). On the finest level we have to solve (6.1) and hence $\underline{L} = \underline{L}_{l_{\max},k_{\max}}$ and $\underline{b} = \underline{b}_{l_{\max},k_{\max}}$. Since every mesh can be derived from the coarse mesh by performing uniform refinements in space and time, the set of all grids is called a *nested* mesh structure. A two-grid cycle only depends on two different levels. Hence this leads to two different settings, with grids $\mathcal{R}_{l,k-1}, \mathcal{R}_{l,k}$ or $\mathcal{R}_{l-1,k}, \mathcal{R}_{l,k}$, respectively. The first one corresponds to a two-grid cycle in time, whereas the second one corresponds to a two-grid cycle in space. In the following we give a short introduction to iterative methods and smoothers. Afterwards we discuss mesh transfer operations for homogeneous polynomial degrees and their extension to the p-adaptive case. Finally, we introduce a coarse grid correction and show how a complete two-grid method can be investigated.

6.2.1 Iterative Methods and Preconditioning

In this section we give a short repetition of iterative methods and the concept of preconditioning. A general introduction and further results can be found for example in [Saa03, Ch. 4] or [Mei11, Ch. 4].

Definition 6.1. Consider a mapping $\Phi\colon \mathbb{R}^N \to \mathbb{R}^N$. A method which solves the linear system (6.1) by computing approximations $\underline{u}^{(i)}$ to $\underline{u}$ for given a given $\underline{u}^{(0)}$ such that

$$\underline{u}^{(i+1)} = \Phi(\underline{u}^{(i)}), \qquad \text{for } i = 0, 1, \ldots, \tag{6.2}$$

is called *iterative method* and (6.2) is called *iteration scheme*. The method is called *linear* if there exists matrices $\underline{G}, \underline{N} \in \mathbb{R}^{N\times N}$ such that

$$\Phi(\underline{u}) = \underline{G}\underline{u} + \underline{N}\underline{b}.$$

The matrix $\underline{G}$ is called *iteration matrix*.

Definition 6.2. An iterative method is called *consistent* with respect to the matrix $\underline{L}$, if the solution $\underline{L}^{-1}\underline{b}$ is a fix point of the iteration scheme (6.2), i.e.,

$$\underline{L}^{-1}\underline{b} = \Phi(\underline{L}^{-1}\underline{b})$$

for every right-hand side $\underline{b} \in \mathbb{R}^N$. Furthermore it is called *convergent*, if for all $\underline{b}, \underline{u}^{(0)} \in \mathbb{R}^N$ there exists a limit

$$\underline{y} = \lim_{i\to\infty} \underline{u}^{(i)}$$

which is independent of the starting vector $\underline{u}^{(0)}$.

Theorem 6.1. *(cf. [Mei11, Thm. 4.4]) A linear iterative method is consistent if and only if*

$$\underline{G} = \underline{\mathrm{Id}} - \underline{N}\underline{L}$$

holds true.

Theorem 6.2. *(cf. [Mei11, Thm. 4.5]) A linear consistent iterative method is convergent if and only if the spectral radius $\rho(\underline{G})$ of the iteration matrix $\underline{G}$ fulfills the condition*

$$\rho(\underline{G}) < 1.$$

Definition 6.3. Let $\underline{B} \in \mathbb{R}^{N \times N}$ be invertible. The linear system

$$\underline{B}\underline{L}\underline{u} = \underline{B}\underline{b} \tag{6.3}$$

is called *preconditioned* with respect to the linear system (6.1) and $\underline{B}$ is called *precondition matrix*. Hence, a *preconditioner* is a method with suitable precondition matrix $\underline{B}$ such that

$$\underline{B}\underline{L} \approx \underline{\mathrm{Id}}$$

where $\underline{B}$ can be computed or applied with much lower computational costs, compared to $\underline{L}$.

The following lemma shows that it is very easy to apply iterative methods as a preconditioner:

Lemma 6.1. *Solving the original linear system* (6.1) *with a consistent linear iterative method*

$$\Phi(\underline{u}) = \underline{G}\underline{u} + \underline{N}\underline{b}$$

is equivalent to the simple (Richardson's) iteration (cf. [Mei11, 4.1.4])

$$\underline{u}^{(i+1)} = (\underline{\mathrm{Id}} - \underline{L})\,\underline{u}^{(i)} + \underline{b}, \qquad \text{for } i = 0, 1, \ldots$$

applied to the preconditioned system (6.3) *with precondition matrix $\underline{B} = \underline{N}$.*

Proof. Apply Richardson's iteration to the preconditioned system yields

$$\underline{u}^{(i+1)} = (\underline{\mathrm{Id}} - \underline{N}\underline{L})\,\underline{u}^{(i)} + \underline{N}\underline{b} = \underline{G}\underline{u}^{(i)} + \underline{N}\underline{b} = \Phi(\underline{u}^{(i)}),$$

where we used the consistency condition in the last step. □

6.2.2 Smoothers

For the smoothing operations, mentioned in Sec. 6.1, one usually considers iterative splitting methods or incomplete LU decomposition and corresponding block-versions (see, e.g., [Saa03, Sec. 13.2.2, Sec. 13.2.3] or [Hac03, Sec. 3.2]). In this work we focus on two iterative splitting methods as smoother, namely the damped block-Jacobi and block-Gauss–Seidel method. We consider the smoothing methods

as preconditioner (cf. Lemma 6.1), where the corresponding precondition matrices are given as

$$\underline{B}_{l,k}^{\mathrm{J}} = \theta_{l,k}\mathrm{block\_diag}_{|\mathcal{R}_{l.k}|}(\underline{L}_{l,k})^{-1}, \tag{6.4}$$

$$\underline{B}_{l,k}^{\mathrm{GS}} = \theta_{l,k}\left(\mathrm{block\_lower}_{|\mathcal{R}_{l.k}|}(\underline{L}_{l,k}) + \mathrm{block\_diag}_{|\mathcal{R}_{l.k}|}(\underline{L}_{l,k})\right)^{-1}. \tag{6.5}$$

Here $\theta_{l,k} \in (0,1]$ denotes a damping factor on level (l,k). Furthermore the functions block_diag and block_lower return the block-diagonal matrix and the block-lower triangular matrix of an input matrix, respectively. Note that the block-methods used as a smoother differ from the block-Gauss–Seidel solver mentioned in (4.24). In this case every block is associated to a space-time cell $R \in \mathcal{R}_{l,k}$, whereas in (4.24) a block is associated to a slice of cells in time. To prevent confusion, $\mathrm{block\_diag}_n$ and $\mathrm{block\_lower}_n$ are equipped with an index $n \geq 1$ which denotes the number of blocks on the diagonal. Hence $n = |\mathcal{R}_{l.k}|$ corresponds to the amount of space-time cells. The corresponding iteration matrices are then given by

$$\underline{S}_{l,k}^{\mathrm{J}} = \underline{\mathrm{Id}}_{l,k} - \underline{B}_{l,k}^{\mathrm{J}}\underline{L}_{l,k},$$
$$\underline{S}_{l,k}^{\mathrm{GS}} = \underline{\mathrm{Id}}_{l,k} - \underline{B}_{l,k}^{\mathrm{GS}}\underline{L}_{l,k}.$$

Since both methods are linear, the iteration scheme can be written as

$$\underline{u}^{(k+1)} = \underline{u}^{(k)} + \underline{B}_{l,k}\left(\underline{b}_{l,k} - \underline{L}_{l,k}\underline{u}^{(k)}\right)$$

and hence the error is given as

$$\underline{u}^{(k+1)} - \underline{u} = \underline{S}_{l,k}\left(\underline{u}^{(k)} - \underline{u}\right) = \ldots = \left(\underline{S}_{l,k}\right)^{k+1}\left(\underline{u}^{(0)} - \underline{u}\right)$$

for some initial vector $\underline{u}^{(0)}$ and $\underline{B}_{l,k} \in \{\underline{B}_{l,k}^{\mathrm{J}}, \underline{B}_{l,k}^{\mathrm{GS}}\}$ and $\underline{S}_{l,k} \in \{\underline{S}_{l,k}^{\mathrm{J}}, \underline{S}_{l,k}^{\mathrm{GS}}\}$.

Remark 6.1. (Parallel Gauss–Seidel smoother) The block-Jacobi method computes the inverse of the block-matrices on the diagonal of $\underline{L}$. Since these blocks are decoupled this can be done completely in parallel. On the other hand the block-Gauss–Seidel methods inverts the block-lower triangular matrix. In this case the single blocks are not decoupled and hence the computation cannot be performed in parallel. For this reason the method is understood as a local block-Gauss–Seidel method on every process throughout this work. Hence the precondition matrix of the parallel version of the block-Gauss–Seidel smoother reads as

$$\underline{B}_{l,k}^{\mathrm{GS}} = \theta_{l,k}\left(\mathrm{block\_lower}_{|\mathcal{R}_{l.k}|}(\widetilde{\underline{L}}_{l,k}) + \mathrm{block\_diag}_{|\mathcal{R}_{l.k}|}(\widetilde{\underline{L}}_{l,k})\right)^{-1},$$

where $\widetilde{\underline{L}}_{l,k} = \mathrm{block\_diag}_P(\underline{L}_{l,k})$. The function $\mathrm{block\_diag}_P$ returns another block-diagonal matrix, where one block is build of all components, which correspond to one process. Here the index P denotes the number of parallel processes and thus the number of blocks (cf. Figure 6.1). In the extremal cases, i.e., where the number

$$\text{block\_lower}(\underline{L}_{l,k}) + \text{block\_diag}(\underline{L}_{l,k}) = \begin{pmatrix} * & & & & & & \\ * & * & & & & & \\ & * & * & & & & \\ & & * & * & & & \\ & & & * & * & & \\ & & & & * & * & \\ & & & & & * & * \end{pmatrix}$$

$$\text{block\_lower}(\widetilde{\underline{L}}_{l,k}) + \text{block\_diag}(\widetilde{\underline{L}}_{l,k}) = \left(\begin{array}{ccc|ccc} * & & & & & \\ * & * & & & & \\ & * & * & & & \\ \hline & & & * & & \\ & & & * & * & \\ & & & & * & * \end{array}\right)$$

Figure 6.1: Visualization of the original block-Gauss–Seidel method and its parallel version on $P = 2$ processes.

of processes is equal to one or corresponds to the number of used space-time cells, respectively, it holds for $P = 1$ that

$$\widetilde{\underline{L}}_{l,k} = \text{block\_diag}_1(\underline{L}_{l,k}) = \underline{L}_{l,k}$$

and

$$\begin{aligned} \text{block\_lower}_{|\mathcal{R}_{l,k}|}(\widetilde{\underline{L}}_{l,k}) &= 0, \\ \text{block\_diag}_{|\mathcal{R}_{l,k}|}(\widetilde{\underline{L}}_{l,k}) &= \text{block\_diag}_{|\mathcal{R}_{l,k}|}(\underline{L}_{l,k}), \end{aligned}$$

for $P = |\mathcal{R}_{l,k}|$. Hence, for $P = 1$ we recover the original version of the block-Gauss–Seidel method (6.5), whereas for $P = |\mathcal{R}_{l,k}|$ the parallel block-Gauss–Seidel is equivalent to the block-Jacobi method (6.4).

6.2.3 Mesh Transfer Operations

In the context of multigrid methods it is necessary to be able to transfer informations from one mesh to another. To do this, mesh transfer operations are needed. Since in this work the meshes are nested, we have that the ansatz and test spaces are also nested with respect to space and time, i.e.,

$$\begin{array}{ccccccccc} V_{0,k} & \subset & V_{1,k} & \subset \ldots \subset & V_{l_{\max}-1,k} & \subset & V_{l_{\max},k}, \\ V_{l,0} & \subset & V_{l,1} & \subset \ldots \subset & V_{l,k_{\max}-1} & \subset & V_{l,k_{\max}}, \\ W_{l_{\max},k} & \supset & W_{l_{\max}-1,k} & \supset \ldots \supset & W_{1,k} & \supset & W_{0,k}, \\ W_{l,k_{\max}} & \supset & W_{l,k_{\max}-1} & \supset \ldots \supset & W_{l,1} & \supset & W_{l,0}, \end{array}$$

where $l = 0, \ldots, l_{\max}$ and $k = 0, \ldots, k_{\max}$. Throughout this section we consider the scalar valued case, e.g., $J = 1$. For the vector valued case ($J > 1$) the mesh transfer operations can be applied component-wise. Moreover, we assume that we use homogeneous polynomial degrees $p_R \equiv p$ and $q_R \equiv q$ on every mesh $\mathcal{R}_{l,k}$ for $l = 0, \ldots, l_{\max}$ and $k = 0, \ldots, k_{\max}$. The extension to the adaptive case is discussed in the subsequent section.

Definition 6.4 (Restriction and prolongation)**.** The surjective mappings from a fine mesh $\mathcal{R}_{l,k}$ to a coarse mesh $\mathcal{R}_{l-1,k}$ or $\mathcal{R}_{l,k-1}$, respectively, are called *restrictions*. They are given as

$$R_{l,k}^{l-1,k} \colon W_{l,k} \to W_{l-1,k} \qquad \text{and} \qquad R_{l,k}^{l,k-1} \colon W_{l,k} \to W_{l,k-1},$$

where $R_{l,k}^{l-1,k}$ is the restriction operator in space and $R_{l,k}^{l,k-1}$ the restriction operator in time. The counterpart to restriction is called *prolongation*. The corresponding injective prolongation operators are given as

$$P_{l-1,k}^{l,k} \colon V_{l-1,k} \to V_{l,k} \qquad \text{and} \qquad P_{l,k-1}^{l,k} \colon V_{l,k-1} \to V_{l,k},$$

respectively. Analogously to the restriction, $P_{l-1,k}^{l,k}$ denotes the prolongation in space, whereas $P_{l,k-1}^{l,k}$ denotes the prolongation in time.

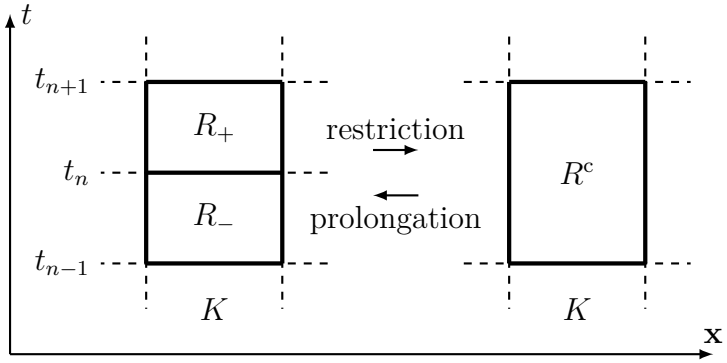

Figure 6.2: Local mesh transfer operations in time for two batched space-time cells $R_- = K \times (t_{n-1}, t_n)$ and $R_+ = K \times (t_n, t_{n+1})$

Due to the nested structure, restriction and prolongation operations can be defined locally, i.e., cell-wise. The local mesh transfer in time and space is illustrated in Figures 6.2 and 6.3, respectively. The following definition and remark give hints how restriction and prolongation operations can be chosen in a reasonable manner.

Definition 6.5. (Trivial injection) In a nested finite element structure coarse nodal points are always nodal points on the fine mesh, too. Hence we can define a restriction by using a point-wise transfer of the coefficient vector of $w_{l,k} \in W_{l,k}$ to the coefficients of $w_{l-1,k} \in W_{l-1,k}$ or $w_{l,k-1} \in W_{l,k-1}$, respectively. This restriction is called *trivial injection*.

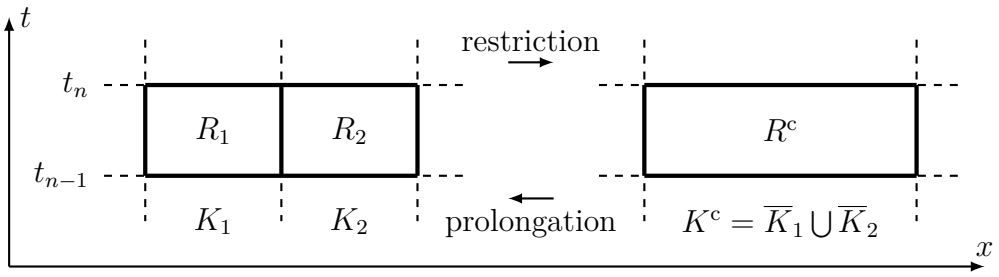

Figure 6.3: Local mesh transfer operations in space for $R_1 = K_1 \times (t_{n-1}, t_n)$ and $R_2 = K_2 \times (t_{n-1}, t_n)$ in one spatial dimension.

Remark 6.2. The trivial injection can be performed without any additional computational costs. However, it contains the danger of loosing or distort informations during the restriction to coarser grids. In a nested finite element context one usually uses interpolation operators for prolongation and the corresponding adjoint operators for restriction, i.e., $R = P^*$ (cf. Definition 5.1). This approach avoids problems, since the restriction and prolongation are chosen in a canonical way. Thus the restriction is equivalent to a projection, see [Hac03, Sec. 3.6].

To reduce computational costs, the canonical projection can be replaced by a reasonable weighted restriction. Due to [Hac03, Sec. 3.8.5] a weighted restriction operation $R_{l,k}^{l-1,k}$ is reasonable if its adjoint $\left(R_{l,k}^{l-1,k}\right)^*$ is a stable prolongation, e.g., the operator norm of $\left(R_{l,k}^{l-1,k}\right)^*$ is bounded independently of the mesh size $h_{l,k}$. It can be shown that the trivial injection is not a stable restriction in general.

In the follow we introduce restriction and prolongation operations in space and time. In doing so we stick to the hints given in the previous remark. Furthermore, we discuss the representation of mesh transfer operations as local restriction and prolongation matrices. We begin with the mesh transfer operations in time. The spatial case is outlined afterwards.

Prolongation and Restriction Operations in Time

We first consider the time-dependent prolongation mapping from $V_{l,k-1}$ to $V_{l,k}$ for $l \geq 0$, $k > 0$. We define the prolongation in time as a nodal interpolation

$$P_{l,k-1}^{l,k} \colon V_{l,k-1} \to V_{l,k}.$$

The prolongation can be evaluated locally (cell-wise). Let $R_- = K \times (t_{n-1}, t_n), R_+ = K \times (t_n, t_{n+1}) \in \mathcal{R}_{l,k}$ be two batched space-time cells on the fine mesh with a common parent cell $R^c = K \times (t_{n-1}, t_{n+1}) \in \mathcal{R}_{l,k-1}$ on the coarse mesh. The local prolongations are then given as

$$P_{R^c}^{R_-} \colon V_{h,R^c} \to V_{h,R_-} \qquad \text{and} \qquad P_{R^c}^{R_+} \colon V_{h,R^c} \to V_{h,R_+}.$$

Since $v_{h,R^c} \in V_{h,R^c}$ and $v_{h,R} \in V_{h,R}$, for $R \in \{R_-, R_+\} \subset \mathcal{R}_{l,k}$, are uniquely defined in terms of their coefficients (cf. Definition 4.5) and nodal bases $\{\psi_{R^c,j}\}_{j=1,\dots,\dim V_{h,R^c}}$ and $\{\psi_{R,j}\}_{j=1,\dots,\dim V_{h,R}}$ we compute

$$v_{h,R} = \sum_{i=1}^{\dim V_{h,R}} v_{h,R^c}(\mathbf{x}_i, t_i)\psi_{R,i}(\mathbf{x}, t) = \sum_{i=1}^{\dim V_{h,R}} \sum_{j=1}^{\dim V_{h,R^c}} \underline{v}_j^c \psi_{R^c,j}(\mathbf{x}_i, t_i)\psi_{R,i}(\mathbf{x}, t)$$

with coefficient vector $\underline{v}^c = \left(\underline{v}_0^c, \dots, \underline{v}_{\dim V_{h,R^c}}^c\right)^\top$. Here $(\mathbf{x}_i, t_i)$, $i = 1, \dots, \dim V_{h,R}$, denote all nodal points associated with the cell R. Note that we have to take the continuity in time into account. Hence the coefficients of $v_{h,R_-,R_+} \in \operatorname{span}\{V_{h,R_-} \cup V_{h,R_+}\} = V_{h,R_-,R_+}$ can be computed by using a $\dim V_{h,R_-,R_+} \times \dim V_{h,R^c}$ interpolation matrix (cf. Section 4.6), i.e.,

$$\underline{v}_{R_-,R_+} = \underline{P}_c^{-,+} \underline{v}^c.$$

The corresponding components are given as $(\underline{P}_c^{-,+})_{i,j} = \psi_{R^c,j}(\mathbf{x}_{R_-,R_+,i}, t_{R_-,R_+,i})$, where $(\mathbf{x}_{R_-,R_+,i}, t_{R_-,R_+,i})$ are the nodal points associated with the cells R_- and R_+, see Section 4.6. This results in a global prolongation matrix $\underline{P}_{l,k-1}^{l,k} \in \mathbb{R}^{N \times N}$ with block-entries, i.e.,

$$\underline{P}_{l,k-1}^{l,k} = \begin{pmatrix} \underline{P}_c^{-,+} & & & \\ & \underline{P}_c^{-,+} & & \\ & & \ddots & \\ & & & \underline{P}_c^{-,+} \end{pmatrix}.$$

In case of standard Galerkin methods, where ansatz and test spaces coincide (cf. Section 4.1), one would define the restriction in time as adjoint prolongation operator $R_{l,k}^{l,k-1} = \left(P_{l,k-1}^{l,k}\right)^*$. Thus, the restriction matrix is the transposed prolongation matrix $\underline{R}_{l,k}^{l,k-1} = \left(\underline{P}_{l,k-1}^{l,k}\right)^\top$. Since we use a Petrov-Galerkin discretization in time this procedure is not possible in our case. To solve this problem we first define an interpolation as prolongation

$$\widetilde{P}_{l,k-1}^{l,k} \colon W_{l,k-1} \to W_{l,k}$$

between the discontinuous test spaces. Afterwards we use the adjoint operator of $\widetilde{P}_{l,k-1}^{l,k}$ as restriction in time. The local interpolation is then similar as before. However, due to the discontinuities in time the interpolation decouples and we achieve

$$\underline{v}_{R_-} = \underline{\widetilde{P}}_c^- \underline{v}^c \qquad \text{and} \qquad \underline{v}_{R_+} = \underline{\widetilde{P}}_c^+ \underline{v}^c$$

for the coefficient vectors $\underline{v}_{R_-}$ and $\underline{v}_{R_+}$ of $v_{h,R_-} \in V_{h,R_-}$ and $v_{h,R_+} \in V_{h,R_+}$, respectively. The components of the $\dim V_{h,R_-} \times \dim V_{h,R^c}$ matrix are given as $\left(\underline{\widetilde{P}}_c^-\right)_{i,j} =$

$\psi_{R^c,j}(\mathbf{x}_{R_-,i}, t_{R_-,i})$ associated nodal points $(\mathbf{x}_{R_-,i}, t_{R_-,i})$ on R_-. The local interpolation matrix $\underline{\widetilde{P}}_{\mathrm{c}}^{+}$ can be computed analogously. Thus the global prolongation matrix is given as

$$\underline{\widetilde{P}}_{l,k-1}^{l,k} = \begin{pmatrix} \underline{\widetilde{P}}_{\mathrm{c}}^{-} & & & \\ \underline{\widetilde{P}}_{\mathrm{c}}^{+} & & & \\ & \underline{\widetilde{P}}_{\mathrm{c}}^{-} & & \\ & \underline{\widetilde{P}}_{\mathrm{c}}^{+} & & \\ & & \ddots & \\ & & & \underline{\widetilde{P}}_{\mathrm{c}}^{-} \\ & & & \underline{\widetilde{P}}_{\mathrm{c}}^{+} \end{pmatrix}.$$

We use $\underline{\widetilde{P}}_{l,k-1}^{l,k}$ to define the restriction in time as

$$R_{l,k}^{l,k-1} \colon W_{l,k} \to W_{l,k-1}, \qquad R_{l,k}^{l,k-1} = \left(\widetilde{P}_{l,k-1}^{l,k}\right)^*$$

with corresponding restriction matrix

$$\underline{\widetilde{R}}_{l,k}^{l,k-1} = \left(\underline{\widetilde{P}}_{l,k-1}^{l,k}\right)^\top = \begin{pmatrix} \underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}} & & & \\ & \underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}} & & \\ & & \ddots \; \ddots & \\ & & & \underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}} \end{pmatrix},$$

where $\underline{R}_-^{\mathrm{c}} = \left(\underline{\widetilde{P}}_{\mathrm{c}}^{-}\right)^\top$ and $\underline{R}_+^{\mathrm{c}} = \left(\underline{\widetilde{P}}_{\mathrm{c}}^{+}\right)^\top$. The following example illustrates the local prolongation and restriction matrices in time, if space is neglected.

Example 6.1. The first three local prolongation matrices in time for $q = 1, \ldots, 3$ can be computed as

$$\underline{P}_{\mathrm{c}}^{-,+} = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}, \quad \underline{P}_{\mathrm{c}}^{-,+} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{8} & \frac{3}{4} & -\frac{1}{8} \\ 0 & 1 & 0 \\ -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \\ 0 & 0 & 1 \end{pmatrix}, \quad \underline{P}_{\mathrm{c}}^{-,+} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{5}{16} & \frac{15}{16} & -\frac{5}{16} & \frac{1}{16} \\ 0 & 1 & 0 & 0 \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ 0 & 0 & 1 & 0 \\ \frac{1}{16} & -\frac{5}{16} & \frac{15}{16} & \frac{5}{16} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

For $q = 1$ the local prolongation is illustrated in Figure 6.4. The corresponding first three local restriction matrices in time can be computed as

$$\left(\underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}}\right) = \begin{pmatrix} 1 & 1 \end{pmatrix}, \; \left(\underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}}\right) = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix},$$

$$\left(\underline{R}_-^{\mathrm{c}} \; \underline{R}_+^{\mathrm{c}}\right) = \begin{pmatrix} 1 & \frac{3}{8} & 0 & 0 & -\frac{1}{8} & 0 \\ 0 & \frac{3}{4} & 1 & 1 & \frac{3}{4} & 0 \\ 0 & -\frac{1}{8} & 0 & 0 & \frac{3}{8} & 1 \end{pmatrix}.$$

See Figure 6.5 for an illustration of local restriction for $q = 2$.

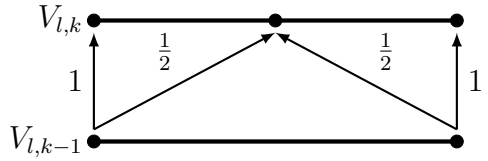

Figure 6.4: Local prolongation in time for polynomial degree $q = 1$.

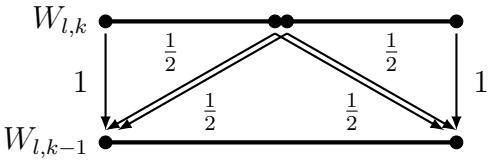

Figure 6.5: Local restriction in time for polynomial degree $q = 2$.

Prolongation and Restriction Operations in Space

Again, we first consider the prolongation mapping in space from $V_{l-1,k}$ to $V_{l,k}$ for $l > 0, k \geq 0$. We define the prolongation in space as an interpolation

$$P_{l-1,k}^{l,k} \colon V_{l-1,k} \to V_{l,k}.$$

The prolongation can be again evaluated locally (cell-wise). Let $R^{\mathrm{c}} = K^{\mathrm{c}} \times (t_{n-1}, t_n) \in \mathcal{R}_{l-1,k}$ be a coarse cell with 2^D child cells $R_m \subseteq R^{\mathrm{c}}$, $m = 1, \ldots, 2^D$. As before $v_{h,k} \in V_{h,R_m}$ is defined uniquely through its coefficient vector $\underline{v}_m \in \mathbb{R}^{\dim V_{h,R_m}}$. Hence we achieve

$$\underline{v}_m = \underline{P}_{\mathrm{c}}^m \underline{v}^{\mathrm{c}},$$

where $\underline{P}_{\mathrm{c}}^m$ is an local $\dim V_{h,R_k} \times \dim V_{h,R^{\mathrm{c}}}$ interpolation matrix. The components are given as $\left(\underline{P}_{\mathrm{c}}^k\right)_{i,j} = \psi_{R^{\mathrm{c}},j}(\mathbf{x}_i, t_i)$ for nodal points $(\mathbf{x}_i, t_i)$ associated with $R_m \in \mathcal{R}_{l,k}$. Hence the global prolongation matrix reads as

$$\underline{P}_{l-1,k}^{k,l} = \begin{pmatrix} \underline{\mathbf{P}}_{\mathrm{c}} & & & \\ & \underline{\mathbf{P}}_{\mathrm{c}} & & \\ & & \ddots & \\ & & & \underline{\mathbf{P}}_{\mathrm{c}} \end{pmatrix}, \qquad \text{where} \quad \underline{\mathbf{P}} = \begin{pmatrix} \underline{P}_{\mathrm{c}}^1 \\ \underline{P}_{\mathrm{c}}^2 \\ \vdots \\ \underline{P}_{\mathrm{c}}^{2^D} \end{pmatrix}.$$

To save computational resources, we apply a weighted restriction in space. We use the trivial injection, introduced in Definition 6.5, and equip it with an additional weight. To circumvent stability problems, mentioned in Remark 6.2, one can use the weight $h_{l-1,k}^D / h_{l,k}^D$, where $h_{l,k}$ is the corresponding mesh size to $\mathcal{R}_{l,k}$ (cf. [Hac03, Ch. 3.5]). Since we have a nested mesh structure, we achieve that $h_{l-1,k}^D / h_{l,k}^D = 2^D$. Thus we define the restriction in space as weighted injection

$$R_{l,k}^{l-1,k} \colon W_{l,k} \to W_{l-1,k}.$$

Locally the restriction is given as

$$R^{\mathrm{c}}_{R_1,\dots,R_{2^D}} : W_{h,R_1,\dots,R_{2^D}} = \operatorname{span}\left\{ \bigcup_{m=1}^{2^D} W_{h,R_m} \right\} \to W_{h,R^{\mathrm{c}}}.$$

As before $w_{h,R^{\mathrm{c}}} \in W_{h,R^{\mathrm{c}}}$ is given uniquely trough its coefficients

$$\underline{w}^{\mathrm{c}}_i = 2^D \left\{ w_h(\mathbf{x}^{\mathrm{c}}_i, t^{\mathrm{c}}_i) \right\}_i ,$$

for a given $w_h \in W_{h,R_1,\dots,R_{2^D}}$. Here $\{.\}_i$ again denotes the averages on the coarse nodal points $(\mathbf{x}^{\mathrm{c}}_i, t^{\mathrm{c}}_i)$, for $i = 1, \dots, \dim W_{R^{h,\mathrm{c}}}$ (cf. Section 5.3.2). Since every nodal point on the coarse mesh corresponds to a nodal point $(\mathbf{x}_{R_m,j}, t_{R_m,j})$ associated with at least one $R_m \subset R^{\mathrm{c}}$ on the fine mesh, the local restriction matrix $\underline{R}^{\mathrm{c}}_m \in \mathbb{R}^{\dim W_{h,R^{\mathrm{c}}} \times \dim W_{h,R_m}}$ can be computed component-wise as

$$(\underline{R}^{\mathrm{c}}_m)_{i,j} = \begin{cases} \frac{2^D}{N_i}, & \text{if } (\mathbf{x}^{\mathrm{c}}_i, t^{\mathrm{c}}_i) = (\mathbf{x}_{R_m,j}, t_{R_m,j}), \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for } m = 1, \dots, 2^D.$$

Here $N_i = |\{R = R_1, R_2, \dots, R_{2^D} \colon (\mathbf{x}^{\mathrm{c}}_i, t^{\mathrm{c}}_i) \text{ is a nodal point of } W_{h,R}\}|$ denotes the number of nodal points on the fine mesh a coarse nodal point $(\mathbf{x}^{\mathrm{c}}_i, t^{\mathrm{c}}_i)$ is connected to. Hence we obtain the global space-time restriction matrix in space

$$\underline{R}^{l-1,k}_{l,k} = \begin{pmatrix} \underline{\mathbf{R}}^{\mathrm{c}} & & & \\ & \underline{\mathbf{R}}^{\mathrm{c}} & & \\ & & \ddots & \\ & & & \underline{\mathbf{R}}^{\mathrm{c}} \end{pmatrix},$$

where $\underline{\mathbf{R}}^{\mathrm{c}} = (\underline{R}^{\mathrm{c}}_1, \underline{R}^{\mathrm{c}}_2, \ \dots, \ \underline{R}^{\mathrm{s}}_{2^D})$ is the corresponding block-matrix vector containing the local restriction matrices. The following example illustrates the resulting local prolongation and restriction matrices in space, if time is neglected.

Example 6.2. For $p = 1, \dots, 3$ and $D = 1$ the local prolongation matrices in space can be computed as

$$\begin{pmatrix} \underline{P}^1_{\mathrm{c}} \\ \underline{P}^2_{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} \underline{P}^1_{\mathrm{c}} \\ \underline{P}^2_{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{8} & \frac{3}{4} & -\frac{1}{8} \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \\ 0 & 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} \underline{P}^1_{\mathrm{c}} \\ \underline{P}^2_{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{5}{16} & \frac{15}{16} & -\frac{5}{16} & \frac{1}{16} \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{16} & -\frac{5}{16} & \frac{15}{16} & \frac{5}{16} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

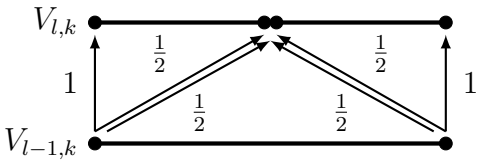

Figure 6.6: Prolongation in space for polynomial degree $p = 1$ in one dimension.

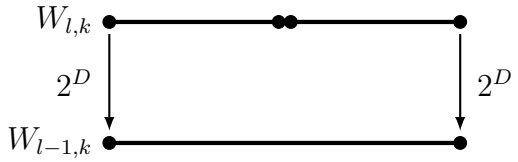

Figure 6.7: Restriction in space for polynomial degree $p = 1$ in one dimension.

Correspondingly the local restriction matrix in space can be computed as

$$(\underline{R}_1^{\mathrm{c}}\ \underline{R}_2^{\mathrm{c}}) = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad (\underline{R}_1^{\mathrm{c}}\ \underline{R}_2^{\mathrm{c}}) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix},$$

$$(\underline{R}_1^{\mathrm{c}}\ \underline{R}_2^{\mathrm{c}}) = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

The local prolongation and restriction in space is illustrated in Figures 6.6 and 6.6 for $p = 1$, respectively.

6.2.4 Extension to the p-Adaptive Case

In case of adaptive polynomial degree refinement we locally interpolate the data on the child cells to homogeneous polynomial degrees p and q in space and time. Then we choose subspaces $W_{h,R^{\mathrm{c}}}$ of the same polynomial degrees. This allows us to apply the same restriction and prolongation matrices as in the homogeneous case if we scale them with the correct interpolation matrices. The precise procedure is given in the following two paragraphs.

Restriction and Prolongation in Time

Let $p_- = p_{R_-}$, $p_+ = p_{R_+}$ and $q_- = q_{R_-}$, $q_+ = q_{R_+}$ be the polynomial degrees on tow batched fine cells R_-, R_+ in space and time, respectively. Determine the highest polynomial degrees $p = \max\{p_-, p_+\}$, $q = \max\{q_-, q_-\}$ and set $p_{R^{\mathrm{c}}} = p$ and $q_{R^{\mathrm{c}}} = q$. Then one can reuse the interpolation matrices $\underline{I}^{p,q}_{p_-,q_-}, \underline{I}^{p,q}_{p_+,q_+}$ and

$\underline{I}^{p_-,q_-}_{p,q}, \underline{I}^{p_+,q_+}_{p,q}$ defined in equation (4.22). Incorporating them into the restriction and prolongation matrices yields

$$\widetilde{\underline{R}}^{l,k}_{l,k-1} = \left(\widetilde{\underline{P}}^{l,k}_{l,k-1}\right)^\top = \begin{pmatrix} \underline{R}^{\mathrm{c}}_- \; \underline{R}^{\mathrm{c}}_+ & & & \\ & \underline{R}^{\mathrm{c}}_- \; \underline{R}^{\mathrm{c}}_+ & & \\ & & \ddots \; \ddots & \\ & & & \underline{R}^{\mathrm{c}}_- \; \underline{R}^{\mathrm{c}}_+ \end{pmatrix},$$

where $\underline{R}^{\mathrm{c}}_- = \left(\widetilde{\underline{P}}^{-}_{\mathrm{c}}\right)^\top \underline{I}^{p,q-1}_{p_-,q_--1}$ and $\underline{R}^{\mathrm{c}}_+ = \left(\widetilde{\underline{P}}^{+}_{\mathrm{c}}\right)^\top \underline{I}^{p,q-1}_{p_+,q_+-1}$. Analogously we achieve

$$\underline{P}^{l,k}_{l,k-1} = \begin{pmatrix} \underline{P}^{+,-}_{\mathrm{c}} & & & \\ & \underline{P}^{+,-}_{\mathrm{c}} & & \\ & & \ddots & \\ & & & \underline{P}^{+,-}_{\mathrm{c}} \end{pmatrix}$$

where $\underline{P}^{+,-}_{\mathrm{c}}$ is replaced by

$$\begin{pmatrix} \underline{I}^{p_-,q_-}_{p,q} & \\ & \underline{I}^{p_+,q_+}_{p,q} \end{pmatrix} \underline{P}^{+,-}_{\mathrm{c}}.$$

Restriction and Prolongation in Space

Let $p_1 = p_{R_1}$, $p_2 = p_{R_2}$, … , $p_{2^D} = p_{R_{2^D}}$ and $q_1 = q_{R_1}$, $q_2 = q_{R_2}$, … , $q_{2^D} = q_{R_{2^D}}$ be the polynomial degrees on the set of fine cells $R_1, \ldots, R_{2^D}$ in space and time, respectively. Determine the highest polynomial degrees $p = \max\{p_1, p_2, \ldots, p_{2^D}\}$, $q = \max\{q_1, q_2, \ldots, q_{2^D}\}$ and set $p_{R^{\mathrm{c}}} = p$ and $q_{R^{\mathrm{c}}} = q$. Again we reuse the interpolation matrices $\underline{I}^{p,q}_{p_1,q_1}$, $\underline{I}^{p,q}_{p_2,q_2}$, … , $\underline{I}^{p,q}_{p_{2^D},q_{2^D}}$ and $\underline{I}^{p_1,q_1}_{p,q}, \underline{I}^{p_2,q_2}_{p,q}, \ldots, \underline{I}^{p_{2^D},q_{2^D}}_{p,q}$ defined in equation (4.22). Incorporating them into the restriction and prolongation matrices yields the new restriction in space

$$\underline{R}^{l,k-1}_{l,k} = \begin{pmatrix} \underline{\mathbf{R}}^{\mathrm{c}} & & & \\ & \underline{\mathbf{R}}^{\mathrm{c}} & & \\ & & \ddots & \\ & & & \underline{\mathbf{R}}^{\mathrm{c}} \end{pmatrix}$$

where $\underline{\mathbf{R}}^{\mathrm{c}} = \left(\underline{R}^{\mathrm{c}}_1 \; \underline{I}^{p,q-1}_{p_1,q_1-1}, \underline{R}^{\mathrm{c}}_2 \; \underline{I}^{p,q-1}_{p_2,q_2-1}, \ldots, \underline{R}^{\mathrm{c}}_{2^D} \; \underline{I}^{p,q-1}_{p_{2^D},q_{2^D}-1}\right)$. Analogously the new prolongation in space reads as

$$\underline{P}^{l,k}_{l-1,k} = \begin{pmatrix} \underline{\mathbf{P}}_{\mathrm{c}} & & & \\ & \underline{\mathbf{P}}_{\mathrm{c}} & & \\ & & \ddots & \\ & & & \underline{\mathbf{P}}_{\mathrm{c}} \end{pmatrix}, \quad \text{where} \quad \underline{\mathbf{P}}_{\mathrm{c}} = \begin{pmatrix} \underline{I}^{p_1,q_1}_{p,q} \; \underline{P}^{1}_{\mathrm{c}} \\ \underline{I}^{p_2,q_2}_{p,q} \; \underline{P}^{2}_{\mathrm{c}} \\ \vdots \\ \underline{I}^{p_{2^D},q_{2^D}}_{p,q} \; \underline{P}^{2^D}_{\mathrm{c}} \end{pmatrix}.$$

6.2.5 Coarse Grid Correction

By having the mesh transfer operations at hand, we can define a coarse grid correction. This is done by computing the residual defect $\underline{r}_{l,k}^{(i)} = \underline{b}_{l,k} - \underline{L}_{l,k}\, \underline{u}_{l,k}^{(i)}$ of the solution $\underline{u}_{l,k}$ in the i-th iteration step. For better readability we omit the iteration index i within a fixed iteration step. The residual defect $\underline{r}_{l,k} = \underline{r}_{l,k}^{(i)}$ is restricted to the coarse mesh $\mathcal{R}_{l-1,k}$ or $\mathcal{R}_{l,k-1}$ respectively. To do this, the restriction matrices in space or time, i.e.,

$$\underline{d}_{l-1,k} = \underline{R}_{l,k}^{l-1,k}\underline{r}_{l,k} \quad \text{or} \quad \underline{d}_{l,k-1} = \underline{R}_{l,k}^{l,k-1}\underline{r}_{l,k}$$

are applied. On this level the coarse problems

$$\underline{c}_{l-1,k} = \underline{L}_{l-1,k}^{-1}\underline{d}_{l-1,k} \quad \text{or} \quad \underline{c}_{l,k-1} = \underline{L}_{l,k-1}^{-1}\underline{d}_{l,k-1} \tag{6.6}$$

are solved exactly, e.g., by applying a direct solver. The computed correction is prolongated back to the fine level by

$$\underline{w}_{l,k} = \underline{P}_{l-1,k}^{l,k}\underline{c}_{l-1,k} \quad \text{or} \quad \underline{w}_{l,k} = \underline{P}_{l,k-1}^{l,k}\underline{c}_{l,k-1}.$$

In a final step an improved solution $\underline{u}_{l,k} = \underline{u}_{l,k} + \underline{w}_{l,k}$ is obtained. Assembling all steps, for $i = 0, 1, \ldots$, results in the iteration schemes

$$\underline{u}_{l,k}^{(i+1)} = \underline{u}_{l,k}^{(i)} + \underline{P}_{l-1,k}^{l,k}\underline{L}_{l-1,k}^{-1}\underline{R}_{l,k}^{l-1,k}\left(\underline{b}_{l,k} - \underline{L}_{l,k}\, \underline{u}_{l,k}^{(i)}\right) \tag{6.7}$$

and

$$\underline{u}_{l,k}^{(i+1)} = \underline{u}_{l,k}^{(i)} + \underline{P}_{l,k-1}^{l,k}\underline{L}_{l,k-1}^{-1}\underline{R}_{l,k}^{l,k-1}\left(\underline{b}_{l,k} - \underline{L}_{l,k}\, \underline{u}_{l,k}^{(i)}\right). \tag{6.8}$$

Hence we can state the following lemma:

Lemma 6.2. *(cf. [Mei11, Lem. 4.41, Lem. 4.42]) The coarse grid corrections themselves are consistent iterative methods with corresponding iteration matrices*

$$\underline{C}_{l,k}^{\mathrm{s}} = \underline{\mathrm{Id}}_{l,k} - \underline{P}_{l-1,k}^{l,k}\underline{L}_{l-1,k}^{-1}\underline{R}_{l,k}^{l-1,k}\underline{L}_{l,k}$$

and

$$\underline{C}_{l,k}^{\mathrm{t}} = \underline{\mathrm{Id}}_{l,k} - \underline{P}_{l,k-1}^{l,k}\underline{L}_{l,k-1}^{-1}\underline{R}_{l,k}^{l,k-1}\underline{L}_{l,k}.$$

However, the methods do no converge to a solution $\underline{u}_{l,k} = \underline{L}_{l,k}^{-1}\underline{b}_{l,k}$.

Proof. Since the statements can be shown analogously in space and time, we set $\underline{R}$ and $\underline{P}$ to be either the mesh transfer operations in space or in time, respectively. Moreover, let $\underline{L}_{\mathrm{c}}$ be the corresponding coarse matrix. Both equations (6.7) and (6.8) ca be written as

$$\begin{aligned}\underline{u}_{l,k}^{(i+1)} &= \left(\underline{\mathrm{Id}}_{l,k} - \underline{P}\underline{L}_{\mathrm{c}}^{-1}\underline{R}\underline{L}_{l,k}\right)\underline{u}_{l,k}^{(i)} + \underline{P}\underline{L}_{\mathrm{c}}^{-1}\underline{R}\underline{b}_{l,k} \\ &= \underline{C}_{l,k} + \underline{P}\underline{L}_{\mathrm{c}}^{-1}\underline{R}\underline{b}_{l,k} = \Phi(\underline{u}_{l,k}^{(i)})\end{aligned}$$

with iteration matrix $\underline{C}_{l,k} = \underline{C}^{\mathrm{s}}_{l,k}$ or $\underline{C}_{l,k} = \underline{C}^{\mathrm{t}}_{l,k}$, respectively. Moreover, we immediately note that the coarse grid correction is a consistent iterative method, due to Theorem 6.1.

To prove that the coarse grid corrections do not converge, we follow a proof which is stated in [Mei11, Lem. 4.42]. Since the restriction is surjective, but not injective, the kernel of the restriction operator is not trivial. As a consequence the restriction and prolongation matrices are not square matrices and hence $\rho\left(\underline{C}_{l,k}\right) \geq 1$. This means that the coarse grid correction is non-contractive and due to Theorem 6.2 does not converge. □

6.2.6 Complete Two-Grid Method

By now, all components of a two-grid cycle can be summarized in a single method. One first performs $\nu^{\mathrm{pre}}_{l,k} \geq 0$ pre-smoothing steps using a smoother defined in Section 6.2.2. Afterwards the coarse grid correction, illustrated in Section 6.2.5, is applied. Finally $\nu^{\mathrm{post}}_{l,k} \geq 0$ post-smoothing steps are performed by using again a smoother. Necessarily the smoothers used in the pre- and post-smoothing steps can be different methods or methods with different damping parameters. The complete two-grid cycles in space and time are illustrated in Figures 6.8 and 6.9 and sketched in Algorithms 6.1 and 6.2. Finally we discuss the convergence of the two-grid cycle.

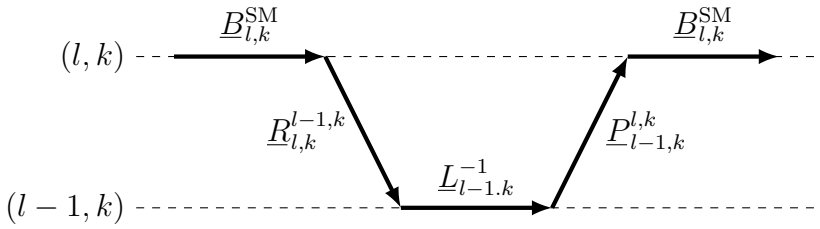

Figure 6.8: Two level coarsening strategy in space represented as sequence of precondition matrices.

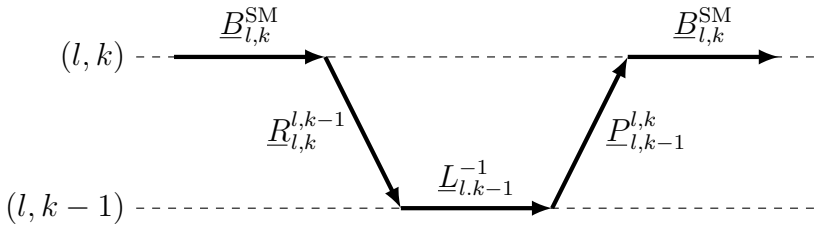

Figure 6.9: Two level coarsening strategy in time represented as sequence of precondition matrices.

Algorithm 6.1 Two-Grid cycle in space with smoother $\underline{B}_{l,k}^{\mathrm{SM}} = \underline{B}_{l,k}^{\mathrm{J}}$ or $\underline{B}_{l,k}^{\mathrm{SM}} = \underline{B}_{l,k}^{\mathrm{GS}}$

(Pre-) smoothing:
for $\nu = 1, \ldots, \nu_{l,k}^{\mathrm{pre}}$ **do**
  $\underline{w}_{l,k} = \underline{B}_{l,k}^{\mathrm{SM}} \underline{r}_{l,k}$
  $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$
Restriction:
  $\underline{d}_{l-1,k} = \underline{R}_{l,k}^{l-1,k} \underline{r}_{l,k}$
Solve coarse grid problem:
  $\underline{c}_{l-1,k} = \textsc{Solve\_Exact}(\underline{A}_{l-1,k}, \underline{r}_{l-1,k})$
Prolongation:
  $\underline{w}_{l,k} = \underline{P}_{l-1,k}^{l,k} \underline{c}_{l-1,k}$
Correction:
  $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$
(Post-) smoothing:
for $\nu = 1, \ldots, \nu_{l,k}^{\mathrm{post}}$ **do**
  $\underline{w}_{l,k} = \underline{B}_{l,k}^{\mathrm{SM}} \underline{r}_{l,k}$
  $\underline{x}_{l,k} = \underline{c}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$

Algorithm 6.2 Two-Grid cycle in time with smoother $\underline{B}_{l,k}^{\mathrm{SM}} = \underline{B}_{l,k}^{\mathrm{J}}$ or $\underline{B}_{l,k}^{\mathrm{SM}} = \underline{B}_{l,k}^{\mathrm{GS}}$

(Pre-) smoothing:
for $\nu = 1, \ldots, \nu_{l,k}^{\mathrm{pre}}$ **do**
  $\underline{w}_{l,k} = \underline{B}_{l,k}^{\mathrm{SM}} \underline{r}_{l,k}$
  $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$
Restriction:
  $\underline{d}_{l,k-1} = \underline{R}_{l,k}^{l,k-1} \underline{r}_{l,k}$
Solve coarse grid problem:
  $\underline{c}_{l,k-1} = \textsc{Solve\_Exact}(\underline{A}_{l,k-1}, \underline{r}_{l,k-1})$
Prolongation:
  $\underline{w}_{l,k} = \underline{P}_{l,k-1}^{l,k} \underline{c}_{l,k-1}$
Correction:
  $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$
(Post-) smoothing:
for $\nu = 1, \ldots, \nu_{l,k}^{\mathrm{post}}$ **do**
  $\underline{w}_{l,k} = \underline{B}_{l,k}^{\mathrm{SM}} \underline{r}_{l,k}$
  $\underline{x}_{l,k} = \underline{c}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k} \underline{w}_{l,k}$

Remark 6.3. The two-grid cycles in space and time are again consistent iterative methods with iteration matrices

$$\underline{G}^{\mathrm{s}}_{l,k} = \left(\underline{S}^{\mathrm{SM}}_{l,k}\right)^{\nu^{\mathrm{post}}_{l,k}} \underline{C}^{\mathrm{s}}_{l,k} \left(\underline{S}^{\mathrm{SM}}_{l,k}\right)^{\nu^{\mathrm{pre}}_{l,k}} \quad \text{and} \quad \underline{G}^{\mathrm{t}}_{l,k} = \left(\underline{S}^{\mathrm{SM}}_{l,k}\right)^{\nu^{\mathrm{post}}_{l,k}} \underline{C}^{\mathrm{t}}_{l,k} \left(\underline{S}^{\mathrm{SM}}_{l,k}\right)^{\nu^{\mathrm{pre}}_{l,k}},$$

see Theorem 6.1. Since for $\nu^{\mathrm{pre}}_{l,k} = \nu^{\mathrm{post}}_{l,k} = 0$ we get that the iteration matrix reduces to $\underline{M}_{l,k} = \underline{C}_{l,k}$ and thus the two-grid cycle is equivalent to the coarse grid correction. This shows that at least one pre- or post-smoothing step is necessary to obtain a convergent method. Due to Theorem 6.2 the two-grid iterations converge if $\rho\left(\underline{G}_{l,k}\right) < 1$ in space and time, respectively. For multigrid methods the solving of the coarse problem (6.6) is recursively replaced by applying one or two additional two-grid cycles, i.e., V-cycle or W-cycle. Hence we cannot deduce the multigrid spectral radius from $\rho\left(\underline{G}^{\mathrm{s}}_{l,k}\right)$ and $\rho\left(\underline{G}^{\mathrm{s}}_{l,k}\right)$ directly (see [Hac03, Sec. 6.1.1]). For that reason we introduce a concept of bounding the spectral radius of the two-grid iteration by investigating the smoother and the coarse grid correction separately.

Definition 6.6 (Smoothing and approximation property)**.** Assume that we only use pre-smoothing within the two-grid cycle, i.e., $\nu^{\mathrm{post}}_{l,k} = 0$ and $\nu^{\mathrm{pre}}_{l,k} = \nu_{l,k} > 0$. The smoother $\underline{S}^{\mathrm{SM}}_{l,k}$ fulfills a so called *smoothing property*, if there exists a positive function $\gamma\colon \mathbb{R} \to \mathbb{R}_+$ with $\gamma(\nu_{l,k}) \to 0$, for $\nu_{l,k} \to \infty$ and a number $\alpha > 0$ such that

$$\|\underline{L}_{l,k} \left(\underline{S}^{\mathrm{SM}}_{l,k}\right)^{\nu}_{l,k} \underline{v}_{l,k}\|_2 \leq c_{\mathrm{SM}} \gamma(\nu_{l,k}) h^{-\alpha}_{l,k} \|\underline{v}_{l,k}\|_2 \tag{6.9}$$

holds uniformly for all $\underline{v}_{l,k} \in \mathbb{R}^N$ and a constant $c_{\mathrm{SM}} > 0$ as $h_{l,k} \to 0$. The coarse grid correction $\underline{C}_{l,k}$ fulfills a so called *approximation property* if

$$\|\underline{C}_{l,k} \underline{L}^{-1}_{l,k} \underline{v}_{l,k}\|_2 \leq c_{\mathrm{approx}} h^{\alpha}_{l,k} \|\underline{v}_{l,k}\|_2 \tag{6.10}$$

holds uniformly for all $\underline{v}_{l,k} \in \mathbb{R}^N$ and a constant $c_{\mathrm{approx}} > 0$ as $h_{l,k} \to 0$. The smoothing properties controls the effect of the applied smoother, whereas the approximation property controls the influence of the mesh transfer operations and the coarse grid correction. Note that the proof of the approximation property requires a certain regularity of the problem (cf. [Hac03, §6.3.1.2]). Moreover, the Euclidean norms can be replaced by suitable weighted norms (cf. [Hac03, §6.3.1.3]). See [Hac03, Sec. 6.2] and [Hac03, Sec. 6.3] for a detailed introduction to smoothing and approximation properties.

The following theorem proves convergence of a two-grid cycle if the smoothing and approximation properties are fulfilled.

Theorem 6.3 (cf. [Hac03, Thm. 6.1.7])**.** *Suppose that the smoothing property* (6.9) *and approximation property* (6.10) *hold for sufficiently large $\nu_{l,k}$. Then we achieve convergence of the two-grid cycle, i.e.,*

$$\rho\left(\underline{G}_{l,k}\right) \leq \|\underline{G}_{l,k}\|_2 = c\gamma(\nu_{l,k}) < 1, \tag{6.11}$$

for a constant $c > 0$ independent of $h_{l,k}$ and the induced matrix norm $\|\,.\,\|_2$.

Remark 6.4. Due to the high complexity, proving the smoothing property (6.9) or approximation property (6.10) explicitly is hardly done for particular applications. Most of the convergence results apply only for elliptic problems (cf. [Hac03, Ch. 11]) and saddle point problems, see, e.g., [BS97], [Zul00], [YZLB07]. For that reason, other techniques are used to predict the behavior of multigrid methods. An example is the so called *local Fourier mode analysis*, where a heuristic local Fourier decomposition of the error is used to investigate the behavior of the smoother and coarse-grid correction with respect to the single basis functions, see, e.g., [Bra94], [TOS01, Ch. 4] or [Neu13, Ch. 4]. Although the obtained results are only valid for very special assumptions, e.g., for periodic boundary conditions, the results can lead to useful predictions for the general case as it is illustrated in the next section.

Remark 6.5 (Application to the multigrid case)**.** The results of the previous theorem can be be transferred to multigrid methods to prove convergence, see [Hac03, Sec. 7.1]. However, the estimates (6.9), (6.10) and (6.11) are not optimal and thus general converge cannot be shown in case of a V-cycle, but require that at least a W-cycle is used [Hac03, Thm. 7.1.7].

A detailed discussion of the V-cycle case is given in [BH83] and [BZ00, Sec. 3]. In case of symmetric positive definite matrices $\underline{L}_{i,j}$ (e.g., as in elliptic problems), duality techniques similar to Chapter 5 can be applied. This leads to a convergence result for the V-cycle and symmetric smoothing with $\nu_{l,k} = \nu_{l,k}^{\text{pre}} = \nu_{l,k}^{\text{post}} \geq 1$, see, e.g., [BZ00, Thm. 3.1] or [Hac03, Thm. 7.2.2]. In particular this means that one pre- and postsmoothing step is already enough to obtain stable multigrid convergence, i.e., constant iteration numbers. Note that this is in contrast to Theorem 6.3, where the number of required smoothing steps depends on the constants c_{SM} and c_{approx}. In the proof of the convergence result, the approximation property (6.10) is replaced by the so called *full regularity and approximation* condition [BZ00, Cond. A.1]. Furthermore, one has to assume that the convergence of the smoother is at least as good as the convergence of the weighted Richardson iteration with iteration matrix

$$\underline{S}_{l,k}^{\text{SM}} = \underline{\text{Id}}_{l,k} - \frac{\theta_{l,k}}{\rho(\underline{L}_{l,k})} \underline{L}_{l,k}$$

for a weight $\theta_{l,k} \in (0, 1)$ (cf. [BZ00, Cond. SM.1] and [BZ00, Rem. 3.2]).

6.3 Block-Jacobi Smoothing in Time

In this section we focus on a heuristic approach to investigate the block-Jacobi smoother within a two-grid cycle in time and derive hints for suitable damping parameters θ^{J}. Since multigrid methods strongly depend on the underlying problem, achieving theoretical convergence results is not an easy task. Therefore we follow the approach of [Neu13, Sec. 4.2] in this section:

We first consider a test equation that depends only on time and prove that the continuous Petrov–Galerkin method in time is equivalent to an implicit Runge–Kutta scheme for all polynomial degrees $q \geq 1$ in this case. Hence we can determine the corresponding stability regions. This result is used to derive a smoothing strategy for the two-grid cycle in time in combination with a block-Jacobi method using a Fourier ansatz (cf. Remark 6.4). Finally, we discuss the possibility of transferring the result to a two-grid cycle in the space-time setting.

6.3.1 Continuous Petrov–Galerkin Method in Time

We consider the one dimensional time-dependent test equation by Dahlquist for an ordinary differential equation

$$\begin{aligned} \partial_t u(t) - \alpha u(t) &= 0, \qquad \text{for } t \in (0,T), \\ u(0) &= u_0, \end{aligned} \tag{6.12}$$

for a given final time $T > 0$ and a constant $\alpha \in \mathbb{R}$ and initial condition u_0 at time $t = 0$. To solve this equation numerically we proceed as in the space-time case and decompose the time interval $(0,T)$ into a finite number N of sub-time intervals $I_n = (t_{n-1}, t_n) \subset I$, where $0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T$. Now we apply a Petrov–Galerkin finite element discretization to the variational formulation of (6.12)

$$\sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} \partial_t uw - \alpha uw \,\mathrm{d}t + (u(0) - u_0)w(0) = 0$$

as described in Section 4.5. On every interval I_n we choose local polynomials of order $q \geq 1$ and $q-1$ for the ansatz and test space, $V_{h,I_n} = \mathbb{P}_q(I_n)$ and $W_{h,I_n} = \mathbb{P}_{q-1}(I_n)$, respectively. This results in the discrete variational formulation of the problem

$$\sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} \partial_t u_{h,I_n} w_{h,I_n} - \alpha u_{h,I_n} w_{h,I_n} \,\mathrm{d}t + (u_h(0) - u_0)w_h(0) = 0 \tag{6.13}$$

for $u_{h,I_n} \in V_{h,I_n}, w_{h,I_n} \in W_{h,I_n}$. The global solution $u_h \in V_h$ is piece-wise given as $u_h(t) = u_{h,I_n}(t)$ for $t \in \overline{I_n}$.

Lemma 6.3. *The continuous Petrov–Galerkin method of polynomial degree q applied to the test equation* (6.12) *is equivalent to the q-stage Gauss–Legendre collocation method.*

Proof. A general proof for ordinary differential equations can be found in [Hul72]. Here we give a short version adapted to Dahlquists test equation (6.12) which shows how Petrov–Galerkin methods are related to time stepping methods. Due to the discontinuous test space the global variational formulation (6.13) decouples and can be solved iteratively on every time interval I_n locally. This results in the following problem.

For $n = 1, \dots, N$ find $u_{h,I_n} \in \mathbb{P}_q(I_n)$ with $u_{h,I_n}(t_{n-1}) = u_{h,I_{n-1}}(t_{n-1})$ such that

$$\int_{t_{n-1}}^{t_n} \partial_t u_{h,I_n} w_h - \alpha u_{h,I_n} w_h \,\mathrm{d}t = 0$$

is fulfilled for all $w_h \in \mathbb{P}_{q-1}(I_n)$.

Since the integrand is at most of polynomial degree $2q-1$, the integral can be replaced by a Gaussian quadrature, with q quadrature points c_k and weights ω_k where $k = 1, \dots, q$ (cf. [HB09, Ch. 40]). We obtain

$$\sum_{k=1}^{q} \omega_k \partial_t u_h(c_k) w_h(c_k) = \sum_{k=1}^{q} \omega_k \alpha u_h(c_k) w_h(c_k).$$

We now test with Lagrange polynomials (cf. Example A.1)

$$\lambda_i(t) = \Pi_{j=1, j\neq i}^{q} \frac{t - c_j}{c_i - c_j}$$

for $i = 1, \dots, q$. Hence it holds that $\lambda_i(c_j) = 0$ for $i \neq j$ and $\lambda_i(c_i) = 1$ for $i = 1, \dots, q$. We obtain $1 + q$ equations

$$\begin{aligned} u_{h,I_n}(t_{n-1}) &= u_{h,I_{n-1}}(t_{n-1}), \\ \partial_t u_{h,I_n}(c_i) &= \alpha u_{h,I_n}(c_i), \qquad \text{for } i = 1, \dots, q. \end{aligned} \tag{6.14}$$

This results in an equivalent problem, which is commonly known as *collocation method* (cf. [Ise09, Sec. 3.4]).

Find a polynomial u_{h,I_n} of degree q on I_n, such that the equations (6.14) *are fulfilled for all collocation parameters c_i.*

Thus the continuous Petrov–Galerkin method is equivalent to the Gauss–Legendre collocation method, with right-hand side $f = \alpha u_{h,I_n}$. □

Theorem 6.4. *(cf. [HW96, Thm. 5.2]) The stability function $R(z)$ for the q-stage Gauss–Legendre collocation method is given by the diagonal Padé approximation of the exponential function* exp. *Furthermore the method is A-stable, i.e., $|R(z)| < 1$ for all $z \in \mathbb{C}$ with negative real part.*

Remark 6.6. Due to Lemma 6.3 the q-stage Gauss–Legendre collocation method coincides with the continuous Petrov–Galerkin method of polynomial degree q for the test equation (6.12). Hence both methods own the same stability function. Due to the previous theorem, the continuous Petrov–Galerkin method is A-stable. The first three diagonal Padé approximations of the exponential function $\exp(z)$ are given as

$$P_{(1,1)}(z) = \frac{2+z}{2-z}, \quad P_{(2,2)}(z) = \frac{12+6z+z^2}{12-6z+z^2}, \quad P_{(3,3)}(z) = \frac{120+60z+12z^2+z^3}{120-60z+12z^2-z^3},$$

see, e.g., [Ise09, Sec. 4.2].

6.3.2 Block-Jacobi Smoother for a Test Equation

The variational formulation of Dahlquists test equation (6.13) leads to the following linear block-system

$$\underline{L}\underline{u} = \underline{b} \quad \Longleftrightarrow \quad \begin{pmatrix} \underline{D}_1 & & & \\ \underline{C}_1 & \underline{D}_2 & & \\ & \ddots & \ddots & \\ & & \underline{C}_{N-1} & \underline{D}_N \end{pmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \vdots \\ \underline{u}_N \end{pmatrix} = \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \vdots \\ \underline{b}_N \end{pmatrix},$$

with $\underline{u}_n, \underline{b}_n \in \mathbb{R}^q$ and blocks $\underline{C}_n, \underline{D}_n \in \mathbb{R}^{q\times q}$. The entries can be again computed as

$$\begin{aligned} (\underline{C}_{n-1})_{i,j} &= \int_{t_{n-1}}^{t_n} (\partial_t - \alpha) \left(\frac{t_n - t}{t_n - t_{n-1}} \lambda_j^{n-1}(t_{n-1}) \right) \lambda_i^n(t) \,\mathrm{d}t, \\ (\underline{D}_n)_{i,j} &= \int_{t_{n-1}}^{t_n} (\partial_t - \alpha) \left(\frac{t - t_{n-1}}{t_n - t_{n-1}} \lambda_j^n(t) \right) \lambda_i^n(t) \,\mathrm{d}t, \end{aligned}$$

for $i, j = 1, \ldots, q$ and nodal shape functions $\lambda_1^n, \ldots, \lambda_q^n \in \mathbb{P}_{q-1}(I_n)$ for $n = 1, \ldots, N$ (cf. Section 4.6). Due to the nodal structure (cf. Lemma 4.8) we can assume that the node t_{n-1} is associated with the q-th degree of freedom. Hence, it hold that

$$(\underline{C}_n)_{i,j} = 0, \qquad \text{for all } i = 1, \ldots, q \text{ and } j = 1, \ldots, q-1, \tag{6.15}$$

i.e., except on the last column, all entries of $\underline{C}_n$ are equal to zero. As described in Remark 3.2 the initial condition is again incorporated into the right hand side by computing

$$(\underline{b}_n)_i = \int_{t_{n-1}}^{t^n} \alpha u_0 \lambda_i^n(t) \,\mathrm{d}t.$$

We now apply the block-Jacobi smoother defined in Section 6.2.2, where a block corresponds to one time interval I_n and achieve the iteration matrix

$$\underline{S}^{\mathrm{J}} = \underline{\mathrm{Id}}_{Nq} - \theta^{\mathrm{J}} \mathrm{block\_diag}_N(\underline{L})^{-1} \underline{L} \tag{6.16}$$

$$= \begin{pmatrix} (1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q & & & \\ \underline{E}_1 & (1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q & & \\ & \ddots & \ddots & \\ & & \underline{E}_{N-1} & (1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q \end{pmatrix}, \tag{6.17}$$

with vectors $\underline{E}_n = -\theta^{\mathrm{J}} \underline{D}_{n+1}^{-1} \underline{C}_n \in \mathbb{R}^{q\times q}$ for $n = 1, \ldots, N-1$. We now analyze the smoother by investigating, how the error coefficients $\underline{e}^{(i)} = \underline{u}^{(i)} - \underline{u}$ on one time interval are damped after one Jacobi iteration. To do so, we use the representation

$$\underline{e}^{(i)} = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} z_l^{(i)} \underline{\phi}_l, \qquad \text{for coefficients } z_l^{(i)} \in \mathbb{C} \tag{6.18}$$

of the error $\underline{e}^{(i)}$ with respect to the Fourier basis $\underline{\phi}_l \in \mathbb{C}^{Nq}$ which is given componentwise as

$$\left(\underline{\phi}_l\right)_j = \exp\left(\mathrm{i}\frac{j}{q}\frac{2\pi l}{N}\right), \quad \text{for} \quad j = 0, \dots, Nq \quad \text{and} \quad l = -\frac{N}{2}, \dots, \frac{N}{2}.$$

The value $\frac{2\pi l}{N}$ is called *frequency* (see, e.g., [HB09, Ch. IX]). Within one separated time interval I_n, for $n = 2, \dots, N$, the local iteration is given as

$$\underline{e}_n^{(i+1)} = \underline{E}_{n-1}\underline{e}_{n-1}^{(i)} + (1-\theta^{\mathrm{J}})\underline{e}_n^{(i)}, \tag{6.19}$$

where $\underline{e}_n^{(i)} \in \mathbb{R}^q$ are the components of $\underline{e}^{(i)}$ associated with I_n. Correspondingly to (6.18) we state the local representation

$$\underline{e}_n^{(i)} = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} z_l^{(i)}\underline{\phi}_{l,n}, \tag{6.20}$$

where the components of the local basis functions read as

$$\left(\underline{\phi}_{l,n}\right)_j = \exp\left(\mathrm{i}\Big(nq+j\Big)\frac{2\pi l}{Nq} - \mathrm{i}\frac{2\pi l}{N}\right), \qquad \text{for } j = 1, \dots, q.$$

A simple calculation shows that $\underline{\phi}_{l,n-1} = \mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}\underline{\phi}_{l,n}$. Inserting (6.20) into the local iteration scheme (6.19) and using (6.15) yields

$$\begin{aligned}
&(1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q\,\underline{\phi}_{l,n} + \underline{E}_{n-1}\underline{\phi}_{l,n-1} \\
=&(1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q\,\underline{\phi}_{l,n} + \underline{E}_{n-1}\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}\underline{\phi}_{l,n} \\
=&\begin{pmatrix}
(1-\theta^{\mathrm{J}}) & & & \mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}\ (\underline{E}_{n-1})_{1,q} \\
& \ddots & & \vdots \\
& & (1-\theta^{\mathrm{J}}) & \mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}\ (\underline{E}_{n-1})_{q-1,q} \\
& & & (1-\theta^{\mathrm{J}}) + \mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}\ (\underline{E}_{n-1})_{q,q}
\end{pmatrix}\underline{\phi}_{l,n} \\
=&\ \widetilde{\underline{S}}_{l,n}\,\underline{\phi}_{l,n},
\end{aligned}$$

with local iteration matrices $\widetilde{\underline{S}}_{l,n} \in \mathbb{R}^{q\times q}$, for $l = -\frac{N}{2}, \dots, \frac{N}{2}$ and $n = 2, \dots, N$. For $n = 1$ we immediately conclude form (6.16) that $\widetilde{\underline{S}}_{l,1} = (1-\theta^{\mathrm{J}})\underline{\mathrm{Id}}_q$. Thus the local iteration (6.19) reads as

$$\underline{e}_n^{(i+1)} = \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} z_l^{(i)}\widetilde{\underline{S}}_{l,n}\,\underline{\phi}_{l,n}, \qquad \text{for } n = 1, \dots, N.$$

Note that $\widetilde{\underline{S}}_{l,n}$ depends on the frequency $\frac{2\pi l}{N}$ and therefore also the efficiency of the block-Jacobi smoother. To analyze the smoothing behavior we compute the spectral radius of the iteration matrix $\widetilde{\underline{S}}_{l,n}$ in the following lemma.

Lemma 6.4. *The local iteration matrix $\underline{\widetilde{S}}_{l,n}$ has the two different eigenvalues $1-\theta^{\mathrm{J}}$ and $1-\theta^{\mathrm{J}}+\theta^{\mathrm{J}}\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}R\left(\alpha(t_n-t_{n-1})\right)$ and hence the corresponding spectral radius with respect to θ^{J} is given as*

$$\begin{aligned}\rho(\underline{\widetilde{S}}_{l,1},\theta^{\mathrm{J}}) &= \left|1-\theta^{\mathrm{J}}\right|,\\ \rho(\underline{\widetilde{S}}_{l,n},\theta^{\mathrm{J}}) &= \max\left\{\left|1-\theta^{\mathrm{J}}\right|,\left|1-\theta^{\mathrm{J}}+\theta^{\mathrm{J}}\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}R\left(\alpha(t_n-t_{n-1})\right)\right|\right\},\end{aligned}$$

where $n=2,\dots,N$ and R is the stability function introduced in Theorem 6.4.

Proof. For $n=1$ the assertion is trivial, since $\underline{\widetilde{S}}_{l,1}$ is a diagonal matrix. In case of $n>1$, the entry $(\underline{E}_{n-1})_{q,q}$ can be computed as $(\underline{E}_{n-1})_{q,q}=\theta^{\mathrm{J}}R\left(\alpha\triangle t\right)$ (see proof of [Neu13, Thm. 4.2.17]). Here it is used that the Petrov–Galerkin method inherits the known stability function from a time stepping scheme. Furthermore all unit vectors $\mathbf{i}_j\in\mathbb{R}^q$ for $j=1,\dots,q-1$ are eigenvectors with corresponding eigenvalues $1-\theta^{\mathrm{J}}$. From the last column of $\underline{\widetilde{S}}_{l,n}$ we conclude, that the components of the last remaining eigenvector are given as

$$\frac{(\underline{E}_{n-1})_{i,q}}{(\underline{E}_{n-1})_{q,q}},\qquad \text{for } i=1,\dots,q$$

with corresponding eigenvalue $1-\theta^{\mathrm{J}}+\theta^{\mathrm{J}}R\left(\alpha\triangle t\right)\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}$. □

Next, we want to determine an optimal damping parameter θ^{J}_* such that the spectral radius gets as small as possible for high frequencies $(\frac{\pi}{2},\pi]$, i.e., $l\in(\frac{N}{4},\frac{N}{2}]$. For this purpose we assume that the step size $\triangle t=t_n-t_{n-1}$ is small such that $|\alpha\triangle t|\ll 1$. Then it holds that $R\left(\alpha\triangle t\right)\approx\exp\left(\alpha\triangle t\right)\approx 1$ and we can apply the following lemma.

Lemma 6.5. *For $R\left(\alpha\triangle t\right)=1$ the optimal damping parameter for high frequencies can be computed as*

$$\theta^{\mathrm{J}}_*=\frac{1}{2}$$

and the spectral radius $\rho(\underline{\widetilde{S}}_{l,n},\theta^{\mathrm{J}}_*)$ is bounded by the constant $\frac{1}{\sqrt{2}}<1$ independent of the step size.

Proof. (cf. [Neu13, Lem. 4.2.24]). Suppose that $l\in\left(\frac{N}{4},\frac{N}{2}\right]$, then it holds that

$$\begin{aligned}&\left|1-\theta^{\mathrm{J}}+\theta^{\mathrm{J}}\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}}R\left(\alpha\triangle t\right)\right|^2\\ &=\left(\theta^{\mathrm{J}}-1\right)^2+2\theta^{\mathrm{J}}\left(\theta^{\mathrm{J}}-1\right)\cos\left(\frac{2\pi l}{N}\right)+\left(\theta^{\mathrm{J}}\right)^2\\ &\le\left(\theta^{\mathrm{J}}-1\right)^2+\left(\theta^{\mathrm{J}}\right)^2\end{aligned}$$

and thus $\rho(\widetilde{\underline{S}}_{l,n}, \theta^{\mathrm{J}})$ attains its infimum at $\theta^{\mathrm{J}} = \theta^{\mathrm{J}}_* = \frac{1}{2}$. This yields

$$\begin{aligned}\rho(\widetilde{\underline{S}}_{l,n}, \theta^{\mathrm{J}}_*) \leq & \max\left\{\left|1-\theta^{\mathrm{J}}_*\right|, \sqrt{\left(\theta^{\mathrm{J}}_*-1\right)^2+\left(\theta^{\mathrm{J}}_*\right)^2}\right\} \\ \leq & \max\left\{\frac{1}{2}, \frac{\sqrt{2}}{2}\right\} \leq \frac{1}{\sqrt{2}}.\end{aligned}$$

□

Remark 6.7. Note that $\theta^{\mathrm{J}}_* = \frac{1}{2}$ holds only for $|\alpha\triangle t| \ll 1$. In the worst case, where $\mathrm{Re}\left(R\left(\alpha\triangle t\right)\right) = 0$ and $\mathrm{Im}\left(R\left(\alpha\triangle t\right)\right) \geq 1$ and $l = \frac{N}{4}$, it holds that

$$\left|\mathrm{e}^{-\mathrm{i}\frac{2\pi l}{N}} R\left(\alpha\triangle t\right)\right| \geq 1,$$

and thus

$$\rho(\widetilde{\underline{S}}_{l,n}, \theta^{\mathrm{J}}) = \Big(\mathrm{Im}\left(R\left(\alpha\triangle t\right)\right) - 1\Big)\theta^{\mathrm{J}} + 1 \geq 1.$$

This means that high frequencies are not damped at all, since the iteration is not contractive.

6.3.3 Possible Transfer to the Space-Time Setting

The presented investigation is far from being complete. We only investigated the smoothing behavior of high oscillating errors and neglected the coarse grid correction. Moreover, transferring the block-Jacobi smoother from the test equation to the space-time setting would result in a structure, where every block corresponds to a complete time slice. The required inversion would be much to costly. Thus, it is not clear how a reduced block-size influences the smoothing behavior.

However, in context of space-time multigrid methods α can be understood as the eigenvalues of the discrete discontinuous operator A_h in space. Hence $\alpha\triangle t \approx z\frac{\triangle t}{\triangle \mathbf{x}}$ where $z \in \mathbb{C}$ is a mesh independent constant. From this we infer that the two-grid cycle in time may converges as long as $\triangle t_k/\triangle \mathbf{x}_l$ and $\triangle t_{k-1}/\triangle \mathbf{x}_l$ are small enough for $l = 1, \ldots, l_{\max}$ and $k = 1, \ldots, k_{\max}$, i.e., the assumptions of Lemma 6.5 are approximately fulfilled. Moreover, it seems that $\theta^{\mathrm{J}} = 0.5$ might be a suitable damping parameter. Despite the discussed obscurities we observe a noticeable accordance of the smoothing results for the test equation and the numerical experiments achieved in the following section for the space-time setting.

A similar investigation of the smoothing behavior in space is more difficult, since the spectral radius of the iteration matrix is not known in case of the DG-FEM method. For that reason, a guess for an optimal damping parameter is computed numerically in the next section.

6.4 Testing different Multilevel Settings

In this section a multigrid preconditioner is developed by using the results from the previous section and numerical test cases. In a first numerical test we will consider the linear transport equation together with a two-grid cycle in time equipped with a block-Jacobi smoother. We will notice that the theoretical results of Section 6.3 for Dahlquists test equation also apply in this test case. Afterwards we test a two-grid cycle in space equipped with a block-Gauss–Seidel smoother and different smoothing factors $\theta^{\mathrm{GS}} \in (0, 1]$ to determine an optimal setting. The results are then used to set up a multilevel strategy. Finally, this strategy is applied to a two dimensional Maxwell test case, too. Numerical results for the multilevel strategy in the adaptive case are given in Chapter 8. All numerical tests in this section where performed on 16 processes on a node of the MA-PDE cluster (cf. A.3).

6.4.1 Linear Transport Equation

We consider the discrete linear transport problem with fixed polynomial degrees $p \equiv q \equiv 2$ (see Section 3.3) with a divergence free vector field $\mathbf{q}(\mathbf{x}) = 2\pi(-x_2, x_1)^\top$ on the space-time domain $Q = (-10, 10)^2 \times (0, 1)$. Furthermore, we choose a homogeneous right-hand side $f = 0$, constant density $\rho = 1$ and a Gaussian pulse

$$u_0(\mathbf{x}) = \exp\Big(-1.4((x_1 - 5)^2 + x_2^2)\Big) \tag{6.21}$$

as initial condition.

We first test the two-grid cycle in time together with a block-Jacobi smoother where we perform $\nu_{l,k}^{\mathrm{pre,J}} = \nu_{l,k}^{\mathrm{post,J}} = 2$ smoothing steps and use the optimal damping parameter $\theta_{\mathrm{J}} = 0.5$ derived in Lemma 6.5. We use different fine and coarse grids $\mathcal{R}_{l,k}$ and $\mathcal{R}_{l-1,k}$, respectively. The corresponding degrees of freedom in space and time are given in Table 6.1. The required number of iteration steps $n_{l,k}$ are given

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$l = 1$	768×16	768×32	768×64	768×128	768×256	768×512
$l = 2$	$3\,072 \times 16$	$3\,072 \times 32$	$3\,072 \times 64$	$3\,072 \times 128$	$3\,072 \times 256$	$3\,072 \times 512$
$l = 3$	$12\,288 \times 16$	$12\,288 \times 32$	$12\,288 \times 64$	$12\,288 \times 128$	$12\,288 \times 256$	$12\,288 \times 512$
$l = 4$	$49\,152 \times 16$	$49\,152 \times 32$	$49\,152 \times 64$	$49\,152 \times 128$	$49\,152 \times 256$	$49\,152 \times 512$

Table 6.1: Degrees of freedom on a space-time mesh $\mathcal{R}_{l,k}$ with polynomial degrees $p \equiv q \equiv 2$ where $\mathcal{R}_{0,0}$ consists of $128 = 16 \times 8$ space-time cells.

in Table 6.2. One notes that for $k - l \geq 2$ the number of iteration steps is always bounded by ten. This corresponds to the observation, made in Remark 6.7 and Section 6.3.3 for the test equation, where the two-grid cycle in time leads to good convergence results only if $\triangle t_k / \triangle \mathbf{x}_l$ and $\triangle t_{k-1} / \triangle \mathbf{x}_l$ are small enough. Furthermore,

the contraction rates of the two-grid cycle can be estimated by computing the average convergence rate

$$d_{l,k} = \left(\frac{\|\underline{b}_{l,k} - \underline{L}_{l,k}\underline{u}^{(n_{l,k})}\|_2}{\|\underline{b}_{l,k} - \underline{L}_{l,k}\underline{u}^{(0)}\|_2} \right)^{1/n_{l,k}} = \left(\frac{r^{(n_{l,k})}}{r^{(0)}} \right)^{1/n_{l,k}}$$

of the preconditioned linear system

$$\underline{u}^{(i+1)} = \underline{u}^{(i)} + \underline{B}_{l,k}(\underline{b}_{l,k} - \underline{L}_{l,k}\underline{u}^{(i)}), \qquad \underline{u}^0 = 0,$$

where $\underline{B}_{l,k}$ is the corresponding precondition matrix for the two-grid cycle in space or time, respectively. The computed averaged rates $d_{l,k}$ and iteration steps $n_{l,k}$ are given in Table 6.2.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$l=1$	27 (4.91e-1)	25 (4.63e-1)	10 (1.43e-1)	7 (6.12e-2)	7 (6.08e-2)	7 (6.08e-2)
$l=2$	34 (5.69e-1)	38 (6.09e-1)	33 (5.63e-1)	7 (6.90e-2)	7 (5.81e-2)	7 (5.83e-2)
$l=3$	45 (6.56e-1)	50 (6.85e-1)	57 (7.19e-1)	49 (6.81e-1)	6 (4.53e-2)	6 (4.32e-2)
$l=4$	65 (7.52e-1)	68 (7.58e-1)	79 (7.86e-1)	106 (8.40e-1)	94 (8.19e-1)	6 (3.54e-2)

Table 6.2: Iteration steps $n_{l,k}$ and averaged rates $d_{l,k}$ for a residual reduction by the factor 10^{-8} of the linear iteration with two-grid cycle preconditioner in time (Smoother: block-Jacobi [$\nu_{l,k} = 2$, $\theta_{l,k} = 0.5$]) and uniform polynomial degrees $(p, q) = (2, 2)$ on every $\mathcal{R}_{l,k}$.

On the other hand block-Gauss–Seidel smoothing with $\nu_{l,k}^{\text{pre,GS}} = \nu_{l,k}^{\text{post,GS}} = 5$ and no damping, i.e., $\theta_{\text{GS}} = 1$, is a suitable choice for the two-grid cycle in space. See Figure 6.10, where we tested the two-grid cycle on several space-time refinements $\mathcal{R}_{l,k}$ with different damping factors $\theta_{\text{GS}} \in (0, 1]$. The results for a test with the two-grid cycle in space on different space-time meshes are given in Table 6.3. We notice that for coarsening in space the number of iteration steps is only bounded in the time level k, but not in the space level l. However, the increase of the steps is small enough to achieve a benefit by using a multilevel method. Another fact is that for higher spatial dimensions $D > 1$ the coarse grid correction in space is cheaper than in time, since coarsening in time reduces the effort by the factor 0.5 whereas coarsening in space decreases the effort by a factor $(2D)^{-1} < 1/2$.

As a conclusion we can state from the numerical tests and Remark 6.7 that coarsening in time is stable as long as $\triangle t_k / \triangle \mathbf{x}_l \lesssim 10^{-2}$ is small enough, i.e., $k - l \geq 2$. Furthermore, coarsening in space is efficient, if the increase of iteration steps is moderate, e.g., logarithmic. This observations motivate a strategy for the space-time multigrid solver, where we at first only coarse in space until the lowest spatial level $k = 0$ is reached. Afterwards we coarse in time up to a lowest temporal level $l = 0$, where $\triangle t_0 / \triangle \mathbf{x}_0$ is barely small enough. On this level we solve the coarse problem exact. The full multilevel V-cycle is illustrated in Figure 6.11 and Algorithm 6.3.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$l=1$	4 (1.31e-4)	4 (1.36e-4)	4 (1.85e-4)	4 (1.79e-4)	4 (1.68e-4)	4 (1.68e-4)
$l=2$	5 (1.35e-2)	5 (5.50e-3)	5 (4.72e-3)	5 (5.50e-3)	5 (5.28e-3)	5 (4.94e-3)
$l=3$	8 (7.57e-2)	7 (3.63e-2)	7 (2.26e-2)	6 (4.24e-2)	6 (4.07e-2)	6 (3.89e-2)
$l=4$	15 (2.70e-1)	11 (1.61e-1)	10 (1.34e-1)	9 (1.27e-1)	9 (1.24e-1)	9 (1.20e-1)

Table 6.3: Iteration steps and averaged rates for a residual reduction by the factor 10^{-8} of the linear iteration with two-grid cycle preconditioner in space (Smoother: block-Gauss–Seidel [$\nu_{l,k} = 5$]) and uniform polynomial degrees $(p, q) = (2, 2)$ on every $\mathcal{R}_{l,k}$.

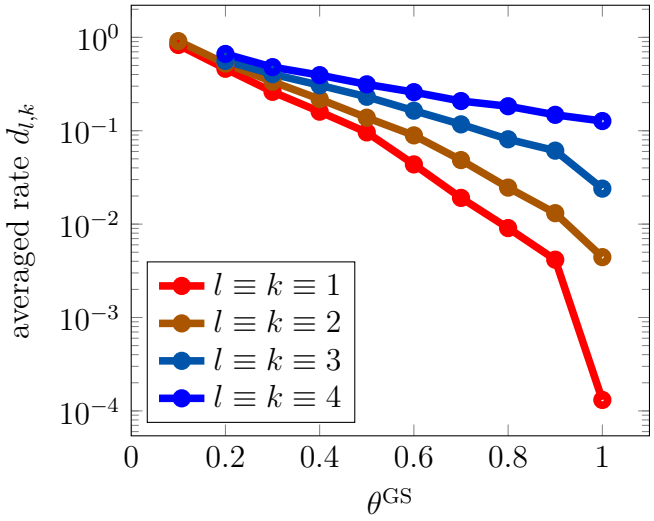

Figure 6.10: Averaged convergence rates $d_{l,k}$ for different damping factors $\theta^{\mathrm{GS}} \in (0, 1]$ on different space-time meshes $\mathcal{R}_{l,k}$ and uniform polynomial degrees $(p, q) = (2, 2)$.

The results for this strategy applied to the test problem are given in Table 6.4. Due to the discussed problems above, we observe a moderate increase in the number of iteration steps, when refining in space.

Algorithm 6.3 Full multilevel preconditioner

function $\underline{x}_{l,k} = \text{PRECONDITION}(l, k, r_{l,k})$
  if $l == k == 0$ **then**
    $\underline{x}_{l,k} = \text{SOLVE\_EXACT}(\underline{A}_{l,k}, \underline{r}_{l,k})$
    return $\underline{x}_{l,k}$
  if $l > 0$ **then**
    $\underline{B}^{\text{SM}}_{l,k} = \underline{B}^{\text{GS}}_{l,k}$
  else
    $\underline{B}^{\text{SM}}_{l,k} = \underline{B}^{\text{J}}_{l,k}$
  (Pre-) smoothing:
  for $\nu = 1, \ldots, \nu^{\text{pre}}_{l,k}$ **do**
    $\underline{w}_{l,k} = \underline{B}^{\text{SM}}_{l,k} \underline{r}_{l,k}$
    $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k}\underline{w}_{l,k}$
  Coarse grid correction in space if possible, otherwise correction in time:
  if $l > 0$ **then**
    $\underline{d}_{l-1,k} = \underline{R}^{l-1,k}_{l,k} \underline{r}_{l,k}$
    $\underline{c}_{l-1,k} = \text{PRECONDITION}(l-1, k, \underline{r}_{l-1,k})$
    $\underline{w}_{l,k} = \underline{P}^{l,k}_{l-1,k} \underline{c}_{l-1,k}$
  else
    $\underline{d}_{l,k-1} = \underline{R}^{l,k-1}_{l,k} \underline{r}_{l,k}$
    $\underline{c}_{l,k-1} = \text{PRECONDITION}(l, k-1, \underline{r}_{l,k-1})$
    $\underline{w}_{l,k} = \underline{P}^{l,k}_{l,k-1} \underline{c}_{l,k-1}$
  Correction:
    $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k}\underline{w}_{l,k}$
  (Post-) smoothing:
  for $\nu = 1, \ldots, \nu^{\text{post}}_{l,k}$ **do**
    $\underline{w}_{l,k} = \underline{B}^{\text{SM}}_{l,k} \underline{r}_{l,k}$
    $\underline{x}_{l,k} = \underline{x}_{l,k} + \underline{w}_{l,k}$ and $\underline{r}_{l,k} = \underline{r}_{l,k} - \underline{L}_{l,k}\underline{w}_{l,k}$
  return $\underline{x}_{l,k}$

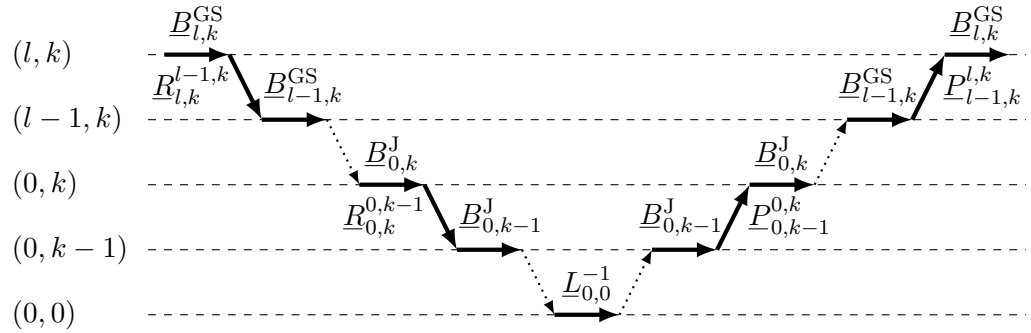

Figure 6.11: Full space-time multilevel strategy in terms of preconditioning matrices: first coarsening in space up to level $(0,k)$, then coarsening in time up to level $(0,0)$. Finally, solve exact on the coarsest level and prolongate back.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$l=1$	4 (1.25e-4)	4 (1.26e-4)	4 (1.80e-4)	4 (1.92e-4)	4 (1.86e-4)	4 (1.75e-4)
$l=2$	5 (1.35e-2)	5 (5.50e-3)	5 (4.71e-3)	5 (5.50e-3)	5 (5.28e-3)	5 (4.94e-3)
$l=3$	8 (7.57e-2)	7 (3.63e-2)	7 (2.25e-2)	6 (4.24e-2)	6 (4.07e-2)	6 (3.89e-2)
$l=4$	15 (2.73e-1)	11 (1.61e-1)	10 (1.34e-1)	9 (1.27e-1)	9 (1.24e-1)	9 (1.20e-1)

Table 6.4: Iteration steps and averaged rates for a full space-time multilevel method for the transport problem. Smoother: Jacobi ($\nu_{k,l}=2$, $\theta_{l,k}=0.5$) in time, Gauss–Seidel ($\nu_{l,k}=5$) in space.

6.4.2 Maxwell's Equations in 2D

We now apply the same multilevel strategy to the Maxwell case, where we take a two-dimensional transversal magnetic test problem in $Q = (0,1)^2 \times (0,1)$ and a homogeneous right-hand side. The initial and boundary conditions are given by the exact solution

$$\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} H_1(\mathbf{x},t) \\ H_2(\mathbf{x},t) \\ E_3(\mathbf{x},t) \end{pmatrix} = \begin{pmatrix} 0 \\ -\sin\left((x_1 - t)\,\pi\right) \\ \sin\left((x_1 - t)\,\pi\right) \end{pmatrix}.$$

The results are given in Table 6.5, where we observe the same behavior as in the linear transport case. Since hyperbolic problems do not fulfill the same regularity

l	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
	degrees of freedom on the space-time mesh $\mathcal{R}_{l,k}$ with polynomial degrees $p \equiv q \equiv 2$					
1	$1\,152 \times 16$	$1\,152 \times 32$	$1\,152 \times 64$	$1\,152 \times 128$	$1\,152 \times 256$	$1\,152 \times 512$
2	$4\,608 \times 16$	$4\,608 \times 32$	$4\,608 \times 64$	$4\,608 \times 128$	$4\,608 \times 256$	$4\,608 \times 512$
3	$18\,432 \times 16$	$18\,432 \times 32$	$18\,432 \times 64$	$18\,432 \times 128$	$18\,432 \times 256$	$18\,432 \times 512$
4	$73\,728 \times 16$	$73\,728 \times 32$	$73\,728 \times 64$	$73\,728 \times 128$	$73\,728 \times 256$	$73\,728 \times 512$
	multilevel iterations in space and time					
1	4 (3.42e-3)	4 (3.93e-3)	4 (4.03e-3)	4 (3.91e-3)	4 (3.70e-3)	4 (3.44e-3)
2	6 (3.18e-2)	6 (3.24e-2)	6 (3.13e-2)	6 (3.00e-2)	6 (2.85e-2)	6 (2.71e-2)
3	10 (1.31e-1)	10 (1.35e-1)	10 (1.31e-1)	10 (1.28e-1)	9 (1.59e-1)	9 (1.53e-1)
4	17 (3.62e-1)	17 (3.50e-1)	17 (3.44e-1)	17 (3.39e-1)	16 (3.68e-1)	16 (3.61e-1)

Table 6.5: Degrees of freedom of the Maxwell example in space-time domain on different space-time meshes $\mathcal{R}_{l,k}$, where $\mathcal{R}_{0,0}$ is decomposed into $64 = 8 \times 8$ space-time cells. Iteration steps and averaged rates for a full space-time multilevel method for the Maxwell example. Smoother: block-Jacobi ($\nu_{l,k} = 2$, $\theta_{l,k} = 0.5$) in time, block-Gauss–Seidel ($\nu_{l,k} = 5$) in space.

assumptions and properties as elliptic problems, applying the techniques and results mentioned in Section 6.2.6 is difficult. As long as corresponding smoothing and approximation properties (cf. Definition 6.6) do not hold, we cannot assume stable multilevel behavior, i.e., bounded iteration numbers, in our case. In the numerical tests (cf. Tables 6.4 and 6.5), we actually observe a logarithmic dependency of iteration numbers on the problem size. However, the numerical effort is still lower compared to other iterative methods, e.g., Krylov–subspace methods, where the iteration numbers grow at least linearly ([Mei11, Sec. 4.3]). As a result, we apply the presented multilevel strategy as a preconditioner to a suitable iterative Krylov–subspace method for our numerical experiments.

In the adaptive case a coarse cell may correspond to a set of space-time cells with different associated polynomial degrees. In order to prevent that the convergence

rate is disturbed, we have to choose a suitable polynomial distribution on the subspaces $V_{l-1,k}$ or $V_{l,k-1}$ and correspondingly on $W_{l-1,k}$ and $W_{l,k-1}$. In this case we proceed as stated in Section 6.2.4. Finally we can apply the restriction and prolongation matrices of the uniform refined case. The convergence behavior of the multilevel preconditioner in the adaptive case is investigated in the last chapter.

7 Implementation

In this work the parallel finite element library M++ was used to implement the presented space-time method (cf. [Wie10]). Additionally to common features of a finite element library, e.g., input-output functions, linear algebra routines, spatial conforming finite elements, time-stepping algorithms, parallel direct and parallel iterative solvers and preconditioners, M++ contains a framework for DG-FEM as well (cf. [Tha15]). Nevertheless, several major changes and extensions were necessary to implement a suitable p-adaptive space-time setting. In the following we give a short overview of the main contributions to the M++ library implemented for this work. This may serves as a basic guideline for the reader to implement parallel and adaptive space-time methods in other finite element libraries.

7.1 Generating a Space-Time Mesh

An admissible spatial mesh $(\mathcal{K}, \mathcal{V}, \mathcal{F})$ in M++ consists of a set of cells $\mathcal{K}$, vertices $\mathcal{V}$ and faces $\mathcal{F}$ (see [Wie10, Sec. 2]). Every geometric object, e.g., a cell $K \in \mathcal{K}$ or face $f \in \mathcal{F}$, is stored in a distributed *hash map*, where its geometric midpoint is used as a *hash key*. Hash maps or *hash tables* are data structures for associative arrays (maps), where hash functions are used to map identifying values, i.e., (hash) keys, to their associative data. These kind of maps are commonly used in computer science since the average time for searching, inserting and deleting operations is almost independent of the map size. Due to this data structure a cell $K \in \mathcal{K}$ is defined uniquely by its midpoint $\mathbf{c}_K$ and corresponding vertices $\mathbf{x}_0, \mathbf{x}_1, \ldots \in \mathcal{V}$. Two neighboring cells $K, K_f \in \mathcal{K}$, $K \neq K_f$, are connected by their joint face $f = \partial K \cap \partial K_f \neq \emptyset$. A face is again defined by its midpoint $\mathbf{c}_f$ and the midpoints $(\mathbf{c}_K, \mathbf{c}_{K_f})$ of both neighboring cells. If $\mathbf{c}_{K_f}$ attains the value ∞ then the cell K is a boundary cell.

We have seen that the spatial mesh structure in M++ depends on the vertices and geometric midpoints. Hence the generation of a space-time mesh is done analogously by introducing coordinates with an additional time component. A space-time vertex

$\mathbf{x} \in \mathcal{V}$ is then defined as vector $\mathbf{x} = (x_1, x_2, \ldots, x_D, x_t)^\top$. We are now able to use the same data structure as before equipped with a slight modification. As discussed above the set of spatial cells $\mathcal{K}$ is replaced by the set of space-time cells $\mathcal{R}$. A single space-time cell $R \in \mathcal{R}$ consists of two underlying objects. A spatial cell $K \in \mathcal{K}$ and a time interval $I \in \mathcal{I}$, where $\mathcal{I}$ is the set of all time intervals $\mathcal{I}$. Hence the resulting space-time mesh is given as $(\mathcal{R}, \mathcal{K}, \mathcal{I}, \mathcal{V}, \mathcal{F})$. To construct a tensor product mesh we first read in the spatial mesh $\mathcal{K}$ and a set of time intervals $\mathcal{I}$. Then we construct space-time cells R and use a routine which inserts the cell into the mesh $(\mathcal{R}, \mathcal{K}, \mathcal{I}, \mathcal{V}, \mathcal{F})$ by extracting vertices and faces from R and filling up the hash maps $\mathcal{V}$ and $\mathcal{F}$, respectively. Afterwards the mesh is distributed to several processes as explained in the following section.

7.2 Cell Distribution and Load Balancing

In this section we introduce a simple distribution and load balancing algorithm, which uses the amount of degrees of freedom associated to a cell as a weight. To distribute a mesh $\mathcal{R}$ on $P \in \mathbb{N}$ processes we use a *recursive coordinate bisection* (RCB), see, e.g., [Wil91] and [MW14], method in space-time which is illustrated in the following. Every space-time cell $R \in \mathcal{R}$ has a unique geometric midpoint $\mathbf{c}_R \in \Omega \times (0, T)$ (see Figure 4.4). Hence we can use the space-time coordinate $\mathbf{c}_R = (c_{1,R}, \ldots, c_{D,R}, c_{t,R})^\top$ of the midpoints to sort the cells. For this purpose a strict order relation in space-time is needed. In every spatial direction and in time we have the strict one dimensional order relation $<$. This relation is known to be irreflexive, antisymmetric and transitive. Moreover we use this relation to define the following strict $D+1$ dimensional ordering relations.

Definition 7.1. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{D+1}$, $D \in \{1, 2, 3\}$. We define the following order relations in every spatial and the time direction:

$$\mathbf{a} <_t \mathbf{b} \quad \Longleftrightarrow \quad \begin{cases} a_t < b_t, & \\ a_1 < b_1, & \text{if } (a_t = b_t), \\ a_2 < b_2, & \text{if } (a_t = b_t) \wedge (a_1 = b_1), \\ a_3 < b_3, & \text{if } (a_t = b_t) \wedge (a_1 = b_1) \wedge (a_1 = b_2), \end{cases}$$

$$\mathbf{a} <_1 \mathbf{b} \quad \Longleftrightarrow \quad \begin{cases} a_1 < b_1, & \\ a_2 < b_2, & \text{if } (a_1 = b_1), \\ a_3 < b_3, & \text{if } (a_1 = b_1) \wedge (a_2 = b_2), \\ a_t < b_t, & \text{if } (a_1 = b_1) \wedge (a_2 = b_2) \wedge (a_3 = b_3), \end{cases}$$

$$\mathbf{a} <_2 \mathbf{b} \quad \Longleftrightarrow \quad \begin{cases} a_2 < b_2, & \\ a_3 < b_3, & \text{if } (a_2 = b_2), \\ a_1 < b_1, & \text{if } (a_2 = b_2) \wedge (a_3 = b_3), \\ a_t < b_t, & \text{if } (a_2 = b_2) \wedge (a_3 = b_3) \wedge (a_1 = b_1), \end{cases}$$

$$\mathbf{a} <_3 \mathbf{b} \quad \Longleftrightarrow \quad \begin{cases} a_3 < b_3, & \\ a_1 < b_1, & \text{if } (a_3 = b_3), \\ a_2 < b_2, & \text{if } (a_3 = b_3) \wedge (a_1 = b_1), \\ a_t < b_t, & \text{if } (a_3 = b_3) \wedge (a_1 = b_1) \wedge (a_2 = b_2). \end{cases}$$

This definition can be used to separate $\mathcal{R}$ into two sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with approximately the same cardinality

$$|\mathcal{R}_1| \approx |\mathcal{R}_2| \tag{7.1}$$

such that

$$\mathbf{c}_{R_1} <_{\text{rel}} \mathbf{c}_{R_2}, \quad \text{for all } R_1 \in \mathcal{R}_1 \text{ and } R_2 \in \mathcal{R}_2,$$

where $<_{\text{rel}} \in \{<_t, <_1, <_2, <_3\}$ is one of the relations defined above. Since both sets $\mathcal{R}_1$ and $\mathcal{R}_2$ can be separated again independent of each other. Thus, we can state a space-time RCB Algorithm 7.1 which generates 2^s disjoint sets $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \ldots \cup \mathcal{R}_{2^s} = \mathcal{R}$ after $s \geq 0$ recursive iterations. Every set $\mathcal{R}_\pi$ is now associated with one process $\pi \in \mathcal{P} = \{1, \ldots, P\}$. Note that this strategy is only efficient if the amount of processes is given as $P = 2^s$. Formally the RCB algorithm defines a mapping

$$\text{RCB}\colon \mathcal{R} \to \mathcal{P}$$

and we obtain the representation $\mathcal{R}_\pi = \{R \in \mathcal{R}\colon \text{RCB}(R) = \pi\}$ for $\pi \in \mathcal{P}$. The parallel implementation is discussed in the following remarks.

Remark 7.1 (Overlapping cells)**.** To be able to interchange informations between neighboring processes, we introduce an additional set of so called *ghost* or *overlapping* cells $\mathcal{O}$ (cf. [Tha15, Ch. 6]). In particular this is necessary to compute the numerical flux $\mathbf{F}^{\text{num}}$ and to ensure continuity in time in our space-time discontinuous Petrov–Galerkin discretization (cf. Section 4.5). Thus the actual mesh is given as $(\mathcal{R}, \mathcal{K}, \mathcal{I}, \mathcal{V}, \mathcal{F}, \mathcal{O})$. The construction of the overlap is done as follows. Let $\mathcal{V}_R$ be the set of vertices associated with the cell $R \in \mathcal{R}$. Then the set of vertices on one process $\pi \in \mathcal{P}$ is given as $\mathcal{V}_\pi = \bigcup_{R \in \mathcal{R}_\pi} \mathcal{V}_R$. This defines a decomposition

Algorithm 7.1 RCB load balancing strategy

1: **function** $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{2^s}\} = \text{RCB\_BALANCE}(\mathcal{R}, \text{rel}, s)$
2: **if** s=0 **then return** R
3: split $\mathcal{R}$ into $\mathcal{R}_1$ and $\mathcal{R}_2$ such that for all $R_1 \in \mathcal{R}_1$, $R_2 \in \mathcal{R}_2$:
4: $\mathbf{c}_{R_1} <_{\text{rel}} \mathbf{c}_{R_2}$ and $|\mathcal{R}_1| \approx |\mathcal{R}_2|$
5: **if** rel $== t$ **then**
6: **return** {RCB_BALANCE$(R_1, 1, s-1)$,RCB_BALANCE$(R_2, 1, s-1)$}
7: **if** $D == 1$ **then**
8: **return** {RCB_BALANCE$(R_1, t, s-1)$,RCB_BALANCE$(R_2, t, s-1)$}
9: **if** rel $== 1$ **then**
10: **return** {RCB_BALANCE$(R_1, 2, s-1)$,RCB_BALANCE$(R_2, 2, s-1)$}
11: **if** $D == 2$ **then**
12: **return** {RCB_BALANCE$(R_1, t, s-1)$,RCB_BALANCE$(R_2, t, s-1)$}
13: **if** rel $== 2$ **then**
14: **return** {RCB_BALANCE$(R_1, 3, s-1)$,RCB_BALANCE$(R_2, 3, s-1)$}
15: **if** rel $== 3$ **then**
16: **return** {RCB_BALANCE$(R_1, t, s-1)$,RCB_BALANCE$(R_2, t, s-1)$}

$\mathcal{V} = \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_P$ (not disjoint). Hence the set of overlapping cells associated with one process $\pi \in \mathcal{P}$ can be defined as

$$\mathcal{O}_\pi = \{R \in \mathcal{R} \setminus \mathcal{R}_\pi \colon \mathcal{V}_\pi \cap \mathcal{V}_R \neq \emptyset\}$$

with corresponding decomposition $\mathcal{O} = \mathcal{O}_1 \cup \ldots \cup \mathcal{O}_P$. The set $\mathcal{O}_\pi$ can be understood as copies of cells on other processes which share joint vertices. Note that this definition follows from the discontinuous Petrov–Galerkin discretization. In case of a discontinuous Galerkin discretization in space-time the amount of overlapping cells can be reduced, since continuity in time is not enforced. As a result

$$\mathcal{O}_\pi = \{R \in \mathcal{R} \setminus \mathcal{R}_\pi \colon \mathcal{F}_\pi \cap \mathcal{F}_R \neq \emptyset\}$$

becomes the set of cells on other processes which share joint faces, where $\mathcal{F}_\pi = \bigcup_{R \in \mathcal{R}_\pi} \mathcal{F}_R$ and $\mathcal{F} = \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_P$ are defined analogously.

Remark 7.2. Since every cell can have a different amount of corresponding degrees of freedom, Algorithm 7.1 will lead to an imbalanced distribution in the adaptive case. To prevent this, we replace (7.1) and the corresponding expression in Algorithm 7.1 (line 4) by the weighting

$$\sum_{R \in \mathcal{R}_1} n_R \approx \sum_{R \in \mathcal{R}_2} n_R,$$

where n_R is the amount of degrees of freedom assigned to the cell R.

7.3 Polynomial Adaptivity

To use different polynomial degrees on every cell $R \in \mathcal{R}$ in space and time we add an additional hash map $\mathcal{D}$ to the M++ DoF-handling class. We only have to store the polynomial degrees of the ansatz space V_h as tuples $(p_R, q_R) \in \mathbb{N}_0 \times \mathbb{N}$, since the degrees of the test space W_h are given as $(p_R, q_R - 1) \in \mathbb{N}_0 \times \mathbb{N}_0$. Again the geometric midpoint of R is used as hash key. The distribution to different processes is implemented as a decomposition

$$\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_P, \quad \text{where } \mathcal{D}_\pi = \{(p_R, q_R) \colon R \in \mathcal{R}_\pi \cup \mathcal{O}_\pi\}\,.$$

After the initialization with homogeneous degrees, e.g., $(p_R, q_R) = (0, 1)$, it is possible to manipulate the degrees $(p_R, q_R) \in \mathcal{D}_\pi$, for $R \in \mathcal{R}_\pi$, on every process $\pi \in \mathcal{P}$ independently. Note that every change of polynomial degrees requires the call of a communication routine, which is necessary to keep informations consistent on overlapping cells $\mathcal{O}_\pi$ associated with other processes. This is done analogously to the linear algebra class, where an accumulation method is called to keep matrix and vector entries consistent on overlapping cells (cf. [Wie10], [Tha15, Sec. 6.3]). Thus the set

$$\{(p_R, q_R) \in \mathcal{D}_\pi \colon R \in \mathcal{O}_\pi\} \subset \mathcal{D}_\pi$$

is updated by requesting the corresponding data from neighboring processes. This two-stage procedure leads to consistent results since $\mathcal{R}_\pi \cap \mathcal{O}_\pi = \emptyset$, for all $\pi \in \mathcal{P}$.

8 Numerical Experiments

8.1 Higher-Order Space-Time Methods

To confirm the theoretical results of [HW08], [MS11] and Theorem 4.2, we consider test case examples, where the solution $\mathbf{u}$ is known and smooth enough. We compute the so called *experimental order of convergence* (EOC). The EOC can be determined by computing the slope of the error graph in a double-logarithmic plot towards the mesh size. Thus we can compute the EOC between two measuring points $(h_1, \|\mathbf{u} - \mathbf{u}_{h_1}\|)$ and $(h_2, \|\mathbf{u} - \mathbf{u}_{h_2}\|)$ as

$$\mathrm{EOC}_{12} = \frac{\log(\|\mathbf{u} - \mathbf{u}_{h_1}\|) - \log(\|\mathbf{u} - \mathbf{u}_{h_2}\|)}{\log(h_1) - \log(h_2)},$$

with respect to a given norm $\|\,.\,\|$. Here h_1 and h_2 are mesh sizes of two different corresponding meshes with $h_1 > h_2$. To evaluate integrals within the norms numerically one uses quadrature formulas of sufficiently high order. In particular one has to ensure that the quadrature error is smaller than the discretization error such that the EOC computation is not distributed.

We first investigate the weighted L_2-norm error $\|\mathbf{e}\|_W = \|\mathbf{u} - \mathbf{u}_h\|_W$ in the space-time domain. Since we are able to choose different polynomial degrees in space and time, we distinguish between convergence in space, time and space-time, respectively. For the spatial discontinuous Galerkin discretization we expect order $p+1$ (cf. [HW08, Ch. 2.2]) in space and order $q+1$ for the continuous Petrov–Galerkin method in time (cf. [Hul72] or [MS11]). Thus the order of our space-time method is expected to be as good as the lowest order in space or in time $\min_{p,q}\{p+1, q+1\}$. In a second test we focus on the error $\|\mathbf{e}\|_{V_h} = \|\mathbf{u} - \mathbf{u}_h\|_{V_h}$ towards the discrete norm $\|.\|_{V_h}$, which is defined in Definition 4.10 and used for stability and convergence analysis in Theorem 4.2.

In a first numerical experiment we investigate the convergence behavior of the Petrov–Galerkin method introduced in Chapter 4 in case of the linear transport example. The convergence in the electromagnetic wave case is investigated afterwards. In both numerical test cases we start with a very coarse mesh of 32×1

space-time cells. To determine the EOC in space, the mesh is refined several times in time such that $\triangle t\, c_{\max} < \triangle \mathbf{x}$ holds true for all used spatial discretizations $\triangle \mathbf{x}$. Here $c_{\max} > 0$ denotes an upper bound for the wave speed in Q for our examples, e.g.,

$$\|\mathbf{q}\|_{\mathrm{L}_\infty(Q)} \leq c_{\max} \qquad \text{and} \qquad \frac{1}{\sqrt{\varepsilon\mu}} = 1 \leq c_{\max}.$$

Furthermore, we use the same polynomial degrees in space and time. This ensures that the temporal error is always smaller than the error in space and hence does not affect the spatial convergence rate. Analogously, the experimental order of convergence in time is measured with interchanged roles of $\triangle \mathbf{x}$ and $\triangle t$. In case of the space-time convergence we again use $p = q$ and refine simultaneously in space and time. All convergence tests are performed on the MA-PDE Cluster (cf. A.3) with up to 128 processes, if possible.

8.1.1 Linear Transport Equation

We consider the exact solution

$$u(\mathbf{x}, t) = \sin\Big((x_1 + x_2 - 2t)\pi\Big)$$

of the linear advection equation (3.8) on a computational domain $Q = \Omega \times (0, T) = (-1, 1)^2 \times (0, 1)$. The parameters are given as $\rho \equiv 1$ and $f \equiv 0$ and $\mathbf{q} = [1, 1]^\top$. Furthermore, the initial and boundary conditions correspond to u and hence $u_0(\mathbf{x}) = u(\mathbf{x}, 0)$ and $u(\mathbf{x}, t)$ for all $(\mathbf{x}, t) \in \Gamma_{\text{in}} \times (0, T)$. We first investigate convergence with respect to the $\|.\|_W$-norm. Here the computed experimental orders of convergence for different polynomial degrees p and q in space, time, space-time are given in Tables 8.1, 8.2, 8.3 and in Figures 8.1, 8.2, 8.3, respectively. For the $\|.\|_{V_h}$-norm the computed experimental orders of convergence in space-time are given in Table 8.4 and Figure 8.4. In every test case, we observe that the EOCs coincide with the expected theoretical order of convergence.

	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC
$\triangle \mathbf{x}$	$p \equiv 1$		$p \equiv 2$		$p \equiv 3$		$p \equiv 4$	
2^{-0}	1.1203e-0	-	1.2162e-1	-	3.0638e-2	-	3.8215e-3	-
2^{-1}	2.8661e-1	1.97	2.2210e-2	2.45	1.9631e-3	3.96	1.5013e-4	4.67
2^{-2}	7.3722e-2	1.96	2.3687e-3	3.23	1.2710e-4	3.95	4.3990e-6	5.09
2^{-3}	1.8269e-2	2.01	2.7825e-4	3.09	7.8038e-6	4.03	1.3309e-7	5.05
2^{-4}	4.4996e-3	2.02	3.4908e-5	2.99	4.7861e-7	4.03	7.8580e-9	5.04
theoretical order		2.00		3.00		4.00		5.00

Table 8.1: Linear transport: Convergence of the error $e = u - u_h$ in space with respect to the $\|.\|_W$-norm for different polynomial degrees.

	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC
$\triangle t$	$q \equiv 1$		$q \equiv 2$		$q \equiv 3$		$q \equiv 4$	
2^{-0}	1.7626e-0	-	1.2287e-0	-	6.1151e-1	-	2.0909e-1	-
2^{-1}	1.3562e-0	0.51	3.5235e-1	1.80	5.3649e-2	3.51	7.3478e-3	4.83
2^{-2}	6.3385e-1	0.93	4.3308e-2	3.02	3.0953e-3	4.12	2.2716e-4	5.02
2^{-3}	1.9581e-1	1.59	4.5838e-3	3.24	1.9073e-4	4.02	7.0715e-6	5.01
2^{-4}	5.1644e-2	1.89	5.2628e-4	3.12	1.1924e-5	4.00	2.1925e-7	5.01
2^{-5}	1.3091e-2	1.97	6.2802e-5	3.07	7.3927e-7	4.01	6.8244e-9	5.01
theoretical order		2.00		3.00		4.00		5.00

Table 8.2: Linear transport: Convergence of the error $e = u - u_h$ in time with respect to the $\|.\|_W$-norm for different polynomial degrees.

	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC	$\|e\|_W$	EOC
$\triangle \mathbf{x}, \triangle t$	$p \equiv q \equiv 1$		$p \equiv q \equiv 2$		$p \equiv q \equiv 3$		$p \equiv q \equiv 4$	
2^{-1}	1.3029e-0	-	3.5435e-1	-	5.3395e-2	-	7.3365e-3	-
2^{-2}	6.3627e-1	1.03	4.3447e-2	3.03	3.0773e-3	4.12	2.2636e-4	5.02
2^{-3}	1.9932e-1	1.67	4.5495e-3	3.26	1.8976e-4	4.02	7.0224e-6	5.01
2^{-4}	5.2700e-2	1.92	5.1740e-4	3.14	1.1857e-5	4.00	2.1870e-7	5.00
2^{-5}	1.3351e-2	1.98	6.2636e-5	3.05	7.3927e-7	4.00	6.8244e-9	5.00
theoretical order		2.00		3.00		4.00		5.00

Table 8.3: Linear transport: Convergence of the error $e = u - u_h$ in space-time with respect to the $\|.\|_W$-norm for different polynomial degrees.

	$\|e\|_{V_h}$	EOC	$\|e\|_{V_h}$	EOC	$\|e\|_{V_h}$	EOC	$\|e\|_{V_h}$	EOC
	space-time convergence							
$\triangle \mathbf{x}, \triangle t$	$p \equiv q \equiv 1$		$p \equiv q \equiv 2$		$p \equiv q \equiv 3$		$p \equiv q \equiv 4$	
2^{-1}	4.1917e-0	-	2.5419e-0	-	6.8845e-1	-	1.4844e-1	-
2^{-2}	4.0844e-0	0.04	7.9058e-1	1.68	1.0670e-1	2.69	1.0565e-2	3.81
2^{-3}	2.1738e-0	0.91	2.1394e-1	1.89	1.3724e-2	2.96	6.7858e-4	3.96
2^{-4}	1.1045e-0	0.98	5.3769e-2	1.99	1.7226e-3	2.99	4.2418e-5	4.00
2^{-5}	5.5479e-1	0.99	1.3339e-2	2.01	2.1545e-4	3.00	2.6421e-6	4.00
theoretical order		1.00		2.00		3.00		4.00

Table 8.4: Linear transport: Convergence of the error $e = u - u_h$ in space-time with respect to the $\|.\|_{V_h}$-norm for different polynomial degrees.

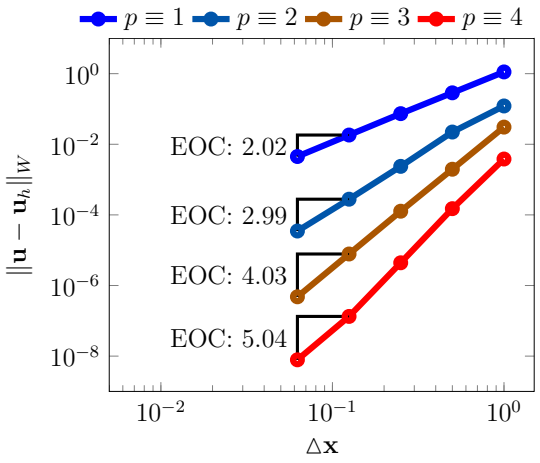

Figure 8.1: Linear transport: Convergence in space for sufficiently small $\triangle t$.

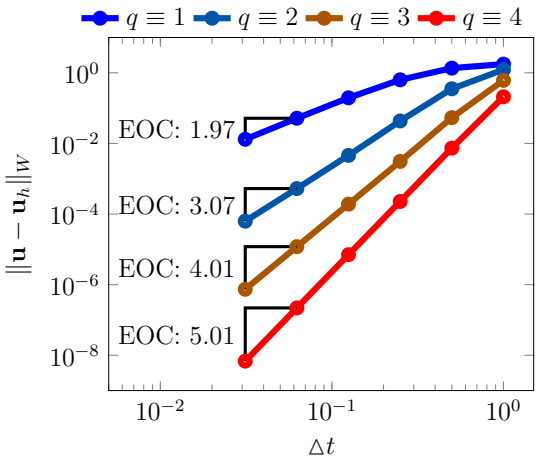

Figure 8.2: Linear transport: Convergence in time for sufficiently small $\triangle \mathbf{x}$.

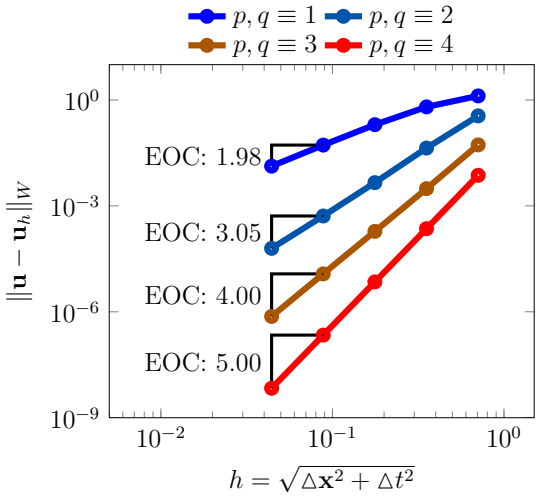

Figure 8.3: Linear transport: Convergence in space-time with respect to the $\|\,.\,\|_W$-norm.

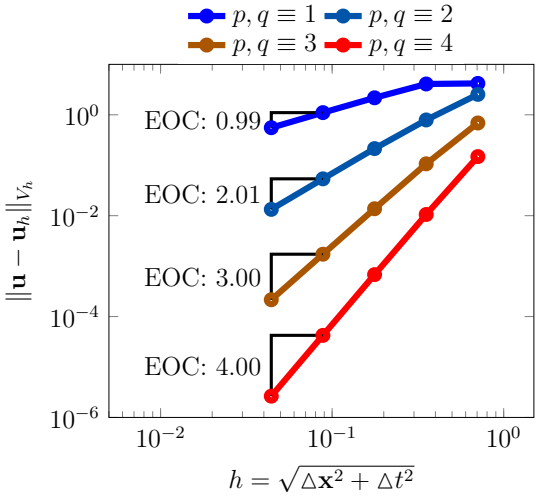

Figure 8.4: Linear transport: Convergence in space-time with respect to the $\|\,.\,\|_{V_h}$-norm.

8.1.2 Maxwell's Equations in 2D

In case of two-dimensional electromagnetic waves (TM waves), we consider the exact two-dimensional solution

$$\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} H_1(\mathbf{x},t) \\ H_2(\mathbf{x},t) \\ E_3(\mathbf{x},t) \end{pmatrix} = \begin{pmatrix} 0 \\ -\sin\left((x_1 - t)\,\pi\right) \\ \sin\left((x_1 - t)\,\pi\right) \end{pmatrix}$$

of linear Maxwell's equations (3.14) on a computational domain $Q = \Omega \times (0,T) = (-1,1)^2 \times (0,1)$. The parameters are given as $\varepsilon \equiv \mu \equiv 1$ and $\mathbf{f} = \mathbf{0}$. Again the initial and boundary conditions are taken from $\mathbf{u}$. Again, we first investigate convergence in space, time and space-time with respect to the $\|.\|_W$-norm. The computed experimental orders of convergence are shown in Tables 8.5, 8.6, 8.7 and in Figures 8.5, 8.6, 8.7. For the $\|.\|_{V_h}$-norm the computed experimental orders of convergence in space-time are given in Table 8.8 and Figure 8.8. As in the previous case, we observe the expected order of convergence in all four test cases.

	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC
$\triangle\mathbf{x}$	$p \equiv 1$		$p \equiv 2$		$p \equiv 3$		$p \equiv 4$	
2^{-0}	7.2477e-1	-	1.1432e-1	-	1.6535e-2	-	2.6628e-3	-
2^{-1}	2.1618e-1	1.75	1.3099e-2	3.13	1.1178e-3	3.89	8.3780e-5	4.99
2^{-2}	5.5389e-2	1.96	1.5297e-3	3.10	7.1642e-5	3.96	2.5857e-6	5.02
2^{-3}	1.3803e-2	2.00	1.8470e-4	3.05	4.5048e-6	3.99	7.9860e-8	5.02
2^{-4}	3.4740e-3	1.99	2.2775e-5	3.02	2.8168e-7	4.00		
theoretical order		2.00		3.00		4.00		5.00

Table 8.5: TM Maxwell's equations: Convergence of the error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ in space with respect to the $\|.\|_W$-norm for different polynomial degrees.

	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC
$\triangle t$	$q \equiv 1$		$q \equiv 2$		$q \equiv 3$		$q \equiv 4$	
2^{-0}	1.1494e-0	-	3.6888e-1	-	7.1625e-2	-	1.0034e-2	-
2^{-1}	5.1559e-1	1.16	5.2232e-2	2.82	4.3247e-3	4.05	3.1775e-4	4.98
2^{-2}	1.6347e-1	1.66	6.0845e-3	3.10	2.6687e-4	4.02	9.9132e-6	5.00
2^{-3}	4.3855e-2	1.90	7.3005e-4	3.06	1.6668e-5	4.00	3.0874e-7	5.00
2^{-4}	1.1184e-2	1.97	8.9588e-5	3.03	1.0422e-6	4.00		
theoretical order		2.00		3.00		4.00		5.00

Table 8.6: TM Maxwell's equations: Convergence of the error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ in time with respect to the $\|.\|_W$-norm for different polynomial degrees.

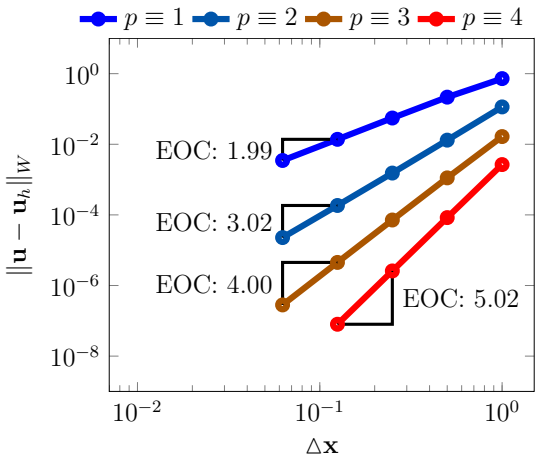

Figure 8.5: TM Maxwell's eq.: Convergence in space for sufficiently small $\triangle t$.

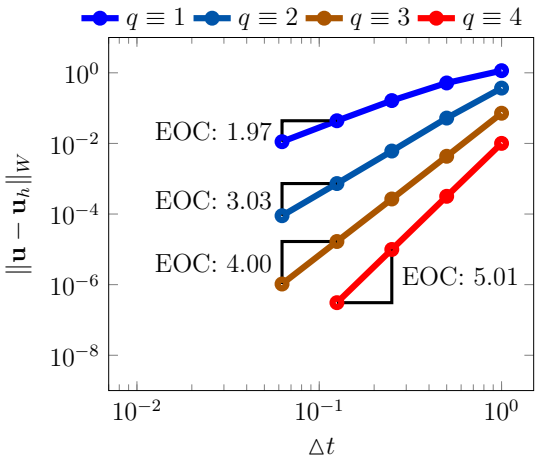

Figure 8.6: TM Maxwell's eq.: Convergence in time for sufficiently small $\triangle \mathbf{x}$.

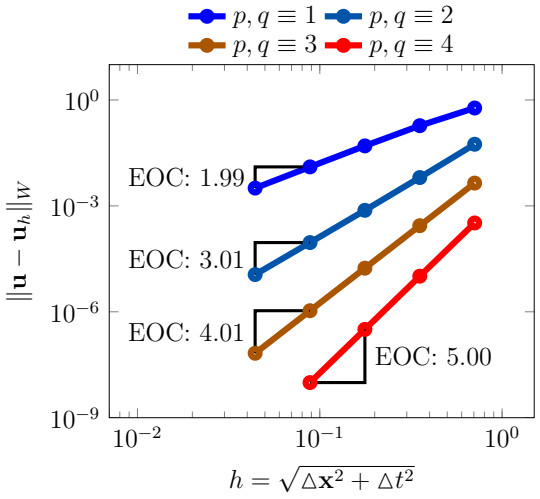

Figure 8.7: TM Maxwell's eq.: Convergence in space-time with respect to the $\|\,.\,\|_W$-norm.

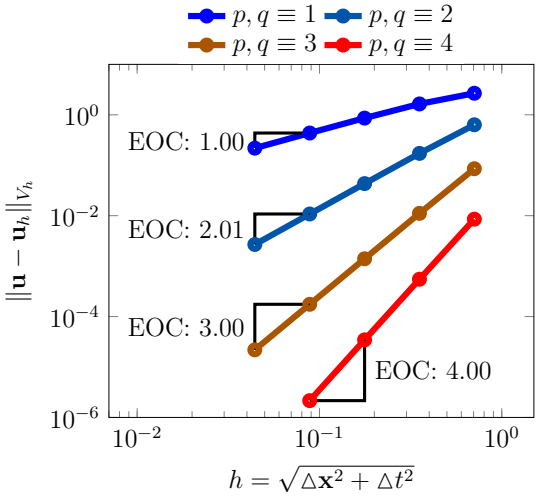

Figure 8.8: TM Maxwell's eq.: Convergence in space-time with respect to the $\|\,.\,\|_{V_h}$-norm.

	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC	$\|\mathbf{e}\|_W$	EOC
$\triangle\mathbf{x}, \triangle t$	$p \equiv q \equiv 1$		$p \equiv q \equiv 2$		$p \equiv q \equiv 3$		$p \equiv q \equiv 4$	
2^{-1}	5.9138e-1	-	5.5654e-2	-	4.4155e-3	-	3.2783e-4	-
2^{-2}	1.8685e-1	1.66	6.3467e-3	3.13	2.7484e-4	4.01	1.0209e-5	5.01
2^{-3}	5.0021e-2	1.90	7.4854e-4	3.08	1.7201e-5	4.00	3.1765e-7	5.01
2^{-4}	1.2715e-2	1.98	9.1341e-5	3.03	1.0735e-6	4.00	9.8990e-9	5.00
2^{-5}	3.1903e-3	1.99	1.1313e-5	3.01	6.6854e-8	4.01		
theoretical order		2.00		3.00		4.00		5.00

Table 8.7: TM Maxwell's equations: Convergence of the error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ in space-time with respect to the $\|.\|_W$-norm for different polynomial degrees.

	$\|\mathbf{e}\|_{V_h}$	EOC	$\|\mathbf{e}\|_{V_h}$	EOC	$\|\mathbf{e}\|_{V_h}$	EOC	$\|\mathbf{e}\|_{V_h}$	EOC
	space-time convergence							
$\triangle\mathbf{x}, \triangle t$	$p \equiv q \equiv 1$		$p \equiv q \equiv 2$		$p \equiv q \equiv 3$		$p \equiv q \equiv 4$	
2^{-1}	2.6766e-0	-	6.3450e-1	-	8.5300e-2	-	8.5373e-3	-
2^{-2}	1.6420e-0	0.70	1.7182e-1	1.88	1.1119e-2	2.94	5.4974e-4	3.96
2^{-3}	8.6077e-1	0.93	4.3259e-2	1.99	1.3980e-3	2.99	3.4516e-5	4.00
2^{-4}	4.3535e-1	0.98	1.0800e-2	2.00	1.7499e-4	3.00	2.1534e-6	4.00
2^{-5}	2.1837e-1	1.00	2.6860e-3	2.01	2.1879e-5	3.00		
theoretical order		1.00		2.00		3.00		4.00

Table 8.8: TM Maxwell's equations: Convergence of the error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ in space-time with respect to the $\|.\|_{V_h}$-norm for different polynomial degrees.

8.2 Parallel Scalability

In this section we test the scaling behavior of the parallel implementation of our multilevel preconditioner for different numbers of processes. To do this we consider the linear transport equation and the same two-dimensional test setting as stated in Section 6.4.1. All numerical tests in this section were performed on the ForHLR I Cluster (cf. A.3).

Weak Scalability We first consider the weak scaling behavior which is defined in the following, see, e.g., [Mag16, Def. 2.6].

Definition 8.1 (Weak scalability)**.** The *weak scalability* of an algorithm measures its efficiency with respect to an increasing number of processes P and an increasing problem size N (with the same rate). In particular the ratio N/P stays constant for increasing P. Moreover, weak scaling behavior is denoted as *perfect* if

$$T_P(N) = \text{const.}$$

where $T_P(N)$ is the runtime of an algorithm on P processes applied to a problem with N unknowns, i.e., degrees of freedom.

This means that the number of processes is doubled, quadrupled or increased eightfold every time we refine in space for one-, two- or three-dimensional problem, respectively. When refining in time, the number of processes is doubled. As a consequence the amount of associated degrees of freedom is fixed on every process. The results are given in Table 8.9 and Figure 8.9, where we refined in time and space alternatingly. Due to the small increase of iteration steps, observed in Section 6.4.1, we do not obtain a perfect weak scaling behavior. Instead we observe a logarithmic growth. To highlight the relation with respect to the increasing number of iteration steps, Figure 8.9 also contains a scaled plot. Here we illustrated the ratio between solving times and iteration steps $T_p(N)/n_{\text{iter}}$ which stays almost constant, i.e., almost perfect weak scaling behavior is achieved.

Strong Scalability As a second test we investigate the so called strong scaling behavior which is defined in the following, see, e.g., [Mag16, Def. 2.4].

Definition 8.2 (Strong scalability)**.** The *strong scalability* of an algorithm measures its efficiency with respect to an increasing number of processes P and a constant problem size N. Moreover, strong scaling behavior is denoted as *perfect* if

$$T_P(N) = \frac{T_1(N)}{P}$$

where $T_P(N)$ is given as in Definition 8.1. In particular we expect that the solving time scales with the reciprocal number of processes.

#proc. P	Degrees of freedom N	N/P	iteration steps n_{iter} (rate)	time $T_P(N)$ in [s]
1	$3\,072 \times 32$	$3\,072 \times 32$	4 (3.51e-3)	5.5
2	$3\,072 \times 64$	$3\,072 \times 32$	4 (6.01e-3)	5.8
4	$6\,144 \times 64$	$3\,072 \times 32$	6 (3.81e-2)	6.1
8	$12\,288 \times 64$	$3\,072 \times 32$	6 (4.16e-2)	7.6
16	$12\,288 \times 128$	$3\,072 \times 32$	6 (4.24e-2)	8.2
32	$24\,576 \times 128$	$3\,072 \times 32$	11 (1.31e-1)	15.7
64	$49\,152 \times 128$	$3\,072 \times 32$	10 (1.30e-1)	15.5
128	$49\,152 \times 256$	$3\,072 \times 32$	10 (1.43e-1)	16.1
256	$98\,304 \times 256$	$3\,072 \times 32$	16 (3.11e-1)	27.2
512	$196\,608 \times 256$	$3\,072 \times 32$	16 (2.87e-1)	31.4
1024	$196\,608 \times 512$	$3\,072 \times 32$	16 (3.05e-1)	35.0

Table 8.9: Weak scalability for uniform polynomial degree $(p, q) = (2, 2)$.

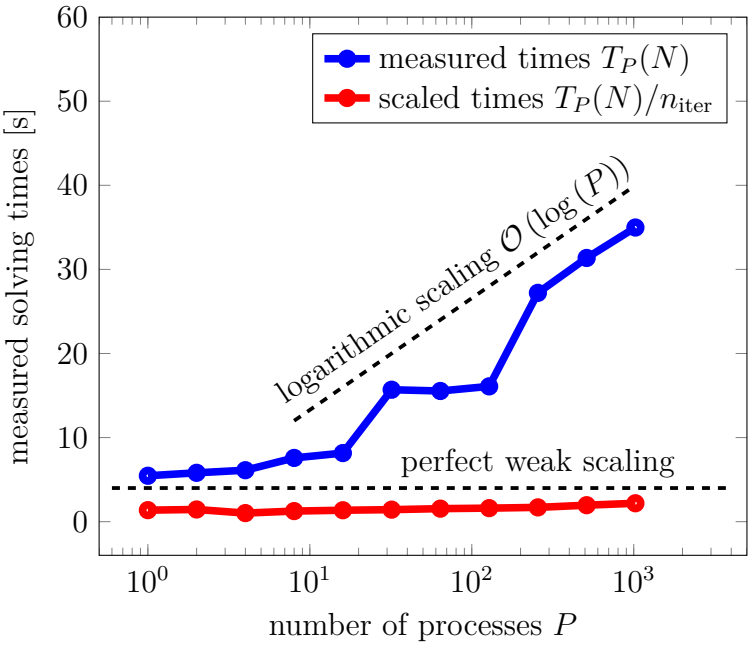

Figure 8.9: Weak scaling behavior for uniform polynomial degrees $(p, q) = (2, 2)$.

The results are given in Table 8.10 and Figure 8.10. For 128 processes and less, we recognize that the solving times decrease by a factor of approximately 0.5, if the number of processes is doubled. Thus the strong scalability is almost optimal in this case. Note that we observe a slight increase of iteration steps in this case, too. This is apparent from the fact that the Gauss–Seidel smoother is applied only locally on every process (cf. Remark 6.1). Additionally, the problem size (amount of DoFs) on every process gets too small for a large number of processes and hence communication time becomes dominant. As a result the strong scaling behavior deteriorates on more than 128 processes.

#proc. P	DoFs N	iter. steps n_{iter} (rate)	time $T_P(N)$ in [s]
1	$49\,152 \times 256$	9 (1.03e-1)	1 100.4
2	$49\,152 \times 256$	9 (1.08e-1)	584.3
4	$49\,152 \times 256$	9 (1.12e-1)	337.4
8	$49\,152 \times 256$	9 (1.16e-1)	194.7
16	$49\,152 \times 256$	10 (1.25e-1)	92.8
32	$49\,152 \times 256$	10 (1.15e-1)	57.0
64	$49\,152 \times 256$	10 (1.28e-1)	31.3
128	$49\,152 \times 256$	10 (1.43e-1)	16.6
256	$49\,152 \times 256$	11 (1.40e-1)	13.0
512	$49\,152 \times 256$	11 (1.63e-1)	9.9
1024	$49\,152 \times 256$	12 (1.79e-1)	9.2

Table 8.10: Strong scaling behavior for uniform polynomial degree $(p, q) = (2, 2)$.

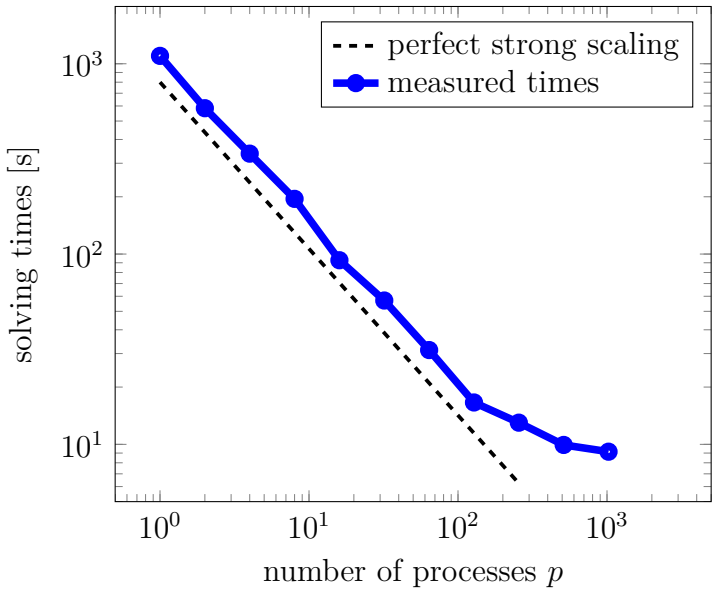

Figure 8.10: Strong scaling behavior for uniform polynomial degrees $(p, q) = (2, 2)$.

8.3 Parallel and Adaptive Space-Time Computation

In this section we present the results for the p-adaptive space-time method. Hence we combine the discontinuous Petrov–Galerkin discretization from Section 4.5 with the dual error estimation introduced in Section 5.4. Finally we solve the resulting linear system with a parallel iterative *generalized minimal residual solver* (GMRES), see, e.g. [Saa03, Ch. 6.5] or [Mei11, Sect. 4.3.2.4]. Moreover, the multilevel preconditioner presented in Chapter 6 is applied to reduce the computational costs. The GMRES method stops if the residual is reduced by a factor of 10^{-8}. Note that this value is chosen arbitrarily for our test cases. In real applications it would be sufficient that the solver terminates if the discretization error is reached. Again we first start with the advection equation (3.8) and a known solution, which serves as a test example to verify our algorithm. Afterwards we consider a more sophisticated application for transversal magnetic waves in two spatial dimensions.

8.3.1 Linear Transport Equation

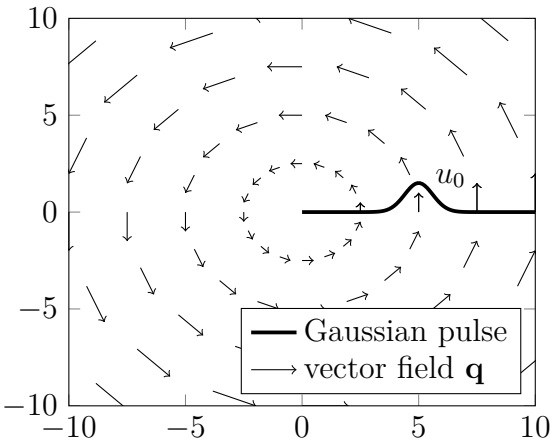

Figure 8.11: Schematic illustration of the rotating cone problem.

In this two-dimensional example we investigate the performance and reliability of our p-adaptive algorithm with respect to a linear advection equation. The results are compared to the case of uniform refinement. All numerical results in this section were computed on 64 processors on two nodes of the MA-PDE cluster (cf. A.3). Again, we choose the setting from Section 6.4.1, where a Gaussian pulse (6.21) in a rotating and divergence-free vector field $\mathbf{q}(\mathbf{x}) = 2\pi(-x_2, x_1)^\top$ is investigated. The space-time domain is given as $Q = (-10, 10)^2 \times (0, 1)$ and moreover a homogeneous right-hand side $\mathbf{f} = \mathbf{0}$ and constant density $\rho = 1$ is assumed (cf. Figure 8.11). The

exact solution is then given as

$$u(\mathbf{x},t) = \exp\Big(-1.4\Big((5\cos(2\pi t) - x_1)^2 + (5\sin(2\pi t) - x_2)^2\Big)\Big), \quad \text{for } (\mathbf{x},t) \in Q.$$

Note that we have $u_0(\mathbf{x}) = u(\mathbf{x},1)$ at final time $T = 1$. In the literature, these kind of test cases are often used as benchmarks for convection diffusion problems and known as *rotating cone problems* (see, e.g., [DH03, Sec. 3.11.3]).

We begin with an initial coarse mesh which consists of $1024 = 64 \times 16$ space-time cells and is refined 3 times in space and time up to 524 288 cells. Moreover we use lowest order polynomial degrees $(p,q) = (1,1)$ as initial distribution. The coarse problems, occurring in the multilevel preconditioner, are solved by using a parallel direct solver [MW11]. Here a block LU decomposition is computed in parallel only once. Then the LU decomposition is used to solve the coarse problem in every multilevel iteration with the cost of two matrix-vector multiplications. The results for the uniform p-refined case are given in Table 8.11.

uniform	GMRES with multilevel preconditioning			energy error
poly. deg. (p,q)	#DoFs	steps (rate)	time [s]	$\triangle\mathcal{J}$
(1,1)	1 585 152	10 (7.19e-2)	31.5	5.10e-2
(2,2)	6 340 608	10 (1.30e-1)	101.0	2.14e-3
(3,3)	15 851 520	11 (1.54e-1)	381.0	3.78e-5
(4,4)	31 703 040	11 (1.67e-1)	1088.3	4.41e-7

Table 8.11: Results for the rotating cone problem on a uniform mesh with $524\,288 = 4\,096 \times 128$ space-time cells and different polynomial degrees.

For the adaptive refinement corresponding to Chapter 5, we choose the quadratic error functional

$$\mathcal{J}(v) = \frac{1}{2}(\rho v, v)_{0,Q}. \tag{8.1}$$

Our aim is to minimize the error $|\triangle\mathcal{J}(u,u_h)| = |\mathcal{J}(u) - \mathcal{J}(u_h)|$ using the dual error indicator introduced in Section 5.3. Thus, the adaptive strategy minimizes the energy error in the space-time domain Q. To verify the results of the error indicator, we need the exact solution u^* of the corresponding dual problem (cf. Definition 5.2). The following result shows that u^* is again a rotating cone solution which travels backwards in time.

Lemma 8.1. *Consider a rotating cone problem with solution $u \in V$. Let $u^* \in V^*$ be the solution of the dual problem with respect to the quadratic error functional*

(8.1), *i.e.*,

$$\begin{aligned} L^*u^* &= \rho u, \quad \text{on } Q = \Omega \times (0,T), \\ u^*(\mathbf{x},T) &= 0, \quad \text{for all } \mathbf{x} \in \Omega, \\ u^*(\mathbf{x},t) &= 0, \quad \text{for all } (\mathbf{x},t) \in \Gamma_{\text{out}} \times (0,T), \end{aligned}$$

with homogeneous Dirichlet boundary conditions on the outflow boundary (cf. Definition 5.2 and Remark 3.3). Then u^* is given as

$$u^*(\mathbf{x},t) = (T-t)\, u(\mathbf{x},t).$$

Proof. For the rotating cone problem it yields that

$$L^*u^* = -Lu^* = -\rho\partial_t u^* - \nabla\cdot(\mathbf{q}u^*) \quad \text{and} \quad \mathcal{J}'(u) = \rho u\,.$$

By inserting u^* into the weak form of the dual problem we conclude that

$$\begin{aligned} (w, L^*u^*)_{0,Q} &= \left(w, -\rho\partial_t u^* - \mathbf{q}\cdot\nabla u^*\right)_{0,Q} = \left(w, \rho u - (T-t)(\rho\partial_t u + \mathbf{q}\cdot\nabla u)\right)_{0,Q} \\ &= \left(\rho u - (T-t)(\rho\partial_t u + \nabla\cdot(\mathbf{q}u)), w\right)_{0,Q} = \left(\rho u - (T-t)f, w\right)_{0,Q} \\ &= \left(\rho u, w\right)_{0,Q} = \langle \mathcal{J}'(u), w\rangle \end{aligned}$$

holds true for all $w \in V$. □

The adaptive results are given in Table 8.12 and Figure 8.12. We observe that the exact dual error $e^* = u^* - u_h^*$ is approximated through $e^*_{h,\mathrm{r}} = u^*_{H,\mathrm{r}} - u_h^*$ quite well. Here we used the patch-wise higher-order approximation introduced in Section 5.3.2. Since we know the exact solution u, the exact error $\triangle\mathcal{J} = \mathcal{J}(u) - \mathcal{J}(u_h)$ can be computed. Moreover, we can compute the approximation $\triangle\mathcal{J}_h$ to $\triangle\mathcal{J}$ by using u_h, u_h^* and $u^*_{H,\mathrm{r}}$ in the error representation (5.15)

$$\triangle\mathcal{J} = \triangle\mathcal{J}_h + \rho(u_h; u^* - u^*_{H,\mathrm{r}}) + \mathcal{T}_2 \approx \triangle\mathcal{J}_h,$$

where we skipped the remainder term $\mathcal{T}_2$ as well as the error term $\rho(u_h; u^* - u^*_{H,\mathrm{r}})$. Note that $\triangle\mathcal{J}_h$ almost coincides with $\triangle\mathcal{J}$ and thus $\triangle\mathcal{J}_h$ is the dominant term, as it is assumed in Section 5.3. Finally, we compare $\triangle\mathcal{J}$ with the sum over all cell-wise estimated errors $\eta_R = \sum_{R\in\mathcal{R}} \eta_R$ computed by (5.14). One observes that due to cancellation effects the true error $\triangle\mathcal{J}$ is overestimated by far, but both have the same asymptotic behavior. This is an essential requirement to achieve a reliable error estimation.

Figure 8.13 and Figure 8.14 show the adaptive solution in the space-time domain and the location of highest polynomial degrees, respectively. We note that highest polynomial degrees $(p,q) = (4,4)$ are only used in areas, where the pulse is actually located. In contrast, lowest polynomial degrees $(p,q) = (1,1)$ are used everywhere

p-ref. level	GMRES #DoFs (effort)	steps (rate)	time [s]	errors $\|e^* - e^*_{h,r}\|_{0,Q}$	$\triangle\mathcal{J}$	$\triangle\mathcal{J}_h$	$\eta_{\mathcal{R}}$
$l=0$	1 585 152	10 (7.19e-2)	31.5	1.78e-1	5.10e-2	4.08e-2	7.60e-1
$l=1$	1 894 176 (30%)	10 (1.00e-1)	39.2	9.98e-3	2.14e-3	2.63e-3	3.59e-2
$l=2$	2 381 589 (15%)	11 (1.41e-1)	89.5	8.41e-4	3.79e-5	4.44e-5	7.33e-4
$l=3$	3 303 819 (10%)	11 (1.67e-1)	185.7	5.29e-4	4.33e-7	4.94e-7	1.47e-5

Table 8.12: Adaptive p-refinement on a mesh with $524\,288 = 4\,096 \times 128$ space-time cells ($\vartheta =$ 1e-4).

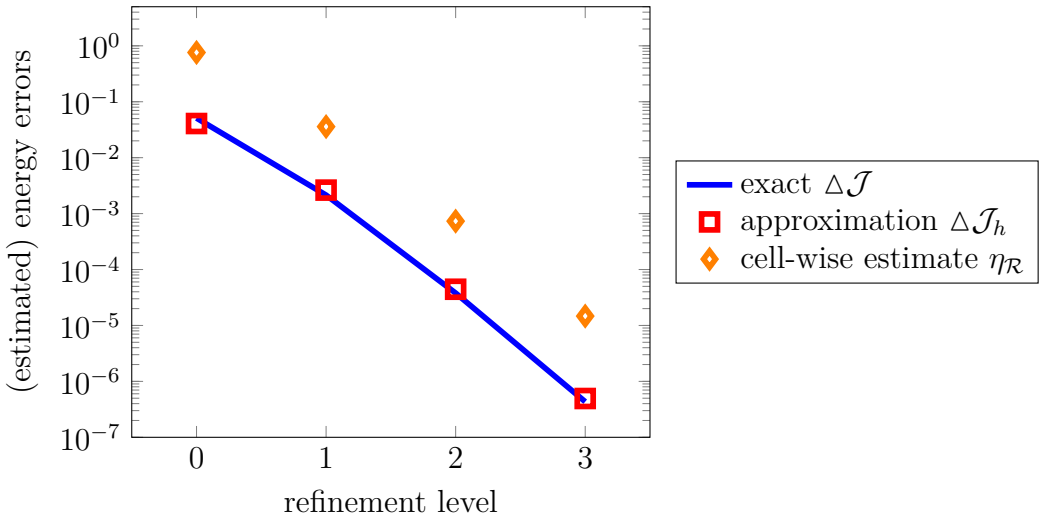

Figure 8.12: Error estimation in the adaptive refined case.

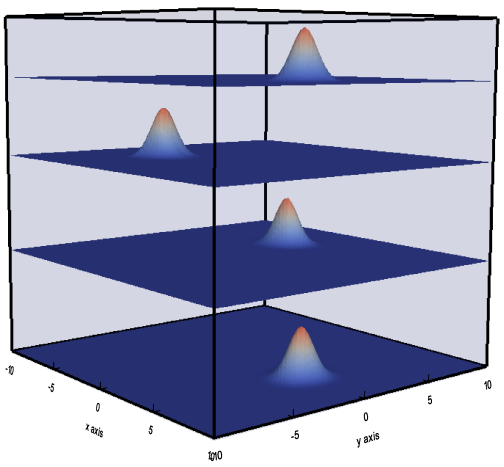

Figure 8.13: Solution of the transport equation in the space-time domain Q, sliced at times $t = 0, 0.3, 0.6, 1$.

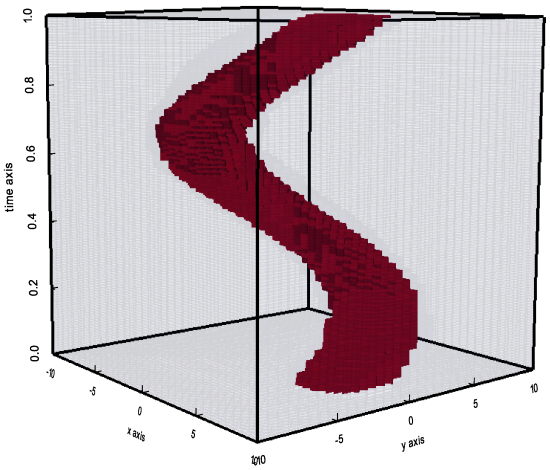

Figure 8.14: Location of the highest polynomial degrees in the space-time domain Q.

else. In Figure 8.15 and Figure 8.16 the benefit of adaptive strategies becomes visible when we compare it to the uniform refined case. First we compare the errors with respect to the degrees of freedom, needed in every refinement step. On the last refinement level one observes that we achieve the same accuracy in both cases, but with a reduction of degrees of freedom of about 90% when using the presented adaptive strategy. This reduced amount of degrees of freedom also results in a decreased amount of computation time. In Figure 8.16 the accumulated solving times of the GMRES solver with multilevel preconditioner are illustrated. Mind that in the adaptive case, we have to solve an additional dual problem of the same complexity as the primal one. Despite this enlarged effort, the uniform refinement is about 3.1 times slower than the adaptive case, i.e., we observe a solving time reduction of about 68%. These results can only be achieved by using a load balancing algorithm which redistributes the space-time cells in every refinement step. In this example we used the weighted RCB method presented in Section 7.2. The redistributed mesh after four refinements is given in Figure 8.17 at different time points.

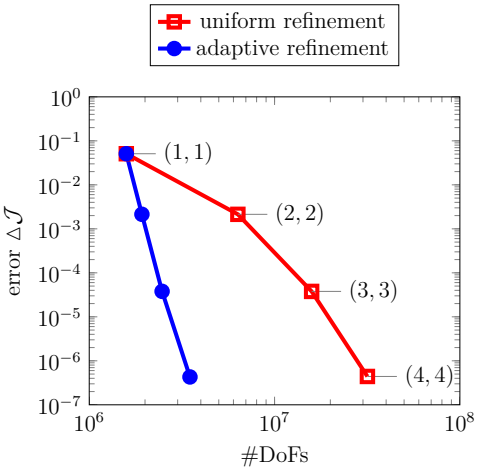

Figure 8.15: Adaptive performance with respect to degrees of freedom.

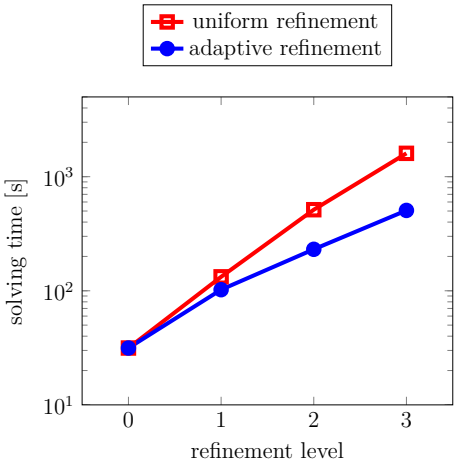

Figure 8.16: Adaptive performance with respect to accumulated times.

In general, the actual benefit of adaptive strategies strongly depends on the underlying problem. Hence no benefit would be expected, if we want to resolve a numerical solution with global support as good as possible over the whole space-time domain. But in applications, where a solution is strongly localized (e.g., in a rotating cone problem or a single wavefront problem) one can save a large amount of computational costs and time. On the other hand it might be possible that one is only interested in small parts of the solution (e.g., point sources and receivers). Such an example is considered in the following section for the Maxwell case.

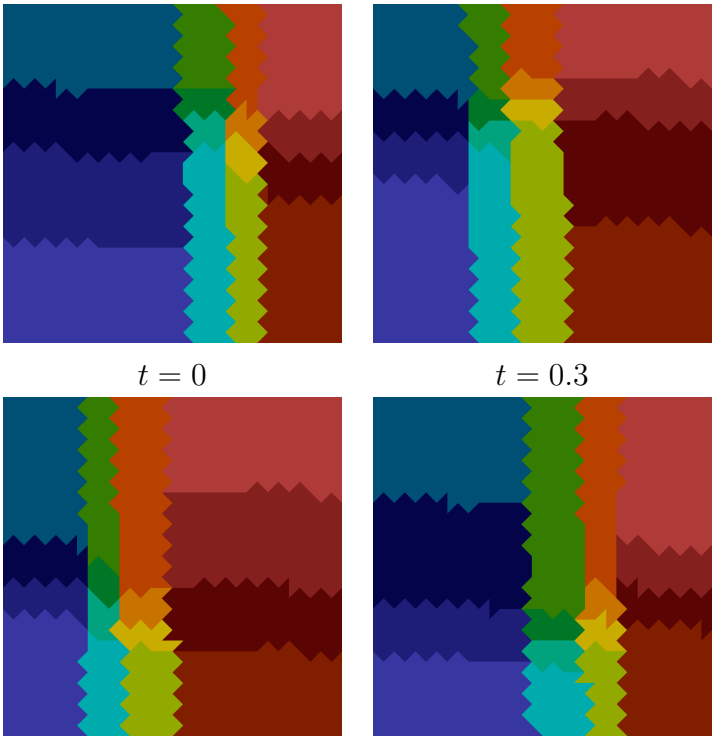

Figure 8.17: Redistributed space-time mesh on 16×4 processes after four refinements using a weighted RCB load balancing method (cf. Section 7.2).

8.3.2 Electromagnetic transverse magnetic Waves

For a more realistic application, we consider the simulation of the so called double-slit experiment. This experiment has been elected to be *the most beautiful experiment* by readers of the *Physics World* journal in 2002 (see [Cre02]).

In this experiment a two-dimensional transverse magnetic wave $\mathbf{u} = \left(\mathbf{H}_1, \mathbf{H}_2, \mathbf{E}_3\right)^\top$ with wavelength $\lambda = 1$ is scattered by a double slit (see Section 3.4). The slit itself has a gap of length $a = 3$ and a width $b = 1$. Moreover constant material parameters $\mu = \varepsilon = 1$ and reflecting boundary conditions are applied. On the left side of the computational domain $Q = (0, 6) \times (-6, 6) \times (0, 8)$ the scattered wave enters (see Figure 8.18) and begins to interfere. One observes a so called

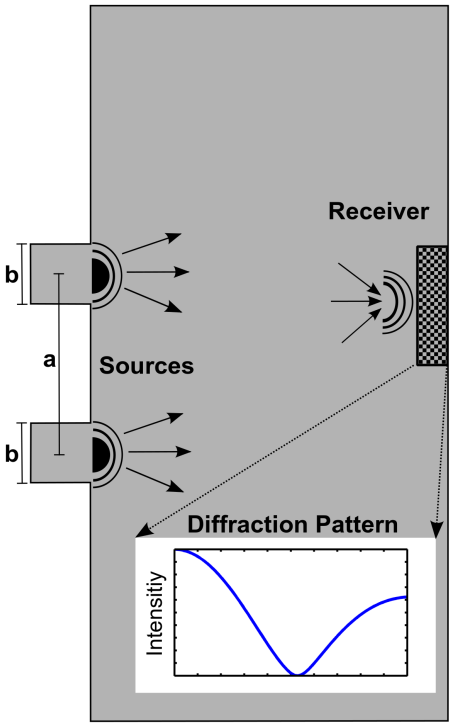

Figure 8.18: Schematic illustration of the double-slit experiment and the expected diffraction pattern at the receiver.

diffraction pattern with several local intensity maxima and minima. In applications one is often only interested in certain (small) parts of the scattered wave, e.g., the first occurring minimum on the right side of the computational domain. For illustration, we therefore choose the region of interest as the space-time domain $S = (5.5, 6) \times (0, 2) \times (0, 8)$. Thus we choose the energy error functional as

$$\mathcal{J}(\mathbf{v}) = \frac{1}{2}(M\mathbf{v}, \mathbf{v})_{0,S},$$

to resolve the minimum as accurate as possible. This corresponds to a screen or receiver which is located in S in the real experiment.

In this setting the exact solutions $\mathbf{u}$ and $\mathbf{u}^*$ are unknown and hence $\mathcal{J}(\mathbf{u})$ cannot be computed. However, we can compute an approximated value $\mathcal{J}_{\text{ex}}$ via extrapolation as

$$\mathcal{J} \approx \mathcal{J}_{\text{ex}} = \frac{r_l}{r_l - 1}\mathcal{J}_l - \frac{1}{r_l - 1}\mathcal{J}_{l-1}, \qquad r_l = \frac{|\mathcal{J}_{l-1} - \mathcal{J}_{l-2}|}{|\mathcal{J}_l - \mathcal{J}_{l-1}|}, \qquad \mathcal{J}_l = \mathcal{J}(\mathbf{u}_{h,l}),$$

where $\mathbf{u}_{h,l}$ are discrete solutions $\mathbf{u}_h$ on the different refinement levels $l = 0, 1, 2, \ldots$ (cf. [HPS+15]). Hence, an approximated error can be computed as $\triangle\mathcal{J} \approx \triangle\mathcal{J}_{\text{ex}} = |\mathcal{J}_{\text{ex}} - \mathcal{J}_l|$.

In the following we show numerical results on two different levels with 256 and 1024 processes, computed on the ForHLR I cluster (cf. A.3). Moreover, we used a GMRES solver with block-Gauss–Seidel preconditioning for the coarse problem within the multilevel preconditioner. First we consider a coarse initial mesh which consists of $9\,472 = 148 \times 64$ space-time cells and is refined 2 times in space and time to a fine mesh with 606 208 space-time cells. The computed results for the uniformly and adaptively refined case on 256 processes is given in Table 8.13. As

		GMRES				
level	(p, q)	#DoFs (effort)	steps (rate)	time [s]	$\mathcal{J}_l$	$\triangle\mathcal{J}_{\text{ex}}$
		uniform refinement				
$l = 0$	$(1, 1)$	5 477 184	10 (1.14e-1)	5.4	8.9307e-2	3.1303e-1
$l = 1$	$(2, 2)$	21 908 736	17 (2.84e-1)	56.3	3.7612e-1	2.6220e-2
$l = 2$	$(3, 3)$	54 771 840	24 (4.48e-1)	437.4	4.0089e-1	1.4502e-3
$l = 3$	$(4, 4)$	109 543 680	34 (5.68e-1)	2154.0	4.0226e-1	8.0209e-5
		adaptive refinement				
$l = 0$		5 477 184	10 (1.14e-1)	5.4	8.9307e-2	3.1303e-1
$l = 1$		9 645 930 (44%)	13 (2.33e-1)	31.0	3.7611e-1	2.6230e-2
$l = 2$		17 309 043 (32%)	18 (3.37e-1)	153.8	4.0089e-1	1.4502e-3
$l = 3$		29 064 348 (27%)	23 (4.39e-1)	710.8	4.0226e-1	8.0209e-5

Table 8.13: Uniform vs. adaptive refinement on $606\,208 = 2\,368 \times 256$ space-time cells distributed to 256 processes (ϑ = 1e-3, E_{ex} = 4.0234e-1).

observed in the previous example, we note that the computed values of $\mathcal{J}_l$ and the approximated error $\triangle\mathcal{J}_{\text{ex}}$ almost coincides when comparing the uniform and the adaptive refined case. But only 27% of the degrees of freedom are used in the adaptive case compared with the full high order discretization. Although we have to solve the dual problem on every refinement level additionally, we observe that the solving is about 2.4 times faster in the adaptive refinement compared to the uniform case (cf. Figure 8.19). This corresponds to a solving time reduction of almost 58%.

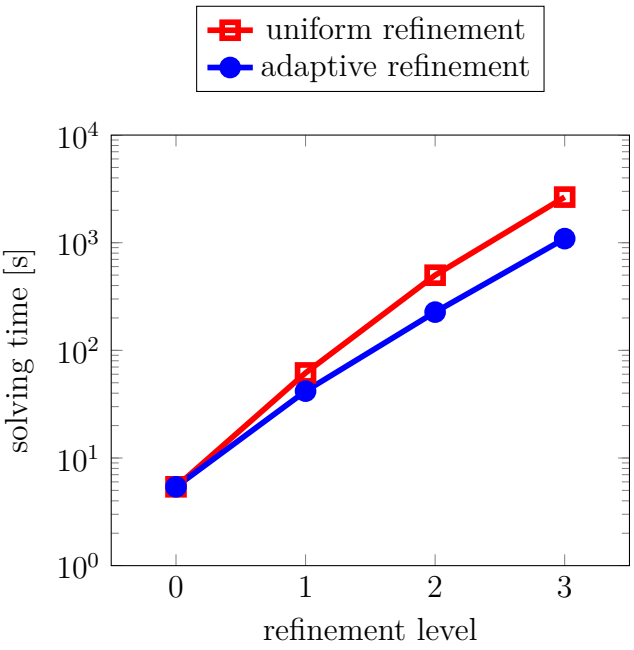

Figure 8.19: Adaptive performance with respect to accumulated solving times. We used 256 processes to solve the double-slit Maxwell.

Decreasing computation times is not the only benefit of adaptive strategies as one observes when performing an additional space-time refinement. The fine mesh with 4 849 664 space-time cells is distributed on 1 024 processes. Correspondingly the coarse mesh is refined, too. This is necessary, since the load balancing is done on the coarse mesh (cf. Section 7.2). We notice that we run out of computational resources, i.e., internal memory, in the uniform refined case. But in the adaptive case we are still able to compute an accurate solution on the highest refinement level since we save 79% of the degrees of freedom compared to a possible uniform refined case (see Table 8.14).

Another point of view is the capability of parallel space-time methods compaced to implicit Runge–Kutta time steppings schemes. Although explicit run time comparisons are missing, we can estimate the numerical effort in this case. To achive the same accurate results as presented in Table 8.14, one would have to apply 512 iteration steps of a 4-stage Gauss–Legendre collocation method (cf. Lemma 6.3). Hence one would have to solve a linear system of equations with about 1.44 million unknowns on average in every time step (cf. [HB09, Sec. 78]). To surpass the space-time method, this has to be done in less than $1543.5/512 \approx 3$ seconds. Even on $P = 1024$ processes this can be a challenging goal.

Finally, we visualize the time evolution of the scattered wave in Figure 8.20. We observe the expected diffraction pattern and several local extrema. Furthermore, one notices that highest polynomial degrees are only used in areas, where it is

level	(p,q)	GMRES #DoFs (effort)	steps (rate)	time [s]	$\mathcal{J}_l$	$\triangle\mathcal{J}_{\text{ex}}$
		uniform refinement				
$l=0$	$(1,1)$	43 732 224	17 (3.04e-1)	16.6	3.0520e-1	9.7373e-2
$l=1$	$(2,2)$	174 928 896	31 (5.36e-1)	187.4	4.0081e-1	1.7630e-3
$l=2$	$(3,3)$	437 322 240	out of memory			
$l=3$	$(4,4)$	874 644 480	out of memory			
		uniform refinement				
$l=0$		43 732 224	17 (3.04e-1)	16.6	3.0520e-1	9.7373e-2
$l=1$		68 437 899 (39%)	21 (4.01e-1)	76.5	4.0082e-1	1.7530e-3
$l=2$		115 207 920 (26%)	28 (5.07e-1)	361.4	4.0250e-1	7.3043e-5
$l=3$		184 208 094 (21%)	37 (5.82e-1)	1543.5	4.0257e-1	3.0435e-6

Table 8.14: Uniform vs. adaptive refinement on $4\,849\,664 = 9\,472 \times 512$ space-time cells distributed to 1024 processes ($\vartheta =$ 1e-3, $E_{\text{ex}} =$ 4.0257e-1).

necessary to have a high resolution in S and hence a minimal error $\triangle\mathcal{J}$. On the other hand lowest polynomial degrees are used everywhere else in Q. In Figure 8.21 the accumulated intensity of the scattered wave at $x_1 = 6$ is shown. Furthermore, we compare the results with a physical approximation to the far field intensity

$$I(\alpha) = \operatorname{sinc}^2\left(\frac{\pi}{\lambda} b \sin(\alpha)\right) \cos^2\left(\frac{\pi}{\lambda} a \sin(\alpha)\right),$$

which depends on the observation angle α. Note that only the extrema are supposed to coincide. The actual graph shape of the exact solution $\mathbf{u}$ is different. But we observe that the minimum of $\mathbf{u}_h$ is indeed resolved quite well.

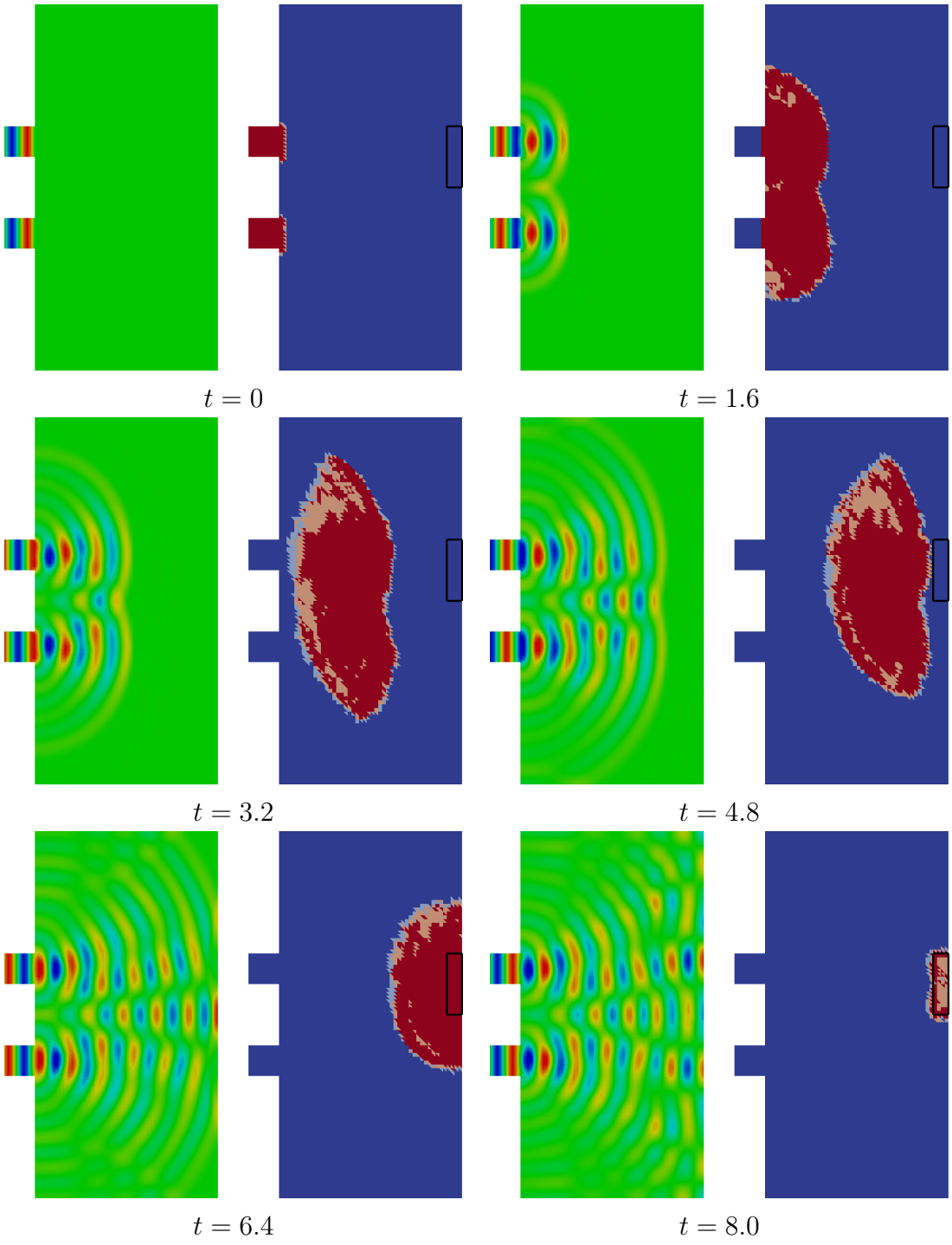

Figure 8.20: Scattered wave solution $\mathbf{u}_h$ (left) and used polynomial degrees up to $(p, q) = (4, 4)$ (right) at times $t = 0, 1.6, 3.2, 4.8, 6.4, 8.0$. Solved on a mesh with $4\,849\,664 = 9\,472 \times 512$ with 1024 processes. The region of interest is additionally highlighted (black box).

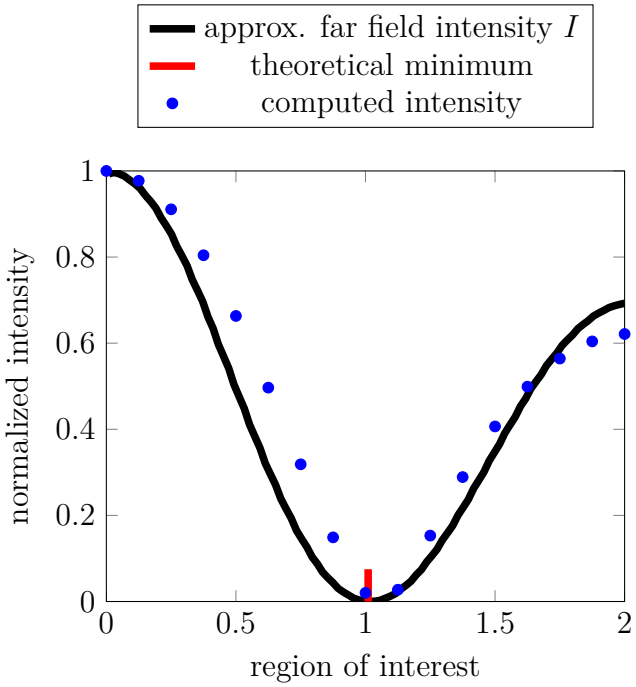

Figure 8.21: Computed intensity compared with the approximated far field intensity I. Extrema are supposed to coincide.

9 Summary, Conclusion and Outlook

9.1 Summary

In this work a discontinuous Petrov–Galerkin space-time discretization for hyperbolic evolution equations had been presented. In particular we focused on a linear transport problem and the linear Maxwell's equations. We investigated the hyperbolic models in a variational setting and stated existence and uniqueness proofs. Moreover, we analyzed the space-time discretization and proved existence and uniqueness of a discrete solution in case of tensor product space-time meshes. It was illustrated that this approach is similar to time stepping schemes, constructed from a slice-wise space-time discretization. However, considering the entire space-time cylinder $Q = \Omega \times (0, T)$ offered the possibility of applying adaptivity to the finite element setting in a "natural" way. We highlighted that in case of time-dependent problems dual weighted residual methods can lead to a significant reduction of computational costs. Here we used that a dual formulation of the problem can be used to trace the relevant parts in the primal solution. Depending on the underlying problem, this advantage is able to compensate the fact that an additional system of equations has to be solved to obtain an approximation to the dual solution. Another advantage of considering the discretization on the entire space-time cylinder is the possibility of using new parallel solving techniques in space **and** time. As an example a multilevel preconditioner with suitable convergence rates was presented. The corresponding adjustments of its parameters were achieved by performing several numerical test. Finally, we described how a space-time data structure can be implemented into existing finite element libraries. Numerical experiments for the linear transport problem and electromagnetic TM waves verify not only the theoretical results, but also indicate that our implementation is parallelized well. Furthermore, we used the test examples in Chapter 8 to illustrated the opportunities of space-time adaptive methods and parallel multilevel solvers.

9.2 Conclusion and Outlook

We presented a holistic approach to solve an electromagnetic wave problem efficiently. Therefore, we equipped a discontinuous Petrov–Galerkin space-time discretization with an adaptive dual error estimator technique and solved the resulting linear systems by using a parallel in space and time multilevel preconditioner. However, the reader will have noticed that there is still a vast variety of possibilities for further developments and improvements for space-time methods. Regarding the discretization one can use a more general space-time discretization to overcome the tensor product approach and use general space-time finite elements, see, e.g. [NS11] or [Neu13, Ch. 3]. Moreover, a rigorous error analysis and a better understanding of the recovery operator would help to improve the error estimation. So far, only p-adaptivity has been implemented and tested. Together with a general space-time finite element discretization, h-adaptivity would lead to more flexibility and better adaptation to the problems. In terms of hp-adaptivity an even higher reduction of computational costs seems possible. On the other hand a rigorous analysis of the smoothing and approximation property, introduced in Chapter 6, may help to find better parameters to improve the multilevel preconditioner.

Despite the indications given in this work, the evidence that parallel space-time methods are faster than standard space-parallel explicit or implicit time stepping methods is still missing. In [FFK$^+$14] several strong scaling test results are given for parabolic test equations and a parallel in space and time multigrid solver. They indicate that several hundreds (for $D = 2$) or thousands (for $D = 3$) of parallel working processes are needed to defeat implicit time stepping schemes. For realistic applications, a need of several tens of thousands processes is predicted. This requires that the used finite element library provides the tools to handle the resulting problems efficiently. Thus, applying the introduced methods to a realistic application (where $D = 3$) and testing with several thousands of processes is one of the next possible steps. With the results of this work, it is justified to hope that parallel space-time methods will outperform standard time stepping methods on massive parallel machines in the future.

A Appendix

A.1 Barycentric Coordinates on Tetrahedral Cells

Here we describe a short definition of barycentric coordinates as given in [EG04, Sec. 1.2.3].

Definition A.1 (Barycentric coordinates)**.** Let $\widehat{K} \in \mathbb{R}^D$ be the reference tetrahedral cell with vertices $\widehat{\mathbf{x}}_0 = (0, \ldots, 0), \widehat{\mathbf{x}}_1 = (1, 0, \ldots, 0), \ldots, \widehat{\mathbf{x}}_D = (0, \ldots, 0, 1) \in \mathbb{R}^D$. Moreover, let $K \in \mathcal{K}$ be a cell of a mesh $\mathcal{K}$, with vertices $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_D\}$. For every point $\mathbf{x} \in K$ there exists a unique vector $\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \ldots, \lambda_D)$, with $\lambda_i \geq 0$ such that

$$\mathbf{x} = \lambda_0 \mathbf{x}_0 + \lambda_1 \mathbf{x}_1 + \ldots + \lambda_D \mathbf{x}_D \quad \text{and} \quad \sum_{i=0}^{D} \lambda_i = 1.$$

The set $\lambda_0, \lambda_1, \ldots, \lambda_D$, i.e., $\boldsymbol{\lambda}$, are called *barycentric coordinates* of the point $\mathbf{x}$. Hence we can define the affine linear mapping

$$\phi \colon \widehat{K} \to K, \qquad \phi(\widehat{\mathbf{x}}) = \lambda_0(\widehat{\mathbf{x}})\mathbf{x}_0 + \lambda_1(\widehat{\mathbf{x}})\mathbf{x}_1 + \ldots + \lambda_D(\widehat{\mathbf{x}})\mathbf{x}_D$$

and the barycentric coordinates became affine linear functions. Let K be an arbitrary tetrahedral cell with vertices $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_D\}$, then

$$\lambda_i \colon \mathbb{R}^D \to \mathbb{R}, \qquad \lambda_i(\mathbf{x}) = 1 - \frac{(\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{n}_i}{(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_i},$$

where $\mathbf{n}_i$ is the outer unit normal to the face $f_i \subset \partial K$ which is opposite to $\mathbf{x}_i$ and $\mathbf{x}_j \in f_i$ can be chosen arbitrary in f_i.

Example A.1 (Lagrange polynomials)**.** For $p_K = 0, 1, 2$ the Lagrange shape functions are given in Table A.1 and illustrated for $D = 1$ in Figure A.1.

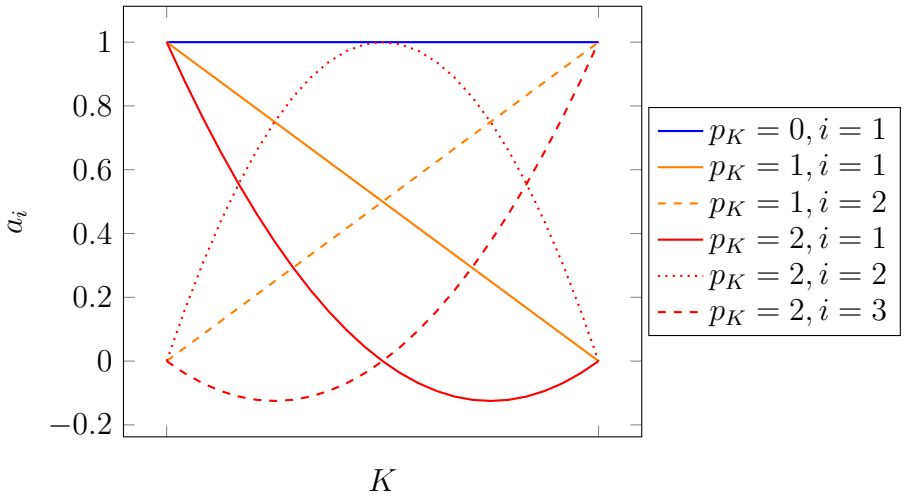

Figure A.1: First three sets of Lagrange polynomials on $K \subset \mathbb{R}$.

	shape functions
$p_K = 0$	1
$p_K = 1$	$\lambda_i, \quad \text{for } 1 \leq i \leq D+1$
$p_K = 2$	$\begin{cases} \lambda_i(2\lambda_i - 1), & \text{for } 1 \leq i \leq D+1 \\ 4\lambda_i\lambda_j, & \text{for } 1 \leq i < j \leq D+1 \end{cases}$

Table A.1: First three sets of Lagrange shape functions, defined through the barycentric coordinates λ_i (see Definition A.1).

A.2 Legendre Polynomials

Here we give a short definition of Legendre polynomials, which can be found for example in [HB09, Ex. 33.2]. Afterwards we give a result stated in [AS64] and show Lemma A.2.

Definition A.2 (Legendre polynomials)**.** We consider the interval $[-1, 1]$. The sequence of real polynomials $(\lambda_p)_{p\in\mathbb{N}_0}$ with $\lambda_p \in \mathbb{P}_p([-1, 1])$ and

$$\lambda_p(t) = \frac{1}{2^p p!}\partial_t^p (t^2 - 1)^p \tag{A.1}$$

is called *Legendre polynomials*. Legendre polynomials are orthogonal, i.e.

$$\int_{-1}^{1} \lambda_p(t)\lambda_q(t)\,\mathrm{d}t = 0, \quad \text{for } p \neq q.$$

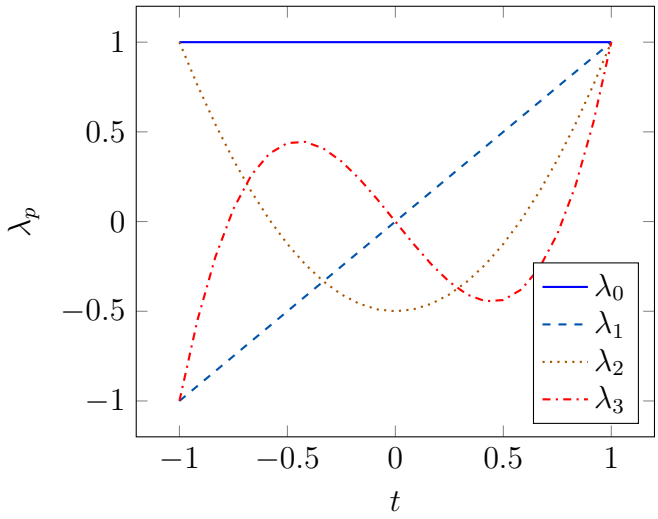

Figure A.2: Fist four Legendre polynomials on $I = [-1, 1]$.

We extend this definition to arbitrary intervals $I = [a, b] \subset \mathbb{R}$ by using the scaling

$$\lambda_{I,p}(t) = c_p \sqrt{\frac{1}{b-a}} \lambda_p \left(2\frac{t-a}{b-a} - 1 \right), \tag{A.2}$$

where the normalization constant c_p is determined by the additional requirement

$$\int_a^b \lambda_{I,p}(t)\lambda_{I,q}(t)\,\mathrm{d}t = \delta_{p,q}, \quad \text{for } p, q \in \mathbb{N}_0.$$

Hence c_p is independent of I and we achieve

$$1 = \int_a^b \lambda_{I,p}^2(t)\,\mathrm{d}t = \int_{-1}^1 \frac{c_p^2}{2}\lambda_p^2(t)\,\mathrm{d}t \quad \Longleftrightarrow \quad c_p = \left(\frac{1}{2} \int_{-1}^1 \lambda_p^2(t)\,\mathrm{d}t \right)^{-\frac{1}{2}}, \quad \text{for } p \in \mathbb{N}_0.$$

We denote the sequence of real polynomials $(\lambda_{I,p})_{p\in\mathbb{N}_0}$ with $\lambda_p \in \mathbb{P}_p(I)$ as *general orthonormal Legendre polynomials*.

Example A.2. The first four Legendre polynomials are given as

$$\lambda_0(t) = 1, \quad \lambda_1(t) = t, \quad \lambda_2(t) = \frac{1}{2}(3t^2 - 1), \quad \lambda_3(t) = \frac{1}{2}(5t^3 - 3t).$$

and illustrated in Figure A.2

Lemma A.1. *Let $(\lambda_p)_{p\in\mathbb{N}_0}$ be the sequence of Legendre polynomials defined in* (A.1) *and set $\lambda_{-1} \equiv 0$, then the relation*

$$(p - q + 1)\partial_t^q \lambda_{p+1}(t) = (2p + 1)t\partial_t^q \lambda_p(t) - (p + q)\partial_t^q \lambda_{p-1}(t)$$

holds true for all $p, q \geq 0$.

Proof. See [AS64, Lem. 8.5.3]. □

Lemma A.2. *Let $\lambda_{p,I} \in \mathbb{P}(I)$ be a general orthonormal Legendre polynomial on the interval $I = [a, b]$ with $p \in \mathbb{N}_0$ and c_p as in Definition A.2. It yields*

$$(t\partial_t \lambda_{I,p}, \lambda_{I,p})_{0,I} = \frac{c_p^2}{2}(t\partial_t \lambda_p, \lambda_p)_{0,[-1,1]} = p.$$

Proof. For the first identity it holds that

$$\begin{aligned}(t\partial_t \lambda_{I,p}, \lambda_{I,p})_{0,I} &= \int_a^b \frac{2c_p^2}{(b-a)^2} t\lambda_p' \left(2\frac{t-a}{b-a} - 1\right) \lambda_p \left(2\frac{t-a}{b-a} - 1\right) \mathrm{d}t \\ &= \int_{-1}^1 \frac{c_p^2}{2} \left(s + \frac{a+b}{b-a}\right) \lambda_p'(s)\lambda_p(s)\,\mathrm{d}s \\ &= \int_{-1}^1 \frac{c_p^2}{2} s\lambda_p'(s)\lambda_p(s)\,\mathrm{d}s = \frac{c_p^2}{2}(t\partial_t \lambda_p, \lambda_p)_{0,[-1,1]}.\end{aligned}$$

In case of constant polynomials, i.e., $p = 0$ we have that $\partial_t \lambda_0 = 0$ and hence the second identity is fulfilled, too. For $p > 0$ we obtain from the previous Lemma (with $q = 0$) that

$$(p+1)\lambda_{p+1}(t) = (2p+1)t\lambda_p(t) - p\lambda_{p-1}(t)$$

and hence

$$(p+1)\partial_t \lambda_{p+1}(t) = (2p+1)\lambda_p(t) + (2p+1)t\partial_t \lambda_p(t) - p\lambda_{p-1}(t). \tag{A.3}$$

Furthermore, Lemma A.1 provides

$$p\partial_t \lambda_{p+1}(t) = (2p+1)t\partial_t \lambda_p(t) - (p-1)\partial_t \lambda_{p-1}(t) \tag{A.4}$$

for $q = 1$. By subtracting (A.4) from (A.3) yields

$$\partial_t \lambda_{p+1}(t) = (2p+1)\lambda_p(t) + \partial_t \lambda_{p-1}(t).$$

Finally, we show the second identity by using the previous results and the explicit representation of c_p:

$$\begin{aligned}\frac{c_p^2}{2}(t\partial_t \lambda_{p+1}, \lambda_{p+1})_{0,[-1,1]} &= \frac{c_p^2}{2}(t(2p+1)\lambda_p, \lambda_{p+1})_{0,[-1,1]} \\ &= \frac{c_p^2}{2}(t(2p+1)\lambda_p - p\lambda_{p-1}, \lambda_{p+1})_{0,[-1,1]} \\ &= (p+1)\frac{c_p^2}{2}(\lambda_{p+1}, \lambda_{p+1})_{0,[-1,1]} = p+1.\end{aligned}$$

□

A.3 Specifications of Computational Resources

All numerical experiment in this work where performed on one of the two following computer clusters:

DELTA-Cluster MA-PDE

- Manufacturer: DELTA Computer Products GmbH
- Nodes: 6
- Cores (total): 192
- Memory (total): 768 GB RAM
- Interconnection: InfiniBand QDR (40 GBit)
- Operating system: openSUSE Linux 12.1
- Operator: Institute for Applied and Numerical Mathematics at KIT
- Node specifications:
  - Processors: 2 × AMD Opteron 6274 (2.2 GHz)
  - Cores: 32
  - Memory: 128 GB RAM
  - Memory per core: 4 GB RAM

Forschungshochleistungsrechner ForHLR I

- Manufacturer: MEGWARE Computer Vertrieb und Service GmbH
- Nodes: 540
- Cores (total): 10 800
- Memory (total): 32 768 GB RAM
- Interconnection: InfiniBand 4X FDR
- Operating system: Ret Hat Enterprise Linux 6
- Operator: Steinbuch Centre for Computing at KIT
- Node specifications:
  - Processors: 2 × Deca-Core Intel Xeon E5-2670 (2.5 - 3.3 GHz)
  - Cores: 20
  - Memory: 64 GB RAM
  - Memory per core: 3.2 GB RAM

Bibliography

[AK04] T. S. Angell and A. Kirsch. *Optimization Methods in Electromagnetic Radiation*. Springer monographs in mathematics. Springer, New York, NY, 2004.

[AM89] A. K. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Math. Comp.*, 52(186):255–274, 1989.

[AO00] M. Ainsworth and J. T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, New York, NY, 2000.

[ARW95] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995.

[AS64] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied mathematics series. Dover Publications, New York, NY, 1964.

[Aub67] J. P. Aubin. Behavior of the error of the approximate solutions of boundary value problems for linear elliptic operators by Galerkin's and finite difference methods. *Ann. Scuola Norm. Sup. Pisa*, 21(4):599–637, 1967.

[Bab73] I. Babuška. The finite element method with penalty. *Math. Comp.*, 27:221–228, 1973.

[Bak66] N. S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comput. Math. & Math. Phys.*, 6(5):101–135, 1966.

[BGR10] W. Bangerth, M. Geiger, and R. Rannacher. Adaptive Galerkin finite element methods for the wave equation. *Comput. Methods Appl. Math.*, 10(1):3–48, 2010.

[BH83] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the V-cycle. *SIAM J. Numer. Anal.*, 20(5):967–975, 1983.

[BR96] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math*, 4:237–264, 1996.

[BR99] W. Bangerth and R. Rannacher. Finite element approximation of the acoustic wave equation: Error control and mesh adaptation. *East-West J. Numer. Math.*, 7(4):263–282, 1999.

[BR01] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

[BR03] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, Basel, 2003.

[Bra94] A. Brandt. Rigorous quantitative analysis of multigrid, I: Constant coefficients two-level cycle with L_2-norm. *SIAM J. Numer. Anal.*, 31(6):1695–1730, 1994.

[Bra13] D. Braess. *Finite Elemente : Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, Berlin, 5th edition, 2013.

[BS97] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Appl. Numer. Math.*, 23(1):3–19, 1997.

[BS08] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Texts in applied mathematics 15. Springer, New York, NY, 3rd edition, 2008.

[BZ00] J. H. Bramble and X. Zhang. *The Analysis of Mulitgrid Methods*. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of Numerical Analysis*, volume 7, pages 173–415. North-Holland, Amsterdam, 2000.

[Car05] C. Carstensen. Estimation of higher sobolev norm from lower order approximation. *SIAM J. Numer. Anal.*, 42(5):2136–2147, 2005.

[CC89] G. Chavent and B. Cockburn. The local projection $P^0 - P^1$ discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO, Modél. Math. Anal. Numér.*, 23(4):565–592, 1989.

[CFP06] G. Cohen, X. Ferrieres, and S. Pernet. A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell's equations in time domain. *J. Comput. Phys.*, 217(2):340–363, 2006.

[Cre02] R. P. Crease. The most beautiful experiment. *Phys. World*, 9:19–20, 2002.

[CS91] B. Cockburn and C. W. Shu. The Runge–Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws. *RAIRO, Modél. Math. Anal. Numér.*, 25(3):337–361, 1991.

[Dem07] L. F. Demkowicz. *Computing with hp-Adaptive Finite Elements. Volume 1: One and Two Dimensional Elliptic and Maxwell Problems.* Chapman and Hall/CRC, Boca Raton, FL, 2007.

[DF15] W. Dörfler and S. Findeisen. Numerical optimization of a waveguide transition using finite element beam propagation. *Int. J. Numer. Model.*, 28(2):201–212, 2015.

[DFW16] W. Dörfler, S. Findeisen, and C. Wieners. Space-time discontinuous Galerkin discretizations for linear first-order hyperbolic evolution systems. *Comput. Methods Appl. Math.*, 16(3):409–428, 2016.

[DG14] L. F. Demkowicz and J. Gopalakrishnan. *An Overview of the Discontinuous Petrov–Galerkin Method.* In X. Feng, O. Karakashian, and Y. Xing, editors, *Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations*, pages 149–180. Springer, Cham, 2014.

[DH03] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems.* Wiley, Chichester, 2003.

[DKP+08] L. F. Demkowicz, J. Kurtz, D. Pardo, M. Paszenski, W. Rachowicz, and A. Zdunek. *Computing with hp-Adaptive Finite Elements. Volume 2: Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications.* Chapman and Hall/CRC, Boca Raton, FL, 2008.

[DKT07] M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes, V: Local time stepping and p-adaptivity. *Geophys. J. Int.*, 171:695–717, 2007.

[DLP+11] W. Dörfler, A. Lechleiter, M. Plum, G. Schneider, and C. Wieners. *Photonic Crystals: Mathematical Analysis and Numerical Approximation.* Oberwolfach Seminars, 42. Birkhäuser, Basel, 2011.

[Dör96] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[DPE12] D. A. Di Pietro and A. Ern. *Mathematical Aspects of Discontinuous Galerkin Methods.* Springer, Berlin, 2012.

[EDCM14] T. E. Ellis, L. F. Demkowicz, J. L. Chan, and R. D. Moser. Space-time DPG: Designing a method for massively parallel CFD, 2014. ICES REPORT 14–32.

[EG04] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Springer, New York, NY, 2004.

[EJ88] K. Eriksson and C. Johnson. An adaptive finite element method for linear elliptic problems. *Math. Comp.*, 50(182):361–383, 1988.

[EJ91] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems, I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1):43–77, 1991.

[EM12] M. Emmett and M. L. Minion. Toward an efficient parallel in time method for partial differential equations. *Comm. Appl. Math. and Comp. Sci.*, 7:105–132, 2012.

[Eva10] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, RI, 2nd edition, 2010.

[Fed64] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Comput. Math. & Math. Phys.*, 4(3):227–235, 1964.

[FFK+14] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder. Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):C635–C661, 2014.

[Gan15] M. J. Gander. *Multiple Shooting and Time Domain Decomposition Methods*. chapter *50 Years of Time Parallel Time Integration*, pages 69–113. Springer International Publishing, Cham, 2015.

[GG13] M. G. Gander and S Güttel. Paraexp: A parallel integrator for linear initial-value problems. *SIAM J. Sci. Comput.*, 35(2):C123–C142, 2013.

[GHN03] M. J. Gander, L. Halpern, and F. Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5):1643–1681, 2003.

[GSS07] M. Grote, A. Schneebeli, and D. Schötzau. Interior penalty discontinuous Galerkin method for Maxwell's equations: Energy norm error estimates. *J. Comput. Appl. Math.*, 204(2):375–386, 2007.

[GV07] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[Hac85] W. Hackbusch. Parabolic multi-grid methods. In *Proc. of the Sixth Int'l Symposium on Computing Methods in Applied Sciences and Engineering*, pages 189–197. North-Holland Publishing Co., 1985.

[Hac03] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, Berlin, 2nd edition, 2003.

[HB09] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Vieweg + Teubner, Wiesbaden, 3rd edition, 2009.

[Hip02] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11:237–339, 1 2002.

[HPS+15] M. Hochbruck, T. Pažur, A. Schulz, E. Thawinan, and C. Wieners. Efficient time integration for discontinuous Galerkin approximations of linear wave equations. *ZAMM*, 95(3):237–259, 2015.

[HST12] S. Hussain, F. Schieweck, and S. Turek. A note on accurate and efficient higher order Galerkin time stepping schemes for the nonstationary Stokes equations. *Open Numer. Methods J.*, 4:35–45, 2012.

[Hul72] B. L. Hulme. One-step piecewise polynomial Galerkin methods for initial value problems. *Math. Comp.*, 26(118):415–426, 1972.

[HV95] G Horton and S. Vandewalle. A space-time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.*, 16(4):848–864, 1995.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, 2nd edition, 1996.

[HW02] J. S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids. *J. Comput. Phys.*, 181(1):186–221, 2002.

[HW08] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in applied mathematics, 54. Springer, New York, NY, 2008.

[Ise09] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, 2nd edition, 2009.

[Jac99] J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, NY, 3rd edition, 1999.

[Joh93] C. Johnson. Discontinuous Galerkin finite elment methods for second order hyperbolic problems. *Comput. Methods Appl. Mech. Eng.*, 107(1):117–129, 1993.

[KB14] U. Köcher and M. Bause. Variational space-time methods for the wave equation. *J. Sci. Comput.*, 61(2):424–453, 2014.

[LeV08] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Lectures in mathematics. Birkhäuser, Basel, 2nd edition, 2008.

[LMT01] J.-L. Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of PDE's. *C. R. Acad. Sci. Paris, Ser. I*, 332(7):661–668, 2001.

[Mag16] F. Magoulés. *Parallel Scientific Computing*. Wiley, London, 2016.

[Mei11] A. Meister. *Numerik linearer Gleichungssysteme : Eine Einführung in moderne Verfahren*. Vieweg+Teubner, Wiesbaden, 4th edition, 2011.

[Mon04] P. Monk. *Finite Element Methods for Maxwells Equations*. Numerical mathematics and scientific computation Oxford science publications. Clarendon Press, Oxford, 2004.

[MS11] G. Matthies and F. Schieweck. Higher order variational time discretizations for nonlinear systems of ordinary differential equations. *Preprint no. 23, Fakultät für Mathematik, Otto–von–Guericke–Universität Magdeburg*, 2011.

[MSW95] J. A. Mackenzie, E. Süli, and G. Warnecke. A posteriori analysis for Petrov–Galerkin approximations of Friedrichs systems. Technical Report NA-95/01, Oxford University Computing Laboratory, 1995.

[MW11] D. Maurer and C. Wieners. A parallel block LU decomposition method for distributed finite element matrices. *Parallel Comput.*, 37(12):742–758, 2011.

[MW14] D. Maurer and C. Wieners. A scalable parallel factorization of finite element matrices with distributed Schur complements. submitted, 2014.

[Néd80] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numer. Math.*, 35(3):315–341, 1980.

[Neu13] M. Neumüller. *Space-Time Methods: Fast Solvers and Applications*. PhD thesis, Universität Graz, 2013.

[Nie64] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Commun. ACM*, 7(12):731–733, 1964.

[Nit68] J. Nitsche. Ein Kriterium für die Quasi-Optimalität des Ritzschen Verfahrens. *Numer. Math.*, 11(4):346–348, 1968.

[NS11] M. Neumüller and O. Steinbach. Refinement of flexible space-time finite element meshes and discontinuous Galerkin methods. *Comp. Vis. Sci.*, 14(5):189–205, 2011.

[NVV09] R. H. Nochetto, A. Veeser, and M. Verani. A safeguarded dual weighted residual method. *IMA J. Numer. Anal.*, 29(1):126–140, 2009.

[RH73] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-0479, Los Alamos Scientific Laboratory, 1973.

[Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edition, 2003.

[Sch15] A. Schulz. *Numerical Analysis of the Electro-Magnetic Perfectly Matched Layer in a Discontinuous Galerkin Discretization*. PhD thesis, Karlsruher Institut für Technologie, 2015.

[SS98] T. Sonar and E. Süli. A dual graph-norm refinement indicator for finite volume approximations of the euler equations. *Numer. Math.*, 78(4):619–658, 1998.

[Tha15] E. Thawinan. *Numerical approximation of higher-dimensional Continuum Dislocation Dynamics theory in single crystal plasticity*. PhD thesis, Karlsruher Institut für Technologie, 2015.

[TOS01] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, CA, 2001.

[Ver96] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley–Teubner, Chichester, 1996.

[Ver13] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Numerical Mathematics and Scientific Computation. Oxford Univ. Press, Oxford, 2013.

[Wer11] D. Werner. *Funktionalanalysis*. Springer, Berlin, 7th edition, 2011.

[Wie10] C. Wieners. A geometric data structure for parallel finite elements and the application to multigrid methods with block smoothing. *Comput. Visual. Sci.*, 13(4):161–175, 2010.

[Wil91] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Pract. Exper.*, 3(5):457–481, 1991.

[WW14] C. Wieners and B. Wohlmuth. Robust operator estimates and the application to substructuring methods for first-order systems. *ESAIM: Math. Mod. Numer. Anal.*, 48:1473–1494, 2014.

[YZLB07] H. Yang, W. Zulehner, U. Langer, and M. Baumgartner. A robust PDE solver for the 3D Stokes/Navier-Stokes systems on the grid environment. In *2007 8th IEEE/ACM International Conference on Grid Computing*, pages 145–152, 2007.

[Zul00] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.

[ZW14] S. Zhao and G. W. Wei. A unified discontinuous Galerkin framework for time integration. *Math. Methods Appl. Sci.*, 37(7):1042–1071, 2014.