TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Numerische Mechanik

Efficient Discontinuous Galerkin Methods for Wave Propagation and Iterative Optoacoustic Image Reconstruction

Svenja Madeleine Schoeder

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. dr.ir. Daniel J. Rixen
Prüfer der Dissertation:	
	1. Prof. Dr.-Ing. Wolfgang A. Wall
	2. Prof. Vasilis Ntziachristos, Ph.D.

Die Dissertation wurde am 13.08.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 13.03.2019 angenommen.

Ich muss etwas tun. Aber was?

Glück haben, antwortete Fuchur, *was sonst?*

Abstract

The acoustic wave equation describes the propagation of mechanical waves in gases and liquids, for example audible sound signals traveling through air or ultrasound propagating in the human body during a medical examination. Furthermore, it is used to predict room acoustics for construction projects or to optimize urban acoustics in terms of traffic noise. Numerical solution strategies are used to obtain approximate solutions of the acoustic wave equation that enable an accurate prediction of reflection patterns and volume levels. From a computational perspective, hyperbolic partial differential equations, i.e., equations describing wave propagation phenomena, are especially challenging since all physical characteristics like dispersion, diffraction, propagation at finite speed, and many more, must be represented within one numerical framework.

This thesis is divided into two parts, where the first part addresses the development of a highly efficient solver for the acoustic wave equation. The solver relies on a spatial discretization with the hybridizable discontinuous Galerkin method and temporal discretization with explicit Runge–Kutta schemes or arbitrary derivative time integration. The discretization methods allow for high orders of accuracy and yield optimal convergence as well as superconvergence in the L_2 errors of the primary fields. The algorithm utilizes matrix-free operator evaluation with fast quadrature and sum-factorization kernels, which exploits modern hardware efficiently. A high arithmetic density is reached by trading data movement from main memory for element-local operations that are closer to the arithmetic performance limit. Studies on the numerical properties in terms of convergence, temporal stability limits, and dispersion errors as well as on the computational properties in terms of computational timings, throughput, and scalability are presented. The applicability of the derived solver to urban acoustics and room acoustics is demonstrated based on a village and a cathedral, respectively.

The second part of this thesis addresses an inverse problem for which the efficient solution of the acoustic wave equation is a subproblem. Optoacoustic tomography is a comparably young medical imaging technique with applications ranging from small animal imaging to breast cancer detection. An object is illuminated with a laser pulse. The light transforms to heat, which in turn induces thermal expansion and thereby causes local pressure rises. The pressure propagates as ultrasonic wave through the object and is measured. To obtain images from the measurement data, an image reconstruction procedure is carried out. Commonly, the main contrast in the images is the optical absorption. Either the optical absorption coefficient itself or the absorbed energy is reconstructed. In this work, an image reconstruction method is developed that reconstructs the optical absorption and diffusion coefficients as well as the speed of sound and the mass density by exploitation and modeling of all relevant physical phenomena. Increasing the number of reconstruction variables worsens the conditioning of the inverse problem and two methods are developed to oppose the ill-conditioning. The methods are transferable to a wide range of inverse problems. The image reconstruction method is complemented by approaches to reduce the size of the computational domain in order to minimize computational expenses. Validation is carried out based on numerical examples and experimental data obtained with phantoms and in-vivo measurements of a mouse brain.

Zusammenfassung

Die akustische Wellengleichung beschreibt die Ausbreitung von mechanischen Wellen in Gasen und Flüssigkeiten, zum Beispiel die Ausbreitung von hörbarem Schall in Luft oder von Ultraschall im menschlichen Körper bei medizinischen Untersuchungen. Sie wird auch verwendet um die Raumakustik bei Bauvorhaben vorherzusagen oder die Städteplanung bezüglich Verkehrslärm zu verbessern. Näherungslösungen der akustischen Wellengleichung werden mittels numerischer Lösungsverfahren bestimmt, um Reflektionsmuster und Lautstärkepegel akkurat vorherzusagen. Hyperbolische partielle Differentialgleichungen, d.h. Gleichungen, die Wellenausbreitungsphänomene beschreiben, sind numerisch gesehen eine Herausforderung, da alle physikalischen Charakteristiken, wie Dispersion, Diffraktion, Ausbreitung bei endlicher Geschwindigkeit und viele andere, vom numerischen Verfahren abgebildet werden müssen.

Diese Arbeit ist in zwei Teile gegliedert. Der erste Teil adressiert die Entwicklung von einem hoch effizienten Lösungsverfahren für die akustische Wellengleichung. Es basiert auf räumlicher Diskretisierung mittels der hybridisierbaren diskontinuierlichen Galerkin Methode und zeitlicher Diskretisierung mittles expliziten Runge-Kutta-Verfahren oder *arbitrary derivative* Zeitintegration. Diese Methoden ermöglichen hohe Genauigkeitsordnungen und liefern optimale Konvergenz und Superkonvergenz in den L_2 Fehlern der Primärfelder. Der Algorithmus verwendet matrixfreie Operatorauswertungen mit schneller Quadratur und Summenfaktoriserungsroutinen, wodurch moderne Hardware effizient ausgenutzt wird. Eine hohe arithmetische Intensität wird dadurch erreicht, dass statt Datenbewegungen über den Arbeitsspeicher elementlokale Operationen durchgeführt werden, die näher an der arithmetischen Leistungsgrenze liegen. Studien der numerischen Eigenschaften bezüglich Konvergenz, Zeitschrittstabilität und Dispersionsfehler sowie der Recheneigenschaften bezüglich Rechenzeiten, Durchsatz und Skalierung werden präsentiert. Die Anwendbarkeit der entwickelten Methode auf urbane Akustik und Raumakustik wird anhand von einem Dorf und einer Kathedrale demonstriert.

Der zweite Teil der vorliegenden Arbeit beschäftigt sich mit einem inversen Problem, für welches die effiziente Lösung der akustischen Wellengleichung ein Teilproblem ist. Optoakustische Tomographie ist ein vergleichsweise junges medizinisches Bildgebungsverfahren mit Anwendungen von der Bildgebung an Kleintieren bis zur Brustkrebserkennung. Ein Objekt wird mit Laserlicht bestrahlt. Das Licht wird in Wärme umgewandelt, was wiederum zu einer Ausdehnung führt und dadurch lokale Druckanstiege verursacht. Der Druck propagiert als Ultraschallwelle durch das Objekt und wird gemessen. Um Bilder aus den Messdaten zu erhalten, muss ein Bildrekonstrutionsprozess durchgeführt werden. Üblicherweise ist der Hauptkontrast in den Bildern die optische Absorption, weshalb entweder der optische Absorptionskoeffizient oder die absorbierte Energie rekonstruiert werden. In dieser Arbeit wird ein Bildrekonstruktionsalgorithmus entwickelt, der durch die Ausnutzung und Modellierung aller relevanten physikalischen Phänomene neben den optischen Absorptions- und Diffusionskoeffizienten auch die Schallgeschwindigkeit und die Massendichte rekonstruiert. Durch die größere Anzahl von Rekonstruktionsvariablen verschlechtert sich die Konditionierung des inversen Problems und zwei Methoden, welche schlechter Konditionierung entgegenwirken, werden vorgestellt. Die Methoden sind auf eine große Bandbreite von inversen Problem übertragbar. Die Rechenzeit wird durch eine künstliche Verkleinerung des Berechnungsgebiets reduziert. Eine Validierung erfolgt mittels numerischer Beispiele sowie experimentellen Studien anhand von Phantomen und in-vivo Messungen an einem Mäusehirn.

Danksagung

Die vorliegende Arbeit entstand in den Jahren 2013 bis 2018 während meiner Arbeit als wissenschaftliche Mitarbeiterin am Lehrstuhl für numerische Mechanik der Technischen Universität München. Viele Personen haben mich in dieser Zeit begleitet und ich möchte an dieser Stelle meine Dankbarkeit dafür ausdrücken.

An erster Stelle möchte ich mich bei Professor Wolfgang A. Wall bedanken, der es mir ermöglicht hat meine Promotion durchzuführen und mich dabei fachlich und persönlich weiterzuentwickeln. Ich bedanke mich insbesondere für das entgegengebrachte Vertrauen und den Freiraum sowie für die unterstützenden Gespräche und die Möglichkeit offen und ehrlich kommunizieren zu können.

An zweiter Stelle möchte ich mich herzlichst bei Doktor Martin Kronbichler bedanken, der als fachlicher Betreuer bezüglich allen Implementierungs- und Effizienzfragen agiert hat. Die Zusammenarbeit hat mir immer große Freude bereitet und ich bin dankbar für die vielen netten und unkomplizierten Gespräche, die fachliche Beratung und ich betrachte es als eine Bereicherung, dass wir zusammenarbeiten konnten.

Als nächstes möchte ich mich bei Ivan Olefir bedanken. Als Doktorand am Helmholtz Zentrum Neuherbeg, war er mein Ansprechpartner für alle Fragen bezüglich optoakustischer Bildgebung. Ich bedanke mich für die freundschaftliche Zusammenarbeit und die ausdauernde Geduld mit der alle noch so naiven Fragen gewissenhaft beantwortet wurden. Außerdem bedanke ich mich bei Professor Vasilis Ntziachristos, der mir ermöglicht hat, an den Gruppentreffen seiner Forschungsgruppe teilzunehmen, wodurch ich mich fachlich weiterentwickeln konnte.

Außerdem möchte ich mich natürlich bei allen Kollegen am Lehrstuhl bedanken. Großer Dank gebührt auch Renata Nagl, die mir geduldig mit jedem meiner Anliegen geholfen hat.

Ich bedanke mich bei allen Studenten, die bei mir eine studentische Arbeit angefertigt haben, als Tutor oder als Hiwi gearbeitet haben, oder einfach durch den Besuch einer Lehrveranstaltung zu meiner Zeit am Lehrstuhl für Numerische Mechanik beigetragen haben. Die Lehre war ein bereichernder Teil meiner Promotion, durch die ich sehr viel gelernt habe. Die Diskussion mit Studenten hat immer wieder zu neuen Ideen und besserem Verständnis geführt.

Zu guter letzt danke ich meinen Freunden und meiner Familie von ganzem Herzen. Ich hätte meine Promotion niemals ohne den Rückhalt und die Unterstützung von ihnen abschließen können. Hier möchte ich explizit Felix für die erfrischenden Kaffeepausen danken, Felix, Tobi, Rinker und Steffi für die schönen Mittagessen am Mittwoch sowie Jitka, Viki, Natalja, Sybelle, Koral und Emil für den tollen Ausgleich. Ich danke meinen Eltern und meiner Schwester Clara für die guten Ratschläge und den Zuspruch in schwierigen Situationen. Der größte Dank gilt meiner Zwillingsschwester Nora. Deine Unterstützung, die unzähligen Telefonate, die tollen Urlaube und dein Verständnis waren und sind mir unersätzlich.

Mai 2019, Svenja Schoeder

Contents

Nomenclature

Abbreviations

ABC	absorbing boundary condition
ADER	arbitrary derivative
BEM	boundary element method
CFL	Courant–Friedrichts–Lewy condition
DA	diffusion approximation
DG	discontinuous Galerkin method
DIRK	diagonally implicit Runge–Kutta scheme
DOF	degree of freedom
FDTD	finite difference time domain method
FEM	finite element method
FLOP	floating point operation
HDG	hybridizable discontinuous Galerkin method
IR	impulse response
LSRK	low-storage Runge–Kutta
LTS	local time stepping
MPI	message passing interface
PBF	patched basis functions
PML	perfectly matched layer
RTE	radiative transfer equation

General Quantities

d	number of space dimensions
i	imaginary unit
$\hat{s}$	solid angle
t	time
Δt	time step
T	final time
$\boldsymbol{x}$	spatial coordinate vector

Continuous and Discretized Acoustics

a_{jl}, b_j	Butcher coefficients of a Runge–Kutta method
c	speed of sound

Cr	Courant number
d	improved velocity divergence
$\boldsymbol{D}$	DOF values of improved velocity divergence
δ	allowed time step difference between neighboring clusters
$\mathcal{E}_h$	set of all faces of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathcal{E}_h^0$	set of all inner faces of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathcal{E}_h^\partial$	set of all boundary faces of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathcal{E}_h^{\partial,\mathrm{abc}}$	set of all absorbing boundary faces of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathcal{E}_h^{\partial,\mathrm{dir}}$	set of all Dirichlet boundary faces of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathcal{E}_h^{\partial,\mathrm{neu}}$	set of all Neumann boundary faces of tesselation $\mathcal{T}_\mathrm{A}^h$
F	one face of set $\mathcal{E}_h$
$\boldsymbol{g}$	improved pressure gradient
$\boldsymbol{G}$	DOF values of improved pressure gradient
$\Gamma_\mathrm{A}^\mathrm{abc}$	absorbing boundary of Ω_A
$\Gamma_\mathrm{A}^\mathrm{dir}$	Dirichlet boundary of Ω_A
$\Gamma_\mathrm{A}^\mathrm{neu}$	Neumann boundary of Ω_A
h	characteristic element size
k	polynomial degree of shape functions
K	one element of tesselation $\mathcal{T}_\mathrm{A}^h$
∂K	boundary of one element K of tesselation $\mathcal{T}_\mathrm{A}^h$
$\mathbb{K}_\mathrm{impl}$	global system matrix for implicit time integration, defined in (2.17)
$\mathbb{K}$	stiffness matrix, defined in (2.21)
λ	pressure trace
$\boldsymbol{\Lambda}$	DOF values of pressure trace λ
μ	test function for the pressure trace λ
$\boldsymbol{M}$	DOF values of test function μ
$\boldsymbol{n}$	outward pointing normal vector
n_A^e	number of elements in $\mathcal{T}_\mathrm{A}^h$
$\mathbb{N}$	matrix holding the shape functions
Ω_A	acoustical domain
p	pressure
p_D	prescribed pressure values at Dirichlet boundary $\Gamma_\mathrm{A}^\mathrm{dir}$
p_0	initial pressure at $t = 0$
$\boldsymbol{P}$	DOF values of pressure
P	L_2 projection operator
q	test function for the pressure p
$\boldsymbol{Q}$	DOF values of test function q
$\mathbb{Q}$	mass matrix, defined in (2.21)
ρ	mass density
s	number of Runge–Kutta stages
$\mathbb{S}$	wave equation derivative operator
$\mathcal{T}_\mathrm{A}^h$	tesselation of acoustical domain Ω_A
$\partial\mathcal{T}_\mathrm{A}^h$	boundary of the tesselation $\mathcal{T}_\mathrm{A}^h$ containing inner element faces twice
τ	HDG specific stabilization parameter (typically $\tau = {}^1\!/_{c\rho}$)
$\boldsymbol{v}$	velocity

$\boldsymbol{v}_0$	initial velocity at $t = 0$
$\boldsymbol{V}$	DOF values of velocity
$\boldsymbol{w}$	test function for the velocity $\boldsymbol{v}$
$\boldsymbol{W}$	DOF values of test function $\boldsymbol{w}$
$\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E},$ $\mathbb{G}, \mathbb{H}, \mathbb{I}, \mathbb{J}, \mathbb{M}$	HDG specific element matrices

Perfectly Matched Layers (PMLs)

$\boldsymbol{A}$	gradient of the damping function
δ	PML thickness
$\boldsymbol{E}_j$	eigenvectors of $\boldsymbol{A}$ with $j \in [1, d]$
$\boldsymbol{F}_j$	row vectors of $\boldsymbol{G}^{-1} = [E_1 \ldots E_d]^{-1}$
$\boldsymbol{G}$	matrix summarizing the eigenvectors $\boldsymbol{E}_j$ of $\boldsymbol{A}$
$\boldsymbol{\gamma}$	damping function
$\Gamma_{\mathrm{A}}^{\mathrm{PML}}$	outer PML boundary
$\Gamma_{\mathrm{A,in}}^{\mathrm{PML}}$	interface between standard acoustic domain Ω_{A} and PML domain $\Omega_{\mathrm{A}}^{\mathrm{PML}}$
$\boldsymbol{I}$	identity matrix
$\boldsymbol{k}$	wave vector
$\Omega_{\mathrm{A}}^{\mathrm{PML}}$	PML domain
ω	angular frequency
s	number of distinct eigenvalues of $\boldsymbol{A}$
$\sigma, \boldsymbol{\sigma}$	scalar and vectorial absorption function
σ_{const}	constant absorption function
σ_n	polynomial absorption function of degree n
σ_{hyp}	hyperbolic absorption function
σ_{shyp}	shifted hyperbolic absorption function
z_j	auxiliary variable with $j \in [1, s]$

Optoacoustic Imaging

$\boldsymbol{c}$	values of discretized speed of sound
d	characteristic acoustic detector function
D	diffusion coefficient
$\boldsymbol{D}$	values of discretized diffusion coefficient
$\boldsymbol{F}_{\mathrm{L}}$	force vector of discretized optical problem
g	anisotropy of scattering
G	Grüneisen coefficient
$\boldsymbol{G}$	values of discretized Grüneisen coefficient
$\boldsymbol{g}_{\mu_{\mathrm{a}}}$	objective function gradient with respect to the absorption coefficient
$\boldsymbol{g}_D$	objective function gradient with respect to the diffusion coefficient
$\boldsymbol{g}_c$	objective function gradient with respect to the speed of sound
$\boldsymbol{g}_\rho$	objective function gradient with respect to the mass density

$\Gamma_{\mathrm{A}}^{\mathrm{mon}}$	monitored boundary of the acoustical domain Ω_{A}
$\Gamma_{\mathrm{L}}^{\mathrm{dir}}$	Dirichlet boundary of the optical domain Ω_{L}
$\Gamma_{\mathrm{L}}^{\mathrm{rob}}$	Robin boundary of the optical domain Ω_{L}
I	optical radiance
$\mathbb{K}_{\mathrm{A,PA}}, \mathbb{K}_{\mathrm{L,PA}}$	matrices of discretized initial pressure mapping
$\mathbb{K}_{\mathrm{A,M}}^{\mathrm{ac}}, \mathbb{K}_{\mathrm{A,F}}^{\mathrm{ac}}$	matrices of discretized acoustical problem
$\mathbb{K}_{\mathrm{L}}$	stiffness matrix of discretized optical problem
l	speed of light
$\mathcal{L}$	Lagrangian of optoacoustic inverse problem
μ_{a}	absorption coefficient
$\boldsymbol{\mu}_{\mathrm{a}}$	values of discretized absorption coefficient
μ_{s}	scattering coefficient
μ_{s}'	reduced scattering coefficient
Ω_{L}	domain of light propagation
P	probability density function for light scattering
$p_{\mathrm{s}}^{\mathrm{d}_j}(t_k)$	pressure at detector d_j at time t_k
$\boldsymbol{P}_{\mathrm{m},t_k}$	vector summarizing measured pressure values of all detectors at time t_k
$\boldsymbol{P}_{\mathrm{s},t_k}$	vector summarizing simulated pressure values of all detectors at time t_k
ϕ	light flux
ψ	test function for light flux ϕ
$\boldsymbol{\Phi}, \boldsymbol{\Psi}$	DOF values of the light flux and its test function ϕ, ψ
q	test function for projection of initial pressure field
$\boldsymbol{\rho}$	values of discretized mass density
S	emission source
$\mathcal{T}_{\mathrm{L}}^h$	tesselation of Ω_{L}

1 Introduction

In 1880, the photoacoustic effect converting light into acoustic signals has been discovered by Alexander Graham Bell [13], but only in the last three decades, a biomedical imaging modality, namely photoacoustic imaging (or equivalently optoacoustic imaging), emerged. In optoacoustic imaging, an object is illuminated by a short pulsed or intensity-modulated laser. The light distributes in the object according to its optical properties and is partly absorbed. The absorbed light transforms to a local pressure rise via thermal expansion. The pressure propagates through the object according to its mechanical properties and is eventually detected as ultrasound signal from which a two- or three-dimensional image of the object is reconstructed. Figure 1.1 summarizes the course of action of optoacoustic imaging.

In this work, the image reconstruction is going to be in the focus, i.e., the step that converts the detected acoustical signals to an image representing tissue properties. Since several physical effects are part of the image formation, the acoustic signals contain information of optical, thermodynamic, as well as mechanical tissue properties. Commonly, the main contrast in optoacoustic images is caused by the optical absorption: a pressure rise is generated in regions where light arrives and the tissue is optically absorbing. Therefore, optoacoustic images typically represent the optical absorption coefficient or the absorbed energy map.

Generally speaking, image reconstruction is an *inverse problem*. In a standard forward problem, the cause is known and the effect is determined, e.g., by simulation. In an inverse problem, however, the effect or a part of the effect is measured while the determination of the cause is attempted. If an optoacoustic image displays the absorbed energy map, the corresponding inverse problem is a source problem: from pressure measurements, the initial pressure field shall be determined. If the optoacoustic image displays the optical absorption coefficient, the corresponding inverse problem is a parameter problem: from pressure measurements, material parameters shall

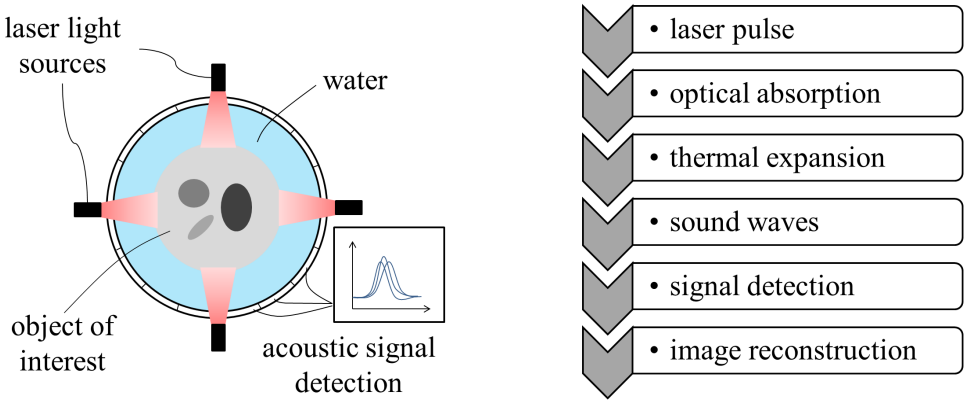


Figure 1.1: The optoacoustic imaging procedure: from laser light excitation to a medical image.

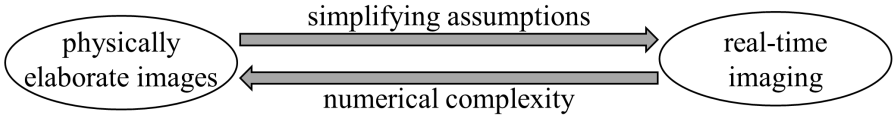


Figure 1.2: General trade-off between costs and benefits for optoacoustic imaging.

be determined. For both types of inverse problems, various solution strategies exist. In general, there is a trade-off between consideration and modeling of relevant physical phenomena on the one hand, versus simplifying assumptions on the other hand, which manifests in terms of image accuracy versus computational time, see Figure 1.2. Often, the assumption of spatially constant acoustical material parameters is made, i.e., speed of sound and mass density are equal in the entire body. This assumption is oversimplifying in several applications but eases the image reconstruction process. If the body is of spatially constant mechanical character, the induced pressure will propagate as circular, spherical, or cylindrical wave and an analytic description of the sound propagation and real-time imaging are possible [159]. If the body however has variations in its mechanical properties, the sound waves will reflect, diffract, refract, and scatter and numerical implementations to find an approximate solution of the acoustic wave equation must be employed in order to accurately predict the sound propagation. In this case, image reconstruction is usually based on an iterative procedure starting from an initial guess of the material properties and repeated evaluations of the forward problem. These algorithms need longer computational times but avoid image artifacts stemming from simplifications.

In this work, an optoacoustic image reconstruction algorithm is derived that is located on the left end of the spectrum shown in Figure 1.2 using a numerical implementation to solve the acoustic wave equation. From the relevant physical phenomena of light propagation, photoacoustic effect, and sound propagation, the sound propagation is numerically the most elaborate and hence the most expensive in terms of computational time. Therefore, an efficient solver for the acoustic wave equation is needed, which is the second main part of this work.

The acoustic wave equation is a hyperbolic partial differential equation that describes sound propagation in general heterogeneous tissues. It is based on the conservation of mass and linear momentum. Absorption due to viscous effects is neglected and only compressible waves are considered. Numerical methods are employed to find an approximate solution to the wave equation to a given level of accuracy and a numerical method is considered especially advantageous the faster the approximate solution is obtained. The acoustic wave equation has already been addressed by many numerical methods. Chronologically, the finite difference method was the first representative in the field of numerical schemes to find an approximate solution to the transient equation. However, with the advancement in computational resources and methodology, the finite element method emerged as a competitive tool applicable also to complex geometries and strongly varying material parameters. In the last two decades, discontinuous Galerkin (DG) methods have become highly popular for the discretization of transport phenomena as they offer good accuracy and are yet more robust than continuous finite elements at a similar computational complexity [74]. DG methods for the wave equation are the basis for the methods to be proposed in this thesis.

1.1 Objectives and Achievements

This thesis makes two major contributions: the development of an optoacoustic image reconstruction algorithm considering all relevant physical phenomena and the development of an acoustical solver that is flexible in terms of geometry, mesh, and material coefficients while it achieves high levels of performance. Thereby, the high performance acoustical solver is absolutely necessary to solve the optoacoustic inverse problem in a reasonable time. Furthermore, the acoustical solver is not only a subproblem of the image reconstruction algorithm but it is developed as a general acoustics solver with applications to various purely acoustic problems like urban and room acoustics. The two main contributions are addressed individually in more detail in the following.

For the development of an **optoacoustic image reconstruction method**, the following achievements are stated:

- The developed algorithm is the first method proposed in literature to model all relevant physical phenomena, namely diffusive light propagation with variable diffusion and absorption coefficient, the photoacoustic effect, and acoustic sound propagation with variable speed of sound and mass density.
- State of the art numerical concepts are used to discretize each of the problems efficiently, i.e., continuous finite elements for the optical problem, a mapping for the photoacoustic effect, and explicit hybridizable DG methods for the acoustical problem.
- The adjoint concept is utilized to derive parameter gradients and thereby, this algorithm is the first that allows for the optimization of absorption coefficient, diffusion coefficient, speed of sound, and mass density in one general framework without algorithmic restrictions on illumination and detector setup.
- The developed model allows to run simulations of the optoacoustic imaging process as forward problem and hence allows to visualize and gain insights into the signal formation process.
- Two approaches to counter ill-conditioning in inverse problems are proposed. One is based on a material identification for objects of known material composition. The other adapts the basis functions for the parameter fields during the iterative optimization. Thereby, distinct material patches are segmented and additionally, information of the body constitution can be transported from the most sensitive parameter to less sensitive parameters. Both methods are not only applicable to optoacoustics but to a variety of inverse problems.

For the development of a **high performance acoustical solver**, the following achievements are stated:

- The high-performance acoustical solver is based on hybridizable DG spatial discretization and explicit Runge–Kutta or arbitrary derivative (ADER) time integration preserving the superconvergence property that is typical for hybridizable DG. The ADER time integration also allows for local time stepping.
- An implementation is proposed that uses matrix-free operator evaluations with sum factorization reaching very high throughput on modern hardware. For ADER time integration,

a flexible basis change and a basis reduction are proposed that increase throughput and reduce operation counts.
- The developed solver is not only applicable for optoacoustic imaging but also to a large variety of real world problems, which is demonstrated for urban acoustics and room acoustics.
- A new perfectly matched layer formulation minimizing the number of auxiliary variables and allowing for complex geometries is proposed.

1.2 Outline

The thesis is organized in two parts with Part I addressing the high performance acoustical solver and Part II addressing the optoacoustic image reconstruction algorithm.

In **Chapter 2**, the acoustic wave equation is presented and a literature overview of numerical solution strategies is given. The spatial discretization of the acoustic wave equation with the hybridizable DG method is introduced and time discretization with implicit Runge–Kutta methods, explicit Runge–Kutta methods, and explicit ADER time integration are derived. An extension of the global ADER time integration to local time stepping is given. Convergence properties and the preservation of the superconvergence property typical for hybridizable DG methods are studied theoretically. Last, a numerical characterization in terms of convergence, temporal stability limits, and dispersion and dissipation error is presented. In **Chapter 3**, the implementation aspects of the high performance acoustic solver are developed. The procedures of basis change and basis reduction for ADER are presented, operation counts are derived and computational performance is measured in terms of solution time, throughput, the roofline performance model, and scalability. The novel perfectly matched layer formulation is given in **Chapter 4**. First, the complex coordinate stretching for arbitrary problem geometries is derived. After that, the formulation is studied in terms of stability and absorption functions. The spatial and temporal discretizations are presented and numerical examples demonstrate basic properties and give rules for the dimensioning of perfectly matched layers. Applications of the acoustical solver to urban acoustics and room acoustics are presented in **Chapter 5**.

Part II starts with an introduction to the functional principle and reconstruction methods for optoacoustic imaging in **Chapter 6**. In **Chapter 7**, the physical and numerical models are derived. The objective function is introduced and the parameter gradients are calculated using the adjoint concept. The solution algorithm is explained including a line search and a checkpointing approach. A proof of concept and numerical examples are given. In **Chapter 8**, a method to reduce the computational domain for tomographs with large void areas is presented. The full view and the limited view scenario are studied. The two approaches opposing the ill-conditioning, namely the variable basis function adaption and the material identification, are described in **Chapter 9**. Applications of the derived image reconstruction algorithm to experimentally obtained signals are presented in **Chapter 10**.

Finally, the conclusion of this work in **Chapter 11** summarizes the results and accomplishments and points out the directions for future developments for the acoustical solver and the optoacoustic image reconstruction method.

I. The Hybridizable Discontinuous Galerkin Method for Waves

2 The Acoustic Wave Equation and its Discretization

Sound propagation is the transport of kinetic and potential energy through a continuum. In frictionless continua, the potential energy takes form of compression, otherwise, e.g. in solid continua, the potential energy can also take form of displacement strains. Physical processes in continua generally obey the conservation of mass, linear momentum, and energy. The following explications show how the conservation laws are stated in the context of acoustics.

The conservation of mass is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0,$$

with the mass density ρ and the particle velocity $\boldsymbol{v}$. The conservation of linear momentum is written as

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla(\rho \boldsymbol{v}) + \nabla p - \nabla \cdot \boldsymbol{\tau} = 0,$$

with the pressure p and the stress tensor $\boldsymbol{\tau}$. If sound propagation in a frictionless fluid is considered, the stress tensor becomes zero $\boldsymbol{\tau} = \mathbf{0}$. For viscous fluids or solids supporting shear waves, the stress tensor can be given as $\boldsymbol{\tau} = \mu \nabla \boldsymbol{v}$ with the dynamic viscosity μ. In this derivation only compressional waves are considered and hence $\boldsymbol{\tau} = \mathbf{0}$.

Thermodynamic considerations on sound propagation reveal that the entropy s is conserved under the assumption that there is no friction and no heat flow,

$$\frac{\partial s}{\partial t} + \boldsymbol{v} \cdot \nabla s = 0.$$

Additionally, thermodynamics provide the general equation of state for the pressure in relation to the current mass density and entropy

$$p = p(\rho, s).$$

Under the assumption of no heat flow, the thermodynamic equation of state for the pressure reduces to

$$p = p(\rho).$$

In the context of acoustics, it is common to split the variable fields $p, \rho, \boldsymbol{v}, s$ into a temporally constant and a fluctuating part. For the velocity, the reference value is assumed to be zero,

i.e., sound propagation in a non-moving fluid is considered:

$$\begin{aligned} p &= p_{\text{ref}} + p' \\ \rho &= \rho_{\text{ref}} + \rho' \\ \boldsymbol{v} &= \boldsymbol{v}' \\ s &= s_{\text{ref}} + s' \end{aligned}$$

In the following, the fluctuation share is assumed to be small compared to the reference share.

The equation of state is linearized in terms of the pressure and density fluctuations

$$p' = c^2 \rho',$$

with the speed of sound c. Introducing this relation and the split of the fields by means of reference and fluctuation shares into the conservation of mass and linear momentum gives

$$\frac{\partial p'}{\partial t} + c^2 \rho_{\text{ref}} \nabla \cdot \boldsymbol{v}' = 0, \tag{2.1}$$

$$\frac{\partial \boldsymbol{v}'}{\partial t} + \frac{1}{\rho_{\text{ref}}} \nabla p' = 0, \tag{2.2}$$

using the linearizing assumption that products of fluctuations are negligible.

Differentiating equation (2.1) with respect to time and subsequently inserting equation (2.2) for the partial time derivative of the velocity yields the acoustic wave equation for heterogeneous frictionless fluids:

$$\frac{\partial^2 p'}{\partial t^2} - c^2 \rho_{\text{ref}} \nabla \cdot \left(\frac{1}{\rho_{\text{ref}}} \nabla p' \right) = 0$$

In case of a spatially constant density, the equation simplifies to

$$\frac{\partial^2 p'}{\partial t^2} - c^2 \nabla \cdot \nabla p' = 0.$$

In the remainder of the work, the fluctuation and reference share will not be explicitly specified and for the sake of convenience and readability, the acoustic wave equation will be written as:

$$\frac{\partial^2 p}{\partial t^2} - c^2 \rho \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = 0 \tag{2.3}$$

In summary, the acoustic wave equation is derived from the Navier–Stokes equations by neglect of viscosity yielding the Euler equations and a subsequent linearization.

The acoustic wave equation is a hyperbolic partial differential equation of second order. In contrast to elliptic equations, it describes the propagation of information at finite speed c and in contrast to parabolic equations, it allows for the transport of discontinuities. The literature on the acoustic wave equation is wide, however the original publication on the one-dimensional wave equation [42], an essay of Stokes on fluids in motion [157], Feynman's lecture on acoustics [58], and a standard text book by Pierce [128] shall be mentioned explicitly. For simple

setups, e.g. spatially constant material properties c, ρ and simple geometries or in one dimension, analytic solutions are available, otherwise numerical solution strategies must be used.

In the derivation of the acoustic wave equation, an infinite medium is assumed. For a multitude of numerical methods, the restriction to a finite domain with boundaries and correspondent boundary conditions is required. Boundaries can be a soft walls, a hard walls, walls with specific acoustic characteristics, or boundaries mimicking an infinite domain. They will be introduced in the following chapters. In order to solve the wave equation, initial conditions must be specified for the pressure and the first time derivative of the pressure (in case the wave equation in its standard form is considered) or for the pressure and the velocity (in case the first order system of equations is considered).

2.1 Numerical Solution Strategies in the Literature

Analytical solutions of the acoustic wave equation (2.3) are only available on simple domains and constant or artificial material parameters. In applications, numerical solution strategies need to be employed in order to quantitatively predict the sound propagation. Since computational resources are sufficiently available and accessible by elementary coding approaches, a huge variety of numerical schemes evolved to supply approximate solutions to the acoustic wave equation and also other hyperbolic problems, like the elastic wave equation or Maxwell's equations. Their high relevance is based on the wide variety of applications, e.g., seismology, room acoustics, electromagnetics, and medicine.

For several decades, wave equations were solved in the frequency domain considering the Helmholtz equation rather than in the time domain, which reduces the problem complexity and hence the computational cost significantly. The Helmholtz equation is derived from the acoustic wave equation assuming that the pressure solution can be separated into a time dependent function $p^{\mathrm{t}}(t)$ and a space dependent function $p^{\mathrm{x}}(\boldsymbol{x})$ as product $p(\boldsymbol{x}, t) = p^{\mathrm{x}}(\boldsymbol{x}) \cdot p^{\mathrm{t}}(t)$. Insertion of this expression into the wave equation (2.3) and rearranging yields

$$\frac{1}{p^{\mathrm{t}}} \frac{\partial^2 p^{\mathrm{t}}}{\partial t^2} = c^2 \rho \frac{\nabla \cdot \left(\frac{1}{\rho} p^{\mathrm{x}} \right)}{p^{\mathrm{x}}},$$

where the left side depends only on the time while the right side depends only on the spatial coordinate. Hence, both sides must equal a constant, which results in an ordinary differential equation in time for p^{t} and the Helmholtz equation for p^{x}. The temporal solutions p^{t} are generally superpositions of sine and cosine functions. The Helmholtz equation can be solved analytically for simple geometries and boundary conditions. Otherwise, direct numerical solution strategies like finite differences, finite elements, or semi-analytic concepts e.g. using Green's functions, or even geometrical approximations are required. The concepts are similar as for the time domain wave equation.

In the time domain, the accurate simulation of high frequency waves is very elaborate because high frequencies do not only require high spatial resolution but also high temporal resolution. One group of solution techniques circumventing the induced difficulties are geometric methods based on ray-tracing. They are applicable to room acoustics but they are not sufficiently accurate for low-frequencies and diffraction [153]. Another approach is to assume that acoustic

waves propagate diffusely in rooms after a sufficient number of reflections and a diffusion equation model is solved, see e.g. [55]. The accurate prediction of sound propagation over a wide frequency range however requires to solve the acoustic wave equation, either with direct numerical schemes with a sufficiently fine discretization or semi-analytic schemes using fundamental solutions of the wave equation.

Boundary element methods (BEM) can be understood as semi-analytic schemes. They employ Green's functions, which are solutions to the wave equation or Helmholtz equation with a point source. Together with Kirchhoff's integral theorem (also known as Green's second theorem), Green's functions are used to relate the solution in a domain to the solution on an enclosing surface, i.e., its boundary. Discretization of the boundary and the integral equation then yields a matrix system to obtain approximate solutions on the enclosing surface. The domain solution is found in a postprocessing step using Green's second theorem once more. One advantage of BEMs is that only the boundary needs discretization with low storage requirements and simple meshing. Thereby, exterior and interior problems are of the same complexity. Disadvantages of the BEM are the difficult mathematical analysis and the treatment of singularities in corners and edges. Also, they are only applicable to problems for which Green's functions are known and calculatable. See [39] for early findings on boundary element methods or [140] for a more recent overview.

Straightforward finite difference time domain (FDTD) methods are representatives for direct numerical schemes and were widely used for lower frequencies [19]. The application to higher frequencies is restricted by the available computational resources. The Yee scheme [179] is a well-known FDTD method, which has been and still is successfully applied to hyperbolic equations and in particular to Maxwell's equations. Finite difference methods naturally comprise diagonal mass matrices. Their drawbacks are the limited applicability to complex geometries, heterogeneous materials, and restrictions in combination with adaptivity. More recently, adaptive rectangular decomposition was proposed, which is a domain decomposition technique relying on the analytic solution of the wave equation in rectangles and additional interface handling [116, 132]. It is for now limited to homogeneous sound speed distributions. Finite element methods (FEM) do not suffer from the drawbacks of FDTD and the adaptive rectangluar decomposition approach. The mass matrix, however, is generally sparse but not diagonal, which motivated the subsequent developments. One approach is the mass lumping introduced in [37] for the one-dimensional wave equation with numerical quadrature and shape functions based on Gauss-Lobatto points, later also called spectral elements, yielding diagonal mass matrices. The other popular approach is the discontinuous Galerkin method (DG) yielding block-diagonal mass matrices as first proposed in [133] and further developed in e.g. [32, 35]. An overview on DG methods is given in [74]. By overcoming the drawbacks of finite difference methods and maintaining the beneficial property of easily inverted mass matrices, DG is by now a favored method to solve hyperbolic problems in the time domain. In the last decades, several contributions further improved DG in terms of computational time and numerical properties. DG can be applied to the spatial coordinates only [66], to the time coordinate only (which is rather uncommon), or to both [127]. If only the spatial coordinate is discretized using DG, the temporal dependency can be discretized e.g. with Runge–Kutta or linear multistep methods. In [118], nodal DG is used for the spatial coordinates while the temporal dependency is discretized using a pseudospectral Fourier method for the linear Euler equations.

Due to the hyperbolicity of the wave equation, the spatial as well as the temporal discretization require high resolution to minimize dissipation and dispersion errors. To fulfill this resolution requirement, high order schemes are suitable, since they generally outperform linear elements in terms of time to solution for given accuracy in the presence of the mentioned numerical challenges. A fair comparison, however, is difficult, even though several treatises addressed the issue [83, 106] or [1] predicting orders $2k+3$ and $2k+2$ for dispersion and dissipation error, respectively, based on analysis of the symbol of the discretized differential error. The effect of integration accuracy in the context of spectral elements is studied in [60] revealing a high impact of underintegration on dispersion and dissipation errors.

Another controversially discussed development is the hybridizable discontinuous Galerkin method (HDG), which introduces a new set of trace degrees of freedom on element faces. The fluxes underlying the HDG method enable more accurate solutions than previously known DG methods. For polynomials of degree k, HDG yields optimal convergence rates of order $k+1$ in the mesh size for both pressure and velocity solutions. By means of an element-wise postprocessing step, a $k+2$ superconvergent pressure solution can be recovered on general meshes. The theoretical fundamentals for the superconvergence property of HDG are given in [34] and specifically for acoustics in [31]. In combination with implicit Runge–Kutta schemes, HDG yields a linear system of equations only for the trace degrees of freedom; for explicit Runge–Kutta schemes, no global system has to be solved and the formulation resembles previously established DG implementations that merely differs in the definition of the numerical flux [97, 120, 155]. In [91, 176], implicit HDG is compared to continuous FEM, showing that their performance is competitive or HDG outperforms continuous FEM. Especially in 3D and for time dependent problems with effective preconditioning, HDG is beneficial. In [97], implicit and explicit Runge–Kutta HDG methods are compared, revealing a clear benefit for explicit time integration with a speed up of a factor between 10 and 1000. Runge–Kutta methods are often used for time integration because they are profoundly understood and easy to implement by a combination of several evalutions of the right hand side. One drawback however is the Butcher barrier stating that the number of stages and thus function evaluations to achieve a given order of accuracy increases overproportionally after order four. Hence, the efficiency of Runge–Kutta time integrators is debatable in combination with high order spatial discretizations. In [36], time integration by Stormer-Numerov schemes in combination with HDG is proposed, which yields an energy conserving method. The computational properties, however, are disadvantageous. An explicit time integration approach overcoming these drawbacks is the arbitrary derivative (ADER) approach presented in [47, 50, 149, 150]. It is based on a truncated Taylor expansion of the solution field in time, where time derivatives are replaced by spatial derivatives using the Cauchy–Kowalevski (or Lax–Wendroff) procedure. ADER allows for arbitrary high order discretization in time since the Taylor series can be truncated at any term. The publications [149, 150] presented the ADER time integration in the context of finite volume spatial discretizations. In those studies, the required higher spatial derivatives are obtained from the cell averages by a reconstruction procedure. An extensive study of the numerical properties of the ADER finite volume scheme can be found in [48]. In [47], the ADER time integration was extended to DG to solve the elastic wave equation. The application to a set of unified first order hyperbolic systems was recently published in [50] using both finite volumes and DG. In [68], ADER time integration is combined with DG and convergence rates of $2k+1$ are claimed due to a specific postprocessing based on convo-

lution with a B-spline kernel. Their method is applied to one-dimensional hyperbolic problems and requires translation invariant meshes though.

For explicit time integrators, the time step size is restricted by the Courant–Friedrichs–Lewy (CFL) condition [40]. Especially for meshes with strongly varying mesh sizes or variations in the speed of sound, this is a clear drawback. Local time stepping (LTS) addresses this issue by advancing each element in time with its optimal time step, thus decreasing the number of element updates. Furthermore, dispersion and dissipation errors are reduced with LTS since large elements are not forced to use a time step much lower than their optimal time step, which is computational expensive and known to introduce dispersion errors [45]. In literature, several LTS schemes for DG have been proposed: In [170], LTS based on Adams–Bashforth time integration was used and combined with a spectral DG method, achieving spectral accuracy in space as well as in time. Runge–Kutta schemes with LTS were proposed in e.g. [61, 65, 129]. For ADER time integration, LTS was proposed in [49] in the context of geophysics and elastic waves. An approach to circumvent strict CFL restrictions on adaptive meshes is given in [126], which is based on a subspace projection step.

Here, the spatial discretization of the wave equation using HDG is described in Section 2.2. Implicit Runge–Kutta time integration as a review of the historical developments is given in Section 2.3.1 followed by the presentation of explicit Runge–Kutta time integration and ADER time integration in Sections 2.3.2 and 2.3.3, respectively. The extension of ADER to LTS is given in Section 2.3.4. After that, the convergence properties of the ADER HDG discretization are examined in Section 2.5.1 and last, numerical examples are given in Section 2.5.

2.2 Spatial Discretization with the Hybridizable Discontinuous Galerkin Method

Starting point for the spatial discretization of the acoustic wave equation (2.3) is the first order formulation in terms of the pressure p and the velocity $\boldsymbol{v}$ on a bounded d-dimensional domain Ω_A and the time interval $[0, T]$ with final time T:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\rho}\nabla p = 0 \qquad \text{in} \quad \Omega_\mathrm{A} \times [0, T], \tag{2.4}$$

$$\frac{\partial p}{\partial t} + c^2 \rho \nabla \cdot \boldsymbol{v} = 0 \quad \text{in} \quad \Omega_\mathrm{A} \times [0, T]. \tag{2.5}$$

It is complemented by boundary conditions on the boundary of the domain $\Gamma_\mathrm{A} = \partial\Omega_\mathrm{A}$ with the parts $\Gamma_\mathrm{A}^\mathrm{dir} \cup \Gamma_\mathrm{A}^\mathrm{neu} \cup \Gamma_\mathrm{A}^\mathrm{abc} = \Gamma_\mathrm{A}$

$$p = p_\mathrm{D} \qquad \text{on} \quad \Gamma_\mathrm{A}^\mathrm{dir} \times [0, T], \tag{2.6}$$

$$\boldsymbol{v} \cdot \boldsymbol{n} = 0 \qquad \text{on} \quad \Gamma_\mathrm{A}^\mathrm{neu} \times [0, T], \tag{2.7}$$

$$\boldsymbol{v} \cdot \boldsymbol{n} - \frac{1}{c\rho} p = 0 \quad \text{on} \quad \Gamma_\mathrm{A}^\mathrm{abc} \times [0, T], \tag{2.8}$$

and initial conditions

$$p(t = 0) = p_0 \quad \text{in} \quad \Omega_\mathrm{A}, \tag{2.9}$$

$$\boldsymbol{v}(t = 0) = \boldsymbol{v}_0 \quad \text{in} \quad \Omega_\mathrm{A}. \tag{2.10}$$

Equations (2.6) and (2.7) are a Dirichlet boundary condition for the pressure and a Neumann boundary condition in terms of the velocity, respectively. Condition (2.8) is an absorbing boundary condition (ABC) of first order [53]. It is a simple approximation of the Sommerfeld radiation condition to mimic an infinite medium. Waves impinging normal to the boundary are perfectly absorbed and no spurious reflections are generated by the artificial boundary. Non-normal waves however can generate reflections with up to 17% amplitude of the original wave [53]. If the simulation is sensitive to reflections at boundaries, higher order approximations to the Sommerfield condition or perfectly matched layers (PMLs) can be utilized. PMLs are addressed in Chapter 4 of this work. For the following derivation of a spatial discretization for the wave equation, boundary conditions are however restricted to the three types mentioned in equations (2.6)–(2.8).

For convenience, the following abbreviations are introduced for vector valued functions $\boldsymbol{a}, \boldsymbol{b} \in (L_2(D))^d$ and scalar valued functions $a, b \in L_2(D)$ on a domain D in d space dimensions and on a domain E in $d-1$ space dimensions

$$\int_D \boldsymbol{a} \cdot \boldsymbol{b} \,\mathrm{d}D = (\boldsymbol{a}, \boldsymbol{b})_D \,, \qquad \int_D ab \,\mathrm{d}D = (a, b)_D \,,$$
$$\int_E \boldsymbol{a} \cdot \boldsymbol{b} \,\mathrm{d}E = \langle \boldsymbol{a}, \boldsymbol{b} \rangle_E \,, \qquad \int_E ab \,\mathrm{d}E = \langle a, b \rangle_E \,.$$

The computational domain Ω_A is tesselated into a triangulation $\mathcal{T}_\mathrm{A}^h$ of n_A^e elements K. In two dimensions, the elements are quadrilaterals or triangles with straight or curved edges; in three dimensions, they are hexahedra or tetrahedra with straight or curved surfaces. The boundary of the tesselation $\partial\mathcal{T}_\mathrm{A}^h$ is the set of all element boundaries ∂K and hence contains inner faces twice and boundary faces once. In contrast, the set of all faces $\mathcal{E}_h$ contains every face only once. It can be split into the set containing all inner faces $\mathcal{E}_h^0$ and the set containing all boundary faces $\mathcal{E}_h^\partial$, which is further divided into the sets containing all faces on the Dirichlet boundary, all faces on the Neumann boundary, and all faces on the absorbing boundary $\mathcal{E}_h^{\partial,\mathrm{dir}}, \mathcal{E}_h^{\partial,\mathrm{neu}}, \mathcal{E}_h^{\partial,\mathrm{abc}}$, respectively, with $\mathcal{E}_h^{\partial,\mathrm{dir}} \cup \mathcal{E}_h^{\partial,\mathrm{neu}} \cup \mathcal{E}_h^{\partial,\mathrm{abc}} = \mathcal{E}_h^\partial$.

In order to derive the HDG spatial discretization for the acoustic wave equation, equations (2.4)–(2.5) are multiplied with test functions $\boldsymbol{w}, q$ and integrated over all elements K of the tesselation $\mathcal{T}_\mathrm{A}^h$,

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{v}}{\partial t}\right)_K + \left(\boldsymbol{w}, \frac{1}{\rho}\nabla p\right)_K = 0,$$
$$\left(q, \frac{\partial p}{\partial t}\right)_K + \left(q, c^2\rho\nabla \cdot \boldsymbol{v}\right)_K = 0.$$

Next, integration by parts is performed assuming elementwise constant material properties

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{v}}{\partial t}\right)_K - \left(\nabla \cdot \boldsymbol{w}, \frac{1}{\rho}p\right)_K + \left\langle \boldsymbol{w} \cdot \boldsymbol{n}, \frac{1}{\rho}p \right\rangle_{\partial K} = 0,$$
$$\left(q, \frac{\partial p}{\partial t}\right)_K - \left(\nabla q, c^2\rho\boldsymbol{v}\right)_K + \left\langle q, c^2\rho\boldsymbol{v} \cdot \boldsymbol{n} \right\rangle_{\partial K} = 0.$$

As in every DG method, the primary quantities evaluated at the element boundaries are replaced by new variables $\hat{p}, \hat{\boldsymbol{v}}$ for the pressure and the velocity, respectively,

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{v}}{\partial t}\right)_K - \left(\nabla \cdot \boldsymbol{w}, \frac{1}{\rho} p\right)_K + \left\langle \boldsymbol{w} \cdot \boldsymbol{n}, \frac{1}{\rho} \hat{p} \right\rangle_{\partial K} = 0,$$
$$\left(q, \frac{\partial p}{\partial t}\right)_K - \left(\nabla q, c^2 \rho \boldsymbol{v}\right)_K + \left\langle q, c^2 \rho \hat{\boldsymbol{v}} \cdot \boldsymbol{n} \right\rangle_{\partial K} = 0.$$

In contrast to standard DG methods, HDG introduces a new variable λ, called trace variable, to replace the pressure at element boundaries $\hat{p}$. The velocity $\hat{\boldsymbol{v}}$ at element boundaries is replaced by the velocity itself plus a stabilization term penalizing the difference between the pressure and the trace field

$$\hat{\boldsymbol{v}} \cdot \boldsymbol{n} = \boldsymbol{v} \cdot \boldsymbol{n} + \tau(p - \lambda).$$

The stabilization parameter τ must be positive to ensure stability (see Section 2.5 in [120]) and is generally chosen as

$$\tau = \frac{1}{c\rho}.$$

Introducing these definitions for the quantities $\hat{\boldsymbol{v}}, \hat{p}$ gives

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{v}}{\partial t}\right)_K - \left(\nabla \cdot \boldsymbol{w}, \frac{1}{\rho} p\right)_K + \left\langle \boldsymbol{w} \cdot \boldsymbol{n}, \frac{1}{\rho} \lambda \right\rangle_{\partial K} = 0,$$
$$\left(q, \frac{\partial p}{\partial t}\right)_K + \left(q, c^2 \rho \nabla \cdot \boldsymbol{v}\right)_K + \left\langle q, c^2 \rho \tau (p - \lambda) \right\rangle_{\partial K} = 0.$$

Since the trace variable λ is an additional variable, an additional equation must be stated to close the problem. The system of equations is completed by weakly enforcing the continuity of the normal component of the velocity across element boundaries with test function μ

$$\langle \mu, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial K} + \langle \mu, \tau(p - \lambda) \rangle_{\partial K} = 0.$$

Element boundaries coinciding with the domain boundary have to take into account the given boundary conditions. Summation over all elements yields the final weak form

$$\left(\boldsymbol{w}, \frac{\partial \boldsymbol{v}}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^h} - \left(\nabla \cdot \boldsymbol{w}, \frac{1}{\rho} p\right)_{\mathcal{T}_\mathrm{A}^h} + \left\langle \boldsymbol{w} \cdot \boldsymbol{n}, \frac{1}{\rho} \lambda \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} = 0,$$
$$\left(q, \frac{\partial p}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^h} + \left(q, c^2 \rho \nabla \cdot \boldsymbol{v}\right)_{\mathcal{T}_\mathrm{A}^h} + \left\langle q, c^2 \rho \tau (p - \lambda) \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} = 0,$$
$$\langle \mu, \boldsymbol{v} \cdot \boldsymbol{n} \rangle_{\partial \mathcal{T}_\mathrm{A}^h} + \langle \mu, \tau(p - \lambda) \rangle_{\partial \mathcal{T}_\mathrm{A}^h} - \left\langle \frac{1}{c\rho} \lambda, \mu \right\rangle_{\Gamma_\mathrm{A}^\mathrm{abc}} = 0.$$

The homogeneous Neumann condition is naturally fulfilled by the third equation, the absorbing boundary condition contributes with an additional term on the relevant faces, and the Dirichlet

boundary conditions for the pressure will be weakly fulfilled by enforcing them on the trace field as explained below.

For the discretized counterparts $\boldsymbol{v}_h, p_h$ of the continuous velocity and pressure fields, $\boldsymbol{v}, p$ the function spaces $\boldsymbol{V}_h$ and P_h are defined

$$\boldsymbol{V}_h = \left\{ \boldsymbol{v}_h \in \left(L_2(\Omega)\right)^d : \boldsymbol{v}_h|_K \in \left(\mathcal{P}_k(K)\right)^d \right\},$$
$$P_h = \left\{ p_h \in L_2(\Omega) : p_h|_K \in \mathcal{P}_k(K) \right\}.$$

In each element K, polynomials $\mathcal{P}_k$ of degree k are utilized. Over element boundaries, no continuity is imposed, making the global function space discontinuous. The function space for the trace field is defined as

$$L_h = \left\{ \lambda_h \in L_2(\mathcal{E}_h) : \lambda_h|_F \in \mathcal{P}_k(F), \forall F \in \mathcal{E}_h \right\},$$

which generally gives discontinuities between faces. To incorporate the Dirichlet boundary condition (2.6), the space is restricted to

$$L_h(p_\mathrm{D}) = \left\{ \lambda_h \in L_h : \ \lambda_h = \mathsf{P} p_\mathrm{D} \text{ on } \Gamma_\mathrm{A}^\mathrm{dir} \right\}.$$

The operator P denotes the L_2 projection. Thereby, the Dirichlet boundary condition on the pressure is weakly enforced employing the trace variable. Figure 2.1 displays node locations representing the degrees of freedom in the typical HDG setup on a two-dimensional mesh of two elements with quadratic polynomials. The approximation of the weighting functions follows the

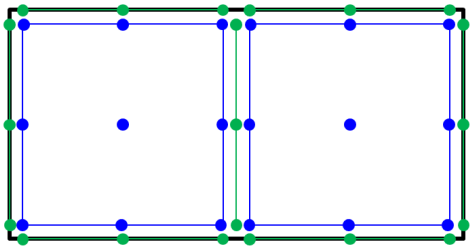


Figure 2.1: Nodes of an HDG discretization consisting of two quadrilateral elements. Blue dots represent the nodes for pressure and velocity while green dots label trace nodes. Taken from [148].

same approach as for the solution fields, i.e., $\boldsymbol{w}_h \in \boldsymbol{V}_h$, $q_h \in P_h$, and $\mu_h \in L_h(0)$, except that the weighting function for the trace field fulfills the homogeneous Dirichlet boundary conditions. With these definitions, the problem statement reads: Find $p_h \in P_h, \boldsymbol{v}_h \in \boldsymbol{V}_h, \lambda_h \in L_h(p_\mathrm{D})$ such that for all $q_h \in P_h, \boldsymbol{w}_h \in \boldsymbol{V}_h, \mu_h \in L_h(0)$

$$\left(\boldsymbol{w}_h, \frac{\partial \boldsymbol{v}_h}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^h} - \left(\nabla \cdot \boldsymbol{w}_h, \frac{1}{\rho} p_h\right)_{\mathcal{T}_\mathrm{A}^h} + \left\langle \boldsymbol{w}_h \cdot \boldsymbol{n}, \frac{1}{\rho} \lambda_h \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} = 0, \tag{2.11}$$

$$\left(q_h, \frac{\partial p_h}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^h} + \left(q_h, c^2 \rho \nabla \cdot \boldsymbol{v}_h\right)_{\mathcal{T}_\mathrm{A}^h} + \left\langle q_h, c^2 \rho \tau (p_h - \lambda_h) \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} = 0, \tag{2.12}$$

$$\left\langle \mu_h, \boldsymbol{v}_h \cdot \boldsymbol{n} \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} + \left\langle \mu_h, \tau (p_h - \lambda_h) \right\rangle_{\partial \mathcal{T}_\mathrm{A}^h} - \left\langle \frac{1}{c\rho} \lambda_h, \mu_h \right\rangle_{\Gamma_\mathrm{A}^\mathrm{abc}} = 0. \tag{2.13}$$

The correspondent matrix form reads

$$\begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{P}} \end{bmatrix} + \begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} + \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{2.14}$$

$$\mathbb{I}\boldsymbol{V} + \mathbb{J}\boldsymbol{P} + \mathbb{G}\boldsymbol{\Lambda} = \mathbf{0}. \tag{2.15}$$

The vectors $\boldsymbol{V}, \boldsymbol{P}, \boldsymbol{\Lambda}$ contain the values for the time-dependent degrees of freedom while the matrices $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{E}, \mathbb{G}, \mathbb{H}, \mathbb{I}, \mathbb{J}$ result from the element and face integrals as shown in equations (2.11)–(2.13).

2.3 Time Discretization

For combined spatial and temporal discretization of the acoustic wave equation, the Courant number *Cr* is a crucial indicator to balance temporal and spatial resolution

$$Cr = \frac{c\Delta t k}{h}$$

in terms of the time step size Δt, the polynomial degree of shape functions k, and the characteristic element size h. It was initially introduced in the context of the Courant–Friedrichs–Lewy stability criterion for finite difference schemes [40]. For higher order finite element schemes, previous work has relied on a quadratic dependence of the critical time step on the element degree, $\Delta t \sim {}^{k^2}/_{h}$ (see e.g. [74]). In [88] the authors show that the growth rate of the maximum eigenvalue is bounded by the quadratic power and their results (Figures 6.11 and 6.18 in [88]) suggest that the dependence is approximately $\Delta t \sim {}^{k^{1.5}}/_{h}$ for moderate polynomial degrees $k < 10$. Therefore, the following definition of the Courant number will be used throughout this work

$$Cr = \frac{c\Delta t k^{1.5}}{h}. \tag{2.16}$$

Many time integration schemes, especially explicit schemes, are generally subject to a stability limit and the discretization parameters must be chosen such that the Courant number is below a method specific critical Courant number

$$Cr \leq Cr_{\text{crit}}.$$

For unconditional stable schemes, a choice of discretization parameters such that

$$Cr \ll 1 \qquad \text{or} \qquad Cr \gg 1$$

is nonetheless disadvantageous since the relation between information transport in space and time would not be in the physical regime, which can yield spurious oscillations [45]. Additionally, discretization errors in space and time should be of similar magnitude considering effectiveness.

The combination of HDG with diagonally-implicit Runge–Kutta methods for the discretization of the acoustic wave equation was proposed in [120] in 2011 and will be briefly explained in Section 2.3.1. In Sections 2.3.2 and 2.3.3, HDG in combination with explicit Runge–Kutta schemes (as proposed in 2016 by [97, 120] and first ideas from 2014 [117]) and with explicit arbitrary derivative schemes (as published in [145] in 2018) are introduced.

2.3.1 Implicit Runge–Kutta Time Integration

The application of implicit Runge–Kutta schemes to discretize the space-discrete equations (2.14) and (2.15) in time is exemplarily shown for the backward Euler method with the abbreviation $p_{t_i} = p(t = i \cdot \Delta t)$ for evaluation at a given time step i

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbb{A} & \mathbb{B} & \mathbb{C} \\ \mathbb{H} & \frac{1}{\Delta t}\mathbb{M} + \mathbb{D} & \mathbb{E} \\ \mathbb{I} & \mathbb{J} & \mathbb{G} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \\ \boldsymbol{\Lambda}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A}\boldsymbol{V}_{t_i} \\ \frac{1}{\Delta t}\mathbb{M}\boldsymbol{P}_{t_i} \\ \boldsymbol{0} \end{bmatrix}.$$

The application of the Schur complement to the system matrix reads

$$\mathbb{K}_{\text{impl}} = \mathbb{G} - \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A} & \mathbb{B} \\ \mathbb{H} & \frac{1}{\Delta t}\mathbb{M} + \mathbb{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix}. \tag{2.17}$$

This Schur complement is cheaply evaluated because the matrices $\mathbb{A}, \mathbb{M}$ and $\mathbb{D}$ are block diagonal and matrices $\mathbb{B}$ and $\mathbb{H}$ are of rectangular blocks due to the discontinuous ansatz spaces for the velocity and pressure field. Hence, equation (2.17) can be understood as a global equation but it can also be understood as an element-wise equation. Subsequently, the inversion is comparably cheap and even cheaper considering that $\mathbb{A}$ consists of d blocks resembling $\mathbb{M}$. The element-wise inversion in equation (2.17) is carried out by an element-internal Schur complement further reducing computational expense, i.e., instead of inverting the matrix as in equation (2.17) at once, the inversion is evaluated according to

$$\begin{aligned} \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A} & \mathbb{B} \\ \mathbb{H} & \frac{1}{\Delta t}\mathbb{M} + \mathbb{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \text{Id} & -\left(\frac{1}{\Delta t}\mathbb{A}\right)^{-1}\mathbb{B} \\ 0 & \text{Id} \end{bmatrix} \\ &\cdot \begin{bmatrix} \left(\frac{1}{\Delta t}\mathbb{A}\right)^{-1} & 0 \\ 0 & \left(\frac{1}{\Delta t}\mathbb{M} + \mathbb{D} - \mathbb{H}\left(\frac{1}{\Delta t}\mathbb{A}\right)^{-1}\mathbb{B}\right)^{-1} \end{bmatrix} \cdot \begin{bmatrix} \text{Id} & 0 \\ -\mathbb{H}\left(\frac{1}{\Delta t}\mathbb{A}\right)^{-1} & \text{Id} \end{bmatrix}, \end{aligned}$$

with Id representing identity matrices of appropriate size. Therein, the inverse of matrix $\mathbb{A}$ is required, which is of size $d \cdot (k+1)^d$ but with d equal blocks. Also, the inverse of a $(k+1)^d$ matrix (bottom right entry of the middle matrix in the above equation) is required. Hence, the computational cost is reduced from the inversion of one $(d+1) \cdot (k+1)^d$ sized matrix to two $(k+1)^d$ sized matrices.

The right hand side for the linear system of equations is calculated according to

$$\boldsymbol{R}_{t_{i+1}} = -\begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A} & \mathbb{B} \\ \mathbb{H} & \frac{1}{\Delta t}\mathbb{M} + \mathbb{D} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A}\boldsymbol{V}_{t_i} \\ \frac{1}{\Delta t}\mathbb{M}\boldsymbol{P}_{t_i} \end{bmatrix},$$

again making use of the Schur complement as explained above. After assembly of $\mathbb{K}$ and $\boldsymbol{R}_{t_{i+1}}$, the global system

$$\mathbb{K}_{\text{impl}} \boldsymbol{\Lambda}_{t_{i+1}} = \boldsymbol{R}_{t_{i+1}} \tag{2.18}$$

is solved for the new trace variable $\boldsymbol{\Lambda}_{t_{i+1}}$, and velocity and pressure are updated according to

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbb{A} & \mathbb{B} \\ \mathbb{H} & \frac{1}{\Delta t}\mathbb{M} + \mathbb{D} \end{bmatrix}^{-1} \left(\begin{bmatrix} \frac{1}{\Delta t}\mathbb{A}\boldsymbol{V}_{t_i} \\ \frac{1}{\Delta t}\mathbb{M}\boldsymbol{P}_{t_i} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \boldsymbol{\Lambda}_{t_{i+1}} \right).$$

This update formula can again be understood as a global or an element-wise equation. The highest computational effort is imposed by the solution of the global system (2.18). The system size depends on the number of element faces and the polynomial degree of the shape functions.

For higher order discretizations, diagonally implicit Runge–Kutta (DIRK) schemes are employed rather than the backward Euler method, see [3, 26, 120]. A general Butcher tableau for a q-stage Runge–Kutta scheme is given by:

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ c_q & a_{q1} & \dots & a_{qq} \\ \hline & b_1 & \dots & b_q \end{array}$$

For implicit schemes, all coefficients a_{ij} can be non-zero. For DIRK schemes, however, the entries a_{ij} with $j > i$ are zero and the diagonal entries a_{ii} are all equal. The size of the system to be solved at each stage of the DIRK scheme is only the original size since sequential stages can be solved one after another and are not coupled in between. Additionally, the system matrix $\mathbb{K}$ (2.17) is the same for each stage and must only be assembled once. Also, the preconditioner must only be applied to the original system and not a larger one or one with changing coefficients. Thereby, the computational effort is significantly reduced by choosing DIRK schemes rather than other implicit Runge–Kutta time integrators.

2.3.2 Explicit Runge–Kutta Time Integration

For temporal discretization of equations (2.14)–(2.15) with explicit Runge–Kutta schemes, the explicit Euler is chosen for demonstration and the update rule reads

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \Delta t \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} + \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \boldsymbol{\Lambda}_{t_i} \right),$$

with the trace variable resulting from

$$\mathbb{I}\boldsymbol{V}_{t_i} + \mathbb{J}\boldsymbol{P}_{t_i} + \mathbb{G}\boldsymbol{\Lambda}_{t_i} = 0. \tag{2.19}$$

The expression (2.19) can be written in terms of the element-wise or face-wise continuous fields

$$\lambda_{h,t_i} = \begin{cases} \frac{1}{2\tau}\left(\boldsymbol{v}^+_{h,t_i} - \boldsymbol{v}^-_{h,t_i}\right)\cdot\boldsymbol{n}^+ + \frac{1}{2}\left(p^+_{h,t_i} + p^-_{h,t_i}\right) & \forall F \in \mathcal{E}^0_h, \\ \mathsf{P} p_{\mathrm{D},t_i} & \forall F \in \mathcal{E}^{\partial,\mathrm{dir}}_h, \\ \frac{1}{\tau}\boldsymbol{v}_{h,t_i}\cdot\boldsymbol{n} + p_{h,t_i} & \forall F \in \mathcal{E}^{\partial,\mathrm{neu}}_h, \\ \frac{1}{\tau + \frac{1}{c\rho}}\boldsymbol{v}_{h,t_i}\cdot\boldsymbol{n} + \frac{\tau}{\tau + \frac{1}{c\rho}} p_{h,t_i} & \forall F \in \mathcal{E}^{\partial,\mathrm{abc}}_h, \end{cases} \tag{2.20}$$

with the plus and minus sign indicating the two limits from both adjacent elements for evaluation of discontinuous fields at interior faces. These expressions result from a point-wise interpretation of equation (2.13). If numerical integration to obtain equation (2.15) from equation (2.13) is exact, equations (2.15) and (2.20) are equivalent. This notation allows the interpretation of HDG in combination with explicit Runge–Kutta time discretization as DG scheme with a specific flux function.

The extension to an s-stage Runge–Kutta scheme is straightforward

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} + \Delta t \sum_{j=1}^{s} b_j \boldsymbol{K}_j \quad \text{and} \quad \boldsymbol{K}_j = -\mathbb{Q}^{-1}\mathbb{K}\left(\begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} + \Delta t \sum_{l=1}^{j-1} a_{jl}\boldsymbol{K}_l\right),$$

with the coefficients a_{jl} and b_j from the Butcher tableau of the respective scheme and the abbreviations $\mathbb{K}$ and $\mathbb{Q}$ for the full stiffness and mass matrix

$$\mathbb{K} = \begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix}, \qquad \mathbb{Q} = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix}. \tag{2.21}$$

For explicit Runge–Kutta schemes, the coefficients a_{jl} are zero for $j \geq l$. The low-storage schemes reducing memory consumption and memory transfer [90] are computationally appealing, but also strong stability preserving Runge–Kutta schemes as in [101]. The computational aspects are studied in detail in Chapter 3.

2.3.3 Explicit Arbitrary Derivative Time Integration

The explicit arbitrary derivative (ADER) time integration has been proposed in [47, 50, 149, 150] for linear hyperbolic problems. The basic steps for the derivation are a Taylor expansion in time and a Cauchy–Kowalevski procedure to express time derivatives in terms of space derivatives. In the following, the ADER time discretization is combined with the HDG space discretization for the acoustic wave equation. The derivation is similar to the work [47], where ADER is used for the elastic wave equation in combination with a standard DG approach. It will be shown that the HDG characteristic property to yield superconvergent pressure solutions is lost by straightforward application of ADER. In Section 2.4.1, a further development is shown to retain the superconvergence. This entire section is written following [145] and parts are quoted literally.

The space-discretized but time-continuous matrix system (2.14)–(2.15) is rewritten by elimination of $\boldsymbol{\Lambda}$

$$\begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{P}} \end{bmatrix} + \left(\begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

or analogously with the abbreviations $\mathbb{K}$ and $\mathbb{Q}$ for the full stiffness and mass matrix from equation (2.21) and the symbolic inversion of the mass matrix as

$$\begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{P}} \end{bmatrix} + \mathbb{Q}^{-1}\mathbb{K} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Time integration of this equation on the interval $t \in [t_i, t_{i+1}]$ yields

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \mathbb{Q}^{-1}\mathbb{K} \int_{t_i}^{t_{i+1}} \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} \mathrm{d}t. \tag{2.22}$$

A common approach, which was also used in the previous sections, is to replace the time integral by evaluations at the start or the end of the interval (yielding the forward or backward Euler schemes, respectively) or by evaluations at several time points in the given interval (yielding Runge–Kutta schemes with several stages). The ADER approach offers another possibility with the beneficial property to overcome the Butcher barriers known for Runge–Kutta schemes, which imply that the number of stages to obtain a desired order grows nonlinearly [25]. Also, the ADER approach has beneficial computational properties compared to Runge–Kutta schemes, which will be explained in detail in Chapter 3.

Starting point to derive an expression for the time integral in equation (2.22) is the temporal Taylor expansion of the velocity and pressure fields at the time instant t_i up to order k

$$\begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \sum_{j=0}^{k} \frac{(t-t_i)^j}{j!} \frac{\partial^j}{\partial t^j} \begin{bmatrix} \boldsymbol{v}_{t_i} \\ p_{t_i} \end{bmatrix}. \tag{2.23}$$

The acoustic wave equation as given in equations (2.4)–(2.5) is equivalently written as

$$\frac{\partial}{\partial t} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = - \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix}.$$

Taking the derivative in time and reintroducing this definition reads

$$\frac{\partial^2}{\partial t^2} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = - \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix}^2 \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix}.$$

This step is repeatable. By induction, the j-th time derivative of velocity and pressure fields is expressed by space derivatives

$$\frac{\partial^j}{\partial t^j} \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = (-1)^j \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix}^j \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix}.$$

This is the Cauchy–Kowalevski procedure, which is used to replace the time derivatives in the Taylor expansion (2.23),

$$\begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix} = \sum_{j=0}^{k} \frac{(t-t_i)^j}{j!} (-1)^j \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix}^j \begin{bmatrix} \boldsymbol{v}_{t_i} \\ p_{t_i} \end{bmatrix} := \sum_{j=0}^{k} \frac{(t-t_i)^j}{j!} (-1)^j \mathbb{S}^j \begin{bmatrix} \boldsymbol{v}_{t_i} \\ p_{t_i} \end{bmatrix}. \tag{2.24}$$

The only time dependent quantity on the right hand side of this equation is the time t itself. For brevity, the derivative operator $\mathbb{S}$ is introduced

$$\mathbb{S} = \begin{bmatrix} 0 & \frac{1}{\rho}\nabla \\ c^2\rho\nabla\cdot & 0 \end{bmatrix}. \tag{2.25}$$

With the matrix $\mathbb{N}$ holding the shape functions of polynomial degree k such that the discretized velocity and pressure field are expressed as

$$\begin{bmatrix} \boldsymbol{v}_h \\ p_h \end{bmatrix} = \mathbb{N} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}, \tag{2.26}$$

the equation (2.24) is projected onto the degrees of freedom

$$\begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix} = \mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(t-t_i)^j}{j!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^j\mathbb{N}\mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$

The final step to derive the time and space discretized ADER HDG method is to introduce this expression into equation (2.22) and evaluate the time integral. The final method is given by

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \mathbb{Q}^{-1}\mathbb{K}\mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^j\mathbb{N}\mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}. \tag{2.27}$$

By construction, equation (2.27) represents an explicit single step scheme.

2.3.4 Local Time Stepping

The Courant number plays a crucial role in the context of numerical solution strategies for hyperbolic problems [40]. Explicit schemes are subject to a stability criterion: the scheme is only stable up to a maximal Courant number. This is often understood as a restriction on the time step size

$$\Delta t \leq \frac{Cr_{\max} h}{ck^{1.5}},$$

with a characteristic element size h, the speed of sound c, and the polynomial degree of the shape functions k, according to the definition of the Courant number in equation (2.16). Especially in scenarios with high variations in the speed of sound or the mesh size, this criterion is too restrictive for most of the elements, which is disadvantageous from a computational perspective but also from a numerical perspective since causality implies that information travels in space as fast as in time scaled with the speed of sound. Using a time step size with $Cr \ll 1$ induces severe computational overhead and can cause spurious oscillations or conditioning problems [45]. This the classical motivation to utilize local time stepping (LTS) methods. The ADER global time integration is easily extended to LTS as shown in [49] in the context of the elastic wave equation. In the following, ADER HDG LTS is presented with reference to [134, 145].

Each element K^e operates with its own time step size Δt^e, which is determined by the element size, the polynomial degree, and the speed of sound. Starting point for the derivation of ADER HDG LTS is the global time stepping scheme given by equation (2.27) with the system matrices as in equations (2.21). The information exchange between elements is the crucial step in LTS. As can be seen from (2.21), the information is passed from one element to the next by application of $\mathbb{G}^{-1}$ on element faces processing information from both adjacent elements. This expression must be generalized such that it works for adjacent elements operating with different time steps and at different time states. For convenience, the element matrices $\mathbb{B}, \mathbb{D}, \mathbb{H}$ are split into terms stemming from element integrals and from element face integrals

$$\begin{bmatrix} 0 & \mathbb{B} \\ \mathbb{H} & \mathbb{D} \end{bmatrix} = \begin{bmatrix} 0 & \mathbb{B}_K \\ \mathbb{H}_K & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \mathbb{H}_{\partial K} & \mathbb{D}_{\partial K} \end{bmatrix}$$
$$\text{with } \mathbb{B} = \mathbb{B}_K, \mathbb{H} = \mathbb{H}_K + \mathbb{H}_{\partial K}, \text{ and } \mathbb{D} = \mathbb{D}_{\partial K}.$$

An element K^e is updated from its current time level t^e to the next time level $t^e + \Delta t^e$ if it fulfills the update criterion

$$t^e + \Delta t^e \leq \min\left(t^{e_n} + \Delta t^{e_n}\right) \quad \forall \text{ neighbor elements } K^{e_n}.$$

Elements are updated once there are no neighbors that could advance to an earlier point in time. The procedure to update element K^e fulfilling the update criterion is as follows:

1. For each attached face (looping n), the time interval for the flux evaluation is determined according to
$$[t_1, t_2] = \left[\max\left(t^e, t^{e_n}\right), \min\left(t^e + \Delta t^e, t^{e_n} + \Delta t^{e_n}\right)\right].$$
   a) In element K^e and the neighbor K^{e_n}, the time integrals of $\boldsymbol{V}$ and $\boldsymbol{P}$ in the given interval $[t_1, t_2]$ are evaluated as
$$\boldsymbol{\Psi}^e = \mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(t_2 - t^e)^{j+1} - (t_1 - t^e)^{j+1}}{(j+1)!} (-1)^j \int_{K^e} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix},$$
$$\boldsymbol{\Psi}^{e_n} = \mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(t_2 - t^{e_n})^{j+1} - (t_1 - t^{e_n})^{j+1}}{(j+1)!} (-1)^j \int_{K^{e_n}} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$
   b) The inverse of the trace mass matrix is applied to the fluxes $\boldsymbol{\Psi}^e$, $\boldsymbol{\Psi}^{e_n}$, all face integrals are evaluated and summed into $\boldsymbol{\Phi}^e$ and into the flux memory variable $\mathcal{M}^{e_n}$ of the neighbor
$$\left\{\boldsymbol{\Phi}^e, \mathcal{M}^{e_n}\right\} \leftarrow \left\{\boldsymbol{\Phi}^e, \mathcal{M}^{e_n}\right\} + \left(\begin{bmatrix} 0 & 0 \\ \mathbb{H}_{\partial K} & \mathbb{D}_{\partial K} \end{bmatrix} - \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \end{bmatrix} \mathbb{G}^{-1} \begin{bmatrix} \mathbb{I} & \mathbb{J} \end{bmatrix} \right) \left\{\boldsymbol{\Psi}^e, \boldsymbol{\Psi}^{e_n}\right\}.$$
2. The time integrals of $\boldsymbol{V}$ and $\boldsymbol{P}$ are evaluated in the considered element K^e for the entire time step and are as well summed into $\boldsymbol{\Phi}^e$
$$\boldsymbol{\Phi}^e \leftarrow \boldsymbol{\Phi}^e + \begin{bmatrix} 0 & \mathbb{B}_K \\ \mathbb{H}_K & 0 \end{bmatrix} \mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(\Delta t^e)^{j+1}}{(j+1)!} (-1)^j \int_{K^e} \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$

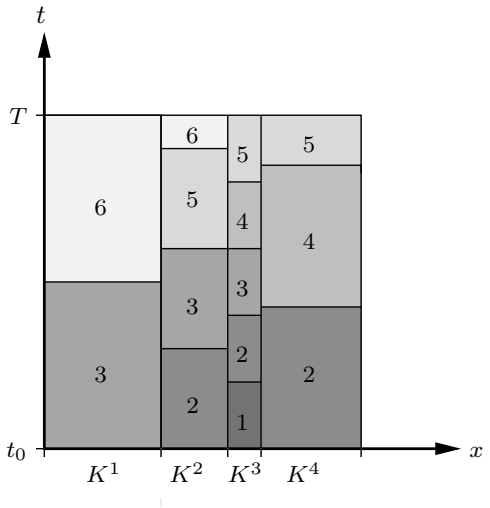


Figure 2.2: One-dimensional example of the ADER LTS update scheme.

3. The update of the state variables is performed

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \mathbb{Q}^{-1} \left(\boldsymbol{\Phi}^e + \mathcal{M}^e \right),$$

   where potentially non-zero entries in the element's memory variable are considered.

4. The element time is updated and its flux memory variable is set to zero

$$t^e \leftarrow t^e + \Delta t^e,$$
$$\mathcal{M}^e \leftarrow 0.$$

The flux memory variable $\mathcal{M}^e$ is crucial for the information transport between elements and reduces the number of required evaluations through storage of previously evaluated contributions. An element with a small time step will be evaluated several times before a neighboring element with a much larger time step is updated. Each evaluation sums contributions to the flux memory variable of elements with larger time steps.

In order to clarify the mode of operation, consider the problem given in Figure 2.2. The example is one-dimensional and consists of four elements of different size, which all have different time step sizes. All elements start at the same time level t_0 and shall advance to the final time T. The procedure is as follows. In each cycle, the elements are looped and their update criterion is checked. In the first cycle, the update criterion is only true for element K^3. It is updated, advances to its next time level and adds contributions to the flux memory variables of elements K^2 and K^4. In the next cycle, elements K^2, K^3, and K^4 are updated: K^2 updates considering the flux contribution from K^3 from the previous cycle and subsequently sets its memory variable to zero, K^3 updates and contributes to the memory variable of K^2 and K^4, and finally, K^4 updates. Note that even though K^2 was active in this cycle, its memory variable is already non-zero again. In cycle 3, element K^1 updates for the first time. Elements K^2 and K^3 update to the same time level, a special case correctly treated by the scheme without additional adjustments. In the

given ordering of the elements, K^2 updates first, writes into the flux memory variable of K^3, and subsequently element K^3 updates taking the flux contribution from element K^2 into account. If K^3 was updated first, K^3 would have contributed to the flux memory variable of K^2 and K^2 could have updated accordingly. Elements K^3 and K^4 are updated in cycle 4. In cycle 5, the first elements reach the final time level T. It might happen (e.g. for element K^4) that the last time step is smaller than the actual time step size. This is easily treated by setting the element time step size for this last step to the remainder. As long as the remainder is not negligibly small, this case is not critical. In the last cycle, K^1 and K^2 reach the final time level and all elements finish their calculation. In total, fourteen element updates were performed. If all elements had had the small time step size of element K^3, twenty updates would have been necessary. Please note that a different numbering of the elements generally yields a different chronology of updates.

To reach better computational efficiency, the algorithmic setup does not allow for arbitrary time steps but only multiples of the smallest time step size, which is called clustering, as explained in [146, 148]. The first cluster works with the minimal time step size Δt_{min}. The i-th cluster works with multiples $((i-1)\cdot\delta+1)\cdot\Delta t_{\text{min}}$, where δ is a user defined input parameter to control the difference of time steps between neighboring clusters. An additional restriction during the setup of the clustering is that cluster i is only allowed to be neighbored by clusters $i-1$ and $i+1$ and that a cluster is at least of two elements thickness, i.e., an element of cluster i can only have one different cluster as neighbor.

2.4 Optimal Convergence and Superconvergence

The HDG spatial discretization yields errors converging with order $\mathcal{O}(h^{k+1})$ in the L_2-norm for the pressure as well as the velocity field and hence the convergence is optimal [33, 34, 120].

In [120], a local postprocessing procedure is proposed to reconstruct a pressure field p_h^* with order $\mathcal{O}(h^{k+2})$ superconvergence. The postprocessing is element-local and does not advance the solution in time. Using the trace field, an improved pressure gradient $\boldsymbol{g}_h \in \boldsymbol{V}_h$ is calculated according to

$$(\boldsymbol{w}_h, \boldsymbol{g}_h)_K = -(\nabla\cdot\boldsymbol{w}_h, p_h)_K + \langle \boldsymbol{w}_h\cdot\boldsymbol{n}, \lambda_h\rangle_{\partial K} \tag{2.28}$$

or analogous in matrix notation

$$\begin{aligned}\boldsymbol{G} &= \mathbb{A}^{-1}\left(\mathbb{B}\boldsymbol{P} + \mathbb{C}\boldsymbol{\Lambda}\right)\\ &= \mathbb{A}^{-1}\left(-\mathbb{C}\mathbb{G}^{-1}\mathbb{I}\boldsymbol{V} + \left(\mathbb{B}-\mathbb{C}\mathbb{G}^{-1}\mathbb{J}\right)\boldsymbol{P}\right).\end{aligned}$$

By usage of the trace field instead of the pressure on the element boundary, the pressure gradient is of order $\mathcal{O}(h^{k+1})$. The superconvergent pressure field $p_h^* \in \mathcal{P}_{k+1}(K)$ is obtained by solving for all $q_h^* \in \mathcal{P}_{k+1}(K)$

$$(\nabla q_h^*, \nabla p_h^*)_K = (\nabla q_h^*, \boldsymbol{g}_h)_K\,, \tag{2.29}$$

$$(1, p_h^*)_K = (1, p_h^*)_K\,. \tag{2.30}$$

Equation (2.29) requires the gradient of the superconvergent pressure field to weakly equal the improved pressure gradient. Equation (2.30) enforces the equality of the pressure averages. The

postprocessing step can also be understood as a least squares fit of the pressure gradients under the constraint of equal pressure averages, which can be shown by variational techniques to have the minimum given by equations (2.29)–(2.30).

Mathematical proof for the superconvergence property of HDG spatial discretization in the context of acoustic wave propagation is given in [31]. Therein, an a priori error analysis for the time continuous case is presented based on a duality argument in combination with a projection-based technique. It is an extension of the derivations for elliptic problems [34] to the hyperbolic acoustic wave equation. Premises for the proof are time continuity, simplices, $\mathcal{C}^1(\Omega_\mathrm{A})$ continuous material properties, and initial conditions imposed by special projections more elaborate than the L_2 projection. The results in [120] indicate, however, that superconvergence is already observed for L_2 projected fields. Theoretical basis and numerical evidence for superconvergence are available for hexahedra in [34, 176], respectively.

In summary it can be said that HDG generally gives optimal $\mathcal{O}(h^{k+1})$ convergence for the primary fields and superconvergence $\mathcal{O}(h^{k+2})$ for the postprocessed quantity. For time discretized problems, the error due to time discretization must not dominate to observe the superconvergence.

2.4.1 Reconstruction for Superconvergence

Application of the local postprocessing (2.29)–(2.30) to the ADER HDG scheme given by equation (2.27) does not yield a superconvergent pressure solution p_h^*. This is due to the fact that time and space discretization are strongly interlinked and the time discretization in this scheme is not of order $\mathcal{O}(\Delta t^{k+2})$ but only $\mathcal{O}(\Delta t^{k+1})$. Simply increasing the upper bound of summation in equation (2.27) from k to $k+1$ does not overcome this issue. To recover the desired superconvergence property, a reconstruction procedure is proposed. The idea is to use the HDG specific trace field to recover derivatives of higher accuracy and reuse them for higher spatial derivatives.

Analogous to the improved pressure gradient $\boldsymbol{g}_h$ in equation (2.28), an improved velocity divergence d_h is calculated according to

$$\left(c^2\rho d_h, q\right)_K = -\left(c^2\rho \boldsymbol{v}_h, \nabla q\right)_K + \left\langle c^2\rho \boldsymbol{v}_h \cdot \boldsymbol{n}, q\right\rangle_{\partial K} + \left\langle c^2\rho\tau(p_h - \lambda_h), q\right\rangle_{\partial K}.$$

In terms of the values of the degrees of freedom at a time instance t_i, the calculation in matrix form reads

$$\begin{bmatrix} \boldsymbol{G}_{t_i} \\ \boldsymbol{D}_{t_i} \end{bmatrix} = \mathbb{Q}^{-1}\mathbb{K} \begin{bmatrix} \boldsymbol{V}_{\boldsymbol{t_i}} \\ \boldsymbol{P}_{t_i} \end{bmatrix}.$$

The ADER HDG scheme (2.27) is repeated but the first term of the sum is split for clarity of the following derivation

$$\begin{aligned} \begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} &- \mathbb{Q}^{-1}\mathbb{K}(t_{i+1} - t_i) \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} \\ &- \mathbb{Q}^{-1}\mathbb{K}\mathbb{Q}^{-1} \sum_{j=1}^{k} \frac{(t_{i+1} - t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^\mathrm{T}\mathbb{S}^j\mathbb{N}\mathrm{d}K \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}. \end{aligned}$$

Application of the derivative operator $\mathbb{S}$ to the velocity and pressure field corresponds to the determination of the pressure gradient and velocity divergence (as can be seen from the definition

of $\mathbb{S}$ in equation (2.25)). But rather than using these quantities, the improved quantities $\boldsymbol{G}_{t_i}$ and $\boldsymbol{D}_{t_i}$ shall be used:

$$\mathbb{S}\begin{bmatrix} \boldsymbol{v}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho}\nabla p_h \\ c^2\rho\nabla\cdot\boldsymbol{v}_h \end{bmatrix} \qquad \text{is replaced by} \qquad \begin{bmatrix} \frac{1}{\rho}\boldsymbol{g}_h \\ c^2\rho d_h \end{bmatrix}.$$

Higher spatial derivatives are based on the improved quantities as well, i.e.,

$$\mathbb{S}^j\begin{bmatrix} \boldsymbol{v}_h \\ p_h \end{bmatrix} \qquad \text{is replaced by} \qquad \mathbb{S}^{j-1}\begin{bmatrix} \frac{1}{\rho}\boldsymbol{g}_h \\ c^2\rho d_h \end{bmatrix}.$$

The resulting ADER HDG scheme with reconstruction is given by

$$\begin{aligned}\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \mathbb{Q}^{-1}\mathbb{K}(t_{i+1}-t_i)\begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} \\ &\quad - \mathbb{Q}^{-1}\mathbb{K}\mathbb{Q}^{-1}\left(\sum_{j=1}^{k+1}\frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j\int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^{j-1}\mathbb{N}\mathrm{d}K\right)\mathbb{Q}^{-1}\mathbb{K}\begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}. \end{aligned} \tag{2.31}$$

The essential ingredient for the reconstruction procedure is to replace the continuous but element-local derivative operator $\mathbb{S}$ by the discrete derivative operator $\mathbb{Q}^{-1}\mathbb{K}$ with its HDG characteristics. In contrast to the scheme without reconstruction (2.27), the upper bound of the sum is $k+1$. Obviously, an element-local derivative evaluation is traded for the application of a global derivative operator. Through the mixture of face and element-blocks in $\mathbb{K}$, the operator corresponds to a sparsely populated global operator without block structure.

A logical further step is to exchange all applications of the continuous derivative operator by the discrete derivative operator $\mathbb{Q}^{-1}\mathbb{K}$ resulting in the following scheme

$$\begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix} - \sum_{j=1}^{k+1}\frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j\left(\mathbb{Q}^{-1}\mathbb{K}\right)^{j+1}\begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \end{bmatrix}. \tag{2.32}$$

This is a scheme with the capability to yield superconvergent pressure solutions p_h^* by means of the local postprocessing as well. Compared to (2.31) it has however disadvantageous computational properties as will be explained in Section 3. The original ADER methods that were applied in the context of finite volume methods were more similar to this setup whereas the element-local scheme was introduced later in [47, 89] in the context of DG methods.

For the LTS variant of ADER HDG, the reconstruction must also be employed to obtain superconvergence. However, the evaluation is slightly more elaborate since one element requires data from its element faces and by means of the trace variable from its adjacent elements. Those, however, are generally on a different time level. To evaluate the face contributions for one element to be updated, the neighboring elements must artificially be brought to the required time level. A dramatic increase of the computational expenses is avoided by the clustering of elements with each cluster operating on the same time step as explained in Section 2.3.4.

2.4.2 Adjoint Consistency

As already mentioned in Section 2.4, the mathematical proof for the superconvergence property of HDG is restricted to the time continuous scenario and other premises. Several reported numerical experiments however indicate that not all of the premises of the proof are necessary to actually obtain superconvergence. One rather crucual premise is the adjoint consistency of the discretization. The adjoint consistency will be studied for the ADER HDG discretization without reconstruction as in equation (2.27), with reconstruction as in equation (2.31), and with the discrete derivative operator replacing all derivative operators as in equation (2.32). The adjoint of the discretized problem will be compared to the discretization of the adjoint problem. For an adjoint consistent scheme, both approaches result in the same expression [72].

The continuous adjoint wave equation with initial and boundary conditions is given by

$$\frac{\partial \boldsymbol{w}}{\partial t} + c^2 \rho \nabla q = \mathbf{0} \qquad \text{in} \qquad \Omega_{\mathrm{A}} \times [0,T], \tag{2.33}$$

$$\frac{\partial q}{\partial t} + \frac{1}{\rho} \nabla \cdot \boldsymbol{w} = 0 \qquad \text{in} \qquad \Omega_{\mathrm{A}} \times [0,T], \tag{2.34}$$

$$\boldsymbol{w}(\boldsymbol{x}, t = T) = \mathbf{0} \qquad \text{in} \qquad \Omega_{\mathrm{A}}, \tag{2.35}$$

$$q(\boldsymbol{x}, t = T) = 0 \qquad \text{in} \qquad \Omega_{\mathrm{A}}, \tag{2.36}$$

$$\frac{1}{\rho} \boldsymbol{w} \cdot \boldsymbol{n} = 0 \qquad \text{on} \qquad \Gamma_{\mathrm{A}}^{\mathrm{neu}} \times [0,T], \tag{2.37}$$

$$q = 0 \qquad \text{on} \qquad \Gamma_{\mathrm{A}}^{\mathrm{dir}} \times [0,T], \tag{2.38}$$

in terms of the testing velocity $\boldsymbol{w}$ and the testing pressure q. For convenience, the absorbing boundary condition is dropped. This adjoint wave equation in first order form results from the wave equation (2.4)–(2.10) by application of Gauss' divergence theorem and integration by parts to spatial as well as temporal derivatives. The boundary evaluation of the temporal derivative terms causes the 'initial conditions' at $t = T$ for the testing velocity and pressure. The solution of the adjoint wave equation requires an integration backwards in time from $t = T$ to $t = 0$. The adjoint problem has no source term which is due to its artificial nature. For an inverse problem, the mismatch between measurements and simulation would be the source term, see for example Section 7.4.

Spatial discretization of (2.33)–(2.38) with HDG yields the following matrix system for the degrees of freedom of the test functions $\boldsymbol{W}, \boldsymbol{Q}, \boldsymbol{M}$ of the testing fields $\boldsymbol{w}, q, \mu$, respectively

$$\begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{M} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{W}} \\ \dot{\boldsymbol{Q}} \end{bmatrix} + \begin{bmatrix} 0 & c^2\rho^2\mathbb{B} \\ \frac{1}{c^2\rho^2}\mathbb{H} & \frac{1}{c^2\rho^2}\mathbb{D}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{W} \\ \boldsymbol{Q} \end{bmatrix} + \begin{bmatrix} c^2\rho^2\mathbb{C} \\ \frac{1}{c^2\rho^2}\mathbb{E}^* \end{bmatrix} \boldsymbol{M} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{2.39}$$

$$\mathbb{I}\boldsymbol{W} + \mathbb{J}^*\boldsymbol{Q} + \mathbb{G}^*\boldsymbol{M} = 0. \tag{2.40}$$

Differences in contrast to equations (2.14)–(2.15) are the prefactors stemming from different positions of the material coefficients and matrices $\mathbb{D}^*, \mathbb{E}^*, \mathbb{J}^*, \mathbb{G}^*$ instead of $\mathbb{D}, \mathbb{E}, \mathbb{J}, \mathbb{G}$. The asterisk indicates the dependence of the matrices on the stabilization parameter τ^* instead of τ in the adjoint problem. Stability analysis reveals that the stabilization parameter τ^* must be negative in order to ensure stability.

Proposition. *If the stabilization parameter τ^* is negative, the HDG discretization for the adjoint wave equation* (2.33)–(2.38) *is stable.*

Proof. The space-discrete, time-continuous version of the adjoint wave equation is similar to equations (2.11)–(2.13) except for the coefficients. Inserting the solution functions for the test functions $\boldsymbol{v}_h = \boldsymbol{w}_h$, $p_h = c^2\rho^2 q_h$, $\lambda_h = -c^2\rho^2\mu_h$, gives the following equations:

$$\begin{aligned}
&\left(\frac{\partial \boldsymbol{w}_h}{\partial t}, \boldsymbol{w}_h\right)_{\mathcal{T}_h} - \left(c^2\rho q_h, \nabla\cdot\boldsymbol{w}_h\right)_{\mathcal{T}_h} + \left\langle c^2\rho\mu_h, \boldsymbol{w}_h\cdot\boldsymbol{n}\right\rangle_{\partial\mathcal{T}_h} &&= 0,\\
&\left(c^2\rho^2\frac{\partial q_h}{\partial t}, q_h\right)_{\mathcal{T}_h} + \left(c^2\rho\nabla\cdot\boldsymbol{w}_h, q_h\right)_{\mathcal{T}_h} + \left\langle c^2\rho\tau^*(q_h-\mu_h), q_h\right\rangle_{\partial\mathcal{T}_h} &&= 0,\\
&-\left\langle c^2\rho\boldsymbol{w}_h\cdot\boldsymbol{n}, \mu_h\right\rangle_{\partial\mathcal{T}_h} - \left\langle c^2\rho\tau^*(q_h-\mu_h), \mu_h\right\rangle_{\partial\mathcal{T}_h} &&= 0.
\end{aligned}$$

Summation of all equations results in:

$$\left(\frac{\partial \boldsymbol{w}_h}{\partial t}, \boldsymbol{w}_h\right)_{\mathcal{T}_h} + \left(\frac{\partial q_h}{\partial t}, c^2\rho^2 q_h\right)_{\mathcal{T}_h} + \left\langle c^2\rho\tau^*(q_h-\mu_h), (q_h-\mu_h)\right\rangle_{\partial\mathcal{T}_h},$$

which is equivalent to

$$\frac{1}{2}\frac{\partial}{\partial t}\|\boldsymbol{w}_h\|^2 + \frac{1}{2}\frac{\partial}{\partial t}\|c\rho q_h\|^2 = -\left\langle c^2\rho\tau^*(q_h-\mu_h), (q_h-\mu_h)\right\rangle_{\partial\mathcal{T}_h}.$$

Since the adjoint equation runs backward in time, the energy is non-increasing if the expression on the right hand side is positive. This is the case for $\tau^* < 0$ and exactly the opposite as when going forward in time. □

With the choice

$$\tau^* = -\rho c, \tag{2.41}$$

the matrices depending on the stabilization parameter fulfill the following conversions

$$\mathbb{D}^* = -c^2\rho^2\mathbb{D}, \qquad \mathbb{E}^* = -c^2\rho^2\mathbb{E}, \qquad \mathbb{J}^* = -c^2\rho^2\mathbb{J}, \qquad \mathbb{G}^* = -c^2\rho^2\mathbb{G}.$$

The adjoint stiffness matrix $\mathbb{K}^*$ is

$$\mathbb{K}^* = \begin{bmatrix} 0 & c^2\rho^2\mathbb{B} \\ \frac{1}{c^2\rho^2}\mathbb{H} & -\mathbb{D} \end{bmatrix} - \begin{bmatrix} c^2\rho^2\mathbb{C} \\ -\mathbb{E} \end{bmatrix} (-c^2\rho^2\mathbb{G})^{-1} \begin{bmatrix} \mathbb{I} & -c^2\rho^2\mathbb{J} \end{bmatrix}. \tag{2.42}$$

Proposition. *With the stabilization parameter $\tau^* = -\rho c$, the stiffness matrices $\mathbb{K}$ and $\mathbb{K}^*$ according to equations* (2.21) *and* (2.42) *fulfill the condition*

$$\mathbb{K}^{\mathrm{T}} = -\mathbb{K}^*.$$

Proof. Expanding equation (2.21) in terms of the face matrices and using $\mathbb{H} = -c^2\rho^2\mathbb{B}^{\mathrm{T}}$, $\mathbb{I} = \rho\mathbb{C}^{\mathrm{T}}$, $\mathbb{J} = -\frac{1}{c^2\rho}\mathbb{E}^{\mathrm{T}}$ results in

$$\mathbb{K} = \begin{bmatrix} 0 & \mathbb{B} \\ -c^2\rho^2\mathbb{B}^{\mathrm{T}} & \mathbb{D} \end{bmatrix} - \begin{bmatrix} \rho\mathbb{C}\mathbb{G}^{-1}\mathbb{C}^{\mathrm{T}} & -\frac{1}{c^2\rho}\mathbb{C}\mathbb{G}^{-1}\mathbb{E}^{\mathrm{T}} \\ \rho\mathbb{E}\mathbb{G}^{-1}\mathbb{C}^{\mathrm{T}} & -\frac{1}{c^2\rho}\mathbb{E}\mathbb{G}^{-1}\mathbb{E}^{\mathrm{T}} \end{bmatrix}.$$

The matrix $\mathbb{K}^*$ given through equation (2.42) with $\tau^* = -c^2\rho^2\tau$ can be expanded by introducing the above mentioned relations to the form

$$\mathbb{K}^* = \begin{bmatrix} 0 & c^2\rho^2\mathbb{B} \\ -\mathbb{B}^{\mathrm{T}} & -\mathbb{D} \end{bmatrix} - \begin{bmatrix} -\rho\mathbb{C}\mathbb{G}^{-1}\mathbb{C}^{\mathrm{T}} & -\rho\mathbb{C}\mathbb{G}^{-1}\mathbb{E}^{\mathrm{T}} \\ \frac{1}{c^2\rho}\mathbb{E}\mathbb{G}^{-1}\mathbb{C}^{\mathrm{T}} & \frac{1}{c^2\rho}\mathbb{E}\mathbb{G}^{-1}\mathbb{E}^{\mathrm{T}} \end{bmatrix}.$$

Comparing these two matrices shows the assertion. □

From the two previous propositions, the following conclusion is drawn:

Proposition. *Since $\mathbb{K}^{\mathrm{T}} = -\mathbb{K}^*$, the discrete adjoint in the forward Euler HDG discretization is a consistent discretization of the adjoint wave equation* (2.33)–(2.38).

The adjoint consistency for the three different types of ADER time discretization will be shown in the following sections.

ADER HDG without Reconstruction

Temporal discretization of the semidiscrete matrix form of the adjoint wave equation (2.39)–(2.40) using ADER without reconstruction yields (*adjoin-then-discretize*)

$$\begin{aligned} \begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} + (t_{i+1} - t_i)\mathbb{Q}^{-1}\mathbb{K}^* \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &\quad - \mathbb{Q}^{-1}\mathbb{K}^*\mathbb{Q}^{-1} \sum_{j=1}^{k+1} \frac{(t_i - t_{i+1})^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^{*j}\mathbb{N}\mathrm{d}K \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}. \end{aligned} \tag{2.43}$$

Therein, the operator $\mathbb{S}^*$ is the derivative operator used for the Cauchy–Kowalevski procedure of the adjoint wave equation

$$\mathbb{S}^* = \begin{bmatrix} 0 & c^2\rho\nabla \\ \frac{1}{\rho}\nabla\cdot & 0 \end{bmatrix}.$$

Contrary, deriving the adjoint of the space and time discrete ADER HDG method (2.27) basically by transposition results in (*discretize-then-adjoin*)

$$\begin{aligned} \begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} - (t_{i+1} - t_i)\mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &\quad - \mathbb{Q}^{-1} \left(\sum_{j=1}^{k+1} \frac{(t_{i+1} - t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}} \left(\mathbb{S}^{\mathrm{T}}\right)^j \mathbb{N}\mathrm{d}K \right) \mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}. \end{aligned} \tag{2.44}$$

ADER HDG with Reconstruction

Analogous to the previous two expressions, the fully discretized problem of the adjoint wave equation using ADER HDG with reconstruction is derived (*adjoin-then-discretize*)

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} + (t_{i+1} - t_i)\mathbb{Q}^{-1}\mathbb{K}^* \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &+ \frac{(t_i - t_{i+1})^2}{2}\mathbb{Q}^{-1}\mathbb{K}^*\mathbb{Q}^{-1}\mathbb{K}^* \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &- \mathbb{Q}^{-1}\mathbb{K}^*\mathbb{Q}^{-1} \left(\sum_{j=2}^{k+1} \frac{(t_i - t_{i+1})^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}} \left(\mathbb{S}^*\right)^{j-1} \mathbb{N}\mathrm{d}K \right) \mathbb{Q}^{-1}\mathbb{K}^* \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}. \end{aligned} \tag{2.45}$$

The adjoint of the fully discrete ADER HDG scheme with reconstruction given by equation (2.31) reads (*discretize-then-adjoin*)

$$\begin{aligned}\begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} - (t_{i+1}-t_i)\mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &+ \frac{(t_{i+1}-t_i)^2}{2}\mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}}\mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} \\ &- \mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}}\mathbb{Q}^{-1} \left(\sum_{j=2}^{k+1} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}} \left(\mathbb{S}^{\mathrm{T}}\right)^{j-1} \mathbb{N}\mathrm{d}K \right) \mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}.\end{aligned} \tag{2.46}$$

ADER HDG with only the Discrete Operator

The fully discretized problem of the adjoint wave equation using ADER HDG and replacing *every* application of the derivative operator $\mathbb{S}^*$ by the space discrete counterpart $\mathbb{Q}^{-1}\mathbb{K}^*$ results in (*adjoin-then-discretize*)

$$\begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} - \sum_{j=0}^{k+1} \frac{(t_i-t_{i+1})^{j+1}}{(j+1)!}(-1)^j \left(\mathbb{Q}^{-1}\mathbb{K}^*\right)^{j+1} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}. \tag{2.47}$$

In contrast, the adjoint of the fully discrete scheme (2.32) is (*discretize-then-adjoin*)

$$\begin{bmatrix} \boldsymbol{W}_{t_i} \\ \boldsymbol{Q}_{t_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix} - \sum_{j=0}^{k+1} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j \left(\mathbb{Q}^{-1}\mathbb{K}^{\mathrm{T}}\right)^{j+1} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \end{bmatrix}. \tag{2.48}$$

Discussion

The scheme (2.32) solely using the discrete derivative operator is adjoint consistent. Equations (2.47) and (2.48) are exactly the same considering that $\mathbb{K} = -(\mathbb{K}^*)^{\mathrm{T}}$. This was expected, since the scheme's structure is similar to that of explicit Runge–Kutta methods, which are known to be adjoint consistent.

Comparison of the two equations (2.43) and (2.44) to advance $\boldsymbol{W}$ and $\boldsymbol{Q}$ backwards in time for the ADER HDG scheme without reconstruction shows that the $k = 0$ term is adjoint consistent, i.e., the term representing the forward Euler. The remaining terms are not equal because the derivative matrix appears in front of the sum if discretization is carried out after adjoining and after the sum if discretization is carried out first.

The adjoint formulations (2.45) and (2.46) resulting from the ADER HDG scheme with reconstruction (2.31) reveal that the same change of position appears for $k \geq 2$ as for the scheme without reconstruction. In the scheme without reconstruction this however already appears for $k = 1$. The term stemming from $k = 1$ is adjoint consistent when the reconstruction is used. Adjoint consistency is taken to the next higher contribution, which appears to be enough to receive superconvergent pressure solutions. Numerical evidence will be given in Section 2.5.

2.5 Numerical Characterization

The numerical characterization of the proposed methods is carried out in terms of a convergence analysis in two and three space dimensions in Section 2.5.1 demonstrating optimal $k+1$ conver-

gence and $k+2$ superconvergence for explicit Runge–Kutta schemes, ADER, and ADER LTS. After that, temporal stability limits are presented in Section 2.5.2. Last, amplitude and phase errors are evaluated in a comparative study based on a one-dimensional problem with periodic boundary conditions in Section 2.5.3.

2.5.1 Numerical Convergence Analysis

In this section, a convergence analysis for all presented methods is carried out. The test case is a vibrating membrane. The analytic solution for the pressure field is

$$p = \cos(m\pi\sqrt{dt}) \cdot \prod_{i=1}^{d} \sin(m\pi x_i),$$

in two and three dimensions. The variable m denotes the number of vibrational modes in the membrane. The computational domain is $\Omega_{\mathrm{A}} = [0,1]^d$ and $p_{\mathrm{D}} = 0$ is prescribed on the boundary.

The mesh is not Cartesian but transformed as shown in Figure 2.3 by moving the nodes of the elements in x_1 direction according to

$$x_1 \leftarrow x_1 + 0.2 \cdot \prod_{i=1}^{d} \sin(\pi x_i).$$

The representative element size h indicated in the following figures and tables is the element extent in x_2 direction. In order to receive error values in a reasonable range, i.e., noticeably below the solution norm and above the machine accuracy, the number of membrane modes m is set to the used polynomial degree k and the coarsest mesh for polynomial degrees $k = 1$ to $k = 4$ is $h = 0.2$ while it is $h = 0.5$ for higher polynomial degrees $k \geq 5$. All simulations are run with a Courant number $Cr = 0.1$ and the error is evaluated in L_2 norm at the final time $T = 1$.

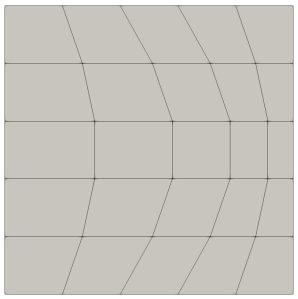


Figure 2.3: Mesh for convergence study.

Results in two dimensions are shown in Figure 2.4 for Runge–Kutta time discretization using a low-storage scheme with five stages of order four as in [90], denoted LSRK4(5), and ADER time integration. Results in three dimensions are plotted in Figure 2.5. As can be seen from the figures, the methods yield optimal $k+1$ convergence for the pressure field p_h and $k+2$

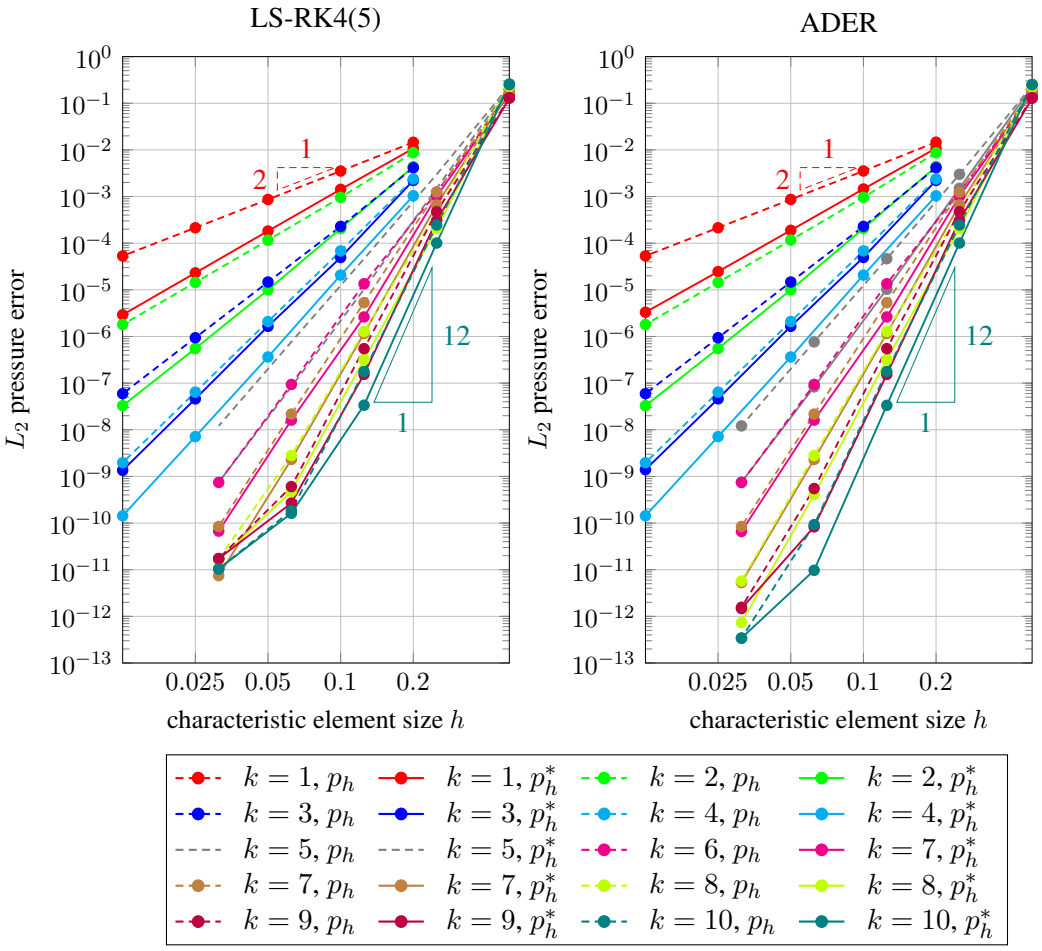


Figure 2.4: Convergence study in two dimensions for time integration with a low-storage Runge–Kutta scheme of order four LSRK4(5) in the left panel and ADER according to (2.31) in the right panel.

superconvergence for the postprocessed pressure field p_h^* for all polynomial degrees. LSRK4(5) and ADER perform very similarly for the tests from $k = 1$ to $k = 8$. For $k = 9$ and $k = 10$ differences are observed for the two finest discretizations. The errors are already close to machine accuracy but between 10^{-10} and 10^{-12} differences occur, which are traced back to a different phenomenon. The Runge–Kutta time discretization with LSRK4(5) is of order four. Even though Runge–Kutta time integrators have a low error constant, the time error dominates over the spatial error within this range. Rerunning the simulation with $k = 10$ and $h = 0.0625$ but with Courant number $Cr = 0.01$ instead of 0.1 yields an error of $9.30 \cdot 10^{-11}$ (for $Cr = 0.1$, the error is $1.86 \cdot 10^{-10}$) confirming the dominance of the time error. Figure 2.5 shows corresponding results for a three-dimensional setup. The same tests are carried out for ADER LTS and results are presented in Figure 2.6 confirming optimal convergence in p_h and superconvergence in p_h^* for all used polynomial degrees $k = 1, ..., 12$.

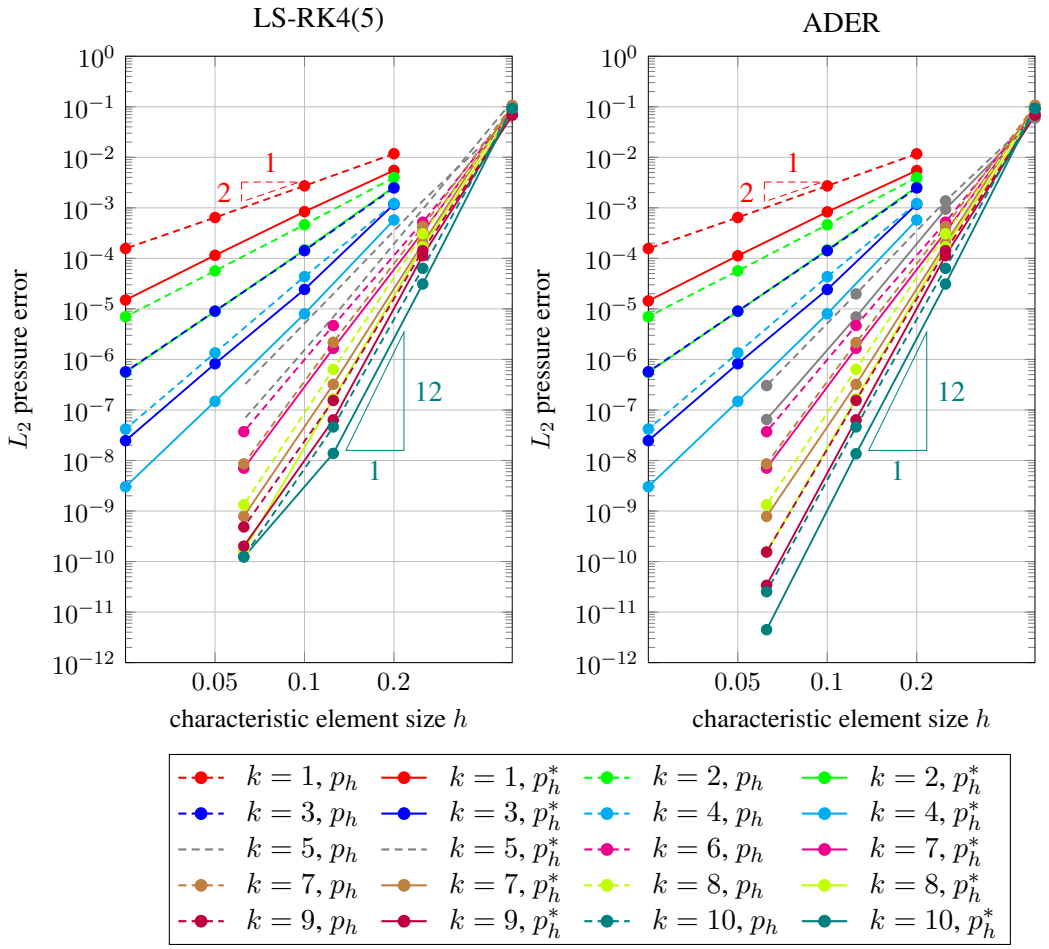


Figure 2.5: Convergence study in three dimensions for time integration with a low-storage Runge–Kutta scheme of order four LSRK4(5) in the left panel and ADER according to (2.31) in the right panel.

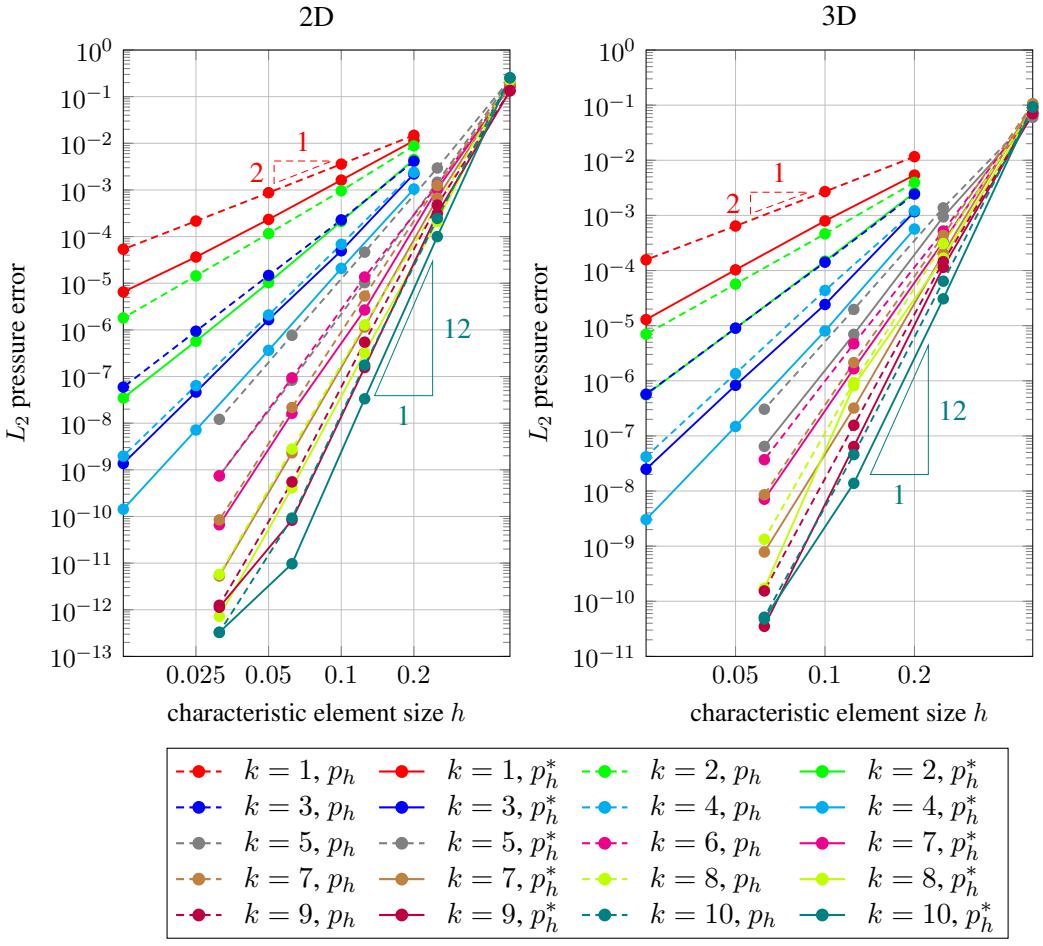


Figure 2.6: Convergence study in two and three dimensions for time integration with ADER LTS.

Numerical Evidence for Adjoint Consistency

To put the gain of the reconstruction procedure into perspective, numerical results on the convergence of the average pressure are presented. As mentioned in [72], adjoint consistency yields order doubling in a target functional. With the average pressure being the target functional, convergence of order $2k$ is expected for the adjoint consistent scheme (2.32) (in the following denoted as ADER adcon full), rate k for the scheme without reconstruction, and something in between for the reconstruction. All integrals are evaluated with quadrature rules enabling exact integration. The numerical example is the same as in the preceding section. The results for the convergence of the average pressure for $k = 2, 3, 4$ and time discretization with the three variants of ADER HDG are presented in Figure 2.7. For comparison, the results for the aforementioned low-storage Runge–Kutta scheme LSRK4(5) of order four with five stages, which is well known to be adjoint consistent, are also shown. Polynomial degrees $k = 2, 3$ are shown for one mode ($m = 1$) and $k = 3, 4$ are shown for two modes ($m = 2$) since the simulation accuracy quickly reaches the level of roundoff errors.

ADER without reconstruction yields pressure average errors of order $k + 1$ in all setups. For $k = 2$ all other schemes converge with order $5 = 2k + 1$. For $k = 3, 4$ the error decline with refinement is not as regular. For $k = 3,\ m = 1$, starting from large h and going to smaller h, the error order starts from 5 and increases to about 16. After that, the error decreases more slowly with orders of about 5 with reconstruction and of about 5 or 6 with the fully adjoint consistent schemes. For small h, the computational roundoff error is reached which is why $k = 3$ is also tested for the next mode $m = 2$. The convergence behavior is similar as for $m = 1$. The error order increases with decreasing h, a kink occurs in the convergence plot and then the order is about $7 = 2k + 1$. The error is slightly higher for ADER HDG with reconstruction compared to ADER adcon full and Runge–Kutta, at a rate of 4 to 5 in the lower part. For $k = 4$ and $m = 2$, all schemes (except the one without reconstruction) perform very similarly. The estimated order is between 6 and 7.

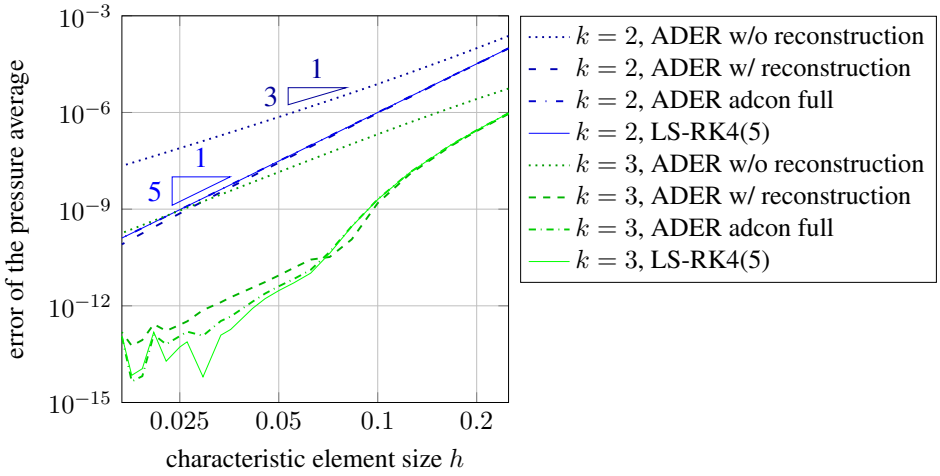
To summarize, the scheme without reconstruction reliably yields order $k + 1$ for the average pressure for all k. For $k = 2$, all other schemes yield errors of order $2k+1$. For $k = 3, 4$, the average pressure does not converge with order $2k$ but clearly better than $k+1$. It is remarkable that the fully adjoint consistent ADER HDG discretization and the Runge–Kutta discretization perform almost equally, showing that the same spatial errors dominate in both cases. ADER HDG with reconstruction comes very close, except for $k = 3$, where slight differences are noted. During these numerical experiments, the L_2 error for the postprocessed pressure p_h^* was also monitored and for all schemes except ADER without reconstruction, it showed $k + 2$ convergence.

2.5.2 Temporal Stability Limits

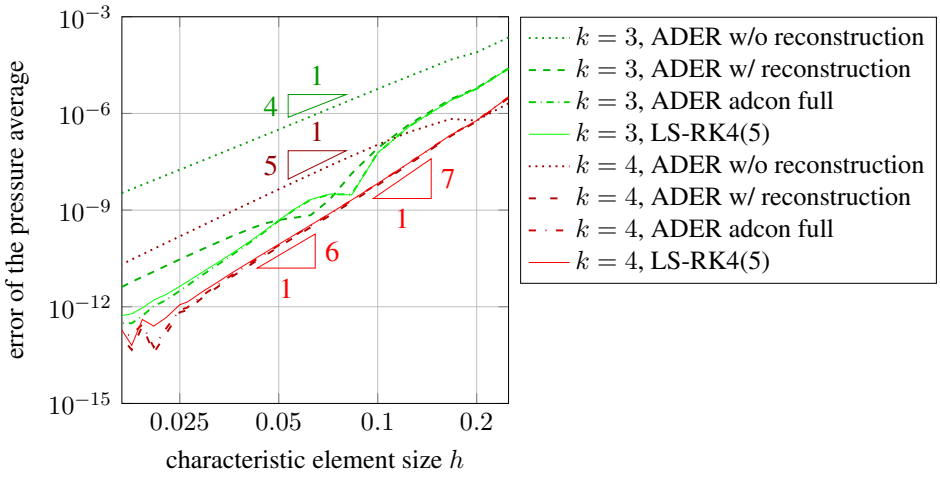
In terms of time to solution, the maximal stable time step size is of crucial relevance. Generally, the critical Courant Cr_{crit} number is known for a numerical method and the time step size is set according to

$$\Delta t \leq Cr_{\text{crit}} \frac{h}{ck^{1.5}}.$$

Here, the critical Courant numbers are measured for representative Runge–Kutta time discretizations, namely the aforementioned low-storage Runge–Kutta scheme LSRK4(5) with five stages



(a) One mode $(m = 1)$



(b) Two modes $(m = 2)$

Figure 2.7: Convergence of the average pressure.

of order four with two registers and the classical Runge–Kutta scheme of four stages of order four, denoted clRK4. Additionally, ADER time discretization is examined with reconstruction according to equation (2.31) and also in the fully adjoint consistent version according to equation (2.32) (denoted ADER adcon full).

The setup is as in the previous chapter, except that the grid is fully Cartesian without interior deformation. In 2D $10^2 = 100$ elements are used. In 3D, $6^3 = 216$ elements are used. The number of membrane modes is set to $m = 2k$.

The critical Courant number is determined iteratively according to the procedure described in [146] starting from an initial Courant number of $Cr = 0.5$ in twelve iterations. Therein, simulations are run and a stability criterion is tested: if the L_2 pressure error at the final time is smaller than 100 times the initial L_2 error of the pressure and if the L_2 pressure error at the final time is smaller than the pressure magnitude at the initial time, a simulation is considered as stable, otherwise as unstable. The results are given in Figure 2.8 for LSRK4(5), clrk4, ADER, and the fully adjoint consistent ADER scheme. Comparison of LSRK4(5) and ADER shows that both require smaller Courant numbers for higher polynomial degrees and that the LSRK4(5) can work with a twice as large time step, with slightly decreasing benefit for increasing polynomial degree. The fully adjoint consistent ADER scheme shows a different qualitative behavior. The critical Courant number increases for increasing polynomial degree. For $k = 1$, the critical Courant number is the same as for the standard ADER scheme, because the reconstruction scheme applies to the $k = 1$ term and both schemes are equal. For $k = 3$, ADER adcon full has the same critical Courant number as the clRK4 scheme because they have the same stability function for this polynomial degree. The trend of the curve indicates that the definition of the Courant number as in (2.16) with the dependency on the polynomial degree $k^{1.5}$ is not suitable for the fully adjoint consistent scheme. Figure 2.9 plots the critical Courant number for an alternative definition with linear dependence on the polynomial degree, i.e., $\tilde{Cr} = {}^{c\Delta tk}/_{h}$, yielding almost constant values.

In general, the LSRK4(5) has weaker restrictions on the time step size compared to ADER. The time step can be chosen between 2.36 and 1.65 times larger than for ADER. In terms of computational expense, this is a benefit for LSRK4(5), if it is reasonable to run the simulation with the maximal allowed time step size. The fully adjoint consistent ADER scheme allows for a larger time steps size than LSRK4(5) for very high polynomial degrees of the shape functions $k \geq 9$.

CFL Stability Limit for ADER LTS

For ADER LTS, the formulation of the time step stability limit is more complex, since the time step differs from element to element. To determine the dependence of the stability limit on the parameter δ controlling the difference of the time step between neighboring clusters, a non-uniform mesh with large variations in the characteristic element size h is used, see Figure 2.10. The mesh is deformed starting from a Cartesian mesh by moving vertices in x_1 direction according to

$$x_1 \leftarrow x_1 + 0.3 \cdot \prod_{i=1}^{d} \sin(\pi x_i).$$

The characteristic element size h is determined as the minimal vertex distance and differs by a factor of about 20 in the mesh correlating to optimal time step sizes between Δt_{min} and $20 \cdot \Delta t_{\text{min}}$.

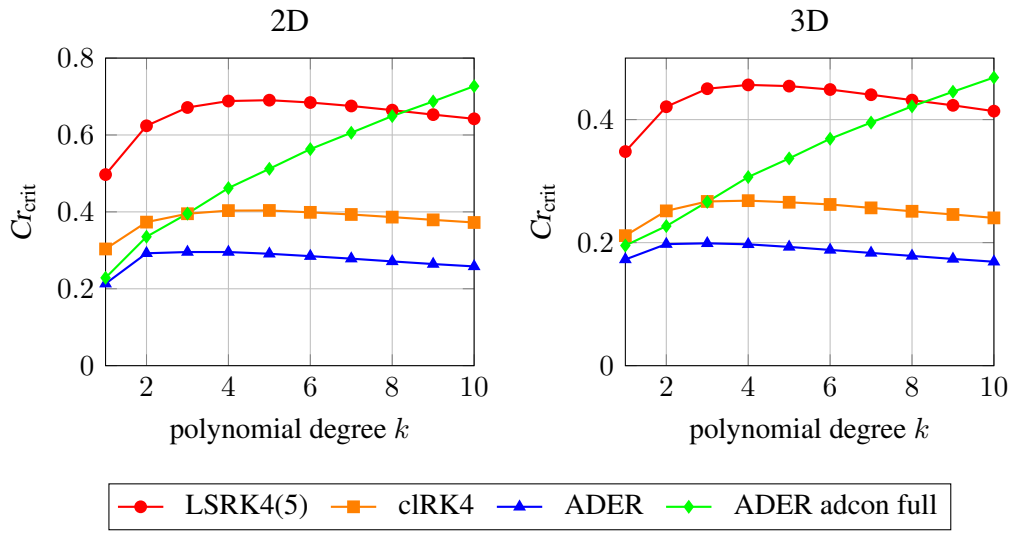


Figure 2.8: Critical Courant number Cr_{crit} in two and three dimensions comparing low-storage Runge–Kutta of order four with five stages LSRK4(5), standard ADER, and fully adjoint consistent ADER (ADER adcon full).

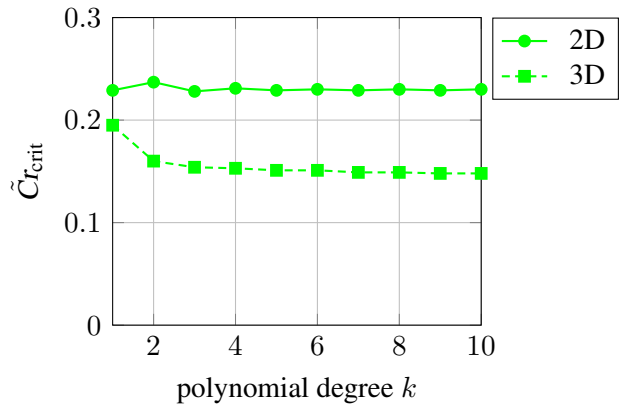
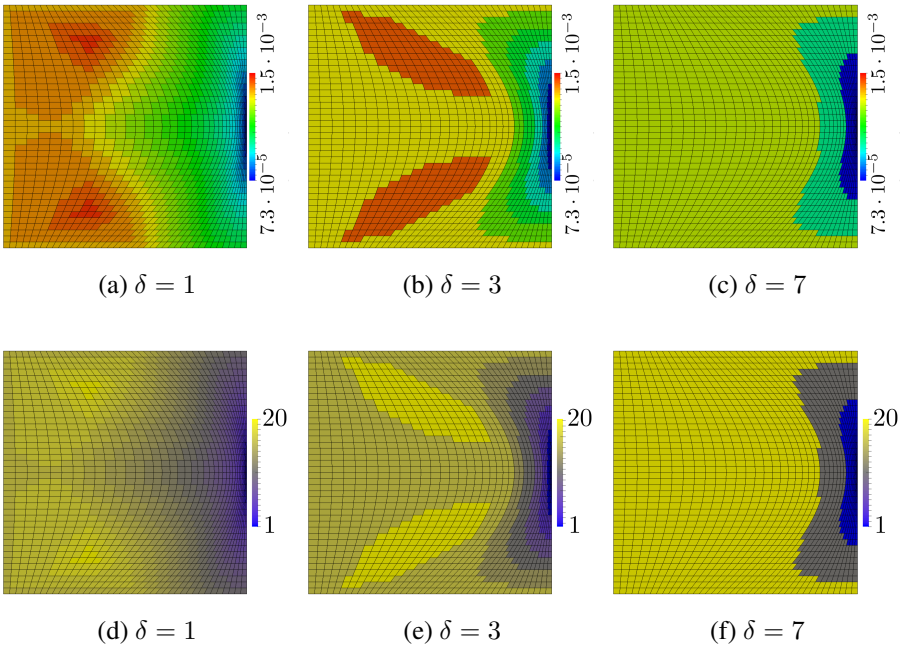


Figure 2.9: Critical Courant number with alternative definition according to $\tilde{C}r = {}^{c\Delta tk}/_{h}$ for the fully adjoint consistent ADER scheme (ADER adcon full).



(a) $\delta = 1$ (b) $\delta = 3$ (c) $\delta = 7$

(d) $\delta = 1$ (e) $\delta = 3$ (f) $\delta = 7$

Figure 2.10: Mesh for CFL stability analysis for ADER LTS. (a)–(c) show time step size distributions while (d)–(f) show the cluster setup.

Exemplary cluster distributions and time step size distributions for $\delta = 1, 3, 7$ are shown in Figure 2.10. As can be seen from the figure, elements are potentially operated on smaller than their optimal time steps.

The critical Courant number Cr_{crit} for various polynomial degrees of the shape functions and for different parameters δ is determined by the iterative procedure mentioned above. The critical Courant number is also determined for the global ADER time stepping for comparison. The reference with the global ADER time stepping can also be understood as ADER LTS with $\delta = 0$. The results from Figure 2.8 are not directly transferable because there, uniform meshes are used and here, elements are strongly distorted. Note that the characteristic element size h as minimum vertex distance is not necessarily the best measure to compare distorted elements to elements of a Cartesian mesh. Figure 2.11 plots the critical Courant numbers and minimal time step sizes for $k = 1, 4$.

The critical Courant number and the time step size vary slightly with the allowed cluster difference δ. Only for $\delta = 0$ representing the results obtained with the global ADER time stepping, the critical Courant number and time step size are significantly higher. This yields the conclusion that the stability critical elements are neither the smallest nor the largest elements, which is due to the fact that the characteristic element size is determined as minimum vertex distance and this quantity does not represent the stability properties of distorted elements accurately. The variations in the critical time steps between $\delta = 1$ and $\delta = 10$ are due to the cluster distributions as shown in Figure 2.10 and if the most critical elements are downcast or closer to their stability limit.

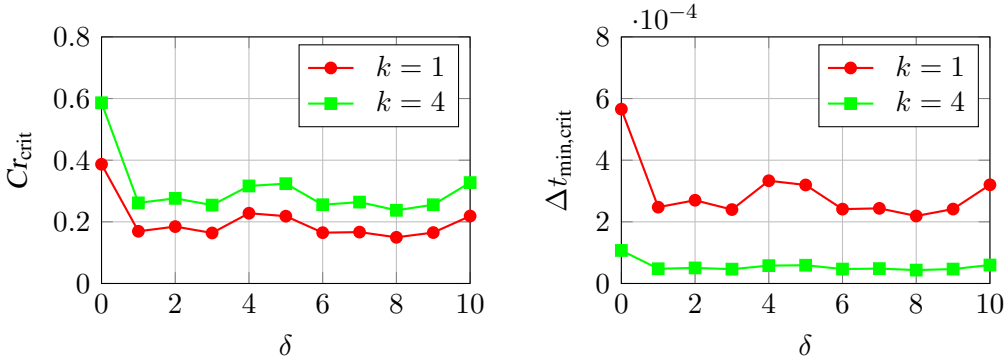


Figure 2.11: Critical Courant number Cr_{crit} in two and three dimensions for ADER LTS (and global ADER for reference as $\delta = 0$).

2.5.3 Amplitude and Phase Error

ADER time discretization in combination with finite volume space discretization is well known for its beneficial dispersion and dissipation behavior [149]. The numerical results from [149] indicate that dispersion errors are partly rectified by dissipation errors because waves traveling with inaccurate wave speed are damped. The work [48] complements the numerical findings with analytic derivations of dispersion and dissipation coefficients.

Here, a simple one-dimensional example is studied to reveal the principal dispersion and dissipation properties of the presented method. This numerical example is based on [149, 177] and is analogously published in [145]. A line of length $L = 2$ is meshed with n_{e} elements of polynomial degree k. The left and right end are connected with a periodic boundary condition. The speed of sound is set to $c = 1$ as well as the mass density $\rho = 1$. The analytic solution for this problem is chosen to be

$$p_0 = \sin(\pi(x - t)),$$
$$v_0 = \sin(\pi(x - t)),$$

and the initial conditions are set accordingly. The end time is set to $T = 1000$, which means that the initial wave traverses the computational domain 500 times. The amplitude error is defined as

$$e_{\text{A}} = \max(p_0(x)) - \max(p_h(x, t = T)). \tag{2.49}$$

For the numerical tests, it is evaluated as the L_∞ norm of the pressure field using a sufficiently dense sampling of the solution. The phase error e_{P} is defined as the position of the root of the pressure distribution, which was initially located at $x = 0$. It is evaluated using a linear interpolation between the points of a sufficiently dense sampling of the solution.

To get a first insight into the dispersion and dissipation behavior of the method at hand, Figure 2.12 shows the solutions at the final time T for quadratic and cubic shape functions on a mesh with eight and four elements, respectively. The time step size was chosen such that $Cr = 0.1$. In Figure 2.12(a), the dissipation error is visualized for the two numerical schemes: the amplitude is decreased in contrast to the analytic solution. In the zoom to the zero-crossing in Figure 2.12(b), the shift of the curves according to the phase error is displayed. Plotting the amplitude and phase

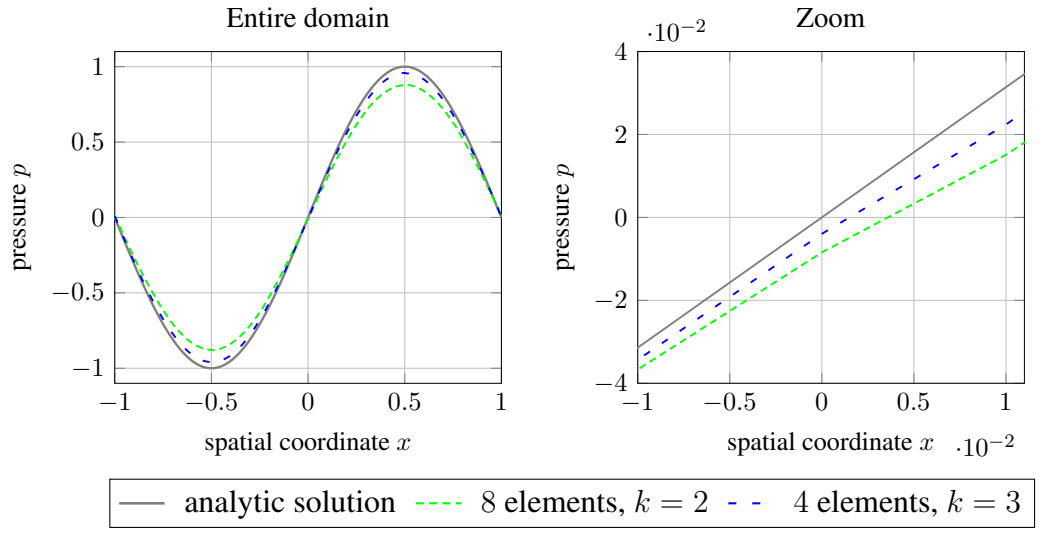
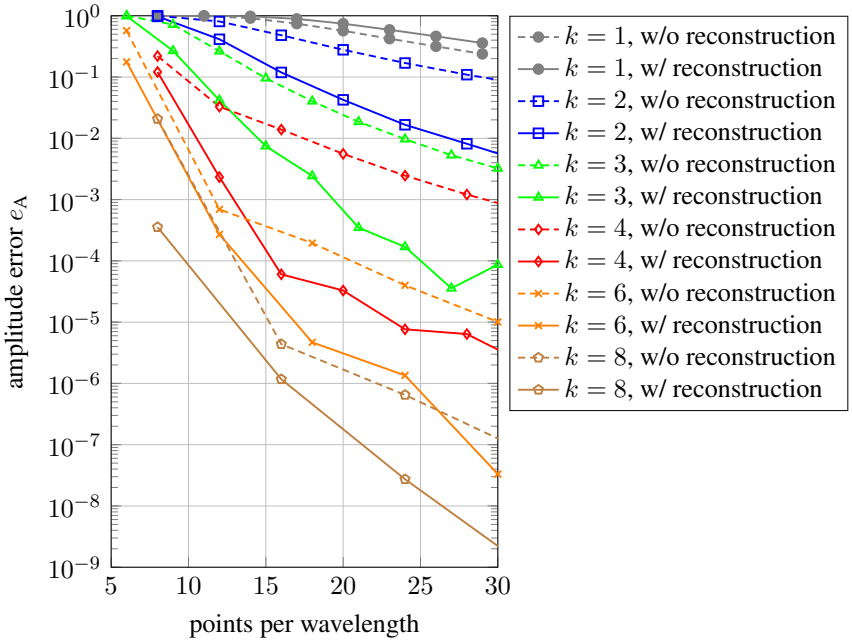


Figure 2.12: Analytic and numerical solution at final time T.

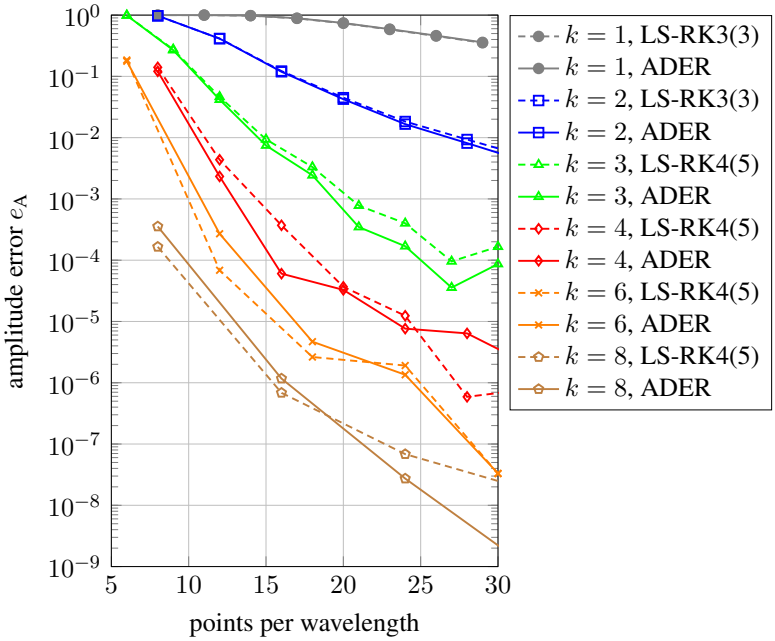
errors over time shows linear growth in time. Despite this theoretically unbounded error increase, no long-time instabilities were observed for any simulation setup.

Figures 2.13 and 2.14 show the amplitude and phase errors in dependence on the number of points per wavelength for the described setup for different discretizations. Results for ADER time integration with polynomials of degrees $k = 1, 2, 3, 4, 6, 8$ are presented for a Courant number of $Cr = 0.1$ with and without reconstruction in panels (a) of Figures 2.13 and 2.14. Additionally, ADER HDG is compared to low-storage Runge–Kutta scheme LSRK3(3) and LSRK4(5) of third and fourth order [90] in panels (b) of Figures 2.13 and 2.14. The qualitative error behavior appears similar to the one presented in [149]. Comparison of ADER with and without reconstruction reveals the improvement of the reconstruction scheme in terms of amplitude and phase errors. With reconstruction, ADER mostly performs better than Runge–Kutta time integration; without reconstruction, the ADER results are worse. An interesting difference is that the phase errors are negative (waves propagate too slowly) without reconstruction but mostly positive with reconstruction and for Runge–Kutta. Even though the Taylor truncation error is not altered between the full method with and without reconstruction, the reconstruction has a significant influence on the amplitude and phase errors. As can be seen from Figure 2.13(a), $k = 3$ with reconstruction gives smaller amplitude errors than $k = 4$ without reconstruction.

The errors decrease for increasing polynomial degree k. The decrease is not as smooth as for the L_2 errors reported in Section 2.5.1 but general trends can be observed. For $k = 1, 2, 3, 4, 6, 8$, the amplitude error seems to converge with orders $2, 4..5, 6..8, 7..9, 8..11, 8..12$. For the absolute phase error it is $4, 4, 6, 7..8, 8..9, 10..11$. In [1], orders of $2k + 3$ and $2k + 2$ are predicted for dispersion and dissipation error, but they are based on analysis of the symbol of the discretized differential operator: for the dispersion error, the real part of the true and the numerical wave number are compared, while the imaginary part of the numerical wave number is considered for the dissipative error.

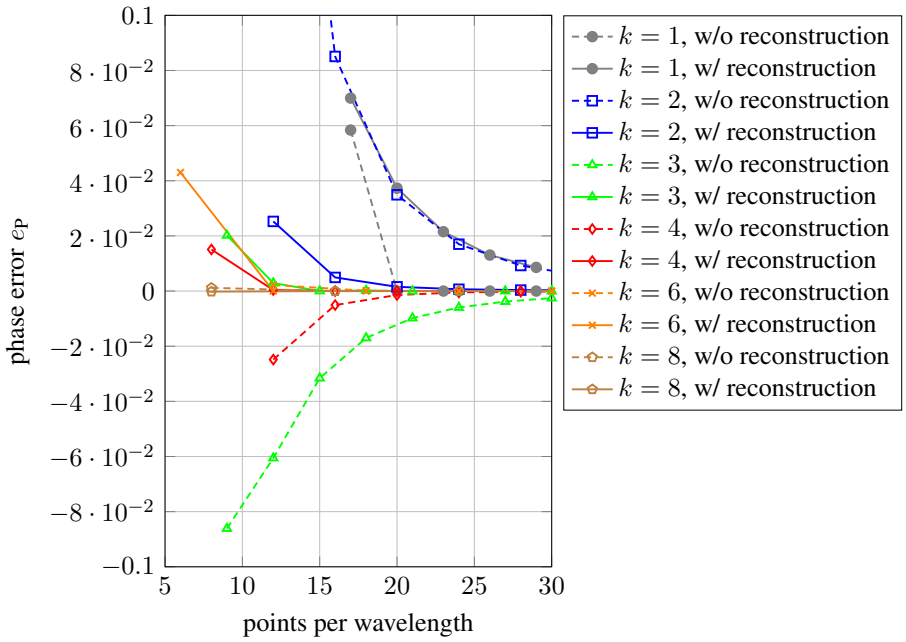


(a) Amplitude error for ADER at $Cr = 0.1$

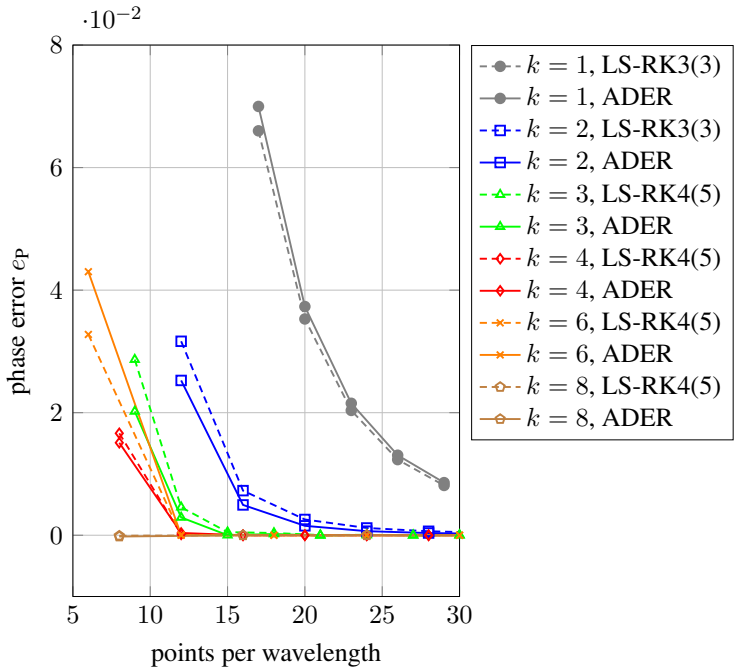


(b) Amplitude error for ADER with reconstruction and Runge–Kutta at $Cr = 0.1$

Figure 2.13: Amplitude error for different discretizations at $Cr = 0.1$.



(a) Phase error for ADER at $Cr = 0.1$



(b) Phase error for ADER with reconstruction and Runge–Kutta at $Cr = 0.1$

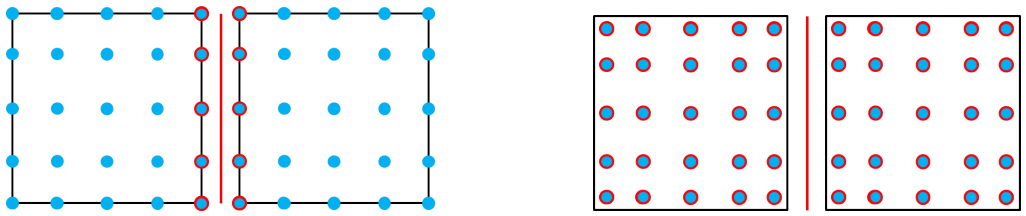
Figure 2.14: Phase error for different discretizations at $Cr = 0.1$.

3 Implementation Aspects

This chapter is written following [144] and most parts are quoted literally.

The description of the methods introduced in Chapter 2 covered the physical and numerical aspects. In this chapter, the algorithmic backgrounds are considered because the implementation of numerical methods on modern hardware is an essential component in terms of efficiency. Modern processors favor arithmetically intense algorithms such as dense matrix multiplications over patterns typical for many solvers of partial differential equations that include kernels of a streaming character with much fewer arithmetics. Thus, algorithmic complexities alone are not enough to judge a method's efficiency as the achievable performance of e.g. a sparse matrix-vector product in terms of floating point operations per second can be almost two orders of magnitude below the advertised peak performance [169].

The algorithmic developments were carried out for the problems and methods derived in Chapter 2, i.e., for the acoustic wave equation discretized with (H)DG and explicit Runge–Kutta methods or ADER time integration and are in parts transferable to other problems. ADER has been successfully applied in the context of finite volume and DG methods [47, 49, 150]. An advantage of ADER over explicit Runge–Kutta schemes is that ADER is not restricted by the Butcher barriers and convergence orders beyond four are not overproportionally expensive. Previous work on ADER DG (as in [21]) relies on triangles or tetrahedra assuming constant coefficients and straight lined boundaries. The operator evaluation in [21] is carried out based on an element matrix with a theoretical complexity per degree of freedom of $\mathcal{O}(k^d)$ in the degree k for spatial dimension d. Here, an ADER DG formulation for quadrilaterals and hexahedra for variable coefficients and curved geometries is proposed. Matrix-free operator evaluation relying on fast quadrature with sum-factorization kernels with a theoretical complexity per degree of freedom of $\mathcal{O}(dk)$ is used. The techniques of sum factorization with fast quadrature have been established by the spectral element community [44, 88, 93, 125] but are also popular in the DG community for explicit time integration [77]. Advances in computer architecture have rendered the matrix-free evaluation, originally targeting high orders beyond around five, also highly competitive at moderate orders, outperforming the memory bandwidth-limited sparse matrix-vector product for second and higher degree polynomials [22, 94]. In terms of algorithmic layout, the sole reduction of arithmetic operations is not advantageous if the memory bandwidth is the performance limiting factor. It is shown that an operator evaluation with sum factorization as in explicit Runge–Kutta schemes is memory bandwidth bound, despite its clear improvement over matrix-based operator evaluation. ADER replaces the global operator application in each Runge–Kutta stage by one global operator application and a completely element-local evaluation routine, the Taylor–Cauchy–Kowalevski procedure, which allows to perform more computations on data read from the global solution vectors. It is shown that ADER does not only employ fewer operations but also supplies a higher arithmetic intensity, which is beneficial on modern cache-based hardware.



(a) Nodal basis with nodes on element faces

(b) Nodal basis with nodes only in the interior of the elements

Figure 3.1: Value access for flux evaluation depending on the nodal positions shown for two elements with $k = 4$. Blue circles indicate nodal positions, the red line represents the face for flux evaluation, and blue circles with red contour show, which node values must be accessed for the flux evaluation.

Another aspect significant for performance is the choice of the shape function nodal points and the choice of the quadrature rule. In case nodal points and quadrature points coincide, interpolation of the solution to quadrature points is avoided and computational expense is decreased. This approach is well known in the context of spectral elements. Usage of the Gauss–Lobatto points for the definition of the nodal points and for integration was shown to degrade the accuracy of the mass matrix and its inverse [51, 161], though. Consistent Gaussian quadrature instead yields full accuracy. A drawback of nodes in the Gauss points, however, is that the flux evaluation on element faces requires an extrapolation accessing all degree of freedom values of both adjacent elements because there are no node points on the faces. For the flux evaluation, nodal points on the element faces ensure that only $(k+1)^{d-1}$ instead of $(k+1)^d$ values must be accessed as demonstrated in Figure 3.1. A new algorithmic method is proposed that changes the polynomial basis and its nodes on the fly depending on the quantity to be evaluated. The standard DG global derivative oper2ator including flux evaluations relies on a Lagrange basis with Gauss–Lobatto points while the ADER specific element local Taylor–Cauchy–Kowalevski procedure relies on a Lagrange basis with nodes in collocated Gauss points. Thereby, cheap element evaluation and flux evaluation with minimal data access are combined. Despite this optimization concerning node and quadrature choice, a second optimization concerning the efficient evaluation of higher order spatial derivatives required in the Taylor–Cauchy–Kowalevski procedure is proposed. Calculation of first order spatial derivatives and successive projection to a lower order basis in combination with the collocated node and quadrature points minimize computational work.

This chapter is structured as follows. The basic algorithmic building blocks are described in the first part of Section 3.1. A quantitative study on the throughput for bases with or without nodes on element faces is given in 3.1.1, which motivates the basis switching approach presented in 3.1.2. The optimization relying on reduced polynomial spaces for higher spatial derivatives is shown in Section 3.1.3. A detailed performance analysis in terms of theoretically derived operation counts, throughput, the roofline model, computational timings, and scalability is given in Section 3.2. A conclusion on the algorithmic developments is draw in Section 3.3.

3.1 Algorithmic Developments

For the sake of clarity and brevity, a more compact notation is used throughout this chapter. The pressure and the velocity are summarized in a vector $\boldsymbol{u}$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{v} \\ p \end{bmatrix},$$

and the values of the degrees of freedom are summarized accordingly

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{P} \end{bmatrix},$$

with the same relation as in equation (2.26)

$$\boldsymbol{u}_h = \mathbb{N}\boldsymbol{U}.$$

With this notation, the ADER HDG time stepping without reconstruction as in equation (2.27) is given as

$$\boldsymbol{U}_{t_{i+1}} = \boldsymbol{U}_{t_i} - \mathbb{Q}^{-1}\mathbb{K}\mathbb{Q}^{-1} \sum_{j=0}^{k} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^j \mathbb{N}\mathrm{d}K \boldsymbol{U}_{t_i}. \tag{3.1}$$

The method with reconstruction as in (2.31) in the new notation reads

$$\begin{aligned} \boldsymbol{U}_{t_{i+1}} = \boldsymbol{U}_{t_i} &- (t_{i+1}-t_i)\mathbb{Q}^{-1}\mathbb{K}\boldsymbol{U}_{t_i} \\ &- \mathbb{Q}^{-1}\mathbb{K}\mathbb{Q}^{-1} \left(\sum_{j=1}^{k+1} \frac{(t_{i+1}-t_i)^{j+1}}{(j+1)!}(-1)^j \int_K \mathbb{N}^{\mathrm{T}}\mathbb{S}^{j-1}\mathbb{N}\mathrm{d}K \right) \mathbb{Q}^{-1}\mathbb{K}\boldsymbol{U}_{t_i}. \end{aligned} \tag{3.2}$$

Now that a brief notation is introduced, the basic underlying algorithmics on which the new developments build are described. Subsequently, a brief preliminary performance evaluation for different polynomial bases is given in Section 3.1.1. The evaluation of the integrals in the weak forms in equations (3.1) or (3.2) is performed by fast integration relying on sum factorization utilizing the structure of tensor product shape functions that are integrated with a tensor product quadrature rule. In the remainder of this work, a Gaussian quadrature with $k+1$ points per direction for polynomials of degree k is chosen, which is enough to integrate bilinear forms with element-wise constant coefficients on affine element shapes exactly. In particular, this choice avoids the accuracy penalty of inexact Gauss–Lobatto quadrature on $k+1$ points as highlighted in [51]. On curved geometries, there is a possible integration error that is often subsumed in the errors from variational crimes [20]. Note that more general hyperbolic problems with nonlinear terms can easily be integrated with more points to avoid aliasing effects in this setup, see for example [56].

For a function described by a basis of tensor degree k and $(k+1)^d$ coefficients, the interpolation onto $(k+1)^d$ quadrature points takes $(k+1)^{2d}$ additions and multiplications in a naive implementation without sum factorization. The evaluation of each component of the gradient takes $2(k+1)^{2d}$ operations. With sum factorization however, the work for the interpolation is

reduced to $2d(k+1)^{d+1}$ operations. The reduction of operations results from d applications of one-dimensional interpolations of cost $2(k+1)^2$ that go through all $(k+1)^{d-1}$ lines of basis functions and quadrature points, respectively. In the remainder of this work, one application of a one-dimensional interpolation over all points, involving $(k+1)^{d+1}$ operations, is refered to as one "tensor product kernel". Note that these kernels can be cast as small matrix-matrix multiplications. For details, see [95].

In case of quadrature over Lagrange polynomials with nodes in the points of the quadrature formula (a so-called collocated setup [93]), the interpolation from coefficients to values in quadrature points is the identity operation and can be skipped. Thus, the evaluation of all d components of the gradient only involves d tensor product kernels for each of the partial derivatives, as opposed to d^2 tensor product kernels for the basic evaluation scheme. As will be elaborated in this work, the optimization of representing the solution coefficients via a collocated basis may be premature as other factors, such as the cost of face integrals, may control the decision about which basis to prefer. Similar kernels are also used for the summation over quadrature points when multiplying with test functions or test function gradients, see e.g. [95] for details.

For the experiments, a state-of-the-art implementation of sum factorization based on the finite element library `deal.II` with support for massively parallel computations and adaptively refined meshes with hanging nodes is used [4]. Since integration involves a series of heavy arithmetics, the use of vectorization (SIMD) is fundamental for getting optimal performance on current architectures. Following the concepts described in [95, 98], this work applies vectorization over several elements which was found to provide best performance on polynomial degrees up to at least 14 and reaches more than 50% of the arithmetic peak performance when considered in isolation, which is an extremely high value for a code compiled from generic C++ code that contains k as a (template) parameter that lets the compiler decide on the loop unrolling and register allocation.

3.1.1 Efficient Face Integral Evaluation

In this section, the impact of the polynomial basis functions on the efficiency of the evaluation of the derivative operator $\mathbb{K}$ and the inverse mass matrix $\mathbb{Q}^{-1}$ are studied. Two contradictory factors are considered: The inverse mass matrix is efficiently evaluated if quadrature points and nodes of the polynomial basis coincide, which results in a diagonal mass matrix and hence simple inversion. On the other hand, the derivative operator $\mathbb{K}$ includes the evaluation of both element and face integrals. For the face integrals, data from both adjacent elements are required. For nodal polynomials with nodes on the element faces, only the values associated to the nodes on the faces must be accessed. If the nodes are only in the interior or polynomials are not nodal, all vector entries of both adjacent elements are required and an extrapolation to the faces must be carried out, see Figure 3.1.

Two variants are examined to demonstrate the effects of the aforementioned contradictory factors. In one case, a polynomial basis of Lagrange functions with nodes in the same Gauss points is used, in the other case a polynomial basis of Lagrange functions with nodes in the Gauss–Lobatto points is used. Quadrature is always based on $(k+1)^d$ Gaussian points. The first case results in a diagonal mass matrix, the second case satisfies the requirement to have nodes on the element faces. Note that the usage of Gauss–Lobatto points as nodes and quadrature points is not considered because it degrades accuracy [51, 161]. Figure 3.2 plots the results in terms of the

number of degrees of freedom processed per second for a three-dimensional geometry consisting of 80^3 Cartesian elements for polynomial degree $k \in \{1, 2, 3\}$ and 40^3 Cartesian elements for $k \in \{4, \dots, 12\}$. Computations are run on the system specified in Table 3.1.

The evaluation of the inverse mass matrix reaches a considerably higher throughput compared to the derivative operator $\mathbb{K}$ because it is completely element-local whereas $\mathbb{K}$ requires the evaluation of face integrals and therefore additional data access from neighboring elements. In fact, the throughput of the former is mostly limited by the memory bandwidth of loading the vector and writing back into the same vector, which is $6.2 \cdot 10^9$ degrees of freedom per second for the system's memory throughput of 115 GB/s when counting $2 \times 8(d+1)$ byte per polynomial (read and write) and 8 bytes for access to the precomputed inverse entry of the diagonal, which is the same for all $(d + 1)$ components. Following [97], the action of the inverse mass matrix can be cheaply evaluated by sum-factorization kernels that first transform into an orthogonal basis (i.e., the Lagrange polynomials in the $(k + 1)^d$ points of Gauss quadrature), apply the inverse diagonal mass matrix, and transform back to the original basis. The comparison to the throughput for collocated nodes of shape functions and quadrature points shows that indeed the evaluation of the non-diagonal mass matrix operation is only up to 15% slower compared to the diagonal mass matrix. Thus, the substantial arithmetic operations can be almost completely hidden behind the unavoidable memory transfer.

The face evaluation for the Lagrange basis in Gauss points does not only require to read the values for the degrees of freedom located on the face but all values of the adjacent elements to allow interpolation onto the face (or to write the face data into a separate global array with additional memory transfer as done in [74, 77]). Comparing the throughput for Lagrange polynomials in Gauss–Lobatto points and Gauss points reveals a significant drop in performance for the derivative operator due to the increased memory access, including substantial indirect addressing components as described in [95]. The direct comparison highlights that using polynomial bases with nodal points on the element faces for the global derivative operator is highly beneficial. This finding applies particularly to Runge–Kutta type time integrators, and it also holds for more general nonlinear operators $\mathbb{S}$.

The main findings are summarized as follows:

- The evaluation of the inverse mass matrix is memory bandwidth bound (especially for moderate order) and a change of basis is for free.
- The throughput for the derivative operator is much lower than for the mass matrix, because values from both adjacent elements must be read in contrast to complete element-local evaluations. Combined with the fact that also the Jacobian of the mapping must be accessed, the memory bandwidth is reached earlier.
- This effect is counteracted by usage of a polynomial basis with nodes on the element boundary, which will be further investigated in the following section.
- Usage of collocated node and quadrature points reduces the number of operations.

From these conclusions, ADER is self-evidently motivated: applications of the global derivative operator $\mathbb{K}$ are traded for element local evaluations in the Taylor–Cauchy–Kowalevski procedure.

3.1.2 Flexible Basis Change

In the previous section, the effect of the memory bandwidth bound became apparent. Despite the memory access, the number of operations is a central quantity of interest to judge code efficiency.

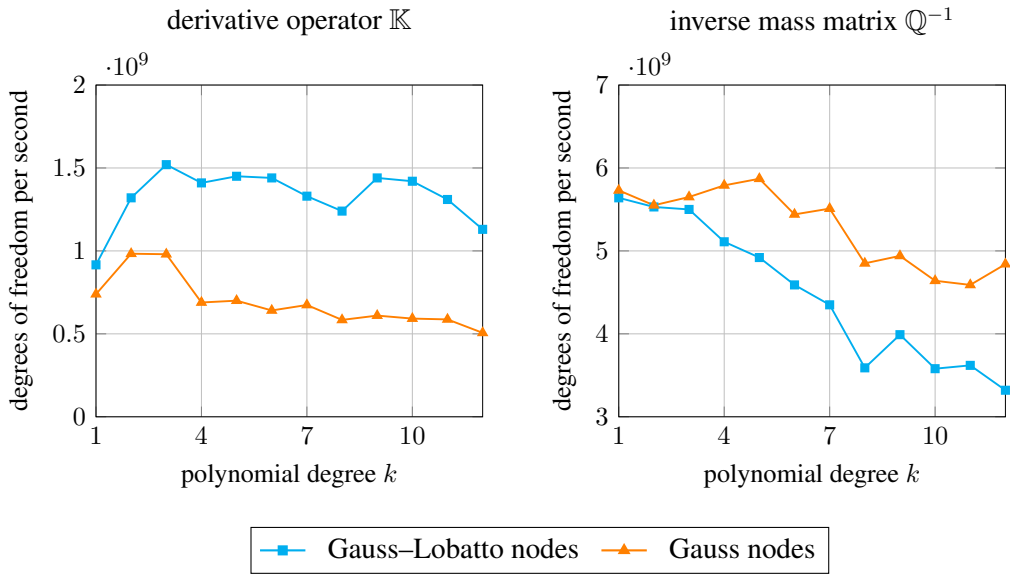


Figure 3.2: Degrees of freedom processed per second for the derivative operator $\mathbb{K}$ and the inverse mass matrix $\mathbb{Q}^{-1}$ on a Cartesian grid for polynomial degrees $k \in \{1, \dots, 12\}$. Gaussian quadrature on $(k+1)^d$ points is used. In one case, the polynomial basis consists of Lagrange polynomials with nodes in Gauss–Lobatto points. In the other case, the Lagrange polynomials have their nodes in the Gauss points yielding a diagonal mass matrix.

As shown in Figure 3.2, a set of shape functions where only $(k+1)^{d-1}$ functions evaluate to non-zero on each of the element faces is beneficial to reduce the vector access for face evaluation. For element local evaluations, however, this involves additional d tensor product kernels per component to interpolate from the solution values to quadrature points as compared to the collocated node and quadrature points commonly used in spectral element solvers [93]. Hence, different bases should be used for the different phases. For element local evaluations, a collocated basis with nodes in Gauss points (denoted by G) is the best choice while a basis with nodes in Gauss–Lobatto points (denoted by GL) is the best choice for the evaluation of face integrals. It is proposed to switch the basis for the different phases on the fly while looping over the elements for the Taylor–Cauchy–Kowaleski evaluation. The solution approximation is expressed either as

$$\boldsymbol{u}_{\mathrm{G}} = \mathbb{N}_{\mathrm{G}} \boldsymbol{U}_{\mathrm{G}} \qquad \text{or} \qquad \boldsymbol{u}_{\mathrm{GL}} = \mathbb{N}_{\mathrm{GL}} \boldsymbol{U}_{\mathrm{GL}},$$

with the matrices $\mathbb{N}_{\mathrm{G}}, \mathbb{N}_{\mathrm{GL}}$ containing the shape functions and the vectors $\boldsymbol{U}_{\mathrm{G}}, \boldsymbol{U}_{\mathrm{GL}}$ containing the degree of freedom values for Gauss and Gauss–Lobatto nodes, respectively. Theoretically, there are no restrictions objecting to change the basis from one evaluation to the other. Since both spaces are of the same degree, the equality $\boldsymbol{u}_{\mathrm{G}} = \boldsymbol{u}_{\mathrm{GL}}$ holds in all cases.

For ADER, the collocated G basis is not only interesting for the application of the inverse mass matrix $\mathbb{Q}^{-1}$ but especially for the Taylor–Cauchy–Kowalevski term

$$\sum_{j=0}^{k} \frac{(t_{i+1} - t_i)^{j+1}}{(j+1)!} (-1)^j \int_K \mathbb{N}^{\mathrm{T}} \mathbb{S}^j \mathbb{N} \mathrm{d}K \boldsymbol{U}_{t_i}$$

summing weighted spatial derivatives from zeroth to k-th order with different prefactors. Higher order derivatives are computed by iterative calculation of a first derivative and a subsequent projection applying the inverse mass matrix. The Taylor–Cauchy–Kowalevski term is completely element local and significantly more calculations are operated on the read data compared to the evaluation of the inverse mass matrix.

A successive evaluation of the gradient of a field and projection instead of direct evaluation of high derivatives is proposed, which significantly improves efficiency in the context of a matrix-free implementation on general meshes. Successively, problems of the form $(w, \widetilde{u}^{i+1})_K = (w, \widetilde{u}^{i})_K$ are solved, where the j-th spatial derivative is denoted $\widetilde{u}^{j} = \nabla^j u$. Algorithm 1 shows how the spatial derivatives are calculated by evaluation of the gradient field and projection onto the degrees of freedom for the non-collocated basis GL. The application of the inverse mass matrix must be understood in the matrix-free sum-factorization context according to [97].

Algorithm 1 Evaluation of j-th order derivatives for the non-collocated basis GL on general non-Cartesian grids.

for $i = 1, \dots, j$ **do**
  evaluate $\nabla \widetilde{u}^{i}_{\mathrm{GL}}$ in the integration points $\boldsymbol{\xi}_{\mathrm{G}}$
  multiply with weighting functions evaluated in integration points $\boldsymbol{\xi}_{\mathrm{G}}$ and sum to get right hand side vector $\boldsymbol{r}_{\mathrm{GL}} = (w_{\mathrm{GL}}, \nabla \widetilde{u}^{i}_{\mathrm{GL}})_K$
  apply inverse mass matrix $\mathbb{Q}^{-1}_{\mathrm{GL}} \boldsymbol{r}_{\mathrm{GL}}$ to get coefficients $\widetilde{\boldsymbol{U}}^{i+1}_{\mathrm{GL}}$ of the field $\widetilde{u}^{i+1}_{\mathrm{GL}}$
end for

The algorithm simplifies drastically if nodes and integration points are collocated because the weighting with test functions and the application of the inverse mass matrix cancel out and all the interpolation matrices are identity matrices. The simplified procedure is shown in Algorithm 2.

Algorithm 2 Evaluation of j-th order derivatives with the collocated basis G.

for $i = 1, \dots j$ **do**
  evaluate $\nabla \widetilde{u}^{i}_{\mathrm{G}}$ in the integration points $\boldsymbol{\xi}_{\mathrm{G}}$
  set values for $\widetilde{\boldsymbol{U}}^{i+1}_{\mathrm{G}}$ of the field $\widetilde{u}^{i+1}_{\mathrm{G}}$
end for

Note that the basis change is done on the fly when processing the data from one element, not on the global solution vector as that would incur additional memory transfer. If the current global solution vector contains the degree of freedom values in the GL basis description, the first evaluation in Algorithm 2 involves the interpolation from GL to G and then all remaining evaluations from $i = 2$ to $i = j$ are computed purely in the collocated basis G. The contribution is finally written in the basis GL:

$$u_{\mathrm{GL}} \xrightarrow{\text{basis change}} u_{\mathrm{G}} \longrightarrow \underset{\text{terms in G}}{\text{Taylor–Cauchy–Kowalevski}} \xrightarrow{\text{basis change}} \text{contribution in GL.}$$

3.1.3 Degree Reduction

A possibility to further reduce the work in the evaluation of the Taylor–Cauchy–Kowalevski sum is to reduce the polynomial degree in the representation of higher order spatial deriva-

tives. This is a well-established method in the ADER DG community for affine element shapes where typically hierarchical bases are used [21, 47]. The higher order spatial derivatives naturally give a contribution only to the lower degree coefficients of the hierarchical basis in the Taylor–Cauchy–Kowalevski procedure. In the case of sum factorization evaluation on general non-Cartesian meshes, however, the embedding to lower polynomial degrees involves additional operations as compared to the spectral evaluation routine from Algorithm 2. In a hierarchical basis, i.e., Legendre polynomials on quadrilaterals or hexahedra, the integrals with a non-affine element geometry are evaluated by quadrature with sum factorization, which in turn must transform between the Legendre basis and the collocation basis. For one spatial derivative i in Algorithm 2, the complexity rises from d tensor product kernels per component in the spectral evaluation to $3d$ tensor product kernels, where d kernels each are needed for the basis change in interpolation and integration, respectively.

A more efficient algorithm can be devised as follows. First, the degree in terms of the Lagrange basis in the respective Gaussian integration points of degree $k-i+1$ and $k-i$ of step i in the Taylor–Cauchy–Kowalevski sum is reduced. The degree reduction is performed by an operation $\mathbb{P}^{k-i}_{k-i+1}\widetilde{u}^{i+1}$ where $\mathbb{P}^{k-i}_{k-i+1}$ is the projection operator from degree $k-i+1$ to $k-i$ on the reference element. Like in the hierarchical case, this setup combined with the additional interpolation of the result into the points of the Taylor–Cauchy–Kowalevski sum involves $3d$ tensor product kernels per component, as compared to only d kernels for the spectral derivative. Thus, as a second ingredient it is proposed to apply the degree reduction only for every second spatial derivative. This step also has the advantage of limiting the extra amount of geometry information, i.e., the inverse Jacobian, that needs to be loaded in each quadrature point, as Gauss formulas of different degrees evaluate the integrands in different positions and the implementation uses pre-computed inverse Jacobians. In other words, the data of the inverse Jacobians loaded into caches is reused once again. The i-th spatial derivative $\tilde{u}^i$ is thus expressed in a basis of polynomial degree $k - \lfloor j/2 \rfloor \cdot 2$. Algorithm 3 details this procedure for $j = 2$.

Algorithm 3 Evaluation of high derivatives with collocated basis G and a degree reduction in every second step.

set $k^{(1)} = k + 1$
for $i = 1, \ldots, j$ **do**
  evaluate $\nabla\widetilde{u}^i_G$ in $\left(k^{(i)}\right)^d$ integration points $\boldsymbol{\xi}^{(i)}_G$
  set values for $\widetilde{\boldsymbol{U}}^{i+1}_G$ of the field $\widetilde{u}^{i+1}_G$ for degree $k^{(i)}$
  if i mod 2 = 0 **then**
    project $\widetilde{u}^{i+1}_G$ to degree $k-i$ by sum-factorized multiplication, $\mathbb{P}^{k-i}_{k-i+2}\widetilde{\boldsymbol{U}}^{i+1}_G$
    set $k^{(i+1)} = k^{(i)} - 2$
  else
    set $k^{(i+1)} = k^{(i)}$
  end if
end for

CPU	2×14 core Intel Xeon Broadwell E5-2690v4, 2.6 GHz
memory	8 channels DDR 4 (2400 MHz) $153^{\text{GB}}/_{\text{s}}$ theoretic bandwidth
compiler	`g++` version 6.2
compiler optimization	`-march=haswell -O3 -funroll-loops`

Table 3.1: System specifications for the numerical tests.

3.2 Performance Evaluation

In the following subsections, the performance of DG with ADER and DG with explicit Runge–Kutta time integration is analyzed in terms of operation counts, computation time, throughput, and scalability. If not specified otherwise, the computational setup as shown in Table 3.1 is used in the numerical examples.

3.2.1 Operation Counts

For the ADER DG method as given in equation (3.1), there are three main contributions to the computational costs, namely the application of the inverse mass matrix with cost $C_{\mathbb{Q}}$, the application of the global derivative operator with cost $C_{\mathbb{K}}$, and the evaluation of the Taylor–Cauchy–Kowalevski sum with cost C_{TCK}, resulting in an overall cost of

$$C_{\text{ADER-DG}} = 2 \cdot C_{\mathbb{Q}} + C_{\mathbb{K}} + C_{\text{TCK}}, \tag{3.3}$$

as can be seen from equation (3.2). For an s-stage Runge–Kutta scheme, the costs are

$$C_{\text{RK}} = s \cdot (C_{\mathbb{Q}} + C_{\mathbb{K}}). \tag{3.4}$$

In ADER, the high order approximations contribute to a sum over all terms in the truncated Taylor series and C_{TCK} involves an additional dependency on the polynomial degree k compared to $C_{\mathbb{Q}}$ and $C_{\mathbb{K}}$. In contrast, high order approximations with Runge–Kutta schemes use more stages s, where the number of stages s has to increase more quickly than the required order of accuracy (for orders larger than four) due to the Butcher barriers. ADER DG as well as Runge–Kutta methods both repeatedly call the application of the derivative operator and of the inverse mass matrices. The main difference is that ADER DG applies the derivative operator locally, i.e., element-wise, while Runge–Kutta integrators rely on the global derivative operator containing both element and face contributions.

The operation counts for $C_{\mathbb{Q}}, C_{\mathbb{K}}$, and C_{TCK} are derived from vector updates, matrix-vector or matrix-matrix multiplications, which in turn rely on the matrix-free implementation of integral evaluation for tensorial shape functions explained in Section 3.1. The cost for the evaluation of one tensor product kernel on a δ-dimensional domain (e.g. $\delta = d$ in an element or $\delta = d - 1$ on element faces) calculates to

$$C_{\text{tensorial}}(\delta) = \left(2 \cdot \frac{k+1}{2} \cdot 2 + (k+1) + 2 \cdot \left\lfloor \frac{(k-1) \cdot (k+1)}{2} \right\rfloor \right) \cdot (k+1)^{\delta - 1},$$

where the three summands in braces represent additions and subtractions (first term), multiplications (second term), and fused multiply-add operations (last term, counted as two arithmetic

Table 3.2: Number of calls of the tensor product kernel for acoustics in terms of cell kernels "C" and face kernels "F".

	mass $\mathbb{Q}^{-1}$	stiffness matrix $\mathbb{K}$	TCK
cell eval.	$2d$	$d^2 + 2d$	$2d$
cell deriv.	$-$	$2d$	$(k-1) \cdot d$
face eval.	$-$	$d \cdot 4(d-1)$	$-$
total	$2d \cdot \mathrm{C}$	$(d^2 + 4d) \cdot \mathrm{C} + 4(d^2 - d) \cdot \mathrm{F}$	$(2d + d(k-1)) \cdot \mathrm{C}$

instructions). Note that an even-odd decomposition of the local coefficients and matrices is used that cuts operation count into approximately one half compared to a usual 1D matrix-vector product [93, 95]. This cost is the main building block to derive the operation counts for $C_{\mathbb{Q}}, C_{\mathbb{K}}$, and C_{TCK} with the number of calls to the tensor product kernels as summarized in Table 3.2. The operation count for ADER with reconstruction is already given in equation (3.3). For the fully adjoint consistent scheme as introduced in Chapter 2.4.1 in equation (2.32), the operation count calculates according to

$$C_{\text{ADER adcon full}} = (k+1) \cdot (C_{\mathbb{Q}} + C_{\mathbb{K}}).$$

Comparison to the cost for the Runge–Kutta update as given in equation (3.4) shows the algorithmic similarity of Runge–Kutta and the fully adjoint consistent ADER scheme. For orders beyond four, the number of Runge–Kutta stages has to increase overproportionally, indicating an advantage for ADER adcon full for high orders of accuracy. Comparison to C_{ADER} shows once more how ADER trades global for local evaluations by using the Taylor–Cauchy–Kowalevski procedure. Note that the additional dependence on k for ADER is hidden in C_{TCK} (see Table 3.2) in contrast to Runge–Kutta and the fully adjoint consistent ADER scheme, where the dependence appears in the final summation of costs. For details on the derivation of operation counts, see [95] and [97].

In Figure 3.3, operation counts for the full schemes are compared, i.e., ADER versus a Runge–Kutta scheme with five stages for all polynomial degrees. To allow for generalization and allow comparability, the number of stages for the Runge–Kutta scheme is kept constant. ADER involves fewer arithmetic operations for all considered polynomial degrees $k \in \{1, \ldots, 12\}$ for both $d = 2, 3$. The figure also shows operation counts in case no basis change to a collocated basis G and no degree reduction for the higher order spatial derivatives are carried out, i.e., if the proposed algorithms of Sections 3.1.2 and 3.1.3 are not realized in the implementation of the Taylor–Cauchy–Kowalevski procedure. Without optimizations, it is apparent that ADER has a higher polynomial dependency on k than Runge–Kutta. The optimizations however compensate this dependency. The basis change is not explicitly applied to the Runge–Kutta discretization because the element-local mass matrix inversion is dominated by the memory bandwidth in contrast to the ADER Taylor–Cauchy–Kowalevski term with higher operational cost, see also Figure 3.2. Within the Runge–Kutta operator evaluation, however, the algorithm initially carries out an evaluation of the shape functions in the quadrature points, followed by operations on the quadrature points, and a closing integration while exploiting the tensor product quadrature formulas and the tensor product shape functions as described in [95].

Figure 3.4 visualizes the gains of the degree reduction approach. As mentioned above, the degree reduction introduces additional cost for the projection from one basis to the other but

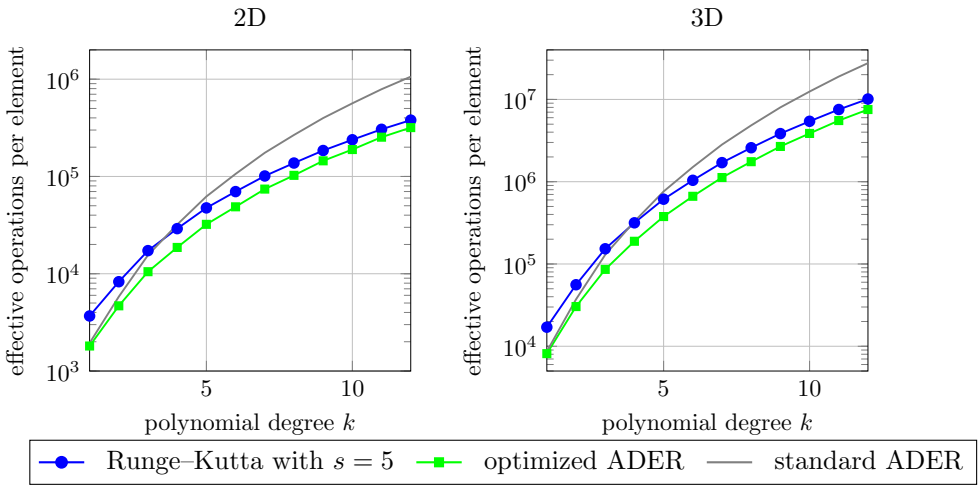


Figure 3.3: Operation counts for evaluation of one element with a five stage Runge–Kutta scheme and ADER DG in two and three dimensions.

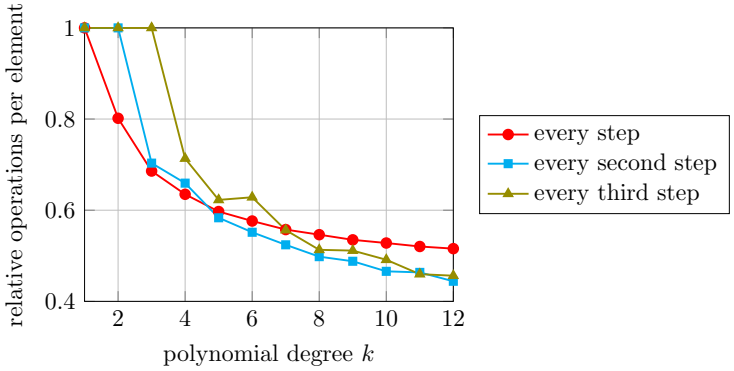


Figure 3.4: Operation counts for the Taylor–Cauchy–Kowalevski procedure with degree reduction every step, every second, and every third step relative to using the degree k for all summands.

also decreases the cost for all following evaluations of the tensor product kernels by reducing the polynomial degree. The cost decrease due to the lower polynomial degree must outrun the projection cost. The Figure shows the operation counts for the Taylor–Cauchy–Kowalevski procedure with degree reduction relative to an implementation without degree reduction and compares the reduction in every step, every second step, and every third step. The operational cost is approximately halved. For higher orders, the reduction in every step is not preferable because the projection introduces overhead. For moderate polynomial degrees, the approach to reduce the basis in every second step of the Taylor–Cauchy–Kowalevski procedure appears most beneficial.

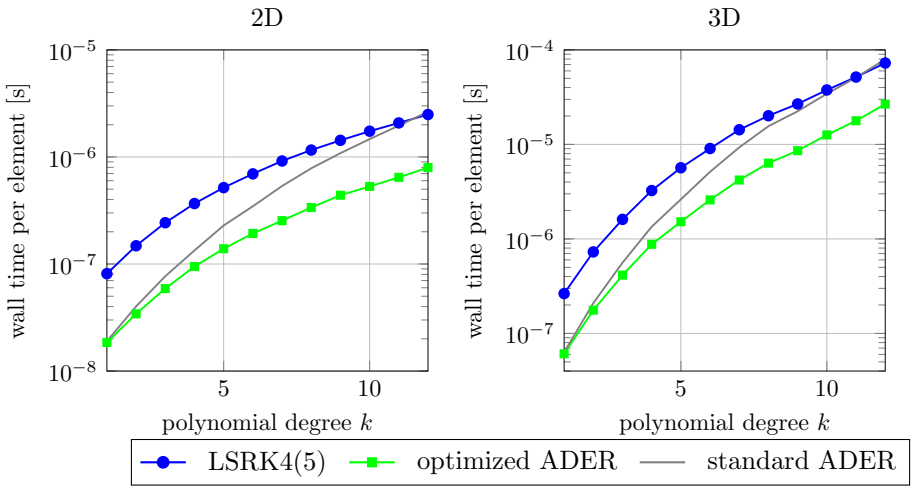


Figure 3.5: Run time per element for an explicit Runge–Kutta scheme with $s = 5$ and ADER DG in two and three dimensions, measured on 28 cores.

3.2.2 Computational Timings

A two- and a three-dimensional example are set up on a domain $\Omega = [0, 1]^d$, solving the acoustic wave equation with vibrational modes as analytic solution as in Section 2.5.1. Tests are run for polynomial degrees $k \in \{1, \dots, 12\}$ on a mesh with slightly deformed elements to prevent the built-in Cartesian mesh optimizations in the code of [95]. In 2D, 1280^2 and 640^2 elements are used for $k \in \{1, 2, 3\}$ and $k \in \{4, \dots, 12\}$, respectively. In 3D, 80^3 and 40^3 elements are used for the respective polynomial degrees.

Figure 3.5 shows measured run times per time step and per element for numerical experiments on 28 cores with LSRK4(5). Comparison between Figure 3.3 and Figure 3.5 reveals that the run time follows the operation counts. ADER is significantly more efficient in terms of wall time per element, though, which is due to the fact that the Taylor–Cauchy–Kowalevski procedure requires one global vector access but Runge–Kutta requires one global vector access for each application of the global derivative operator $\mathbb{K}$ in the stages. In other words, the operation counts that ignore the memory access are pessimistic concerning ADER. This statement also holds for the ADER implementation without basis change with run times between LSRK4(5) and the optimized ADER.

3.2.3 Breakdown into Algorithmic Components

The algorithmic components are examined separately in terms of the computational time per million degrees of freedom. In Figure 3.6, two versions of LSRK4(5) are compared: an optimized vector updater that only reads and writes two vectors per stage by merging the loop over vectors into a single loop and a standard vector updater reading five vectors and writing two vectors per stage, similar to the "merged operations" presented in [52]. Also, the results for the optimized ADER scheme implementing the basis change and degree reduction from Sections 3.1.2 and 3.1.3 and the ADER scheme without the degree reduction are shown in Figure 3.6. The results are obtained from simulations on a three-dimensional non-Cartesian grid. The measured

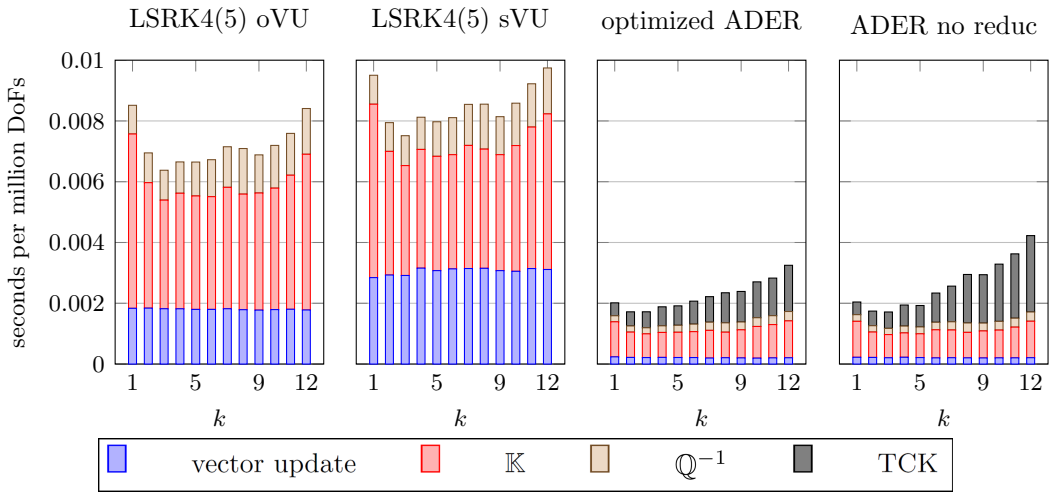


Figure 3.6: The algorithmic components for various polynomial degrees in 3D on 28 cores with the low-storage scheme LSRK4(5) with optimized vector update "oVU" routine and standard vector update "sVU", and ADER time integration with the two optimizations as presented in Sections 3.1.2 and 3.1.3. In the very right panel, results are shown for the ADER algorithm without the degree reduction optimization.

run time is the accumulated time spent in the respective functions, e.g., the contribution of the derivative operator $\mathbb{K}$ appears five times as high for LSRK4(5) compared to ADER, because it is applied in each of the five stages while ADER applies the derivative operator only once per time step.

As can be seen from Figure 3.6, the cost for vector updates in Runge–Kutta schemes cannot be neglected, neither in the standard implementation nor in the optimized variant, where they contribute with about a third or a quarter of the run time, respectively. Likewise, the results document the high level of performance reached in the computations of $\mathbb{K}$ and $\mathbb{Q}^{-1}$, a distinctive feature of the developed implementation. For ADER, the vector updates have a comparably small contribution to the entire cost because only one single update at the end of the method is required.

The application of the derivative operator $\mathbb{K}$ is most expensive for $k = 1$ with its rather disadvantageous ratio between degrees of freedom located on the element faces as compared to the interior. The ratio improves for higher orders and an almost constant throughput per degree of freedom is obtained, despite the theoretical $\mathcal{O}(k)$ increase in arithmetic complexity. The constant throughput is mainly explained by the memory transfer that scales as $\mathcal{O}(1)$ per unknown. The Taylor–Cauchy–Kowalevski procedure shows slightly increasing costs for higher degrees as an additional summand contributes in the Taylor expansion according to equation (3.1). Nonetheless, the increase for higher degrees is moderate due to the proposed degree reduction approach as presented in Section 3.1.3 and efficient algorithms, less than doubling the run time between degrees two and twelve. In the rightmost panel of Figure 3.6, results are shown for a Taylor–Cauchy–Kowalevski procedure that does not reduce the polynomial degree, where the increase in computing time is much more significant. Obviously, the latter reaches higher arithmetic throughput with more than 300 GFLOPs/s, which is a secondary quantity, though.

3.2.4 Roofline Performance Model

Figure 3.7 shows a roofline model according to [169] for LSRK4(5) and ADER in two and three dimensions on a non-Cartesian mesh. It plots the operated FLOPs per second over the arithmetic intensity, which is defined as operated FLOPs per accessed memory in byte. Typical values of the arithmetic intensity for vector updates or matrix algebra are $\approx \frac{1}{8}\frac{\text{FLOP}}{\text{byte}}$ to $\approx \frac{1}{4}\frac{\text{FLOP}}{\text{byte}}$. The roofline plots contain the hardware specific limits in terms of the memory bandwidth limit (diagonal lines) measured with the STREAM triad benchmark and the peak arithmetic performance (horizontal lines). All numbers are based on measured data from hardware performance counters, extracted from monitoring the programs with the `likwid` performance measurement tool, version 4.3, as presented in [163].

Generally, the polynomial degree increases for the points from left to right. The left panel of Figure 3.7 highlights that ADER comes with a higher arithmetic intensity, in particular for the higher degrees, as compared to LSRK4(5) that is clearly in the memory bandwidth bound regime. The computations are made for a non-Cartesian mesh where not only vector entries and some index data must be loaded from main memory but also the inverse of the Jacobian transformation in each quadrature point.

The right panel of Figure 3.7 shows the results for the individual components of the methods. The evaluation of the derivative operator $\mathbb{K}$ that needs to load geometry data is most strongly limited by the memory. Note that loops over cells and faces are interleaved in the implementation to re-use the vector data loaded into caches in cell integrals also for face integrals. More detailed measurements similar to the ones presented in [95] show that the code of face integrals finds more than 90% of the vector data for face integrals already in caches. Nonetheless, the partial indirect addressing with gather/scatter type instructions to rearrange the face data for vectorization over several faces and the remaining cache misses reduce throughput by around 25% as compared to idealized code that only performs the integration, see the experiments in [95] for details. The Taylor–Cauchy–Kowalevski procedure comes along with considerably higher arithmetic intensities, but the high level of arithmetic optimizations in the proposed algorithm still keeps the code in the memory-limited region, in clear contrast to e.g. [21]. Given these detailed results, the behavior of the entire method can be characterized as follows: the Runge–Kutta scheme applies $\mathbb{K}$ and $\mathbb{Q}^{-1}$ five times in each time step, while ADER only applies $\mathbb{K}$ and $\mathbb{Q}^{-1}$ once and the rest is taken care of by the completely element-local Taylor–Cauchy–Kowalevski procedure, enabling a much higher re-use of cached data.

3.2.5 Throughput as Function of Polynomial Degree

Figure 3.8 lists the computational throughput in terms of degrees of freedom processed per second and the actually realized GFLOPs per second rate, measured with the `likwid` tool [163], version 4.3. If not stated otherwise, the optimized implementation of ADER using the basis change and reduction as proposed in Sections 3.1.2 and 3.1.3 are considered.

The performance advantage of ADER compared to the low-storage Runge–Kutta scheme is a factor of around 4 for low polynomial degrees. The advantage decreases to a factor of 1.5 for $k = 12$ owing to the additional computations for the high order of accuracy of the ADER time integration, whereas the LSRK4(5) schemes uses a fixed temporal accuracy of four with five stages. In terms of GFLOPs rates, which lie between 60 and 264 GFLOPs per second,

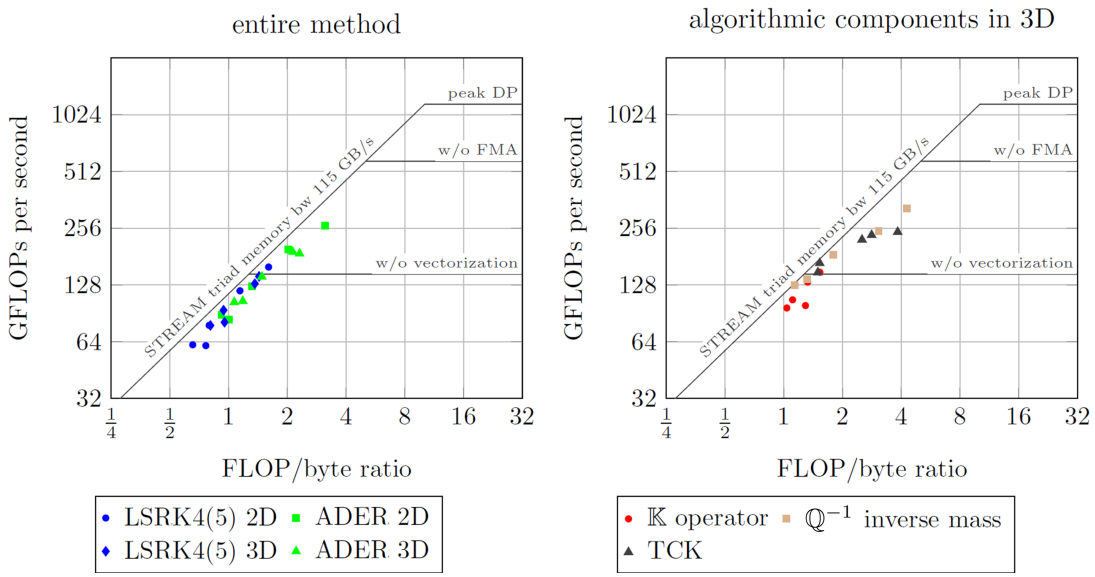


Figure 3.7: Roofline model for polynomial degrees $k = 1, 2, 4, 8, 12$ for the entire method (left panel) and the individual method components in 3D (right panel).

ADER operates 1.6 times more GFLOPs per second due to better caching. For comparison, a matrix based evaluation of a sparse matrix vector product was reported to reach 21 GFLOPs per second [96] on the same hardware.

3.2.6 Throughput as Function of Problem Size

The throughput as a function of the problem size between $4 \cdot 10^3$ and $2 \cdot 10^9$ degrees of freedom in 3D with shape functions of polynomial degree $k = 4$ is shown in Figure 3.9. For small discretizations of size $n^{\text{dof}} < 10^5$, limited parallelism prevents the full exploitation of the 28 cores. Then, the parallel efficiency improves and a first peak is reached at around $n^{\text{dof}} \approx 10^6$ where all data fits into the 70 MB of level 3 cache of the two processors and no access to main memory is needed. Performance drops again once the data structures exceed the caches and the vector and geometry data need to be streamed from main memory. For $n^{\text{dof}} > 10^6$, a slow increase of the throughput is noted, which is related to the decreased influence of the MPI communication for larger discretizations due to a maximal beneficial volume-to-surface ratio.

3.2.7 CPU Time Versus Accuracy

In this section, the time to solution is evaluated with respect to the accuracy on a two-dimensional example. With the standard setup of a vibrating membrane with seven modes, simulations are run on different refinement levels of the mesh between 5^2 and 1280^2 elements for polynomial degrees $k = 1, 4, 7$. For a fair comparison, the number of processors is chosen reasonably for the discretization sizes, e.g., the coarsest discretization is computed on one processor core only while the finest discretization is computed on 28 cores. The final numbers report the accumulated CPU time over all utilized processors. For the accuracy, the L_2 pressure error at the final time $T = 1.0$ is considered. The upper panel of Figure 3.10 plots the results for a Courant number of

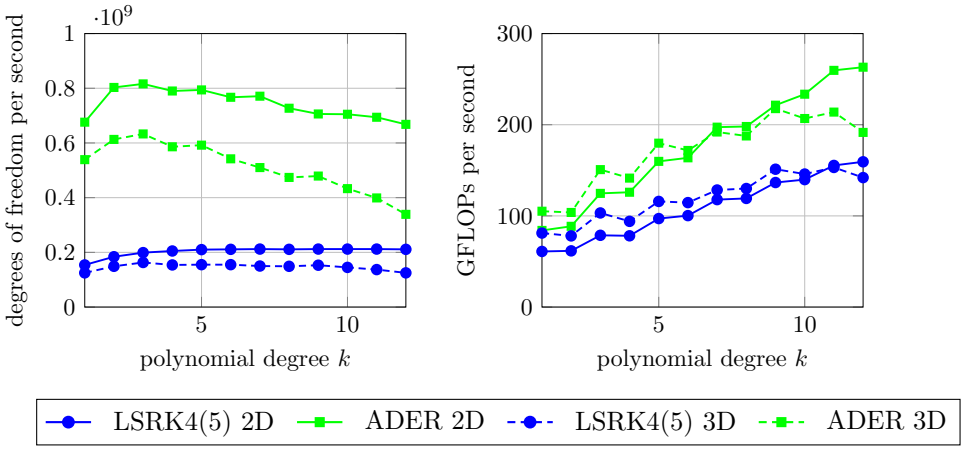


Figure 3.8: Throughput in terms of degrees of freedom per second and GFLOPs per second comparing LSRK4(5) and ADER DG in two and three dimensions.

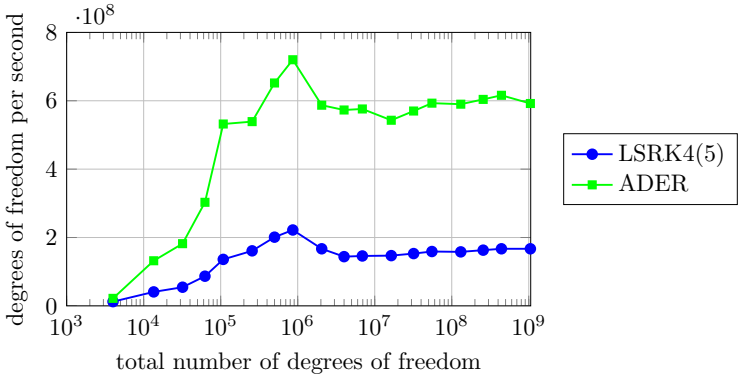


Figure 3.9: Throughput as function of the problem size, displayed as the number of degrees of freedom processed per second for one time step for $k = 4$.

$Cr = 0.1$. Generally, higher orders appear beneficial in case a strict accuracy criterion is applied. ADER yields the same solution quality as LSRK4(5) in less computational time.

Since ADER and Runge–Kutta schemes are subject to different CFL stability limits, the solution accuracy versus the CPU time at $Cr = 0.9 \cdot Cr_{\text{crit}}$ of the respective time integrator is analyzed in the bottom panel of Figure 3.10. The critical Courant numbers are taken from Section 2.5.2. For $k = 1$, ADER is faster for most error tolerances. For $k = 4$, all methods perform similarly. With polynomial degree $k = 7$ for the spatial discretization, ADER outperforms LSRK4(5) clearly because the LSRK4(5) scheme is of order four while the spatial discretization is of order eight, and the temporal error dominates over the spatial error. This is also true for LSRK5(9) at high spatial resolution.

The superconvergence property as explained in Section 2.5.1 is explored and simulations are run with reconstruction for ADER and the postprocessing step to obtain $k + 2$ convergent solutions. The results for $Cr = 0.1$ and $Cr = 0.9 \cdot Cr_{\text{crit}}$ for polynomial shape functions of degree $k = 4$ are presented as green lines in Figure 3.10. Again, for $Cr = 0.1$, ADER performs slightly better compared to LSRK4(5), while a significant performance advantage for ADER is noted for $Cr = 0.9 \cdot Cr_{\text{crit}}$, which is due to better temporal accuracy. Comparing the top and bottom panel for $k = 7$ shows that ADER is also faster than LSRK4(5) with smaller time steps. The discretization with LSRK5(9) is competitive to ADER for the superconvergent pressure result with $k = 4$. For the polynomial degree $k = 7$, a change in slope indicates the turnover from spatial error domination to temporal error domination. Where the temporal error dominates, the advantage for ADER is more distinct. This is due to the fact that ADER DG is automatically $k + 1$ convergent in space and time while Runge–Kutta is limited by the Butcher barriers: either overproportionally more stages are required to match the temporal order of accuracy with the spatial discretization, or a small time step is required.

3.2.8 Scalability

In order to assess the strong scalability of the proposed methods, a two- and a three-dimensional geometry consisting of 640^2 and 40^3 elements of polynomial degree $k = 4$ ($n^{\text{dof}} = 3.1 \cdot 10^7$ and $n^{\text{dof}} = 3.2 \cdot 10^7$, respectively) are used and the solution is computed on 1 to $2^8 = 256$ processors on a parallel cluster of 2×8 core Intel Xeon E5-2630 v3 (Haswell) processors at 2.4 GHz. Additionally, one computation with fewer elements (40^2, $n^{\text{dof}} = 1.2 \cdot 10^5$) in 2D is carried out such that the distribution yields only six elements per processor for the highest level of parallelism. The left panel of Figure 3.11 summarizes the results. In accordance with the previous sections, ADER is consistently faster than LSRK4(5) for the same time step size Δt. The scaling is almost ideal with a slight kink when going from 8 to 16 processors, where the code goes from being compute bound to being memory bound. For the small discretization the scaling deteriorates for high processor numbers due to communication overhead. In the right panel, results are shown for a strong scaling study on the SuperMUC Phase 2 Petascale system with nodes of 2×14 Intel Xeon E5-2697 v3 (Haswell) processors at 2.6 GHz on 1 to 512 nodes. For the 2D LSRK4(5) simulation, a kink can be seen when going from 224 to 448 cores indicating that simulations are memory bandwidth bound on lower core counts but get computation bound at higher core counts. At this processor count, all local data of the time integrators fits into the L3 cache of the individual processors bypassing the slow main memory. The scaling is close

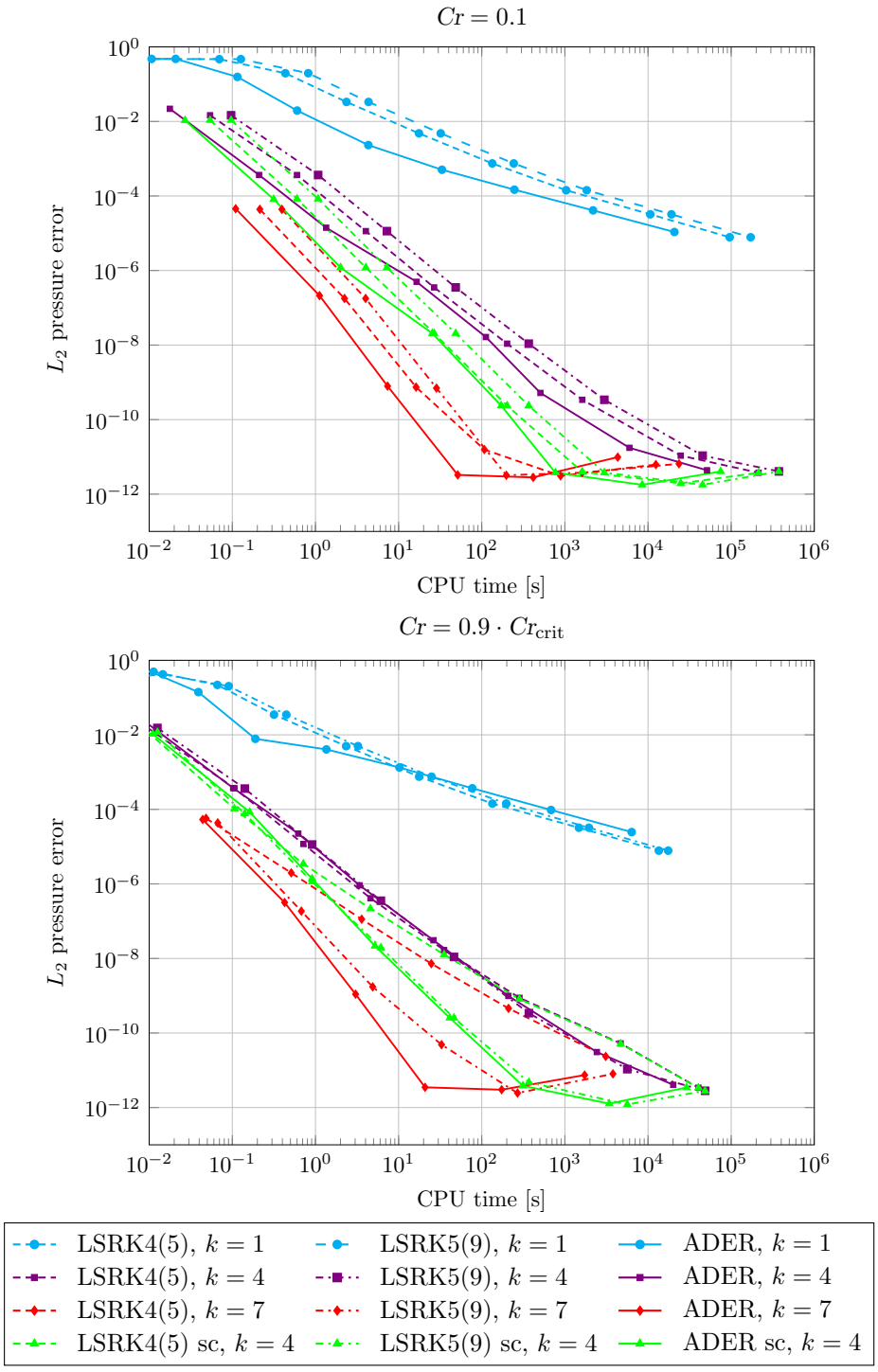


Figure 3.10: Accuracy over wall time accumulated over all MPI ranks comparing low-storage Runge–Kutta and ADER for the same time step size and at their respective critical time step size. Polynomial degrees $k = 1, 4, 7$ are studied. For $k = 4$ the super-convergent pressure solution is also considered for error calculation indicated by "sc".

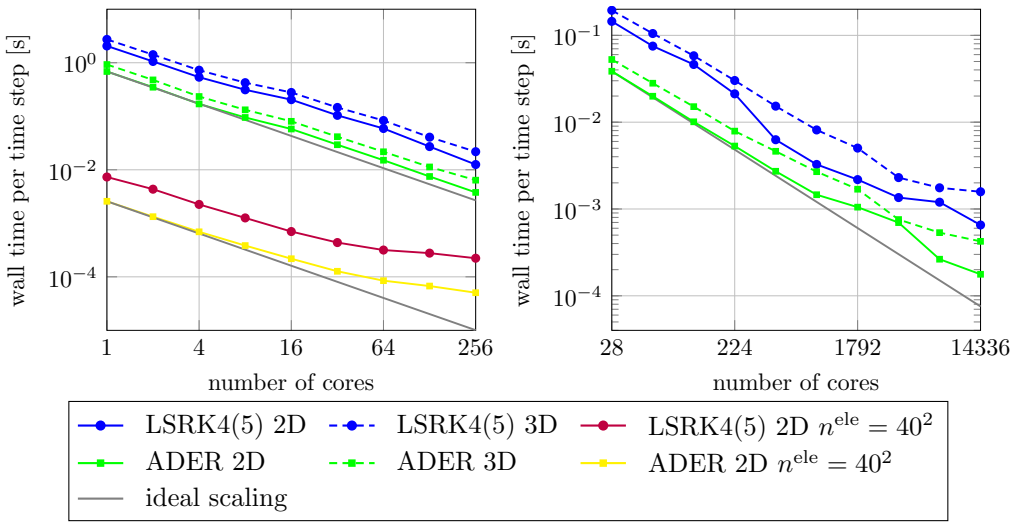


Figure 3.11: Strong scaling for LSRK4(5) and ADER DG in two and three dimensions in terms of wall time per time step over the number of cores. The left panel is obtained on a cluster using 1 to 256 cores of Intel Haswell E5-2630 v3, while the right panel is obtained with simulations on the SuperMUC Phase 2 system using 28 to 14336 cores of Intel Haswell E5-2697 v3.

to ideal considering that the highest level of parallelism corresponds to 28 and 4 elements per processor in 2D and 3D, respectively.

Figure 3.12 plots the strong scalability for the algorithmic components. It can be seen that the vector updates give the largest contribution to the reduced scalability between 8 and 16 processors, which is due to the fact that the utilized cluster possesses 16 cores but only 8 memory channels which can be saturated already with 8 processors. The kink is only due to shared memory effects because the communication is considered as part of the operator application. Since LSRK4(5) spends a larger fraction of time in vector updates, the reduced scalability in the overall method is more pronounced. The inverse mass matrix and the Taylor–Cauchy–Kowalevski procedure scale almost perfectly while the stiffness matrix shows a slight scaling decay due to a worse volume-to-surface effect of the data that must be exchanged with MPI.

In order to illustrate the effect of memory bandwidth on the algorithmic components, the calculations are repeated on an Intel Xeon Phi 7210-F (Knights Landing, KNL) system with 64 cores and 16 GB of high-bandwidth memory delivering up to 420 GB/s. For KNL, the code is vectorized with AVX-512, i.e., eight-wide SIMD lanes, as opposed to AVX2 with four-wide SIMD lanes on Broadwell. In Table 3.3, the computing times per time step are compared to a calculation on 28 Broadwell cores. The KNL system yields a speed up for the vector update of about 4 for LSRK4(5), corresponding to the four times higher memory bandwidth. Also, the application of the inverse mass matrix is substantially faster. The derivative operator $\mathbb{K}$, on the other hand, runs slightly slower on KNL because of the more irregular code patterns in face integrals that favor the more sophisticated CPU cores of Broadwell.

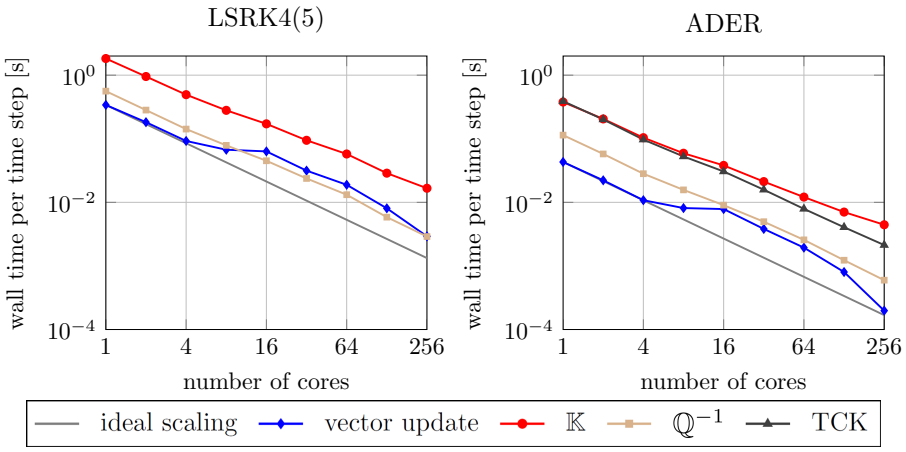


Figure 3.12: Strong scaling for LSRK4(5) and ADER DG in three dimensions for the components of the algorithm in terms of wall time per time step over the number of cores for $n^{\text{dof}} = 3.2 \cdot 10^7$ on up to 16 Haswell E5-2630 v3 nodes.

Table 3.3: Comparison of one node with 28 Broadwell cores and 64 KNL cores in terms of wall time per time step for a system with $3.2 \cdot 10^7$ degrees of freedom. "VU" abbreviates "vector update".

	VU	$\mathbb{K}$	$\mathbb{Q}^{-1}$	TCK	sum
LSRK4(5) Broadwell	$5.4 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	$2.6 \cdot 10^{-2}$	—	$1.9 \cdot 10^{-1}$
LSRK4(5) KNL	$1.3 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$2.0 \cdot 10^{-2}$	—	$1.5 \cdot 10^{-1}$
ADER Broadwell	$4.5 \cdot 10^{-3}$	$2.5 \cdot 10^{-2}$	$6.6 \cdot 10^{-3}$	$1.8 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$
ADER KNL	$1.5 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	$3.6 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$	$4.9 \cdot 10^{-2}$

3.3 Conclusion

A performance analysis for explicit Runge–Kutta and ADER DG implementations was presented. ADER outperforms optimized low-storage Runge–Kutta schemes over a range of test scenarios. The ingredients are fast integration techniques with sum factorization that combine optimal-complexity mathematical algorithms utilizing the tensor product structure of the shape functions with a highly competitive implementation that vectorizes over several elements and faces. The methods have been devised to be applicable also for complex meshes with curved quadrilateral or hexahedral elements. The experiments clearly show that it is most efficient to evaluate operators including face integrals with nodal basis functions that have nodes on the element boundaries, while the cell evaluations in the Taylor–Cauchy–Kowalevski procedure of ADER are best performed with Lagrange polynomials in the points of the quadrature formula. To combine these two, an on-the-fly change between the bases has been proposed in this work. This result is in contrast to the consensus belief in spectral elements that favor collocation of the polynomials nodes and quadrature points. While the theoretically derived operation counts already signify a slight benefit for ADER compared to Runge–Kutta, the actual timings show a distinct benefit, reducing the time to perform one time step by approximately a factor of four, because ADER better suits modern hardware architecture: while Runge–Kutta schemes are mostly limited by the memory bandwidth, ADER performs more operations on the data that is loaded from main memory and thus reaches a higher arithmetic intensity. A detailed analysis of Runge–Kutta versus ADER integration at the CFL stability limit has shown comparable performance where the Runge–Kutta time discretization order matches the spatial discretization order. For approximations with a high order of accuracy where the Butcher barriers set in, ADER exceeds the abilities of Runge–Kutta because its computational cost does not grow overproportionally. While the findings for ADER are limited to linear hyperbolic PDEs, the optimizations regarding the basis functions and reduced vector access for the Runge–Kutta time integrators regarding basis functions are also directly applicable to general nonlinear systems of hyperbolic PDEs.

The work highlights the importance to develop modern DG solvers according to the trends and limits in modern hardware architectures. For common solvers, the memory bandwidth limit is more relevant also when performing the relatively expensive computations of high order DG methods, i.e., it is met earlier than the arithmetic performance limit. Trading global for element-local operations counters this effect, rendering approaches like the Taylor–Cauchy–Kowalevski procedure favorable. Vector updates are inherently memory bandwidth limited and need to be optimized specifically. The experiments and performance models highlight that going significantly beyond the throughput recorded in this work demands either hardware with higher memory bandwidth, such as GPUs or the Xeon Phi, or new software paradigms that reduce the memory access over several stages, such as wavefront blocking that is already commonly used in the finite difference community.

4 Perfectly Matched Layers

A critical issue for the accurate simulation of wave propagation is the presence of artificial boundaries. If the real problem is in infinite space as shown in Figure 4.1(a), the computational domain must introduce an artificial boundary that should be perfectly permeable for outwards traveling waves, see Figure 4.1(b). A fundamental law representing this behavior is the Sommerfeld radiation condition

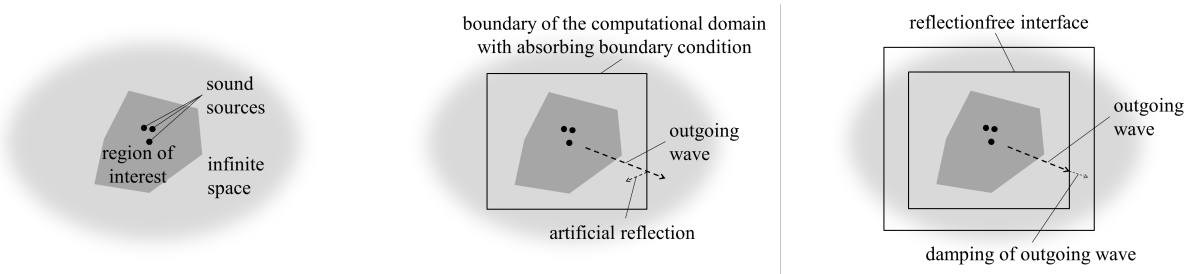
$$\lim_{|r|\to\infty} r^{(d-1)/2} \left(\frac{\partial p}{\partial r} - ikp \right) = 0,$$

with the radial coordinate r and the wave number k [154]. It states that waves radiated from the sources must scatter to infinity. Is is derived for the Helmholtz equation and also holds in the time domain.

An exemplary boundary condition approximately representing the Sommerfeld radiation condition is the first order absorbing boundary condition (ABC)

$$\boldsymbol{v} \cdot \boldsymbol{n} - \frac{1}{c\rho} p = 0, \tag{4.1}$$

which was already metnioned in Chapter 2 of this work and was derived in [53]. In one dimension, the absorbing condition (4.1) is exact and outward traveling waves do not cause reflections at the artificial boundary. In two or three dimensions, reflections occur for waves that do not hit the boundary orthogonally. For a wave with an angle of incidence of $45°$, reflections have an amplitude of 17% of the original wave [53]. To increase accuracy and decrease artifical reflections, high order ABCs were proposed in [53] and in [75, 76, 158]. A great variety of high order ABCs exists (see e.g. [63, 70] for reviews), however, they suffer either from non-locality or require the evaluation of high derivatives in space or time. Also, edge and corner treatment and the application to curved boundaries are not straight forward. The approach overcoming these drawbacks is the perfectly matched layer (PML). The PML was initially introduced in [14] in the context of electromagnetic waves and the finite difference time domain method. The computational domain is surrounded by an additional layer with the purpose to absorb waves from the actual physical domain without reflections at the interface, see Figure 4.1(c) and 4.2. The name “perfectly matched” states the property that the continuous formulation of the PMLs is exactly non-reflecting at the interface. In the interior of the PML, the waves are attenuated by a complex coordinate transformation enforcing exponential damping [30]. The original publication [14] formulates the PML using a split-field approach. Later the uniaxial PML formulation [138] emerged. However, both formulations can also be seen as a result of a complex coordinate stretching [30]. General PML formulations and a review on the developments of PMLs are given in [6]. A drawback of PMLs is that their accuracy depends on the discretization as well as on the selection of several parameters. Good parameter choices are obtained through experience,



(a) Sound sources in a region of interest in infinite space.

(b) Truncation of the infinite space by an artificial boundary and an absorbing boundary condition.

(c) Truncation of the infinite space by a perfectly matched layer.

Figure 4.1: The solution of the wave equation in an infinite space or approximations of the infinite space.

experimentation, or automated routines [112]. Note that the standard formulation for PMLs is either for straight boundaries on cuboidal domains (with overlap in the corners) or for cylindrical or spherical domains (see e.g. [38]). To the author's knowledge, there are no publications on general PML geometries. A comparison between PMLs and high order ABCs in the frequency domain is presented in [131] with the main conclusion that they perform equally efficient in the high accuracy regime. Later, even more efficient evaluations of high order ABCs with an optimized GPU implementation were proposed [113]. However, until this day, the question whether PMLs or ABCs should be used is not easily answered.

In this work, the first order ABC and PMLs are used because they are easily combined with the DG spatial discretization of the acoustic wave equation as presented in Chapter 2. Parts of this chapter are based on the Master's Thesis by M. Kufner [102]. In Section 4.1, a very general PML formulation is derived based on the first order formulation of the acoustic wave equation. Section 4.2 addresses the stability of the PML configuration and in Section 4.3, the absorption function is introduced. Spatial and temporal discretization are given in Section 4.4 and numerical examples are presented in Section 4.5. The chapter is concluded in Section 4.6.

4.1 Derivation

The formulation presented in this work is based on the common stretched coordinate approach as e.g. in [112]. The derivation is however more general and allows for higher flexibility. The formulation allows for PMLs on convex and concave polygons while keeping the number of auxiliary variables to a minimum. Also, it enables the combination of circular or spherical PMLs with straight PMLs as shown in Figure 4.2.

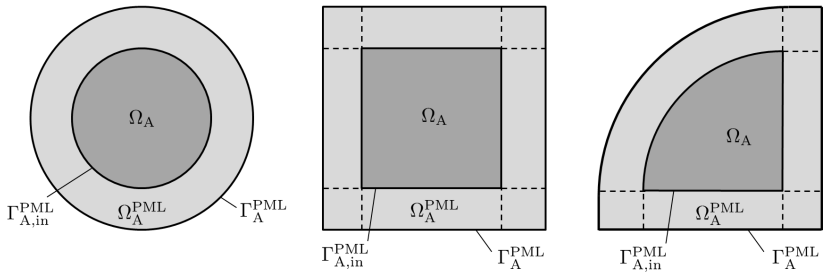


Figure 4.2: Exemplary configurations of an acoustical domain surrounded by PMLs.

Starting point for the new formulation is the acoustic wave equation written as first order system (2.4)–(2.5), which is repeated here for convenience

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\rho}\nabla p = \mathbf{0}, \tag{4.2}$$

$$\frac{\partial p}{\partial t} + c^2\rho\nabla\cdot\boldsymbol{v} = 0, \tag{4.3}$$

with the sound pressure p, the acoustic particle velocity $\boldsymbol{v}$, the mass density ρ, and the speed of sound c. This system is solved on the domain Ω_A in the time interval $[0, T]$. The computational domain is now extended by the PMLs $\Omega_\mathrm{A}^{\mathrm{PML}}$ with the new boundary $\Gamma_\mathrm{A}^{\mathrm{PML}}$ as sketched in Figure 4.2. On the boundary $\Gamma_\mathrm{A}^{\mathrm{PML}}$ either a Dirichlet, a Neumann, or an ABC is enforced:

Dirichlet	$p = p_\mathrm{D},$
Neumann	$\boldsymbol{v}\cdot\boldsymbol{n} = 0,$
first order absorbing	$\boldsymbol{v}\cdot\boldsymbol{n} - \dfrac{1}{c\rho}p = 0.$

In Section 4.5.4, it will be identified, which of these three boundary conditions gives the best results. Of course it is also possible to surround the domain Ω_A only partially by PMLs. The extension is straightforward. The equations presented in the following are for acoustical domains completely surrounded by PMLs to keep notation clear.

A general solution of the undamped wave equation as in (4.2)–(4.3) is the superposition of plane pressure waves and plane velocity waves

$$p = \hat{p}e^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t)},$$
$$\boldsymbol{v} = \hat{\boldsymbol{v}}e^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t)},$$

with the amplitudes $\hat{p}, \hat{\boldsymbol{v}}$ for pressure and velocity, the wave vector $\boldsymbol{k}$, and the angular frequency ω. In the artificial PML domain $\Omega_\mathrm{A}^{\mathrm{PML}}$, *damped* plane waves shall be solutions

$$\tilde{p} = \hat{p}e^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t-\frac{1}{i\omega}\boldsymbol{k}\cdot\boldsymbol{\gamma}))}, \tag{4.4}$$

$$\tilde{\boldsymbol{v}} = \hat{\boldsymbol{v}}e^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t-\frac{1}{i\omega}\boldsymbol{k}\cdot\boldsymbol{\gamma}))}, \tag{4.5}$$

with the damping function $\boldsymbol{\gamma} = \boldsymbol{\gamma}(\boldsymbol{x})$ defining the direction and the magnitude of the damping. The damping is represented by a real part in the exponent. The damping function must vanish on the interface between standard domain and PML domain $\boldsymbol{\gamma} = \mathbf{0}$ on $\Omega_\mathrm{A} \cap \Omega_\mathrm{A}^\mathrm{PML}$ to obtain the perfectly matched property and have a reflection free interface. For convenience, the abbreviation

$$\boldsymbol{A} = \left(\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}}\right)^\mathrm{T} \tag{4.6}$$

is introduced for the transposed partial derivative matrix of the damping function with respect to the spatial coordinates.

A wave equation is sought for which the damped plane waves are a solution or more specifically, a differential operator $\tilde{\nabla}$ is sought such that the damped waves (4.4) and (4.5) are a solution to

$$\frac{\partial \tilde{\boldsymbol{v}}}{\partial t} + \frac{1}{\rho}\tilde{\nabla}\tilde{p} = \mathbf{0}, \tag{4.7}$$

$$\frac{\partial \tilde{p}}{\partial t} + c^2\rho\tilde{\nabla}\cdot\tilde{\boldsymbol{v}} = 0. \tag{4.8}$$

The application of the modified differential operator $\tilde{\nabla}$ to the damped solution shall have the same effect as the application of the standard operator to the undamped solution:

$$\tilde{\nabla}\tilde{p} \stackrel{!}{=} i\boldsymbol{k}\tilde{p} \qquad \text{and} \qquad \tilde{\nabla}\cdot\tilde{\boldsymbol{v}} \stackrel{!}{=} i\boldsymbol{k}\cdot\tilde{\boldsymbol{v}}.$$

The modified differential operator must be

$$\tilde{\nabla} = \left(\boldsymbol{I} - \frac{1}{i\omega}\boldsymbol{A}\right)^{-1}\nabla, \tag{4.9}$$

which can be seen from the gradient of $\tilde{p}$ derived from equation (4.4)

$$\begin{aligned}\nabla\tilde{p} &= \nabla\left(\hat{p}e^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t-\frac{1}{i\omega}\boldsymbol{k}\cdot\boldsymbol{\gamma}))}\right)\\ &= \left(i\boldsymbol{k} - \frac{1}{i\omega}\boldsymbol{k}\cdot\boldsymbol{A}\right)\tilde{p}\\ &= \left(\boldsymbol{I} - \frac{1}{i\omega}\boldsymbol{A}\right)i\boldsymbol{k}\tilde{p},\end{aligned}$$

and the following expansion

$$\tilde{\nabla}\tilde{p} = i\boldsymbol{k}\tilde{p} = \underbrace{\left(\boldsymbol{I} - \frac{1}{i\omega}\boldsymbol{A}\right)^{-1}\left(\boldsymbol{I} - \frac{1}{i\omega}\boldsymbol{A}\right)}_{\boldsymbol{I}} i\boldsymbol{k}\tilde{p} = \left(\boldsymbol{I} - \frac{1}{i\omega}\boldsymbol{A}\right)^{-1}\nabla\tilde{p},$$

where the gradient expression from above is inserted for $\nabla\tilde{p}$. With the differential operator as in (4.9), equations (4.7) and (4.8) yield the same dispersion and amplitude relation as the original wave equation.

Writing equations (4.7) and (4.8) in frequency domain allows to introduce the operator

$$
\begin{aligned}
&- i\omega\tilde{p} + \rho c^2 \nabla \cdot \tilde{\boldsymbol{v}} + \rho c^2 \left((i\omega \boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{A} \right) : (\tilde{\boldsymbol{v}} \otimes \nabla) = 0, \\
&- i\omega\tilde{\boldsymbol{v}} + \frac{1}{\rho} (i\omega \boldsymbol{I} - \boldsymbol{A})^{-1} \nabla \tilde{p} = \boldsymbol{0},
\end{aligned}
$$

with the symbol $\otimes$ representing a dyadic product and the symbol : representing a tensor contraction. In order to enable the transformation back to the time domain, the auxiliary variables z_j are introduced in the frequency domain

$$
\sum_{j=1}^{s} z_j = \left((i\omega \boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{A} \right) : (\tilde{\boldsymbol{v}} \otimes \nabla) .
$$

If the matrix $\boldsymbol{A}$ is diagonizable, this equation can be reformulated. Therefore, the eigenvalues λ_i of $\boldsymbol{A}$ are determined as the roots of the characteristic polynomial. The matrix $\boldsymbol{A}$ is of size $d \times d$ and hence, d eigenvalues can be found, however some eigenvalues can have a multiplicity greater than one and in general s denotes the number of distinct eigenvalues of $\boldsymbol{A}$. The distinct eigenvalues are summarized in μ_j with $j = 1, ..., s$ while λ_i with $i = 1, ..., d$ lists all eigenvalues. The eigenvectors $\boldsymbol{E}_i$ to the eigenvalues λ_i are summarized in $\boldsymbol{G} = [E_1 \dots E_d]$ and $\boldsymbol{F}_j$ denotes the row-vectors of $\boldsymbol{G}^{-1}$. With the introduced notation, the equation defining the auxiliary variables is reformulated

$$
\sum_{j=1}^{s} z_j = \sum_{j=1}^{s} \frac{\mu_j}{i\omega - \mu_j} (\boldsymbol{E}_j \otimes \boldsymbol{F}_j) : (\tilde{\boldsymbol{v}} \otimes \nabla) , \tag{4.10}
$$

with the notation $\boldsymbol{E}_j \otimes \boldsymbol{F}_j = \sum_{k=1}^{r} \boldsymbol{E}_k \otimes \boldsymbol{F}_k$ for r-fold eigenvalues. Therefore, eigenvalues with algebraic multiplicity greater than one imply a reduction of the number of required auxiliary variables. Stipulating the equality for each summand in equation (4.10), the following s equations must be fulfilled in the frequency domain

$$
-i\omega z_j + \mu_j z_j = -\mu_j (\boldsymbol{E}_j \otimes \boldsymbol{F}_j) : (\tilde{\boldsymbol{v}} \otimes \nabla) \qquad \text{for } j = 1, ..., s. \tag{4.11}
$$

With this definition of the auxiliary variables z_j, the transformation of the modified wave equation from the frequency domain back to the time domain is possible. After several algebraic manipulations, the following time domain formulation is found

$$
\frac{\partial \tilde{\boldsymbol{v}}}{\partial t} + \frac{1}{\rho} \nabla \tilde{p} = -\boldsymbol{A} \tilde{\boldsymbol{v}}, \tag{4.12}
$$

$$
\frac{\partial \tilde{p}}{\partial t} + c^2 \rho \nabla \cdot \tilde{\boldsymbol{v}} = -\rho c^2 \sum_{j=1}^{s} z_j, \tag{4.13}
$$

$$
\frac{\partial z_j}{\partial t} + \mu_j z_j = \mu_j (\boldsymbol{E}_j \otimes \boldsymbol{F}_j) : (\tilde{\boldsymbol{v}} \otimes \nabla) \quad \text{for } j = 1, ..., s. \tag{4.14}
$$

The system (4.12)–(4.14) is complemented by initial conditions for $\tilde{p}$, $\tilde{\boldsymbol{v}}$, and z_j. The auxiliary variables z_j are always initialized to zero. By construction, the superposition of damped plane waves for pressure and velocity are solutions to this equation. The interface between Ω_{A}, where the standard wave equation is fulfilled, and $\Omega_{\mathrm{A}}^{\mathrm{PML}}$, where the derived modified wave equation is fulfilled, is reflection free in the continuous context because the differential operator $\tilde{\nabla}$ yields the same dispersion and amplitude relations as the original wave equation.

4.2 Stability

The modulus of the pressure field $\tilde{p}$ is studied to analyze the stability properties of the derived equations

$$|\tilde{p}| = \left| \hat{p} e^{i(\boldsymbol{k}\cdot\boldsymbol{x} - \omega t - \frac{1}{i\omega}\boldsymbol{k}\cdot\boldsymbol{\gamma}))} \right| = |\hat{p}| \left| e^{-\frac{1}{\omega}\boldsymbol{k}\cdot\boldsymbol{\gamma}} \right|.$$

For $\boldsymbol{k}\cdot\boldsymbol{\gamma} > 0$, the plane wave $\tilde{p}$ is decaying. The product $\boldsymbol{k}\cdot\boldsymbol{\gamma}$ is abbreviated by

$$g = \boldsymbol{k}\cdot\boldsymbol{\gamma}.$$

As stated above, the damping function $\boldsymbol{\gamma}$ is zero on the interface between the physical acoustical domain Ω_A and the PML domain $\Omega_\mathrm{A}^\mathrm{PML}$ per premise and hence is g. To ensure a monotonically decaying wave, the following relation must hold for the directional derivative of g in the propagation direction given by $\boldsymbol{k}$

$$\nabla g \cdot \frac{\boldsymbol{k}}{|\boldsymbol{k}|} \geq 0.$$

Expanding the spatial derivatives gives

$$\nabla g \cdot \frac{\boldsymbol{k}}{|\boldsymbol{k}|} = (\boldsymbol{k}\cdot(\nabla\boldsymbol{\gamma}) + (\nabla\boldsymbol{k})\cdot\boldsymbol{\gamma}) \cdot \frac{\boldsymbol{k}}{|\boldsymbol{k}|} \geq 0.$$

For constant material properties in terms of speed of sound c and mass density ρ, the spatial derivative of the wave vector vanishes and the result is

$$\boldsymbol{k}\cdot(\nabla\boldsymbol{\gamma})\cdot\boldsymbol{k} = \boldsymbol{k}^\mathrm{T}\left(\frac{\partial\boldsymbol{\gamma}}{\partial\boldsymbol{x}}\right)^\mathrm{T}\boldsymbol{k} = \boldsymbol{k}^\mathrm{T}\boldsymbol{A}\boldsymbol{k} \geq 0,$$

where the conversions are only changes in notation and insertion of the abbreviation $\boldsymbol{A}$ as introduced in equation (4.6). The derived condition holds true if $\boldsymbol{A}$ is positive semidefinite, i.e., all eigenvalues of $\boldsymbol{A}$ must be greater or equal to zero: $\mu_j \geq 0$. This conditions enables checks for stability on the potential damping functions and layer configurations by calculation of the eigenvalues of $\boldsymbol{A}$. The matrix $\boldsymbol{A}$ is in general not symmetric and for given geometries and profile functions, the eigenvalues of $\boldsymbol{A}$ must be determined in order to be able to judge on the stability of the configuration.

4.3 The Absorption Function

Last, the damping function $\boldsymbol{\gamma}$ and thus the matrix $\boldsymbol{A}$ need specification. They depend on the user input of a function $\boldsymbol{\sigma}$ which is often denoted as absorption function or profile function. Several possibilities to define $\boldsymbol{\sigma}$ are known and some common definitions are introduced in the following.

The damping function $\boldsymbol{\gamma}$ calculates as the integral over the absorption function $\boldsymbol{\sigma}$ starting at the interface between the physical and the PML domains. The definition of the absorption function is not a simple task because it determines the magnitude of the damping. A straightforward

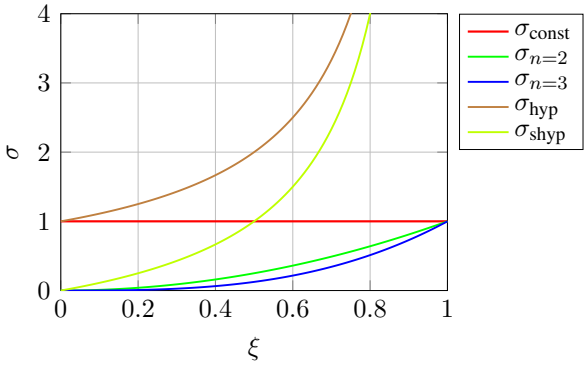


Figure 4.3: Different absorption functions all with $\alpha = 1$ and $\delta = 1$.

approach would be to choose $\boldsymbol{\sigma}$ as a constant function of high absolute value in order to damp the outgoing waves as quickly as possible

$$\sigma_{\text{const}} = \alpha,$$

with α as scaling parameter. The problem, however, is that a PML is only perfectly matched in the continuous case. As soon as the wave equation and the modified wave equation are discretized, reflections can occur at the interface. The magnitude of these reflections depends amongst other things on the absolute value of $\boldsymbol{\sigma}$. Therefore, polynomial functions have been proposed (see e.g. [62])

$$\sigma_n(\xi) = \alpha \left(\frac{\xi}{\delta}\right)^n,$$

where n is the degree of the polynomial, α is a scaling parameter, ξ is defined in $\Omega_{\text{A}}^{\text{PML}}$ and describes the normal distance to the interface between physical domain and PML. The value δ is the PML thickness and $\xi \in [0, \delta]$. The polynomial degree is commonly chosen as $n = 2$ or $n = 3$. The optimal choice for α is problem dependent because α balances the damping of the outgoing waves and the reflections from the PML. There is no general rule how to choose α. Additional possibilities to define the absorption function are the hyperbolic and the shifted hyperbolic absorption functions from [15]

$$\sigma_{\text{hyp}}(\xi) = \frac{\alpha}{\delta - \xi},$$
$$\sigma_{\text{shyp}}(\xi) = \frac{\alpha}{\delta - \xi} - \frac{\alpha}{\delta}.$$

They tend to infinity at the outer boundary of the PML and hence theoretically absorb waves entirely. In the discrete context this property is again only approximated. The parameter α is commonly chosen as $\alpha \approx c$ for the hyperbolic function and $\alpha \approx 1$ for the shifted hyperbolic function. Figure 4.3 summarizes the presented possibilities to define profile functions.

4.3.1 Straight-Lined Boundaries

Different PML geometries require different definitions of the vector valued function $\boldsymbol{\sigma}$. First, a straight-lined boundary is considered as shown in Figure 4.4. At the interface $\Gamma_{\text{A,in}}^{\text{PML}}$, the co-

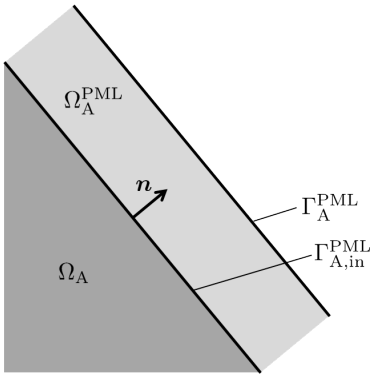


Figure 4.4: Acoustical domain with straight-lined PML.

ordinate ξ is zero, since it describes the distance to the interface in normal direction $\boldsymbol{n}$. In real coordinates, points on the interface are denoted $\boldsymbol{x}_0$. The normal vector $\boldsymbol{n}$ is of length $|\boldsymbol{n}| = 1$. The vector valued absorption function is given by

$$\boldsymbol{\sigma}(\xi) = \sigma(\xi) \cdot \boldsymbol{n}.$$

The absorption function has different contributions for the coordinate directions, depending on the orientation of the boundary. If the boundary is orthogonal to the x_1-direction, only the first entry of $\boldsymbol{\sigma}$ is non-zero and only waves propagating in x_1-direction are damped. For the above definition of $\boldsymbol{\sigma}$, the damping function $\boldsymbol{\gamma}$ calculates as

$$\boldsymbol{\gamma}(\boldsymbol{x}) = \int_0^{\boldsymbol{n}\cdot(\boldsymbol{x}-\boldsymbol{x}_0)} \boldsymbol{\sigma}(\xi)\mathrm{d}\xi,$$

with $\boldsymbol{x}_0$ as closest point on the interface to $\boldsymbol{x}$, thereby integrating over the normal distance to the interface. Consequently, the derivative of $\boldsymbol{\gamma}(\boldsymbol{x})$ is

$$\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{n} \otimes \boldsymbol{\sigma}.$$

It has only one non-zero eigenvalue $\mu = \boldsymbol{n}\cdot\boldsymbol{\sigma}$ and hence only one auxiliary variable z is required. The wave equation in a general PML as in (4.12)–(4.14) adapted to the straight-lined PML reads

$$\begin{aligned}
&\frac{\partial \tilde{\boldsymbol{v}}}{\partial t} + \frac{1}{\rho}\nabla\tilde{p} = -\left(\boldsymbol{n} \otimes \boldsymbol{\sigma}\right)\tilde{\boldsymbol{v}},\\
&\frac{\partial \tilde{p}}{\partial t} + c^2\rho\nabla\cdot\tilde{\boldsymbol{v}} = -\rho c^2 z,\\
&\frac{\partial z}{\partial t} + \boldsymbol{n}\cdot\boldsymbol{\sigma} z = -\left(\boldsymbol{n} \otimes \boldsymbol{\sigma}\right) : \left(\tilde{\boldsymbol{v}} \otimes \nabla\right).
\end{aligned}$$

4.3.2 Circular and Spherical Boundaries

For domains of circular or spherical shapes, the point $\boldsymbol{x}_0$ denotes the center of the domain. The vector $\boldsymbol{n}_{\mathrm{r}}$ denotes the outward pointing normal vector of unit length as a function of $\boldsymbol{x}$ with $\boldsymbol{x} \in \Gamma_{\mathrm{A,in}}^{\mathrm{PML}}$ as shown in Figure 4.5. The absorption function is chosen as

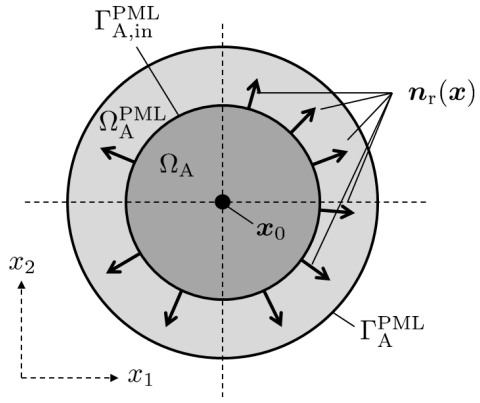


Figure 4.5: Circular acoustical domain with center point $\boldsymbol{x}_0$.

$$\boldsymbol{\sigma}(\xi) = \sigma(\xi) \cdot \boldsymbol{n}_{\mathrm{r}}(\boldsymbol{x}),$$

where ξ is again the shortest distance to $\Gamma^{\mathrm{PML}}_{\mathrm{A,in}}$ in the domain $\Omega^{\mathrm{PML}}_{\mathrm{A}}$. The damping function subsequently is

$$\boldsymbol{\gamma}(\boldsymbol{x}) = \int_0^{|\boldsymbol{x}-\boldsymbol{x}_0|} \sigma(\xi) \cdot \boldsymbol{n}_{\mathrm{r}}(\boldsymbol{x}) \mathrm{d}\xi.$$

During the calculation of $\boldsymbol{A}$, the spatial dependency of the normal vector must be considered, yielding

$$\begin{aligned}
\boldsymbol{A} &= \eta(\boldsymbol{I} - \boldsymbol{n}_{\mathrm{r}} \otimes \boldsymbol{n}_{\mathrm{r}}) + \sigma \boldsymbol{n}_{\mathrm{r}} \otimes \boldsymbol{n}_{\mathrm{r}}, \\
\eta &= \frac{1}{|\boldsymbol{x} - \boldsymbol{x}_0|} \int_0^{|\boldsymbol{x}-\boldsymbol{x}_0|} \sigma(\xi) \mathrm{d}\xi.
\end{aligned}$$

In two dimensions, η is a 1-fold eigenvalue of $\boldsymbol{A}$, in three dimensions, it is 2-fold belonging to eigenvectors pointing in circumferential direction, while σ is always a 1-fold eigenvalue belonging to an eigenvector pointing in radial direction. In case an analytic indefinite integral is available for the function $\sigma(\xi)$, the eigenvalue η can be calculated analytically.

The wave equation as in (4.12)–(4.14) adapted to the circular and spherical PML reads

$$\begin{aligned}
&\frac{\partial \tilde{\boldsymbol{v}}}{\partial t} + \frac{1}{\rho} \nabla \tilde{p} = -(\eta \boldsymbol{I} + (\sigma - \eta) \boldsymbol{n}_{\mathrm{r}} \otimes \boldsymbol{n}_{\mathrm{r}}) \tilde{\boldsymbol{v}}, \\
&\frac{\partial \tilde{p}}{\partial t} + c^2 \rho \nabla \cdot \tilde{\boldsymbol{v}} = -\rho c^2 (z_{\mathrm{r}} + z_\eta), \\
&\frac{\partial z_{\mathrm{r}}}{\partial t} + \sigma z_{\mathrm{r}} = -\sigma (\boldsymbol{n}_{\mathrm{r}} \otimes \boldsymbol{n}_{\mathrm{r}}) : (\tilde{\boldsymbol{v}} \otimes \nabla), \\
&\frac{\partial z_\eta}{\partial t} + \eta z_\eta = -\eta (\boldsymbol{I} - \boldsymbol{n}_{\mathrm{r}} \otimes \boldsymbol{n}_{\mathrm{r}}) : (\tilde{\boldsymbol{v}} \otimes \nabla).
\end{aligned}$$

Note that the number of auxiliary variables is $s = 2$ for the two-dimensional as well as the three-dimensional case.

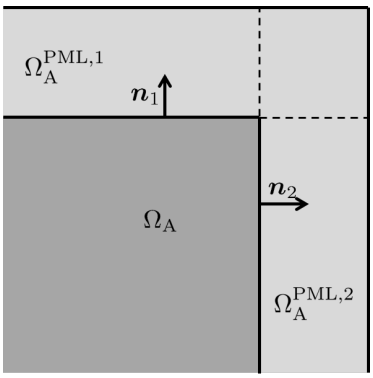


Figure 4.6: Acoustical domain Ω_{A} with two overlapping PMLs.

4.3.3 Overlapping Perfectly Matched Layers

In the corners of the computational domain, PMLs can overlap, see Figure 4.6. Two PMLs with orthogonal normal vectors are shown. Since the PML wave equation (4.12)–(4.14) is linear, the principle of superposition applies. In the single PMLs, the equations as derived in Section 4.3.1 apply. In the overlapping region $\Omega_{\mathrm{A}}^{\mathrm{PML},1} \cap \Omega_{\mathrm{A}}^{\mathrm{PML},2}$, the damping function results from a simple addition

$$\boldsymbol{\gamma}_{\Omega_{\mathrm{A}}^{\mathrm{PML},1} \cap \Omega_{\mathrm{A}}^{\mathrm{PML},2}} := \boldsymbol{\gamma}_{12} = \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2 = \int_0^{\boldsymbol{n}_1 \cdot (\boldsymbol{x} - \boldsymbol{x}_0)} \boldsymbol{\sigma}_1(\xi) \mathrm{d}\xi + \int_0^{\boldsymbol{n}_2 \cdot (\boldsymbol{x} - \boldsymbol{x}_0)} \boldsymbol{\sigma}_2(\xi) \mathrm{d}\xi.$$

For absorption functions which are parallel to the respective normal vector, the matrix $\boldsymbol{A}_{12}$ is symmetric and has real eigenvalues. The matrix $\boldsymbol{A}_{12}$ then calculates as

$$\boldsymbol{A}_{12} = \boldsymbol{A}_1 + \boldsymbol{A}_2 = \sigma_1 \boldsymbol{n}_1 \otimes \boldsymbol{n}_1 + \sigma_2 \boldsymbol{n}_2 \otimes \boldsymbol{n}_2.$$

In case the normal vectors are orthogonal the PML wave equation in the overlapping regions is given by

$$\begin{aligned}
&\frac{\partial \tilde{\boldsymbol{v}}}{\partial t} + \frac{1}{\rho} \nabla \tilde{p} = -\left(\sum_{j=1}^{r} \sigma_j \boldsymbol{n}_j \otimes \boldsymbol{n}_j \right) \tilde{\boldsymbol{v}}, \\
&\frac{\partial \tilde{p}}{\partial t} + c^2 \rho \nabla \cdot \tilde{\boldsymbol{v}} = -\rho c^2 \sum_{j=1}^{r} z_j, \\
&\frac{\partial z_j}{\partial t} + \sigma_j z_j = -\sigma_j \left(\boldsymbol{n}_j \otimes \boldsymbol{n}_j \right) : \left(\tilde{\boldsymbol{v}} \otimes \nabla \right) \qquad \text{for } j = 1, ..., r.
\end{aligned}$$

Therein, r is the number of overlapping regions. As displayed in Figure 4.4, two PMLs overlap and $r = 2$. In three dimensions, at the corner of a cube, three PMLs overlap and $r = 3$. If the normal vectors are not orthogonal, the more general equations (4.12)–(4.14) must be used. Stability must be checked by testing the sign of the non-zero eigenvalues of $\boldsymbol{A}$.

4.3.4 General Shapes

For PMLs which are neither straight-lined, nor spherical, cylindrical, nor an overlap of those, the function $\boldsymbol{\sigma}$ must be defined according to the requirement that $\boldsymbol{\gamma}$ vanishes on the interface

$\Gamma_{\mathrm{A,in}}^{\mathrm{PML}}$ and such that the eigenvalues of $\boldsymbol{A}$ are greater or equal to zero $\mu_j \geq 0$ and the general equations (4.12)–(4.14) must be used.

4.4 Spatial and Temporal Discretization

The numerical solution of the PML equations (4.12)–(4.14) is based on HDG spatial discretization and is achieved following the same approach as in Chapter 2.2 of this work. The problem can either be understood as to solve problem (4.12)–(4.14) in $\Omega_{\mathrm{A}}^{\mathrm{PML}}$ and problem (2.4)–(2.5) in Ω_{A} or to solve only problem (4.12)–(4.14) in $\Omega_{\mathrm{A}}^{\mathrm{PML}} \cup \Omega_{\mathrm{A}}$ with $\boldsymbol{\gamma} = \mathbf{0}$ in Ω_{A}. Both approaches result in the same computational procedure, because the coupling between elements is carried out by the trace variable playing the same role in both problem statements. For ease of notation, the first approach is presented in the following without explicitly repeating the coupling criterion $\lambda = \tilde{\lambda}$ on $\Omega_{\mathrm{A}}^{\mathrm{PML}} \cap \Omega_{\mathrm{A}}$. The tesselation of the domain $\Omega_{\mathrm{A}}^{\mathrm{PML}}$ into elements is denoted $\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}$.

The weak form is derived by multiplication with weighting functions $\tilde{\boldsymbol{w}}, \tilde{q}, y_j, \tilde{\mu}$ for the velocity, pressure, auxiliary, and trace field, respectively, integration over one element, partial integration, summation over all elements in the domain, and replacement of the element boundary terms stemming from partial integration by the trace variable and stabilization term:

$$
\begin{aligned}
&\left(\tilde{\boldsymbol{w}}, \frac{\partial \tilde{\boldsymbol{v}}}{\partial t}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} - \left(\nabla \cdot \tilde{\boldsymbol{w}}, \frac{1}{\rho}\tilde{p}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left\langle \tilde{\boldsymbol{w}} \cdot \boldsymbol{n}, \frac{1}{\rho}\tilde{\lambda}\right\rangle_{\partial\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} = -\left(\tilde{\boldsymbol{w}}, \boldsymbol{A}\tilde{\boldsymbol{v}}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}}, \\
&\left(\tilde{q}, \frac{\partial \tilde{p}}{\partial t}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left(\tilde{q}, c^2\rho\nabla\cdot\tilde{\boldsymbol{v}}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left\langle \tilde{q}, c^2\rho\tau(\tilde{p}-\tilde{\lambda})\right\rangle_{\partial\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} = -\left(\tilde{q}, \rho c^2 \sum_{j=1}^{s} z_j\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}}, \\
&\left(y_j, \frac{\partial z_j}{\partial t}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left(y_j, \mu_j z_j\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} = \left(y_j, \mu_j\left(\boldsymbol{E}_j \otimes \boldsymbol{F}_j\right) : \left(\tilde{\boldsymbol{v}} \otimes \nabla\right)\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} \quad \text{for } j = 1, ..., s, \\
&\left\langle \tilde{\mu}, \tilde{\boldsymbol{v}} \cdot \boldsymbol{n}\right\rangle_{\partial\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left\langle \tilde{\mu}, \tau(\tilde{p}-\tilde{\lambda})\right\rangle_{\partial\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} = 0,
\end{aligned}
$$

where a Neumann boundary condition is assumed on the outer PML boundary $\Gamma_{\mathrm{A}}^{\mathrm{PML}}$. Application of the first order ABC or a Dirichlet condition on the outer PML boundary is analogous as in Section 2.2, i.e., for the ABC an additional term is added to the last of the equations analogous to the term in equation (2.13) and a Dirichlet condition would be weakly imposed on the pressure field by setting the trace variable to the L_2 projection of the prescribed value on the faces along the boundary.

The discretized problem reads: Find $\tilde{p}_h \in P_h, \tilde{\boldsymbol{v}}_h \in \boldsymbol{V}_h, z_{j,h} \in P_h, \tilde{\lambda}_h \in L_h(p_{\mathrm{D}})$ such that for all $\tilde{q}_h \in P_h, \tilde{\boldsymbol{w}}_h \in \boldsymbol{V}_h, y_{j,h} \in P_h, \tilde{\mu}_h \in L_h(0)$

$$
\left(\tilde{\boldsymbol{w}}_h, \frac{\partial \tilde{\boldsymbol{v}}_h}{\partial t}\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} - \left(\nabla \cdot \tilde{\boldsymbol{w}}_h, \frac{1}{\rho}\tilde{p}_h\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} + \left\langle \tilde{\boldsymbol{w}}_h \cdot \boldsymbol{n}, \frac{1}{\rho}\tilde{\lambda}_h\right\rangle_{\partial\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}} = -\left(\tilde{\boldsymbol{w}}_h, \boldsymbol{A}\tilde{\boldsymbol{v}}_h\right)_{\mathcal{T}_{\mathrm{A}}^{h,\mathrm{PML}}}, \tag{4.15}
$$

$$\left(\tilde{q}_h, \frac{\partial \tilde{p}_h}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} + \left(\tilde{q}_h, c^2\rho\nabla\cdot\tilde{\boldsymbol{v}}_h\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} + \left\langle \tilde{q}_h, c^2\rho\tau(\tilde{p}_h - \tilde{\lambda}_h)\right\rangle_{\partial\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} = -\left(\tilde{q}_h, \rho c^2 \sum_{j=1}^{s} z_{j,h}\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}}, \tag{4.16}$$

$$\left(y_{j,h}, \frac{\partial z_{j,h}}{\partial t}\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} + \left(y_{j,h}, \mu_j z_{j,h}\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} = \left(y_{j,h}, \mu_j \left(\boldsymbol{E}_j \otimes \boldsymbol{F}_j\right) : \left(\tilde{\boldsymbol{v}}_h \otimes \nabla\right)\right)_{\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} \quad \text{for } j = 1, ..., s\,, \tag{4.17}$$

$$\left\langle \tilde{\mu}_h, \tilde{\boldsymbol{v}}_h \cdot \boldsymbol{n}\right\rangle_{\partial\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} + \left\langle \tilde{\mu}_h, \tau(\tilde{p}_h - \tilde{\lambda}_h)\right\rangle_{\partial\mathcal{T}_\mathrm{A}^{h,\mathrm{PML}}} = 0. \tag{4.18}$$

In contrast to the material parameters c, ρ, which are assumed to be element-wise constant, the eigenvalues μ_j in equation (4.17) are generally space dependent, which can introduce an integration error in combination with the standard quadrature. However, since absorption functions are smooth by construction and reasonable PMLs span several elements, effects like aliasing are assumed to be negligible.

The semidiscret system in matrix form reads

$$\begin{bmatrix} \mathbb{A} & 0 & 0 \\ 0 & \mathbb{M} & 0 \\ 0 & 0 & \mathbb{P} \end{bmatrix} \begin{bmatrix} \dot{\tilde{\boldsymbol{V}}} \\ \dot{\tilde{\boldsymbol{P}}} \\ \dot{\tilde{\boldsymbol{Z}}} \end{bmatrix} + \begin{bmatrix} \mathbb{L} & \mathbb{B} & 0 \\ \mathbb{H} & \mathbb{D} & \mathbb{R} \\ \mathbb{S} & 0 & \mathbb{T} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{V}} \\ \tilde{\boldsymbol{P}} \\ \tilde{\boldsymbol{Z}} \end{bmatrix} + \begin{bmatrix} \mathbb{C} \\ \mathbb{E} \\ 0 \end{bmatrix} \tilde{\boldsymbol{\Lambda}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \tag{4.19}$$

$$\mathbb{I}\tilde{\boldsymbol{V}} + \mathbb{J}\tilde{\boldsymbol{P}} + \mathbb{G}\tilde{\boldsymbol{\Lambda}} = \boldsymbol{0}. \tag{4.20}$$

The last equation enforces the continuity between elements and is the same as in equation (2.15). Thereby, also the continuity between PML and non-PML region is enforced. The vector $\boldsymbol{Z}$ summarizes the degrees of freedom for all auxiliary variables z_j. As in Section 2.3.2, time integration is based on explicit Runge–Kutta time stepping. The derivation of the fully discrete system is analogous and is not repeated here.

4.5 Numerical Examples

In the following, several examples are presented, showing basic functionality and properties on the one hand (Sections 4.5.1-4.5.4) and the advantages of the proposed scheme over common schemes on the other hand (Section 4.5.5).

4.5.1 Convergence Behavior in One Dimension

For a convergence study of the PML formulation, a semi-one-dimensional domain of length $l = 1$ with PML of length δ is created as shown in Figure 4.8. On the boundary along the x_1 direction, homogeneous Neumann boundary conditions are applied creating the one-dimensional behavior. On the left and right boundary along the x_2 direction, the first order ABC is applied for the study at hand but various boundary conditions are studied in the subsequent sections. In a first setup, the PML length is set to $\delta = 1$. The domain is meshed with 200 linear elements in

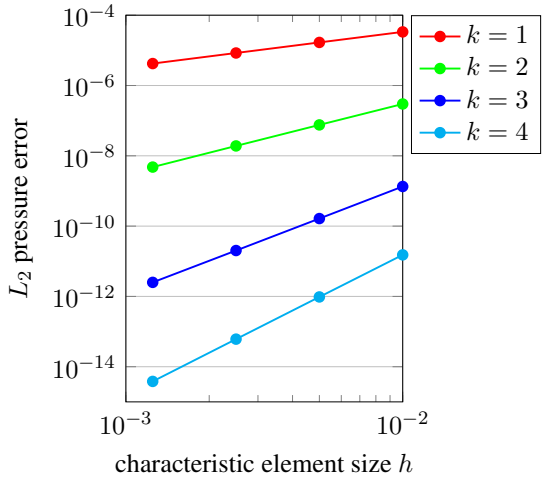


Figure 4.7: Convergence results for one-dimensional PML example.

x_1 direction. The material parameters are $\rho = 1$ and $c = 1$. For time integration, the LSRK3(3) integrator is used and the time step size is $\Delta t = 0.001$ with a Courant number of $Cr = 0.1$. Initial pressure and velocity are

$$p_0 = e^{-100\cdot(x_1-0.5)^2},$$
$$v_0 = e^{-100\cdot(x_1-0.5)^2},$$

representing a Gaussian hill propagating in positive x_1 direction. For $t = 0$, the Gaussian hill is located at $x_1 = 0.5$.

The convergence behavior is studied in terms of the reflection at the interface between PML and physical domain. As mentioned earlier, the interface between PML and physical domain is only reflection free in the continuous context. In the following, a numerical convergence study on the reflection is carried out for polynomial degrees $k = 1, 2, 3, 4$ of the shape functions. The measured quantity is the L_2 pressure error at $t = 1.0$ in the physical part of the domain. Theoretically, the pressure is zero in the entire physical part but due to reflections at the interface, pressure amplitudes are measured. Figure 4.7 shows the convergence on four uniformly refined meshes. From the HDG context and as examined in Chapter 2, optimal convergence with order $k + 1$ accuracy is expected. The measured results indicate convergence of order k, which is assumed to result from the equation for the auxiliary variable z, which always starts from a zero field and has the velocity divergence, i.e., a derived quantity, as input.

4.5.2 Quantitative Study of the Absorption Functions

The setup is the same as in the preceding section with the initial fields representing a Gaussian hill and $k = 3$. Here, different absorption functions as introduced in Section 4.3 are studied. To prevent infinite values of σ, the hyperbolic function and the shifted hyperbolic function assume 1.25δ for PML width. Large values of σ can yield instabilities in the time integrator. The scaling parameter in the absorption function is set to $\alpha = 100$.

Figure 4.9 shows snapshots of the pressure field for various points in time. For $t = 0.5$, the peak of the Gaussian hill hits the interface $\Gamma^{\text{PML}}_{\text{A,in}}$. At this point in time, it is already apparent

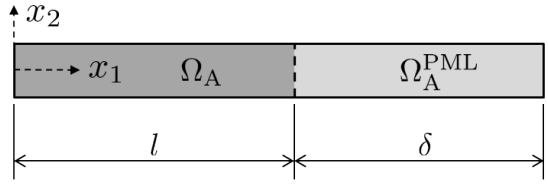


Figure 4.8: Computational semi-one-dimensional domain with PML on the right boundary.

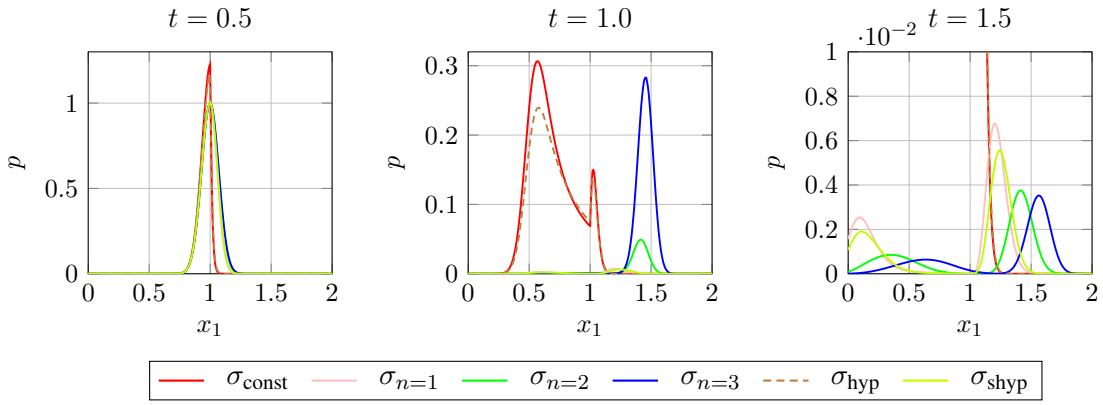


Figure 4.9: Pressure along the x_1 axis for three points in time simulated with six different choices for the absorption functions.

that high reflections occur for the constant profile function σ_const as well as for the hyperbolic absorption function σ_hyp. As stated in the derivation, the PML is only perfectly matched in the continuous case. Also for the other absorption functions, reflections occur at the interface but they are of much lower amplitude as can be seen from the snapshots at $t = 1.0$ and $t = 1.5$. In the PML, the wave travels more slowly and is damped. For $t = 1.0$, the cubic absorption function shows low damping compared to the quadratic, the linear, and the shifted hyperbolic function. This is due to the fact that the parameter α is the same for all, and the integral over σ starting from the interface $\Gamma_\mathrm{A,in}^\mathrm{PML}$ is lower for the cubic profile. However, the reflections traveling back through the physical domain are lower for the cubic absorption function. The linear and the shifted hyperbolic absorption function perform similarly in quality and quantity. From this comparison, constant and hyperbolic functions appear disadvantageous and quadratic and cubic absorption functions perform better compared to linear and shifted hyperbolic function in terms of reflections at the interface.

In the next test, the quadratic profile function is used but the parameter α to scale the absorption function is varied. Apart from that, all other settings are as in the preceding simulations. Figure 4.10 displays the pressure along the x_1 axis at time $t = 0.9$. In the PML domain (right panel of Figure 4.10), pressure amplitudes decrease for increasing α. For $\alpha = 1$ almost no damping occurs and the maximal pressure amplitude is 0.977. For $\alpha = 10, 100, 200, 400$, the maximal pressure amplitudes in the PML domain are $0.81, 0.16, 0.04, 0.01$, respectively. The qualitative behavior is opposite in the physical domain: a part of the Gaussian hill is reflected at the interface between physical and PML domain and propagates back into the physical domain. For higher values of the parameter α, higher reflections occur. Maximal pressure amplitudes in the physical

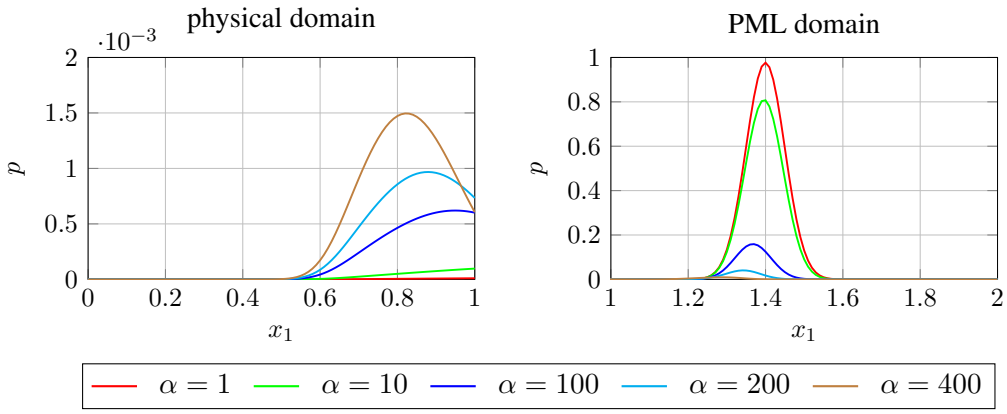


Figure 4.10: Pressure along the x_1 axis at time $t = 0.9$ for a quadratic profile function with different values for the parameter α scaling the absorption function. Notice the different scaling of the pressure axis in the two panels.

domain are $1.02 \cdot 10^{-5}, 9.72 \cdot 10^{-5}, 6.20 \cdot 10^{-4}, 9.68 \cdot 10^{-4}, 1.50 \cdot 10^{-3}$ for $\alpha = 1, 10, 100, 200, 400$, respectively. This example highlights a typical trade off when configuring a PML: high values for α induce higher damping but also increase unphysical reflections at the interface. The situation is even more intricate, if the PML width is smaller as will be shown in Section 4.5.3.

4.5.3 The Layer Width

In this section, the layer width δ is examined quantitatively. Therefore, the same setup as in the previous sections is chosen except that the boundary at $x_1 = l + \delta$ is not applied with the first order absorbing boundary condition but a Neumann boundary condition that causes reflections. Therefore, all wave absorption has to be carried out by the PML. Shape functions of $k = 3$ and a mesh with characteristic element size $h = 0.01$ are chosen. Quadratic as well as shifted hyperbolic absorption functions are used with $\alpha = 100$ for both cases. Additionally, the quadratic profile function is tested with $\alpha = 400$. To allow for a fair comparison between the different setups, the maximum of the modulus of the pressure is measured at $t = {}^1\!/_c\,(l + 2\delta)$. At this time, the Gaussian hill traveled to the right boundary, was reflected and returned to the initial position. The shortest tested layer width corresponds to the size of one finite element, i.e., $\delta = h$. The highest tested value corresponds to $\delta = 20h$, which is beyond the range of reasonable choices in practical applications but is presented here to gain a deep understanding of the underlying methodical and numerical properties.

Figure 4.11 plots the results. In the semi-logarithmic plot, the quadratic absorption function with $\alpha = 100$ yields a straight line indicating an exponential relation between layer width and reflection amplitude. For thin PML layers, the quadratic profile with $\alpha = 400$ yields the lowest reflections. For $\delta \in [0.05, 0.1]$, the shifted hyperbolic profile function performs best. For the thick layer with $\delta = 0.2$, the quadratic profile with $\alpha = 100$ appears best. At this stage, the layer is thick enough to absorb the wave almost completely and the dominant contribution to the measured pressure amplitude stems from the reflection at the interface, which is why $\alpha = 100$ gives a better result than $\alpha = 400$. It is important to keep in mind that the spatial discretization uses shape functions of polynomial degree $k = 3$, which yields comparably low errors from the

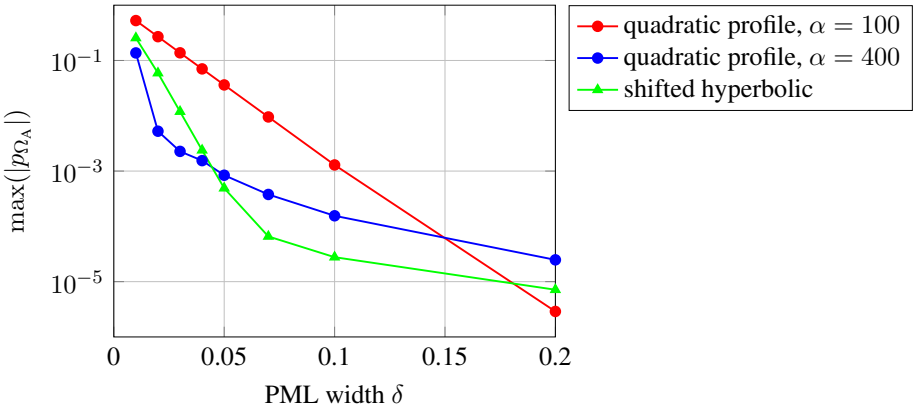


Figure 4.11: Reflections in the physical domain as function of the layer width δ for a one-dimensional setup comparing quadratic and shifted hyperbolic absorption functions.

reflection at the interface (see Figure 4.7). Most of the errors measured for the quadratic profile with $\alpha = 100$ are caused by too low damping. For $\alpha = 400$, effects of both the reflection at the interface as well as the reflection at the outer boundary are seen. In applications, a reasonable trade off must be found between reflections at the interface and damping in the PML.

4.5.4 The Outer Boundary Conditions

A possibility to improve the absorption is to apply the first order absorbing boundary condition to the PML boundary Γ_A^{PML}. It is expected that this allows for smaller PMLs for a given level of accuracy. In this section, the effect of the outer boundary conditions is quantified. Since the first order absorbing boundary condition is exact in one dimension and perfectly absorbs orthogonally impacting waves, the effect of the outer boundary conditions is studied on a quadratic domain with a spherically propagating pressure wave. The physical domain is $\Omega_A = [-1, 1] \times [-1, 1]$ and surrounded by PMLs of variable width. The mesh is Cartesian with quadratic elements ($k = 2$) of size $h = 0.02$, the time step size is set to $\Delta t = 0.001$, and the initial pressure and velocity distributions are given by

$$p_0 = e^{-100 \cdot (x_1^2 + x_2^2)},$$
$$\boldsymbol{v}_0 = \mathbf{0}.$$

For the PML, a quadratic absorption function with $\alpha = 100$ is chosen. The tested widths correspond to one to ten layers of PML elements. Numerical experiments are run with the first order absorbing boundary condition, a Neumann condition, or a Dirichlet boundary condition on Γ_A^{PML}. Since the reflections from the outer boundary are the quantity of interest, the maximal pressure value in the physical domain at $t = 2 + 2 \cdot \delta$ is measured.

Figure 4.12 plots the maximal of the modulus of the pressure for the different configurations over the PML width δ. For thick layers, all configurations perform similarly because the PML behavior and reflections at the PML interface dominate over the effect of the outer boundary condition. The measured reflections stem from the interface between physical domain and PML

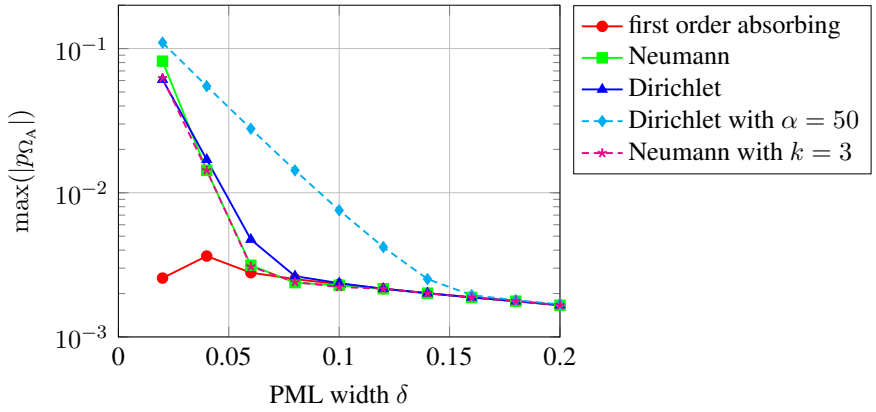


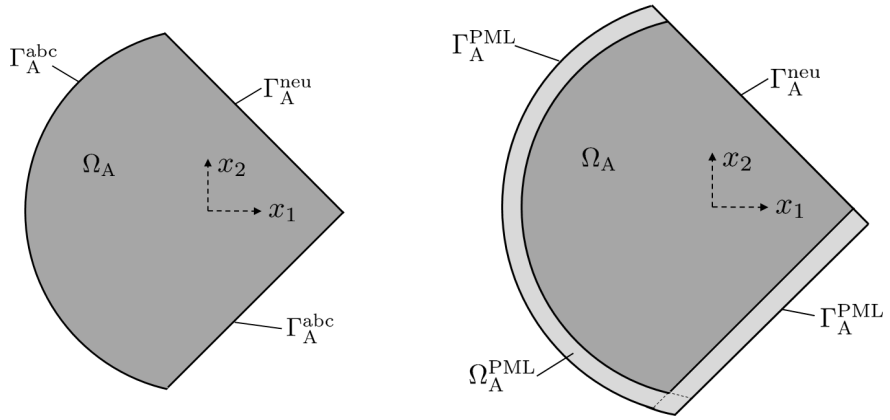
Figure 4.12: Reflections in the physical domain as function of the layer width δ for a two-dimensional setup visualizing the influence of the outer boundary condition.

and not from the outer boundary conditions. For PML width $\delta < 0.1$ and $\alpha = 100$, the effect of the outer boundary condition is visible. The transition from dominating errors at the interface to dominating errors at the outer boundary manifests as kink between $\delta = 0.05$ and $\delta = 0.1$ for $\alpha = 100$. For Neumann and Dirichlet boundary conditions, reflections occur at the outer boundary, while the absorbing boundary condition dampens most of these reflections. For comparison, the Dirichlet test case is also evaluated for $\alpha = 50$ plotted as dashed cyan line in Figure 4.12. The decrease of the reflections with increasing PML width is significantly slower due to the reduced damping. The transition from domination of reflections stemming from the outer boundary to a damping within the PML is slower. To elaborate on the discretization error, the simulation with Neumann boundary condition is repeated but with cubic shape functions $k = 3$ instead of $k = 2$ as in the preceding section. Figure 4.12 shows the result as magenta dashed line. Comparing both setups with $k = 2$ and $k = 3$ reveals that the curves are very similar. Only for the smallest PML width of $\delta = 0.02$, a difference is seen. The dominating error is hence not due to the discretization error but due to the general reflection and damping properties of the PMLs.

For PMLs of sufficient width or with a sufficiently high damping parameter, the choice of the outer boundary condition does not influence the simulation accuracy significantly. In cases of thin PMLs or unsuitable damping or absorption function, the absorbing boundary can enhance the accuracy of the simulation. In this setup, the maximal deviation of the incident angle from orthogonal incident onto the outer boundary is $45°$. If this deviation was higher, the gains from the absorbing boundary condition would be lower.

4.5.5 A General Setup

The general applicability of the PML approach presented herein is demonstrated using a two-dimensional geometry as shown in Figure 4.13. The physical domain Ω_A consists of a circle of radius $R = 1$ which is cut by planes in a $45°$ angle through the point $(x_1, x_2) = (0.8, 0)$. The upper straight boundary represents a hard wall while the other two boundaries should be perfectly absorbing. Two setups are compared, see Figure 4.13. In the first setup, the absorbing boundaries are mimicked by the first order ABC. In the second setup, PMLs of width $\delta = 0.1$ are used. The reflecting hard wall is represented by a Neumann boundary. The initial fields are



(a) Physical geometry (b) Physical geometry with PMLs

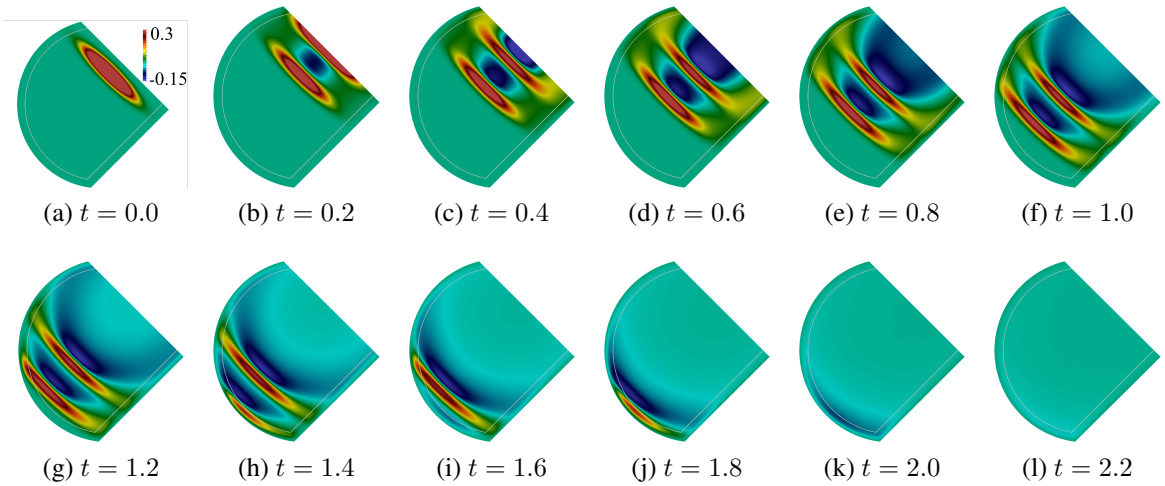
Figure 4.13: Setup consisting of a cut circle with one reflecting and two absorbing boundaries.

given by

$$p_0 = e^{-100\cdot(x_1+x_2-0.5)^2-4\cdot(x_1-x_2+0.3)^2},$$
$$\boldsymbol{v}_0 = \mathbf{0}.$$

The computational domain is discretized with 5676 and 7190 quadratic elements ($k = 2$) in the ABC and PML case, respectively. The material parameters are $c = 1, \rho = 1$ and the LSRK3(3) with $\Delta t = 0.001$ is used.

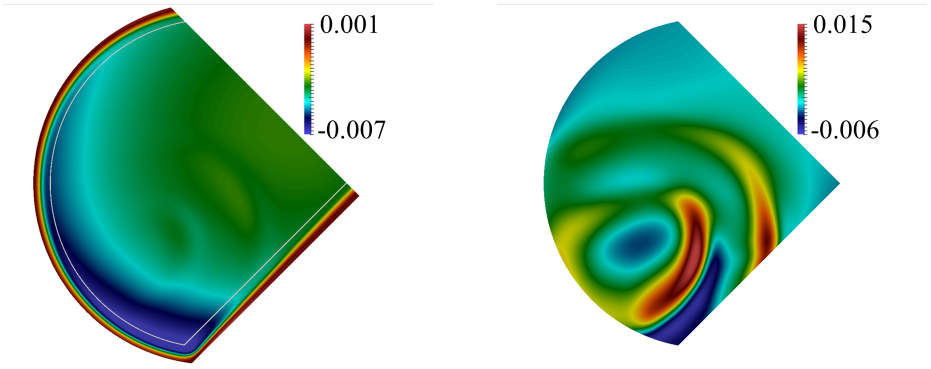
This example puts high demands on the PML formulation because two PMLs, a spherical and a straight lined overlap as shown in 4.13(b). In the overlapping region, the general formulation according to equations (4.12)–(4.14) with spectral decomposition is applied, while the other regions allow for the specific formulations for spherical and straight lined boundaries. Figure 4.14 shows snapshots of the pressure fields at various points in time to give an impression of the wave propagation pattern. Figure 4.15 compares the pressure fields at the final time $T = 2.6$ between ABC and PML setup. Apparently, the setup with ABCs yields higher artificial reflections. In the physical domain Ω_A, the maximal pressure is $\max|p| = 0.0157$ and $\max|p| = 0.0076$ for ABC and PML, respectively.



(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.4$ (d) $t = 0.6$ (e) $t = 0.8$ (f) $t = 1.0$

(g) $t = 1.2$ (h) $t = 1.4$ (i) $t = 1.6$ (j) $t = 1.8$ (k) $t = 2.0$ (l) $t = 2.2$

Figure 4.14: Snapshots of the pressure field at various points in time. The grey lines separate PML and physical domain. The color scale is the same for all images.



(a) With PML (b) Without PML

Figure 4.15: Pressure fields at final time $t = 2.6$.

4.6 Conclusion

In contrast to the common stretched coordinate formulation, the transformation into the time domain is based on a spectral decomposition of $\boldsymbol{A}$. One advantage of the derived formulation is that only as many auxiliary variables z_i must be introduced as the matrix $\boldsymbol{A}$ has distinct non-zero eigenvalues. This is especially useful in settings where a PML is not aligned with the coordinate axes: in a two-dimensional as well as in a three-dimensional setting, only one auxiliary variable must be introduced within a plane PML. Also, this formulation is easily applied to circular and spherical PMLs in Cartesian coordinates. It turns out that the circular as well as the spherical PML require two auxiliary variables (for the sphere, the second eigenvalue has algebraic multiplicity of two), thus reducing computational expense.

Another advantage is that the eigenvalues of $\boldsymbol{A}$ indicate non-stable configurations of PMLs before executing the numerical simulation. When constructing a computational domain, choosing profile functions, and PML geometries, the calculation of the eigenvalues of $\boldsymbol{A}$ reveal the stability of the configuration. It turns out that it is possible to surround even concave polygons with PMLs and the eigenvalues remain positive. In contrast, a concave sector of a circle as PML part turns out to be unstable.

The presented numerical examples demonstrate convergence properties and give indications on the choice of parameters for a PML configuration. A common drawback of PMLs is that several parameters must be determined by the user, namely the outer boundary conditions, the layer width, and the profile function (shape and magnitude). The choice of a set of parameters is difficult and depends heavily on the example at hand, which is why no general rules can be provided. The given numerical examples supply a first impression on the impact of the parameter choices on the solution quality. For the derived method and the alogrithmic framework at hand, one can state that the first order ABC should always be applied to the outer boundary because it can improve solution quality in case of thin layers without increasing the computational cost. The layer width should at least span two element layers but computational cost increases with the layer width. The definition of a good profile function for a given problem is most challenging. Quadratic profile functions are most common while shifted hyperbolic functions are preferable for thin PMLs. An extensive study on profile functions is given in [112].

An aspect that should be addressed by future work is the convergence behavior. As shown in Chapter 2, the spatial discretization of the acoustic wave equation yields optimal convergence of order $k + 1$. In combination with PMLs, however, convergence of order k is observed, which is traced back to the formulation of the equations for the auxiliary variables. Enhancing the DG formulation for the system (4.12)–(4.14) through integration by parts in equation (4.14) and introduction of a numerical flux might recover optimal convergence.

The herein presented PML formulation allows for PMLs surrounding general prismatic bodies and also for PMLs combining spherical or cylindrical boundaries with straight lined boundaries, which (to the author's knowledge) has previously not been possible.

5 Acoustical Solver Applications

The accurate prediction of sound propagation is a challenging task and until now, there is no method yielding accurate results over the entire frequency range and for early as well as late reflections. In [153], a comprehensive summary of methods and their applicability to specific use cases is given. Numerical methods to find an approximate solution of the acoustic wave equation, like finite difference, finite element, or boundary element methods allow for the prediction of diffraction, early and late reflections, but they are comparably slow for high frequencies because of high resolution demands [54]. For high frequencies, geometrical methods like ray tracing are more suited, which however fail to capture diffraction effects occurring especially in low frequency regimes [153]. In case multiple reflections occur, sound loses its directionality and a diffusive sound propagation can be assumed. Then, a parabolic diffusion equation is solved [17], which is numerically cheaper compared to the hyperbolic acoustic wave equation. The accuracy of this method however is limited.

In this work, the acoustic wave equation is solved using DG methods and explicit time integration as presented in Chapter 2. Hence, the explicit DG approach is put into perspective with competitive methods and relevant applications. Historically, the finite difference time domain method (FDTD) is most popular with the first implementations for the acoustic wave equation in 1994 described in [18, 142]. Back then, however, the computational resources only allowed for coarse grids and very low frequencies ($< 150\,\mathrm{Hz}$). With the advances of computational resources and developments of new numerical methods or further developments of basic numerical methods, nowadays a large variety of complex real world problems can be solved and sound propagation prediction through numerical simulations plays an important role in design processes. Relevant problem classes among others are general urban acoustics [78, 115] and more specifically city planning [78], or street canyon design [139] as well as room acoustics, e.g. class rooms, living rooms, concert halls, corridors, open space offices, see [16, 123, 132]. Another topic subject to vivid research is auralization [130, 135, 141], i.e., the simulation of sound propagation in specific scenarios to generate acoustical signals and make the modeling results audible. Auralization improves the user experience of virtual reality, movies, and computer games. In the last years, efforts were made to introduce benchmark examples for linear and nonlinear acoustics; until now, however, most examples for linear acoustics refer to the frequency domain [79, 80].

In the following sections, representatives of urban acoustics and room acoustics are presented in order to demonstrate the applicability of the developed code to real world problems, but also to compare the computational expenses with the results reported in literature. In Section 5.1, sound propagation in a village is shown, with reference to [115]. In Section 5.2, a cathedral like geometry is studied as in [16].

5.1 Urban Acoustics

In [78], a review on the recent questions that are relevant to computational urban acoustics is given. It highlights the importance to be able to accurately predict time dependent sound fields in urban environments. Here, the applicability of the proposed acoustical solvers to answer questions of urban acoustics is demonstrated based on a three-dimensional training village, which has already been used in order to verify an acoustical solver [115]. Consideration of this example is representative for general outdoor acoustics, which is relevant for urban planning and city design [87] or useful for gun shot localization in the context of crime control [103].

The geometry under consideration is based on the artificial village presented in [2, 115], where a FDTD method and an adaptive rectangular decomposition approach is used to solve the acoustic wave equation in the training village. The geometry consists of fifteen buildings of different height (here, flat roofs are used, which is in contrast to the original publication where not only flat roofs but also peeked roofs are used). Figure 5.1(a) depicts the geometry of the buildings. The computational domain is a cuboid of size $175 \times 140 \times 14$ with the buildings cut out. Surfaces corresponding to walls, roofs, and the ground are assumed to be perfectly reflecting, whereas the first order absorbing boundary condition is applied on all other boundaries. A source is located at $(62, 104, 1)$ corresponding to SP1 from [115].

In [115], simulations were run on a mesh consisting of 11 million grid points, a time step size of $\Delta t = 3.85 \cdot 10^{-4}$ with 2000 time steps, which corresponds to a simulation frequency of 450Hz. They state that a simulation took 20 minutes on a single core CPU machine. Here, simulations are run on several discretizations as listed in Table 5.1, which approximately recreate the same number of grid points as in [115]. The final time is $T = 0.77$. For time integration, ADER without reconstruction is used with a Courant number of $Cr = 0.2$ in the smallest element. The average Courant number is $Cr \approx 0.05$. Figure 5.2 shows pressure snapshots at various points in time to give an impression of the sound propagation patterns and Figure 5.3 gives a three-dimensional impression of the sound field.

Simulations are run on a two socket Intel Xeon E5-2690 v4 Broadwell 2.6GHz system, compiled with the `g++` compiler, version 6.2, at optimization level `-march=haswell -O3 -funroll-loops`. Table 5.2 summarizes the computational timings. The number of processors and the wall time for the specified number of processors are shown. The CPU time calculates as product of processors and wall time. Last, the CPU time for 2000 time steps is given in min-

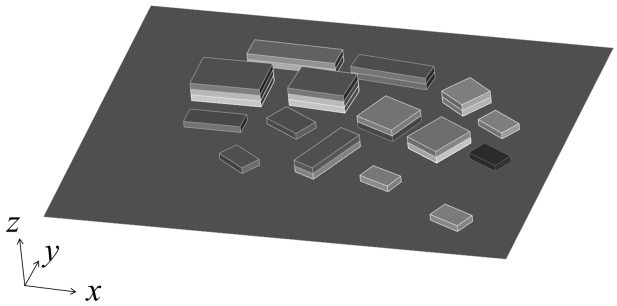
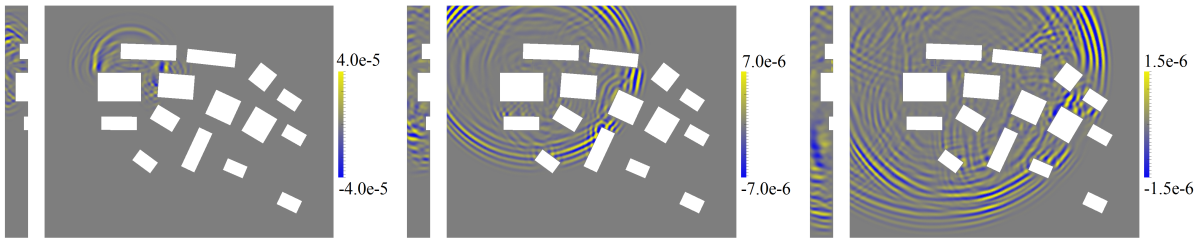


Figure 5.1: Geometry of the training village.



(a) Pressure at $t = 0.1$ (b) Pressure at $t = 0.2$ (c) Pressure at $t = 0.3$

Figure 5.2: Pressure snapshots in the training village on the xy plane at $z = 1$ and on the yz plane at $x = 62$.

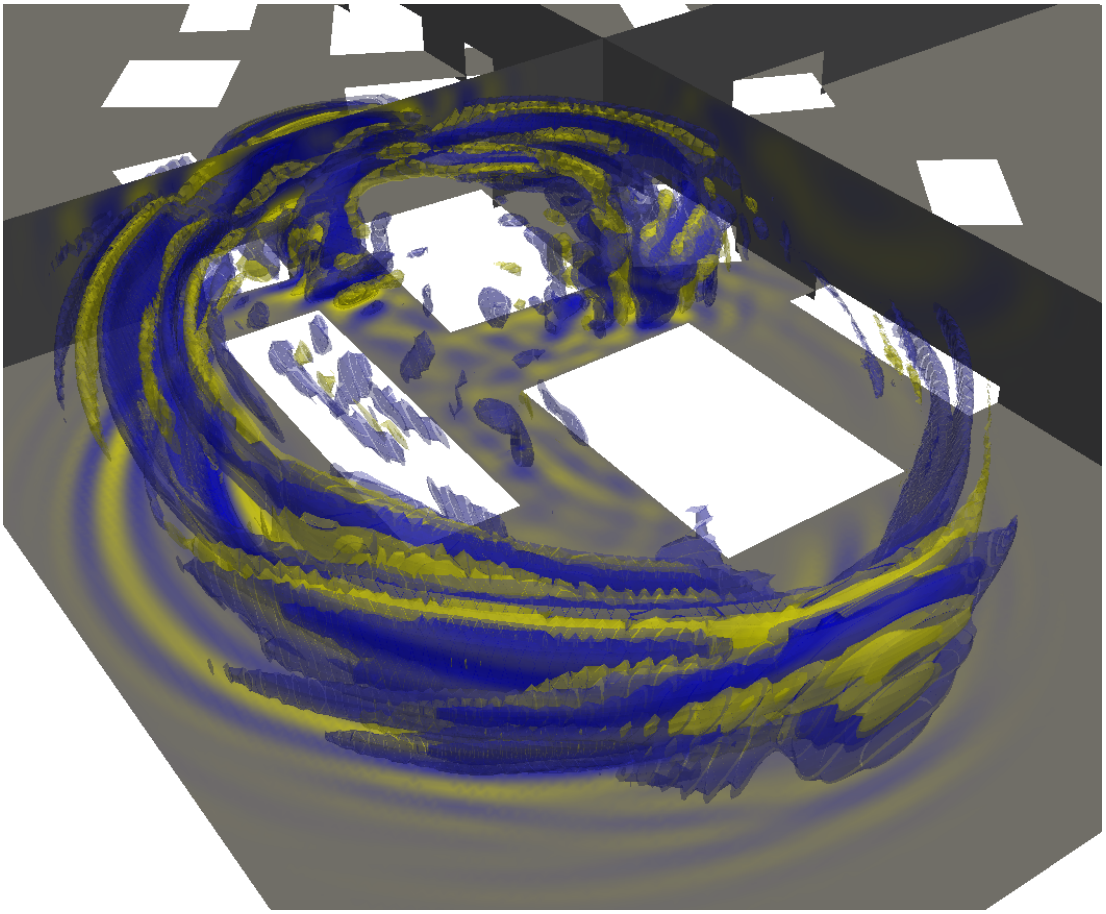


Figure 5.3: 3D view of the pressure at $t = 0.12$ on three planes and as isosurfaces.

	setup I	setup II	setup III
grid spacing	0.6	1.2	2.4
degree k	1	3	7
effective resolution	0.3	0.3	0.3
#grid points	$1.20 \cdot 10^7$	$1.15 \cdot 10^7$	$1.21 \cdot 10^7$
Δt	$7.8 \cdot 10^{-5}$	$2.45 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
#time steps	6436	19704	26180

Table 5.1: Discretizations used for the urban acoustics simulation.

	setup I	setup II	setup III
#processors	28	28	28
wall time [s]	$1.2 \cdot 10^3$	$2.5 \cdot 10^3$	$4.6 \cdot 10^3$
CPU time for all time steps [s]	$3.4 \cdot 10^4$	$6.9 \cdot 10^4$	$1.3 \cdot 10^5$
CPU time for 2000 time steps [min]	114	76	106

Table 5.2: Discretizations used for the urban acoustics simulation.

utes to allow comparison to [115]. Setup I requires the least CPU time for all time steps. For 2000 time steps, setup II is the fastest. Comparison to [115] with approximately the same number of grid points and the same number of time steps shows that the computational time is only 3.8 times higher for setup II. This is a very good result, considering that the adaptive rectangular decomposition is based on a semi-analytic approach using the discrete cosine transform in the decomposed rectangles and considering that their time integration is not of order five as in setup II but a two step type and hence of lower order.

The presented comparison does only consider computational time for a given number of grid points. The accuracy of the results is not taken into account. In [115], it is stated that the adaptive rectangular decomposition requires only 2.6 time samples per wavelength (compared to ten or twenty samples used in earlier FDTD methods). Here, a consistent discretization is supplied with accuracy order $\mathcal{O}(h^{k+1})$, which does only partly coincide with the old rules of thumb on how many nodes or samples must be used per wavelength. The source functions used in [115] were developed in [104], where frequency contents of $150\,\mathrm{Hz}$ corresponding to a wave length $\lambda = 2\,\mathrm{m}$ were specified. With the discretizations listed in Table 5.2, between three and seven nodes are used per wavelength depending on the considered element and because nodes are not equidistantly distributed within elements. However, one can clearly say that setup III is expected to yield the most accurate results. Although the effective resolution $h/k+1$ is the same in all three setups, setup III allows for an order eight approximation between nodes and therefore supplies the highest accuracy. A comparison of the computational performance considering accuracy and temporal stability should be addressed by future work.

5.2 Room Acoustics

Room acoustics is an area of intensive ongoing research because it is impacting human task performance [135] e.g. at workplaces. Also, computational prediction of acoustic characteristic of rooms is used in the design of concert halls. In the context of auralization, the acoustic characterization of indoor scenery is required as well.

In the literature, cathedrals and concert halls are common examples to demonstrate accuracy and performance of solvers for the acoustic wave equation, see e.g. [16, 116, 132]. The applicability of the proposed algorithm to a cathedral-like geometry is demonstrated in analogy to the example presented in [16], where it was studied in the context of finite volume and FDTD methods and claimed to be representative for a complex room geometry in three dimensions. The geometry consists of three overlapping blocks of length 40, height 10, and width 10. Additionally a half sphere of radius 10 is on top of the bricks as depicted in Figure 5.4. The initial pressure distribution is given by

$$p_0 = \exp\left(-n \cdot \left((x_1 - 15)^2 + x_2^2 + x_3^2\right)\right),$$

which corresponds to a Gaussian pulse excitation at the location indicated by S in Figure 5.4. Values $n = 0.5, n = 5$, and $n = 50$ will be studied to obtain signals of different frequency composition. All boundaries are assumed to be perfectly reflecting hard walls, except for the boundaries at the six ends of the blocks, where a first order ABC is applied. The geometry is discretized with 89880 elements of polynomial degree $k = 8$ or $8 \cdot 89880 = 719040$ elements of polynomial degree $k = 4$ obtained by uniform refinement. Both discretizations result in an effective resolution of approximately 0.0625. For time integration, ADER without reconstruction at a Courant number of $Cr = 0.2$ is used. The final time is $T = 0.2$. With the rule of thumb to use five nodes per wavelength[1], this spatial discretization allows for frequencies of up to $1\,\mathrm{kHz}$ and hence is in the low frequency simulation regime.

Figure 5.5 plots several pressure snapshots for $n = 50$ to give an impression of the sound propagation patterns. Figure 5.6 compares the reflection patterns for the different widths of the initial pulse at time $t = 0.07$ on the discretization of 89880 elements with $k = 8$. Only for the initial pulse with high frequency content, a clear image of all occurring reflections is found.

Pressure curves over time are monitored at the three receiver locations L_1, L_2, and L_3 as indicated in Figure 5.4(a), which are located at $(7.51, 0.01, 13.00), (-7.49, 0.01, 13.00)$, and $(-14.99, 0.01, 0.01)$. They are shown in Figure 5.7 for $n = 5$. These so-called impulse responses of a room can be used to determine the reverberation time, i.e., the time for reflected sound to become inaudible. This is an important quantity for the design of rooms because it determines the speech intelligibility. Impulse responses in rooms are also used in music or movie production, e.g., to artificially create the impression of reverberated sound matching the scenery by auralization. In Figure 5.7, the different arrival times of the initial signal are clearly visible. Since most walls are perfectly reflecting and no sound absorption is simulated, strong signals also arrive after the initial signal.

[1] Note that several rules of thumb exist specifying different values for the required nodes per wavelength. In [1], about four nodes per wavelength are suggested for high spatial approximation orders ($k > 10$), while [60] suggest about 4.6 nodes for $k = 10$ and six nodes for $k = 4$. A discussion of the rules of thumb is given in [111].

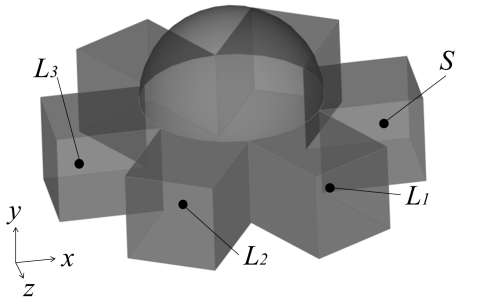
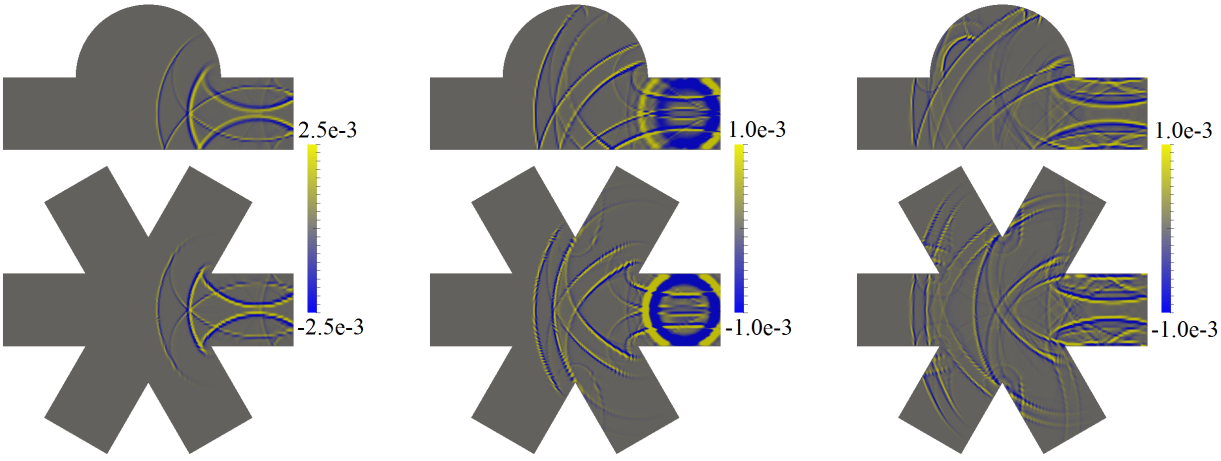
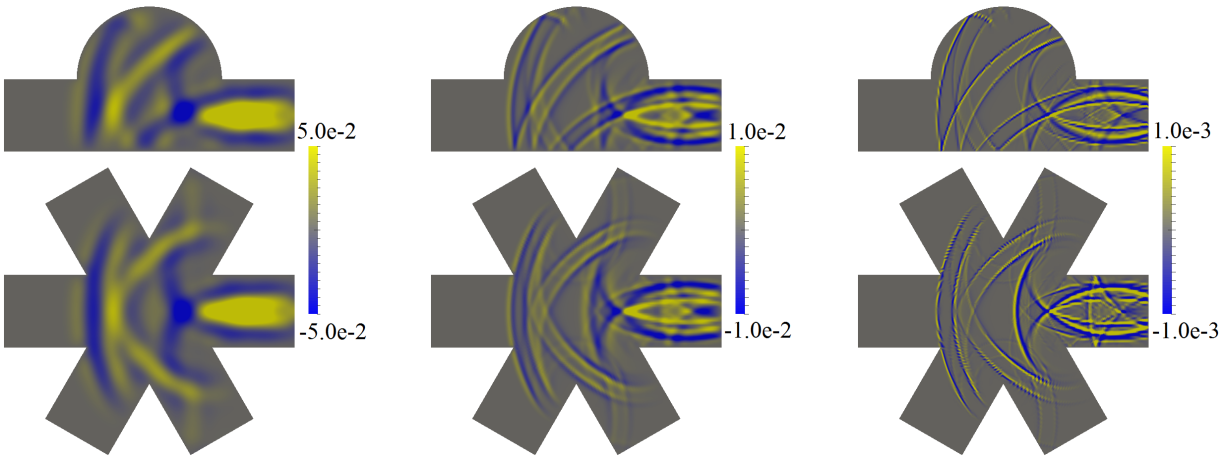


Figure 5.4: Complex three-dimensional room geometry.



(a) $t = 0.04$ (b) $t = 0.06$ (c) $t = 0.08$

Figure 5.5: Pressure snapshots in the room at various points in time for $n = 50$. Plots in the top row show the xy plane at $z = 0$ and plots at the bottom show the xz plane at $y = 0$.



(a) $n = 0.5$ (b) $n = 5$ (c) $n = 50$

Figure 5.6: Pressure snapshots in the room at $t = 0.07$ for the different initial pressure impulse widths. Plots in the top row show the xy plane at $z = 0$ and plots at the bottom show the xz plane at $y = 0$.

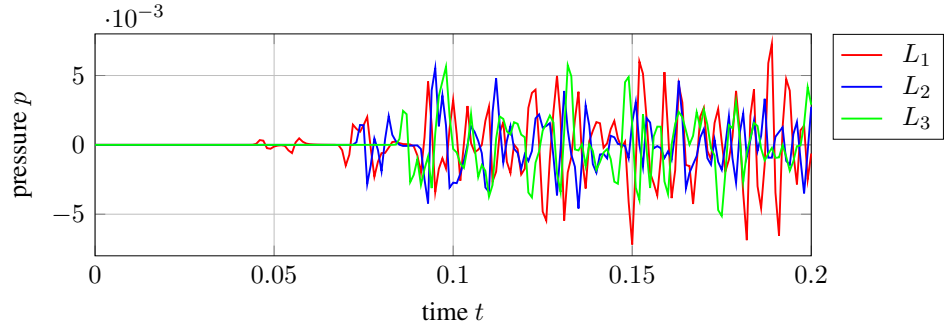


Figure 5.7: Pressure over time for the three locations as indicated in Figure 5.4 for $n = 5$.

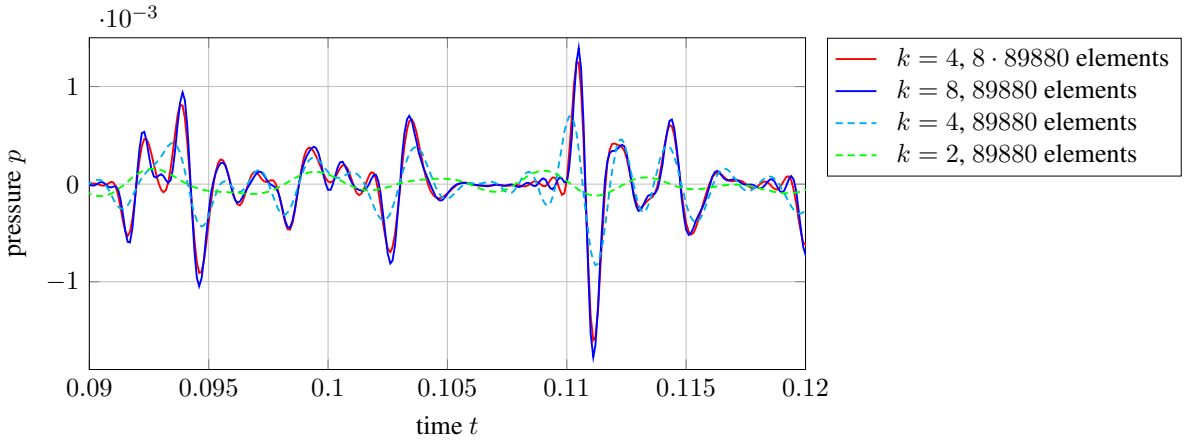


Figure 5.8: Pressure over time with $n = 50$ for detector location L_1 comparing different discretizations.

For the two given discretizations ($k = 8$ with 89880 elements versus $k = 4$ with one refinement, i.e., $8 \cdot 89880 = 719040$ elements), the pressure curves at detector location L_1 are compared. For wide initial pressure impulses ($n = 0.5, n = 5$), no differences are apparent. Only for the thin initial pressure impulse with $n = 50$, the two discretizations yield slightly different results as shown in Figure 5.8. The discretization with $k = 4$ and the coarse mesh shows lower amplitude peaks and slightly smoother curves, which is in accordance with the findings presented in Section 2.5.3. For comparison, pressure signals are also obtained with two coarser discretizations using $k = 2$ and $k = 4$ both with 89880 elements. The results are also shown in Figure 5.8. For $k = 2$, the signal is very smeared and fails to recover qualitative solution features because the high frequency components cannot be captured by the discretization. For $k = 4$, the distinct peaks of the signal are represented, however, with significantly lower amplitudes and a visible phase error.

II. The Optoacoustic Inverse Problem

6 The Optoacoustic Imaging Technique

The photoacoustic effect was first described by Alexander Graham Bell in 1880 [13]. Ever since, it was initially applied for the characterization of gases [164] and only in the early nineties its capabilities for photoacoustic (or equivalently optoacoustic) biomedical imaging were discovered and explored [99, 100, 124]. In the last decades, great advancements were achieved and the methodology has evolved to a sophisticated imaging technique with a wide range of applications [166]. An exhaustive overview of the historical developments is given in [110]. Here, the physical imaging principle is briefly introduced and afterwards, a brief literature review on optoacoustic image reconstruction methods is presented.

6.1 Functional Principle

In optoacoustic imaging, an object of interest is illuminated with a short pulse of laser light. The laser light is typically chosen in the near infrared range to maximize the penetration depth. The light propagates within the given object according to its optical properties and is partially absorbed. The absorption causes a temperature rise and hence thermal expansion, which induces local pressure rises within the tissue. The pressure propagates through the object according to its mechanical properties and is eventually detected as ultrasound signal. The detected ultrasound signals are used to reconstruct images of the object. See Figure 6.1 for a visualization of the procedure. Classically, the reconstructed images show either the optical absorption coefficient or the absorbed optical energy density map [166, 174]. However, there are also other methods reconstructing or estimating optical scattering properties as well as acoustical properties [143, 160], which will be addressed in more detail in Section 6.2.

Tomographic setups range from mounted tomographs [46], over self-made constructions, to handheld systems [23]. Thereby, different illumination scenarios are combined with different detector geometries and types. Doing so, either cross section images, entire three-dimensional reconstructions, or specific regions of interest are resolved and reconstructed. To bridge the gap between the object and the detectors, a coupling medium is required. Suited media are water or ultrasonographic coupling gel because their acoustic impedances are similar to the impedance of soft tissue [165].

The advantages of optoacoustic imaging over common imaging techniques are manifold. Compared to ultrasonography, optoacoustic imaging yields different contrast, because the images show optical properties rather than mechanical properties [41]. Compared to purely optical imaging techniques, optoacoustic imaging provides images with higher resolution because sound propagation is a wave propagation phenomenon unlike light propagation in biological tissue which is rather diffusive [24, 84]. In contrast to magnetic resonance imaging, the tomographic

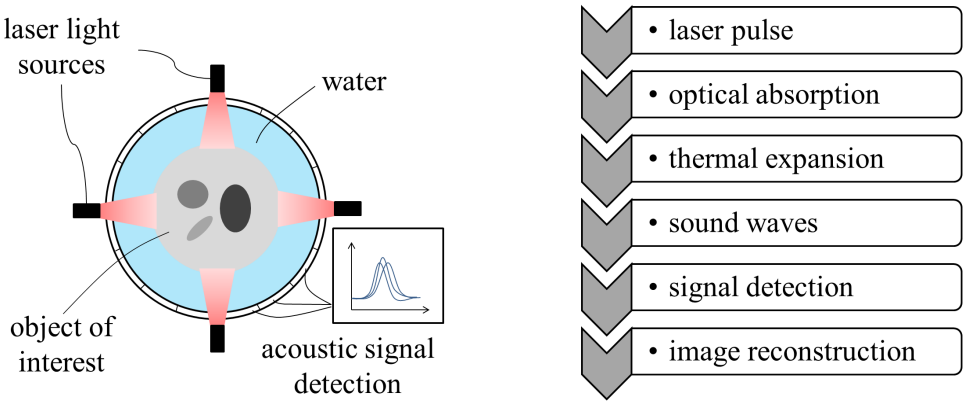


Figure 6.1: The optoacoustic imaging procedure: An object of interest is illuminated with a short impulse of laser light. Acoustic signals are generated by the photoacoustic effect and are detected at the boundary.

setup is less elaborate and less expensive. Last, the used excitation in terms of laser light is non-ionizing and therefore less hazardous than radiography. On the downside, optoacoustic imaging does not reach high penetration depths, such that it can only be used in regions close to the skin and not for imaging e.g. across the entire torso [41]. This is due to the fact that laser light is used for the excitation, which is strongly scattered and absorbed in biological tissue and penetrates only the first one or two centimeters of tissue. This drawback is overcome if microwaves are used for the excitation of signals. The imaging is then referred to as thermoacoustic imaging [174]. In that case however, the reconstructed images do not represent chromophore distributions as in optoacoustic imaging but dielectric tissue properties.

6.2 Optoacoustic Image Reconstruction Methods

Optoacoustic image reconstruction methods can be classified according to the quantity to be reconstructed: either the absorbed optical energy density which is assumed to be proportional to the initial pressure rise is reconstructed (case I), the optical absorption coefficient is reconstructed (case II), or other quantities are additionally reconstructed by specialized algorithms (case III). See Figure 6.2 for an overview. In case I, only the acoustical part of the imaging procedure is inverted, i.e., the acoustical source is reconstructed from the signals measured at the boundary. Three main approaches to solve the acoustic inversion exist, namely back-projection, time-reversal, and model-based inversion. An overview of acoustic inversion methods in optoacoustic tomography is given in [137]. The fundamental back-projection algorithm for optoacosutics is described in [173] for three-dimensional setups with planar, spherical, and cylindrical detection surfaces. Later, this algorithm was extended e.g. for limited view scenarios [105] or in the context of quality enhancement through weighting functions [73]. Time-reversal is another approach to solve the inverse acoustic source problem, which propagates the pressure measurements from the detectors backwards in time into the region of interest. An analytic time-reversal formula (which is equivalent to the universal back-projection formula of [173] under certain

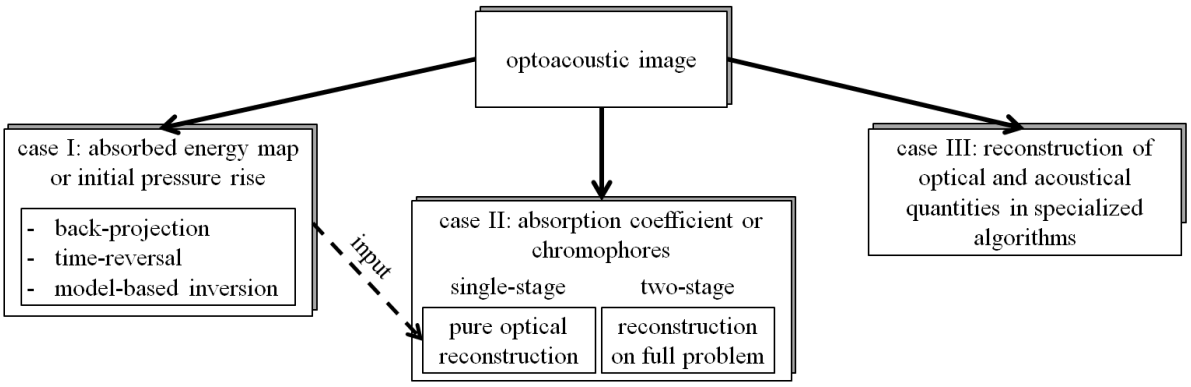


Figure 6.2: Optoacoustic imaging is classified according to the quantity to be reconstructed and according to the inverted parts of the imaging procedure.

assumptions) is derived from Green's functions [175]. Most time-reversal algorithms, however, are based on numerical implementations [67, 81, 162]. Model-based inversion techniques build a numerical model of the acoustic problem that maps an initial pressure field to boundary measurements and subsequently inverts the relation to solve for the initial pressure [114, 136]. In contrast to back-projection and time reversal methods, model-based inversion techniques are more flexible and allow to introduce additional modeling aspects like the detector frequency response [122].

In case II, the absorption coefficient is reconstructed from the absorbed energy map by solving the pure optical inverse problem and using case I as input (two-stage scheme), or the entire imaging procedure is covered by one model (single-stage scheme). The first of these two approaches is more common because of the lower problem complexity and lower computational expense with representatives [5, 10–12, 28, 41, 109, 119, 160, 178]. Conceptually, these methods are also able to reconstruct the diffusion or scattering coefficient or even the Grüneisen coefficient characterizing the thermodynamic tissue properties and are therefore also known as quantitative optoacoustic image reconstruction methods. One important question for this step of the optoacoustic imaging inverse problem is uniqueness. The reconstruction of absorption and scattering coefficient is non-unique in case only one illumination pattern is used [41]. In case an arbitrary number of illumination patterns is available, two optical properties can be determined uniquely and stably but not three as shown in [10]. For illuminations with different colors of laser light (multi-spectral illumination), all three coefficients can be reconstructed simultaneously [11]. As shown in [119], unique reconstruction of all three parameters is also possible with only one wavelength if the solution is restricted to piecewise constant parameters. These results hold for optical light transport modeled by the diffusion approximation. Results on the uniqueness of reconstruction for the radiative transport equation with multi-specral illumination are given in [109], and the reconstruction is again stable if several wavelengths are used. The reconstruction of optical properties based on the full problem is less common due to higher computational expenses but allows for new levels of flexibility in terms of setup or consideration of acoustical heterogeneities [71, 82, 143]. The work [143] bridges the gap to the specialized algorithms of case III that additionally reconstruct acoustical material properties like speed of sound, mass density, or impedance. In [85], a reconstruction algorithm is presented that is based on a finite element dis-

cretization of the Helmholtz equation and an adjoint sensitivity analysis. A statistical approach for strong acoustic heterogeneities is described in [43], which weighs signal parts with the probability of affection by acoustical heterogeneities for correction. Exact reconstruction formulas for speed of sound and absorption coefficient are presented in [92] with the restriction to sliced illumination experiments. A recent publication [107] proposes a segmentation method that uses characteristic features of the acoustical signals to determine acoustical properties prior to the optical image reconstruction. It is limited concerning variations in acoustical properties and setup. One general problem for the reconstruction of not only the initial energy distribution but also the speed of sound is that the solution is generally not unique. However, a '(weak) local uniqueness result' is presented for odd dimensions and restrictions on the parameters and the geometry in [81]. In [156], it is shown that the linearized problem for recovery of both speed of sound and initial energy distribution is unstable. Another important aspect is that most of the reviewed methods are only applied in a theoretical context and were not combined with experimentally obtained data.

The progress in numerical modeling and implementation as well as the increase in computing power make it possible to model the physical processes unfolding in an optoacoustic scanner during acquisition more accurately. As indicated in Section 1.1, an optoacoustic image reconstruction algorithm is derived that does not only recover the optical properties of an object, namely absorption and diffusion coefficients, but also the mass density and speed of sound without restrictions on the underlying geometry, the tomographic setup, or assumptions on parameter distributions. The algorithm includes an iterative gradient-based optimization scheme and is based on a physical description of the underlying problems taking the primary physical effects into account by means of the optical diffusion approximation and the acoustical wave equation.

7 The Optoacoustic Image Reconstruction Method

In this chapter, the developed image reconstruction algorithm is presented. In Section 7.1 and Section 7.2, the physical model and the numerical model are described, respectively. In Section 7.3, the objective function is presented and in Section 7.4 the parameter gradients are derived. The solution algorithm is given in Section 7.5. A proof of concept and a numerical example are shown in Section 7.6 and Section 7.7, respectively.

7.1 Physical Model

A general description for light transport in scattering tissue is provided by the radiative transfer equation (RTE), which takes the effects of absorption, emission, and scattering into account [167]. It describes the evolution of the radiance $I(\boldsymbol{x}, t, \hat{\boldsymbol{s}})$ as function of position $\boldsymbol{x}$, time t, and solid angle $\hat{\boldsymbol{s}}$. The dependence on the solid angle is due to the directionality of general light transport. The RTE reads

$$\frac{1}{l}\frac{\partial I(\boldsymbol{x}, t, \hat{\boldsymbol{s}})}{\partial t} + \hat{\boldsymbol{s}} \cdot \nabla I(\boldsymbol{x}, t, \hat{\boldsymbol{s}}) - (\mu_\mathrm{a} + \mu_\mathrm{s}) I(\boldsymbol{x}, t, \hat{\boldsymbol{s}}) = \mu_\mathrm{s} \int_{4\pi} I(\boldsymbol{x}, t, \hat{\boldsymbol{s}}) P(\hat{\boldsymbol{s}}' \cdot \hat{\boldsymbol{s}}) \mathrm{d}\Theta' + S(\boldsymbol{x}, t, \hat{\boldsymbol{s}}),$$

with the speed of light l, the absorption coefficient μ_a, the scattering coefficient μ_s, the emission source term S, and the probability density function $P(\hat{\boldsymbol{s}}' \cdot \hat{\boldsymbol{s}})$ describing the probability of scattering from direction $\hat{\boldsymbol{s}}'$ to direction $\hat{\boldsymbol{s}}$ [167]. Absorption and scattering coefficient describe the probability for photons to be absorbed or scattered when traveling a unit length, respectively. A common approach to avoid the probabilistic character of the RTE in the context of biomedical optics is the diffusion approximation (DA). The DA is based on the assumption that the propagating light looses its directionality due to sufficiently many scattering events. Decisive parameters to determine the validity of the DA are the absorption coefficient μ_a, the scattering coefficient μ_s, and the parameter g describing the anisotropy of scattering. The reduced scattering coefficient calculates as

$$\mu_\mathrm{s}' = (1 - g)\mu_\mathrm{s}.$$

It is related to the probability density function $P(\hat{\boldsymbol{s}}' \cdot \hat{\boldsymbol{s}})$ and corrects the scattering description if photons are not scattered to arbitrary angles but preferably to a direction similar to the original direction. A rule of thumb for the applicability of the DA in dependence on absorption and scattering coefficient according to [84] is

$$\mu_\mathrm{s}' \geq 10 \cdot \mu_\mathrm{a}. \tag{7.1}$$

This condition requires that the scattering is dominant over the absorption. If this criterion is fulfilled, light loses its directionality after a relative short propagation distance and hence, light transport in an optical medium is reliably described by the diffusion theory. However, the DA can yield unsatisfactory results near sources and near the boundary and results should be scrutinized for their accuracy requirements in these regions. The DA partial differential equation is given by

$$\frac{1}{l}\frac{\partial \phi}{\partial t} + \mu_\mathrm{a}\phi - D\Delta\phi = S.$$

It is derived from the RTE by integration over all directions and assuming direction independent photon movement. The diffusion coefficient D is an abbreviation for

$$D = \frac{1}{3(\mu_\mathrm{a} + \mu_\mathrm{s}')},$$

which is a third of a transport mean free path.

In a typical optoacoustic imaging setup, the illumination by a laser light source is applied for several nanoseconds. Considering the speed of light l and typical length scales of the imaging objects, the steady-state DA is applicable with the illuminating light source represented by a Dirichlet boundary condition. The remainder of the boundary is subject to a Robin boundary condition, which models photons leaving the body without being scattered back. Volume source terms are uncommon in optoacoustic imaging and therefore neglected. Hence, the description of light propagation in biological tissue is given by

$$\mu_\mathrm{a}\phi - D\Delta\phi = 0 \qquad \text{in} \qquad \Omega_\mathrm{L}, \tag{7.2}$$

$$\phi = \hat{\phi} \qquad \text{on} \qquad \Gamma_\mathrm{L}^\mathrm{dir}, \tag{7.3}$$

$$\phi + 2D\nabla\phi \cdot \boldsymbol{n} = 0 \qquad \text{on} \qquad \Gamma_\mathrm{L}^\mathrm{rob}, \tag{7.4}$$

with the optical domain Ω_L, its Dirichlet boundary $\Gamma_\mathrm{L}^\mathrm{dir}$, and Robin boundary $\Gamma_\mathrm{L}^\mathrm{rob}$. As in the previous chapters, the vector $\boldsymbol{n}$ denotes the outward pointing normal vector. In [167], a detailed derivation of the DA from the RTE is given. In [29], a review on the optical properties of biological tissues is given, listing material properties from several references. Representative values for the material parameters in biological tissue are summarized in Table 7.1. The values indicate the applicability of the DA, since most materials fulfill the criterion given by equation (7.1). The values are subject to high variations due to the experimental setup, the species, and of course the natural variations within one tissue type.

Table 7.1: Representative values for optical tissue properties [29].

tissue type	$\mu_\mathrm{a}\ [\frac{1}{\mathrm{cm}}]$	$\mu_\mathrm{s}\ [\frac{1}{\mathrm{cm}}]$	$\mu_\mathrm{s}'\ [\frac{1}{\mathrm{cm}}]$
aorta	$0.52 - 18.10$	$171 - 410$	$25.70 - 69.40$
blood	$1.30 - 4.87$	$505 - 1413$	$2.49 - 6.11$
heart	$0.07 - 0.35$	$136 - 167$	$-$
lung	$8.10 - 25.50$	$35 - 356$	$-$
muscle	$0.12 - 11.20$	$4 - 530$	$1.20 - 8.00$

With equations (7.2)–(7.4) representing a description of light propagation in biological tissue, which is sufficiently accurate in typical optoacoustic imaging setups, the next step is to find a description of the photoacoustic effect. Light heats tissue, especially in regions with high optical absorption. The heating causes a thermal expansion and hence an induction of pressure. The pressure propagates according to the laws of acoustics and is eventually measured with acoustical detectors. For optoacoustic signal generation, the light source must be temporally varying, i.e., pulsed or modulated. A common setup is to use pulsed laser light, and a typical temporal length of a light pulse is several nanoseconds (e.g. $15\,\mathrm{ns}$). This is a comparably long time considering the speed of light and allows to describe the light propagation with the steady-state DA as given in (7.2). Considering heat propagation and pressure propagation however, several nanoseconds is very short and it is assumed that neither heat nor pressure propagate significantly within this time period. This assumption is called thermal and stress confinement [167]. Hence, the pressure p_{t_0} and the velocity $\boldsymbol{v}_{t_0}$ induced by the illumination calculate according to

$$p_{t_0} = -G\mu_\mathrm{a}\phi \qquad \text{in} \qquad \Omega_\mathrm{L}, \tag{7.5}$$
$$p_{t_0} = 0 \qquad \text{in} \qquad \Omega_\mathrm{A} \setminus \Omega_\mathrm{L}\,, \tag{7.6}$$
$$\boldsymbol{v}_{t_0} = \mathbf{0} \qquad \text{in} \qquad \Omega_\mathrm{A}\,, \tag{7.7}$$

with the Grüneisen coefficient G summarizing thermodynamic material properties. The object of interest is generally surrounded by an acoustical coupling medium, either water or ultrasonic gel, to overcome the distance between the object and acoustical transducers with minimal acoustical signal deterioration. The union of the object of interest and the coupling medium is denoted Ω_A. The coupling medium is generally assumed to be transparent $\mu_\mathrm{a} = 0$, which is represented by equation (7.6).

The sound propagates in the object of interest and in the coupling medium according to the conservation of mass, linear momentum, and energy. In optoacoustics, different assumptions on the sound propagation can be made, as already mentioned in Chapter 6. One important modeling step is the consideration of damping effects. Sound propagation can be lossless, i.e., a plane wave travels with constant amplitude, or it can be subject to viscosity, i.e., a plane wave travels with decreasing amplitude and energy is dissipated. In the present model, viscous effects are neglected because the propagation distances are limited. Another common assumption in the context of optoacoustic imaging is acoustic homogeneity, i.e., spatially constant speed of sound and mass density. This assumption prevents consideration of typical effects of sound propagation like reflection, diffraction, or refraction and is rough as the natural variations in soft tissue can already yield reflections with up to 30% signal amplitude [143]. In the presence of bones or air (e.g. in the lung), reflections can even be higher, which is why spatially varying material properties are considered in this work. The lossless propagation of sound in a heterogeneous medium is described by the *acoustic wave equation* derived in equation (2.3) in Chapter 2 and repeated here as first order system

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\rho}\nabla p = 0 \qquad \text{in} \quad \Omega_\mathrm{A} \times [0,T], \tag{7.8}$$
$$\frac{\partial p}{\partial t} + c^2\rho\nabla\cdot\boldsymbol{v} = 0 \quad \text{in} \quad \Omega_\mathrm{A} \times [0,T], \tag{7.9}$$
$$p = p_\mathrm{D} \qquad \text{on} \quad \Gamma_\mathrm{A}^\mathrm{dir} \times [0,T], \tag{7.10}$$

$$\boldsymbol{v} \cdot \boldsymbol{n} = 0 \qquad \text{on} \quad \Gamma_{\mathrm{A}}^{\mathrm{neu}} \times [0, T], \tag{7.11}$$

$$\boldsymbol{v} \cdot \boldsymbol{n} - \frac{1}{c\rho} p = 0 \qquad \text{on} \quad \Gamma_{\mathrm{A}}^{\mathrm{abc}} \times [0, T], \tag{7.12}$$

with Dirichlet, Neumann, and first order absorbing boundary condition. The applicability of the boundary conditions depends on the tomographic setup. For generality, all three types are mentioned at this point.

The last relevant physical process in the optoacoustic signal forming process is the signal detection. The acoustical detectors are located at (a part of) the boundary of the acoustical domain $\Gamma_{\mathrm{A}}^{\mathrm{mon}} \subseteq \partial\Omega_{\mathrm{A}}$. Many different types of acoustic detectors are employed in optoacoustic tomographs. The simplest model of an acoustical detector corresponds to the evaluation of the pressure field in one point and at a specific time $p_{\mathrm{s}}^{\mathrm{d}_j}(t_k) = d(p(\boldsymbol{x}^{\mathrm{d}_j}, t))$, with j numbering the detectors d_j and k numbering the sampling times $k \in [0, n_{\mathrm{D}}^{\mathrm{step}}]$ with the number of sampling times $n_{\mathrm{D}}^{\mathrm{step}}$. A more accurate model considers that detectors are of finite size and evaluate the pressure on a surface by integration or averaging $p_{\mathrm{s}}^{\mathrm{d}_j}(t_k) = d(p(\boldsymbol{x}, t_k))$. Additionally, transducers can be focused, such that the measured pressure also depends on the wave's direction of travel, i.e., the detected pressure values are also a function of the boundary geometry and the velocity vector, $p_{\mathrm{s}}^{\mathrm{d}_j}(t_k) = d(p(\boldsymbol{x}, t_k), \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x}, t_k))$. Another characteristic measure for a detector is its impulse response (IR): a detector is impinged with a Dirac impulse signal and the measurement signals are monitored. The impulse response can be a smeared or scaled Dirac impulse, but it can also include negative values depending on the physical principle the detector is based on. The effects mentioned before (i.e., finite detector size and focusing) can also affect the impulse response. If the impulse response is considered in the function d, the measured value at time t_k depends not only on the current state but also on previous pressure values and the pressure detection function follows $p_{\mathrm{s}}^{\mathrm{d}_j}(t_k) = d(p(\boldsymbol{x}, t), \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x}, t))$. In the model derived in this work, the function d is chosen to represent the pressure evaluation in one point with consideration of the impulse response

$$p_{\mathrm{s}}^{\mathrm{d}_j}(t_k) = d(p(\boldsymbol{x}^{\mathrm{d}_j}, t)), \tag{7.13}$$

with t in the proximity of the evaluation time t_k. Thereby, the directional dependence and the integration over the detector surface are neglected.

In summary, the optoacoustic imaging procedure is modeled by four distinct contributions, namely the light transport, the photoacoustic effect, the sound propagation, and the sound detection. The physical description of the imaging process is summarized as follows:

- **P1** The DA given by equations (7.2)–(7.4) describes the light transport in the object of interest Ω_{L}. The Dirichlet boundary condition represents the applied laser light source.
- **P2** A mapping describes how optical quantities are converted to acoustical quantities by the photoacoustic effect, see equations (7.5)–(7.7).
- **P3** The acoustic wave equation (7.8)–(7.12) describes the sound propagation in a lossless, heterogeneous medium.
- **P4** The function d describes how an acoustical detector measures pressure.

The modeling assumptions are:

M1 Light transport is diffusive ($\mu_\mathrm{s}' \geq 10 \cdot \mu_\mathrm{a}$). This approximation yields errors near boundaries and near source terms.
M2 Heat and sound transport during the illumination are neglected, i.e., the excitation is in thermal and stress confinement.
M3 Damping of acoustic waves due to viscous effects is neglected.
M4 An acoustic detector is assumed to be infinitely small and independent of direction.

From a mathematical viewpoint, the describing equations are complete. In terms of modeling error versus algebraic complexity, the trade-off is reasonable considering the typical noise levels in optoacoustic imaging and typical artifacts, e.g. due to the assumption of acoustic homogeneity.

The evaluation of the optoacoustic model given by equations (7.2)–(7.4) for the optical problem, equations (7.5)–(7.7) for the photoacoustic effect, equations (7.8)–(7.12) for the sound propagation, and equation (7.13) for the detection with given material parameters is denoted as the *forward problem*. The *inverse problem* of image reconstruction concerns the determination of material parameters from pressure measurements.

7.2 Numerical Model

In this section, the numerical treatment of the physical model derived in Section 7.1 will be explained. The occurring physical phenomena are one-way coupled, i.e., they can be solved one after another and hence are easily treated with different numerical methods.

First, the diffusive light transport is considered given by equations (7.2)–(7.4). For spatially varying material parameters μ_a, D, arbitrary boundary conditions, and arbitrary domain geometries Ω_L, analytic solutions are not available and numerical solution strategies are required. Since the light transport is described by an elliptic partial differential equation with reactive and diffusive term, the standard continuous finite element method is suitable to find an approximate solution. The tessellation of the domain Ω_L is denoted $\mathcal{T}_\mathrm{L}^h$. The function spaces for the solutions ϕ and the weighting functions ψ are defined as

$$\Phi = \left\{ \phi \in H^1(\mathcal{T}_\mathrm{L}^h) : \phi = \hat{\phi} \text{ on } \Gamma_\mathrm{L}^\mathrm{dir} \right\},$$
$$\Psi = \left\{ \psi \in H^1(\mathcal{T}_\mathrm{L}^h) : \psi = 0 \text{ on } \Gamma_\mathrm{L}^\mathrm{dir} \right\}.$$

With the notation for domain and boundary integrals as specified in Sections 2.2, the weak form is derived by multiplication of the given problem with the weighting functions and integration by parts

$$(\psi, \mu_\mathrm{a}\phi)_{\mathcal{T}_\mathrm{L}^h} + (\nabla\psi, D\nabla\phi)_{\mathcal{T}_\mathrm{L}^h} + \left\langle \psi, \frac{1}{2}\phi \right\rangle_{\Gamma_\mathrm{L}^\mathrm{rob}} = 0. \tag{7.14}$$

The boundary term is transformed according to the zero weighting functions on the Dirichlet boundary and the given expression on the Robin boundary $\Gamma_\mathrm{L}^\mathrm{rob}$. In contrast to the method derived in Section 2.2, a separate integration over each element boundary is not necessary, because weighting and solution spaces are continuous.

For the discretized problem, the following solution and weighting spaces are defined

$$\Phi_h = \left\{ \phi_h \in \mathcal{S}^1(\mathcal{T}_\mathrm{L}^h) : \phi_h = \mathsf{P}\hat{\phi} \ \text{ on } \ \Gamma_\mathrm{L}^\mathrm{dir} \right\},$$
$$\Psi_h = \left\{ \psi_h \in \mathcal{S}^1(\mathcal{T}_\mathrm{L}^h) : \psi_h = 0 \ \text{ on } \ \Gamma_\mathrm{L}^\mathrm{dir} \right\},$$

where $\mathcal{S}_1(\mathcal{T}_\mathrm{L}^h)$ denotes the finite element space for linear finite elements associated with the triangulation $\mathcal{T}_\mathrm{L}^h$. The space $\mathcal{S}_1(\mathcal{T}_\mathrm{L}^h)$ is $n_\mathrm{L}^\mathrm{dof}$-dimensional and the solution ϕ_h is spanned by the basis functions associated to the nodes of the mesh. The values of the degrees of freedom scaling the basis functions are summarized in the vector $\boldsymbol{\Phi}$, which is again of length $n_\mathrm{L}^\mathrm{dof}$ and analogous for the field ψ_h with the values summarized in $\boldsymbol{\Psi}$. The material parameters μ_a, D, and G are potentially spatially varying. They are discretized as element-wise constant $\mu_{\mathrm{a}h}$, D_h, and G_h and the describing coefficients are summarized in the vectors $\boldsymbol{\mu}_\mathrm{a}$, $\boldsymbol{D}$, and $\boldsymbol{G}$. With these notations and assumptions, the discretized weak form in matrix notation reads

$$\boldsymbol{\Psi}^\mathrm{T} \left(\mathbb{K}_\mathrm{L} \boldsymbol{\Phi} - \boldsymbol{F}_\mathrm{L} \right) = 0.$$

The vector $\boldsymbol{F}_\mathrm{L}$ results from the non-zero Dirichlet boundary values and $\mathbb{K}_\mathrm{L}$ denotes the assembled system matrix stemming from the three terms in equation (7.14) in discretized fashion. Obviously, the weighting function values are arbitrary and the system

$$\mathbb{K}_\mathrm{L} \boldsymbol{\Phi} = \boldsymbol{F}_\mathrm{L}$$

must be solved to obtain the approximate solution to the light transport problem. The numerical cost to solve this system depends mainly on the size of the system matrix $\mathbb{K}_\mathrm{L}$ and hence the number of nodes in the tessellation $\mathcal{T}_\mathrm{L}^h$.

Next, a discretization of the equations describing the photoacoustic energy conversion is sought. Equations (7.6) and (7.7) are trivial. Only equation (7.5) requires special attention. For all elements K of the tessellation $\mathcal{T}_\mathrm{A}^h$ of the acoustical domain that have an intersection with the optical tessellation $K \cap \mathcal{T}_\mathrm{L}^h \neq \emptyset$, the following L_2 projection is defined for the discretized pressure $p_{h,t_0} \in P_h$ as in Section 2.2,

$$\left(q_{h,t_0}, p_{h,t_0} \right)_K = - \left(q_{h,t_0}, G_h \mu_{\mathrm{a}h} \phi_h \right)_K \qquad \forall q_{h,t_0} \in P_h. \tag{7.15}$$

This mapping requires the evaluation of the fields G_h, $\mu_{\mathrm{a}h}$, and ϕ_h in the quadrature points of the acoustical element K. In case a given quadrature point does not lie in $\mathcal{T}_\mathrm{L}^h$, no energy is disposed because no absorption is present and the product $G_h \mu_{\mathrm{a}h} \phi_h$ is zero. The projection can analogously be written in matrix form

$$\boldsymbol{Q}_{t_0}^\mathrm{T} \left(\mathbb{K}_\mathrm{A,PA} \boldsymbol{P}_{t_0} - \mathbb{K}_\mathrm{L,PA} \boldsymbol{\Phi} \right) = 0$$

with arbitrary weighting, such that the initial pressure field is determined by solution of the system

$$\mathbb{K}_\mathrm{A,PA} \boldsymbol{P}_{t_0} = \mathbb{K}_\mathrm{L,PA} \boldsymbol{\Phi}.$$

The matrix $\mathbb{K}_{\text{A,PA}}$ is a mass matrix on the acoustical mesh while $\mathbb{K}_{\text{L,PA}}$ combines weighting functions based on the acoustical discretiztaion with an evaluation of a field defined on the optical discretization.

For the description of efficient solution strategies of the acoustic wave equation, the reader is referred to Chapters 2 and 3 of this work. Here, only an abbreviated notation shall be introduced generalizing the different discretization approaches presented previously.

The discretization of the acoustic wave equation (7.8)–(7.12) with HDG for spatial discretization and explicit Runge–Kutta schemes or ADER for temporal discretization is generally given as

$$\begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \\ \boldsymbol{M}_{t_{i+1}} \end{bmatrix}^{\mathrm{T}} \left(\mathbb{K}^{\text{ac}}_{\text{A,M}} \begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \\ \boldsymbol{\Lambda}_{t_{i+1}} \end{bmatrix} - \mathbb{K}^{\text{ac}}_{\text{A,F}} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \\ \boldsymbol{\Lambda}_{t_i} \end{bmatrix} \right) = 0 \qquad \forall\, i = 0 \text{ to } i+1 = n^{\text{step}}_{\text{A}},$$

where the specific discretization determines the acoustical matrices $\mathbb{K}^{\text{ac}}_{\text{A,M}}$ and $\mathbb{K}^{\text{ac}}_{\text{A,F}}$ scaling the unknowns and the quantities from the preceding time step, respectively.

The last discretization to be specified is the discretization of the detection, which is given by

$$p_{\text{s}}^{\text{d}_j}(t_k) = d(p(\boldsymbol{x}^{\text{d}_j}, t)).$$

A general discretization of this equation is given by

$$\mathbb{K}_{\text{D}} \boldsymbol{P}_{\text{s},t_k} = \boldsymbol{F}^k_{\text{D}} \left(\boldsymbol{V}_l, \boldsymbol{P}_l \right) \qquad \forall k = 0, ..., n^{\text{step}}_{\text{D}} \quad \text{with} \quad l = 0, ..., n^{\text{step}}_{\text{A}}, \tag{7.16}$$

with the vector valued function $\boldsymbol{F}_{\text{D}}$ taking into account the impulse response of the detectors.

7.3 Objective Function

The experiments with an optoacoustic tomograph yield time-sampled pressure curves from several detectors. The values from all detectors for the time points t_k are summarized in the k vectors $\boldsymbol{P}_{\text{m},t_k}$. The simulated pressure curves are calculated according to equation (7.16) and the preceding equations and are summarized in the k vectors $\boldsymbol{P}_{\text{s},t_k}$. The difference between the measured and simulated pressure values determines the error e

$$e = \frac{1}{2} \sum_{k=0}^{n^{\text{step}}_{\text{D}}} \left\| \boldsymbol{P}_{\text{m},t_k} - \boldsymbol{P}_{\text{s},t_k} \right\|^2 .$$

The objective for solving the inverse problem of optoacoustic imaging is to minimize the difference between measured and simulated pressure values by adapting the discretized material parameters

$$\min_{\boldsymbol{\mu}_{\text{a}}, \boldsymbol{D}, \boldsymbol{c}, \boldsymbol{\rho}} e. \tag{7.17}$$

Since this is an inverse problem that suffers from ill-conditioning, a regularization in form of a Tikhonov regularization r_{Tikh} or a total variation regularization r_{TV} can be added to the error and the objective function is defined as

$$J = e + r_{\text{TV}} + r_{\text{Tikh}} =: e + r, \tag{7.18}$$

with

$$\begin{aligned} r_{\text{TV}} &= \frac{1}{2}\omega_{\text{TV}}^{\mu_\text{a}}\|\nabla_{\text{TV}}\boldsymbol{\mu}_\text{a}\|^2 + \frac{1}{2}\omega_{\text{TV}}^{D}\|\nabla_{\text{TV}}\boldsymbol{D}\|^2 + \frac{1}{2}\omega_{\text{TV}}^{c}\|\nabla_{\text{TV}}\boldsymbol{c}\|^2 + \frac{1}{2}\omega_{\text{TV}}^{\rho}\|\nabla_{\text{TV}}\boldsymbol{\rho}\|^2, \\ r_{\text{Tikh}} &= \frac{1}{2}\omega_{\text{Tikh}}^{\mu_\text{a}}\|\boldsymbol{\mu}_\text{a}\|^2 + \frac{1}{2}\omega_{\text{Tikh}}^{D}\|\boldsymbol{D}\|^2 + \frac{1}{2}\omega_{\text{Tikh}}^{c}\|\boldsymbol{c}\|^2 + \frac{1}{2}\omega_{\text{Tikh}}^{\rho}\|\boldsymbol{\rho}\|^2. \end{aligned}$$

and ∇_{TV} approximating the gradient on the discretized quantities. The weights $\omega_{\text{TV,Tikh}}^{\mu_\text{a},D,c,\rho}$ are user-defined. The regularized objective is to minimize

$$\min_{\boldsymbol{\mu}_\text{a},\boldsymbol{D},\boldsymbol{c},\boldsymbol{\rho}} J, \tag{7.19}$$

in contrast to minimization of the error as in equation (7.17). In case all regularization weights $\omega_{\text{TV,Tikh}}^{\mu_\text{a},D,c,\rho}$ are zero, equations (7.17) and (7.18) are equivalent.

The problem given in equation (7.19) is denoted an *inverse problem*, because material parameters are sought for given geometry, boundary conditions, and measurement data. The solution of the correspondent *forward problem* as introduced in Section 7.1 is only a subproblem.

7.4 Parameter Gradients

The minimization of the objective function J requires information about the dependence on the material parameters, i.e., the gradients of the objective function with respect to the material parameters are sought

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_\text{a}} \coloneqq \boldsymbol{g}_{\mu_\text{a}}, \qquad \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{D}} \coloneqq \boldsymbol{g}_{D}, \qquad \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{c}} \coloneqq \boldsymbol{g}_{c}, \qquad \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\rho}} \coloneqq \boldsymbol{g}_{\rho}.$$

The dependency of J on the material parameters is strongly nonlinear. By changing one material parameter in one element, all solution fields are altered. To enable the evaluation of the gradients, a Lagrangian functional $\mathcal{L}$ is constructed such that its partial derivatives with respect to the material parameters equal the absolute parameter derivatives of the objective function

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}_\text{a}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_\text{a}}, \qquad \frac{\partial\mathcal{L}}{\partial\boldsymbol{D}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{D}}, \qquad \frac{\partial\mathcal{L}}{\partial\boldsymbol{c}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{c}}, \qquad \frac{\partial\mathcal{L}}{\partial\boldsymbol{\rho}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\rho}}. \tag{7.20}$$

The Lagrangian is given by the sum of the objective function itself and all discretized weak forms of the optoacoustic imaging model

$$\begin{aligned} \mathcal{L} &= J && (7.21)\\ &+ \boldsymbol{\Psi}^\text{T}\left(\mathbb{K}_\text{L}\boldsymbol{\Phi} - \boldsymbol{F}_\text{L}\right) && (7.22)\\ &+ \boldsymbol{Q}_{t_0}^\text{T}\left(\mathbb{K}_{\text{A,PA}}\boldsymbol{P}_{t_0} - \mathbb{K}_{\text{L,PA}}\boldsymbol{\Phi}\right) && (7.23)\\ &+ \sum_{i=0}^{n_\text{A}^\text{step}-1} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \\ \boldsymbol{M}_{t_{i+1}} \end{bmatrix}^\text{T} \left(\mathbb{K}_{\text{A,M}}^{\text{ac}} \begin{bmatrix} \boldsymbol{V}_{t_{i+1}} \\ \boldsymbol{P}_{t_{i+1}} \\ \boldsymbol{\Lambda}_{t_{i+1}} \end{bmatrix} - \mathbb{K}_{\text{A,F}}^{\text{ac}} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \\ \boldsymbol{\Lambda}_{t_i} \end{bmatrix} \right) && (7.24)\\ &+ \sum_{k=0}^{n_\text{D}^\text{step}} \boldsymbol{Q}_{\text{s},t_k}^\text{T}\left(\mathbb{K}_\text{D}\boldsymbol{P}_{\text{s},t_k} - \boldsymbol{F}_\text{D}^k\left(\boldsymbol{V}_l, \boldsymbol{P}_l\right)\right). && (7.25) \end{aligned}$$

Therein, the discretized weighting functions act as Lagrange multipliers to enforce the physical problem as constraint for the minimization of the objective function. As long as the solution variables solve the systems of equations, the residuals of the weak forms are zero, and the Lagrangian reduces to $\mathcal{L} = J$. For the derivatives however, the correlation is not as obvious. To show that the gradient relations of equation (7.20) are fulfilled, an abbreviated notation is introduced. The values of the degrees of freedom of all solution fields ($\boldsymbol{\Phi}, \boldsymbol{V}_{t_i}, \boldsymbol{P}_{t_i}, \boldsymbol{\Lambda}_{t_i}, \boldsymbol{P}_{\mathrm{s},t_k}$) are summarized in the vector $\boldsymbol{Z}$ and the corresponding values of the weighting degrees of freedom ($\boldsymbol{\Psi}, \boldsymbol{W}_{t_i}, \boldsymbol{Q}_{t_i}, \boldsymbol{M}_{t_i}, \boldsymbol{Q}_{\mathrm{s},t_k}$) are summarized in $\boldsymbol{Y}$. The values describing the discretized parameter fields ($\boldsymbol{\mu}_\mathrm{a}, \boldsymbol{D}, \boldsymbol{c}, \boldsymbol{\rho}$) are contained in the vector $\boldsymbol{m}$. With these abbreviations, the Lagrangian is compactly written as

$$\mathcal{L}(\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{m}) = J(\boldsymbol{Z}, \boldsymbol{m}) + \mathcal{W}(\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{m})$$

explicitly stating the dependencies and $\mathcal{W}$ representing all discretized weak forms with

$$\mathcal{W} = \boldsymbol{Y}^\mathrm{T} \left(\mathbb{K}_\mathcal{W} \boldsymbol{Z} - \boldsymbol{F} \right) = 0.$$

The total derivative of the Lagrangian with respect to the material parameters calculates as

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{m}} &= \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{m}} + \frac{\mathrm{d}\mathcal{W}}{\mathrm{d}\boldsymbol{m}} \\
&= \frac{\partial J}{\partial \boldsymbol{m}} + \frac{\partial J}{\partial \boldsymbol{Z}} \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Y}} \frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}\boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Z}} \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}} \\
&= \frac{\partial J}{\partial \boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Y}} \frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}\boldsymbol{m}} + \left(\frac{\partial J}{\partial \boldsymbol{Z}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Z}} \right) \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}} \\
&= \frac{\partial \mathcal{L}}{\partial \boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Y}} \frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}\boldsymbol{m}} + \left(\frac{\partial J}{\partial \boldsymbol{Z}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Z}} \right) \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}}.
\end{aligned}$$

From the first to the second line, the derivatives are expanded. From the second to the third line, the terms are reordered and last, the partial derivative terms are summarized as partial derivative of the Langrangian itself. Using the right hand side of the first line and the last line, the equality

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{m}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{m}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Y}} \frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}\boldsymbol{m}} + \left(\frac{\partial J}{\partial \boldsymbol{Z}} + \frac{\partial \mathcal{W}}{\partial \boldsymbol{Z}} \right) \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}} - \frac{\mathrm{d}\mathcal{W}}{\mathrm{d}\boldsymbol{m}} \tag{7.26}$$

follows. In case the last three terms on the right hand side are shown to be zero, the equalities from equations (7.20) are confirmed. In the following, the three terms are examined one after another.

The derivative of the weighting degrees of freedom with respect to the material parameters $\mathrm{d}\boldsymbol{Y}/\mathrm{d}\boldsymbol{m}$ is zero, because the weighting is arbitrary and does not depend on the material parameters. The derivative of the entire weak form with respect to the material parameters $\mathrm{d}\mathcal{W}/\mathrm{d}\boldsymbol{m}$ is zero as well, because the weak form must be zero even for disturbed material parameters $\boldsymbol{m} + \boldsymbol{\delta}$:

$$\begin{aligned}
&\mathcal{W}(\boldsymbol{Y}, \boldsymbol{Z}(\boldsymbol{m}), \boldsymbol{m}) = 0 \\
&\mathcal{W}(\boldsymbol{Y}, \boldsymbol{Z}(\boldsymbol{m} + \boldsymbol{\delta}), \boldsymbol{m} + \boldsymbol{\delta}) = 0
\end{aligned}$$

Evaluation of the limit of the difference quotient defining a derivative reveals that

$$\begin{aligned}\frac{\mathrm{d}\mathcal{W}(\boldsymbol{Y},\boldsymbol{Z}(\boldsymbol{m}),\boldsymbol{m})}{\mathrm{d}\boldsymbol{m}} &= \lim_{|\boldsymbol{\delta}|\to 0}\frac{\mathcal{W}(\boldsymbol{Y},\boldsymbol{Z}(\boldsymbol{m}+\boldsymbol{\delta}),\boldsymbol{m}+\boldsymbol{\delta})-\mathcal{W}(\boldsymbol{Y},\boldsymbol{Z}(\boldsymbol{m}),\boldsymbol{m})}{\boldsymbol{\delta}}\\ &= \lim_{|\boldsymbol{\delta}|\to 0}\frac{0-0}{\boldsymbol{\delta}} = 0.\end{aligned}$$

Hence, the last term in equation (7.26) vanishes. Last, the term

$$\underbrace{\left(\frac{\partial J}{\partial \boldsymbol{Z}}+\frac{\partial \mathcal{W}}{\partial \boldsymbol{Z}}\right)}_{(*)}\frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}\boldsymbol{m}} = 0$$

must be zero. The derivative of the solution values with respect to the material parameters is not zero, since the solution generally changes when the material parameters change. Consequently, the contribution $(*)$ must vanish in order to make the product equal to zero. The term $(*)$ is denoted the *adjoint problem*. The weak form partially derived with respect to the solution values gives rise to a discrete problem in terms of the weighting functions with the derivative of the objective function as source term. For the forward model derived in the preceding sections, the adjoint problem reads

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{\mathrm{s},t_k}} \qquad \to \mathbb{K}_{\mathrm{D}}^{\mathrm{T}}\boldsymbol{Q}_{\mathrm{s},t_k} = \left(\boldsymbol{P}_{\mathrm{s},t_k}-\boldsymbol{P}_{\mathrm{m},t_k}\right), \tag{7.27}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{t_{\mathrm{A}}^{\mathrm{step}}}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{t_{\mathrm{A}}^{\mathrm{step}}}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Lambda}_{t_{\mathrm{A}}^{\mathrm{step}}}} \to {\mathbb{K}_{\mathrm{A,M}}^{\mathrm{ac}}}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{W}_{t_{\mathrm{A}}^{\mathrm{step}}}\\ \boldsymbol{Q}_{t_{\mathrm{A}}^{\mathrm{step}}}\\ \boldsymbol{M}_{t_{\mathrm{A}}^{\mathrm{step}}}\end{bmatrix} = \sum_{k=0}^{n_{\mathrm{D}}^{\mathrm{step}}}\begin{bmatrix}\left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{V}_{t_i}}\right)^{\mathrm{T}}\\ \left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{P}_{t_i}}\right)^{\mathrm{T}}\\ \mathbf{0}\end{bmatrix}\boldsymbol{Q}_{\mathrm{s},t_k}, \tag{7.28}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{t_i}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{t_i}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Lambda}_{t_i}} \to {\mathbb{K}_{\mathrm{A,M}}^{\mathrm{ac}}}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{W}_{t_i}\\ \boldsymbol{Q}_{t_i}\\ \boldsymbol{M}_{t_i}\end{bmatrix} = {\mathbb{K}_{\mathrm{A,F}}^{\mathrm{ac}}}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{W}_{t_{i+1}}\\ \boldsymbol{Q}_{t_{i+1}}\\ \boldsymbol{M}_{t_{i+1}}\end{bmatrix} + \sum_{k=0}^{n_{\mathrm{D}}^{\mathrm{step}}}\begin{bmatrix}\left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{V}_{t_i}}\right)^{\mathrm{T}}\\ \left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{P}_{t_i}}\right)^{\mathrm{T}}\\ \mathbf{0}\end{bmatrix}\boldsymbol{Q}_{\mathrm{s},t_k},$$
$$\forall i = n_{\mathrm{A}}^{\mathrm{step}}-1 \ \ldots \ 1, \tag{7.29}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{V}_{t_0}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{P}_{t_0}},\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Lambda}_{t_0}} \to \mathbb{K}_{\mathrm{A,PA}}^{\mathrm{T}}\boldsymbol{Q}_{t_0} = {\mathbb{K}_{\mathrm{A,F}}^{\mathrm{ac}}}^{\mathrm{T}}\begin{bmatrix}\boldsymbol{W}_{t_1}\\ \boldsymbol{Q}_{t_1}\\ \boldsymbol{M}_{t_1}\end{bmatrix} + \sum_{k=0}^{n_{\mathrm{D}}^{\mathrm{step}}}\begin{bmatrix}\left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{V}_{t_0}}\right)^{\mathrm{T}}\\ \left(\frac{\partial \boldsymbol{F}_{\mathrm{D}}^{k}}{\partial \boldsymbol{P}_{t_0}}\right)^{\mathrm{T}}\\ \mathbf{0}\end{bmatrix}\boldsymbol{Q}_{\mathrm{s},t_k}, \tag{7.30}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Phi}} \qquad \to \mathbb{K}_{\mathrm{L}}^{\mathrm{T}}\boldsymbol{\Psi} = \mathbb{K}_{\mathrm{L,PA}}^{\mathrm{T}}\boldsymbol{Q}_{t_0}. \tag{7.31}$$

The order in which these equations are written down is a guideline for the solution procedure of the adjoint problem. The adjoint problem is solved in reverse order. First, the adjoint measurement values $\boldsymbol{Q}_{\mathrm{s},t_k}$ are evaluated from simulated and measured pressure values. Next, the adjoint

wave equation problem is considered, which is solved backwards in time: the 'initial conditions' for the last time step are set according to the adjoint detector function. Then, the adjoint sound propagation is solved backwards in time until an adjoint pressure at time step zero $\boldsymbol{Q}_{t_0}$ is calculated and used in order to calculate the adjoint photoacoustic effect. Last, the adjoint light flux $\boldsymbol{\Psi}$ is determined.

With weighting functions fulfilling equations (7.27)–(7.31), the third term on the right hand side of equation (7.26) vanishes and the desired link (7.20) is actually true such that the objective function gradients are straightforwardly evaluated from the Lagrangian by partial differentiation:

$$\boldsymbol{g}_{\mu_\mathrm{a}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_\mathrm{a}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_\mathrm{a}} = \frac{\partial r}{\partial \boldsymbol{\mu}_\mathrm{a}} + \boldsymbol{\Psi}^\mathrm{T} \frac{\partial \mathbb{K}_\mathrm{L}}{\partial \boldsymbol{\mu}_\mathrm{a}} \boldsymbol{\Phi} + \boldsymbol{Q}_{t_0}^\mathrm{T} \frac{\partial \mathbb{K}_\mathrm{L,PA}}{\partial \boldsymbol{\mu}_\mathrm{a}} \boldsymbol{\Phi} \tag{7.32}$$

$$\boldsymbol{g}_{D} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{D}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{D}} = \frac{\partial r}{\partial \boldsymbol{D}} + \boldsymbol{\Psi}^\mathrm{T} \frac{\partial \mathbb{K}_\mathrm{L}}{\partial \boldsymbol{D}} \boldsymbol{\Phi} \tag{7.33}$$

$$\boldsymbol{g}_{c} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{c}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{c}} = \frac{\partial r}{\partial \boldsymbol{c}} - \sum_{i=0}^{n_\mathrm{A}^\mathrm{step}-1} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \\ \boldsymbol{M}_{t_{i+1}} \end{bmatrix}^\mathrm{T} \frac{\partial \mathbb{K}_\mathrm{A,F}^\mathrm{ac}}{\partial \boldsymbol{c}} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \\ \boldsymbol{\Lambda}_{t_i} \end{bmatrix} \tag{7.34}$$

$$\boldsymbol{g}_{\rho} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\rho}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} = \frac{\partial r}{\partial \boldsymbol{\rho}} - \sum_{i=0}^{n_\mathrm{A}^\mathrm{step}-1} \begin{bmatrix} \boldsymbol{W}_{t_{i+1}} \\ \boldsymbol{Q}_{t_{i+1}} \\ \boldsymbol{M}_{t_{i+1}} \end{bmatrix}^\mathrm{T} \frac{\partial \mathbb{K}_\mathrm{A,F}^\mathrm{ac}}{\partial \boldsymbol{\rho}} \begin{bmatrix} \boldsymbol{V}_{t_i} \\ \boldsymbol{P}_{t_i} \\ \boldsymbol{\Lambda}_{t_i} \end{bmatrix} \tag{7.35}$$

In case regularization is enabled, the objective function depends on the material parameter itself and this has to be considered in the gradients. The remaining contributions stem from the weak form. For the acoustical gradients, solution vectors of the forward and adjoint problem from all time steps have to be summed, which is explained in Section 7.5.2.

The gradients of the objective function as listed above could alternatively be evaluated using a finite difference approach. For each value in the material vectors, one evaluation of the entire forward problem with a disturbed material value has to be carried out to determine the correspondent entry in the gradient vector. The computational expense relates to the number of material parameters n_param and the solution of $n_\mathrm{param} + 1$ forward problems is required. Evaluation of the gradients by solution of one forward problem and one adjoint problem is as expensive as two evaluations of the forward problem. Calculation of the gradients with the adjoint approach hence is $\frac{n_\mathrm{param}+1}{2}$ times faster compared to finite differences.

7.5 Solution Algorithm

To solve the optimization problem given by equation (7.19), gradient based solution strategies are chosen. Depending on the application and problem setup, the user can choose between a gradient descent or a low-storage BFGS procedure. Also, the user can chose which of the material parameters should be optimized. The algorithmic framework allows for the optimization of all four material parameters $\mu_\mathrm{a}, D, c, \rho$. For specific applications however, the optimization of the diffusion coefficient D or the mass density ρ is not needed.

Input parameters for the solution algorithm are an optical as well as an acoustical discretization. For the optical problem, this concerns the mesh consisting of elements and nodes. For the acoustical problem, this concerns also the mesh, the polynomial degree of the shape functions,

the time integration scheme, and the time step. Boundary conditions for the optical domain (a Dirichlet boundary condition where the laser is applied, and Robin boundary conditions for the remainder) and boundary conditions for the acoustical domain (ABCs, PMLs, reflecting Neumann or Dirichlet boundary conditions) must be specified as input. Most important is the input of initial material parameters for $\mu_\mathrm{a}, D, c, \rho$, which are subsequently optimized. A reasonable choice of initial material parameters can reduce the overall computational time significantly and can even change the result, since the objective function generally has more than one local minimum. Additionally, the user has to specify the maximum number of global iterations as well as the number of sequential operations, i.e., the number of iterations to optimize the absorption coefficient, diffusion coefficient, speed of sound, and mass density, separately, denoted by $i_{\mu_\mathrm{a}}^{\mathrm{seq}}, i_D^{\mathrm{seq}}, i_c^{\mathrm{seq}}, i_\rho^{\mathrm{seq}}$. A tolerance must be specified to check convergence by comparison of the tolerance to the objective function. With all input parameters available, the minimization problem is solved as shown in Algorithm 4. After the initialization phase, the entire forward problem is solved and the objective function is evaluated. Then, the entire adjoint problem is solved and the required gradients are evaluated. The optimization loop is entered. It consists of four subproblems, each looping the line search for one material parameter for the given number of sequential iterations. It is important to note that the absorption coefficient is optimized first, followed by the speed of sound, mass density, and finally diffusion coefficient. This order is according to the general sensitivities of the parameters from high to low. The overall optimization loop is repeated until convergence is achieved or the maximal number of iterations is reached.

Note that an optimization procedure summarizing all material parameters and all gradients to optimize them altogether in one line search is not reasonable, because they scale differently and because they have different sensitivities.

Algorithm 4 Global solution algorithm

```
solution of the forward problem
evaluation of the objective function J = e + r
solution of the adjoint problem
evaluation of the gradients g_{μa}, g_D, g_c, g_ρ
repeat
  for i = 0, i < i_{μa}^seq do
    run line search to update the absorption coefficient μa
  end for
  for i = 0, i < i_c^seq do
    run line search to update the speed of sound c
  end for
  for i = 0, i < i_ρ^seq do
    run line search to update the mass density ρ
  end for
  for i = 0, i < i_D^seq do
    run line search to update the diffusion coefficient D
  end for
until convergence or maximal number of iterations is reached
```

7.5.1 Line Search

As can be seen from Algorithm 4, a line search method is required. The version chosen herein is based on the description in [121], Chapter 3.5. Here, the main aspects of the line search are repeated. A line search procedure is concerned with the optimization problem along a given direction $\boldsymbol{d}$, i.e., to find the step length β such that the update of the quantity to be optimized (here exemplary for the absorption coefficient)

$$\boldsymbol{\mu}_{\mathrm{a}}^{l} = \boldsymbol{\mu}_{\mathrm{a}} + \beta^{l} \boldsymbol{d}_{\mu_{\mathrm{a}}}$$

yields a sufficient progress for the optimization problem. The Wolfe conditions determine if the progress is sufficient. The first Wolfe conditions requires a sufficient decrease of the objective function

$$J(\boldsymbol{\mu}_{\mathrm{a}}^{l+1}) \leq J(\boldsymbol{\mu}_{\mathrm{a}}) + c_1 \beta^{l} \boldsymbol{g}_{\mu_{\mathrm{a}}}^{l} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}},$$

where l denotes the line search iteration. The coefficient c_1 is a user specified constant $c_1 \in (0, 1)$ that is usually chosen small. Since the directional derivative $\boldsymbol{g}_{\mu_{\mathrm{a}}} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}^{l}$ is negative (for a gradient descent approach it is $\boldsymbol{d}_{\mu_{\mathrm{a}}}^{l} = -\boldsymbol{g}_{\mu_{\mathrm{a}}}$), the stated condition requires a decrease that is proportional to the step length as well as the directional derivative. It is also denoted Armijo condition. It is easily fulfilled for small step lengths. To prevent too short steps and obtain sufficient progress, the second Wolfe condition (also denoted as curvature condition) tests the new gradient

$$\boldsymbol{g}_{\mu_{\mathrm{a}}}^{l+1} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}} \geq c_2 \boldsymbol{g}_{\mu_{\mathrm{a}}}^{l} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}},$$

with the user specified constant $c_2 \in (c_1, 1)$. This condition requires the new gradient to be less steep. To enforce beneficial updates, the second Wolfe condition is formulated in strong form

$$|\boldsymbol{g}_{\mu_{\mathrm{a}}}^{l+1} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}| \leq |c_2 \boldsymbol{g}_{\mu_{\mathrm{a}}}^{l} \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}|.$$

An update fulfilling the Wolfe conditions does not necessarily yield a minimizer. Only the repeated update with each update fulfilling the Wolfe conditions can result in a parameter distribution minimizing the objective function along the direction $\boldsymbol{d}_{\mu_{\mathrm{a}}}$. The line search procedure to determine a step length β^{*} fulfilling the strong Wolfe conditions is shown in Algorithm 5. It starts with the choice of an initial step length. If a gradient descent scheme is used, the initial step length depends on the scaling of the parameters. If low-storage BFGS is used, the initial step length is set to one. The line search contains a loop in which the forward problem and the objective function are evaluated for the updated parameters. The first Wolfe condition is checked and the zoom function is called if the condition is violated because this corresponds to a too long step. If the first Wolfe condition is fulfilled, the adjoint problem and the gradient are evaluated to enable verification of the second Wolfe condition. If it is fulfilled, a suitable step length is found. If the directional derivative is positive, the step length is too long and is reduced in the zoom function. The line search terminates after a user defined maximal number of iterations and checks the success of the step length determination. The line search procedure makes use of the zoom function as specified in Algorithm 6. It is called as soon as the general line search procedure sets a too long step length. Its structure resembles the general line search except that it can additionally change the direction of search.

Algorithm 5 Line search

set initial step length β^0
for $l = 0, l < l^{\max}$ **do**
  $\boldsymbol{\mu}_{\mathrm{a}}^l = \boldsymbol{\mu}_{\mathrm{a}} + \beta^l \boldsymbol{d}_{\mu_{\mathrm{a}}}$
  solution of the forward problem
  evaluation of the objective function J^l
  if $J^l > J^0 + c_1 \beta^l \boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}$ **then**
    determine β^* by call of zoom with $\beta^{\min} = \beta^{l-1}, \beta^{\max} = \beta^l$
    break
  end if
  solution of the adjoint problem
  evaluation of the gradient $\boldsymbol{g}_{\mu_{\mathrm{a}}}^l$
  if $|\boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}| \leq |c_2 \boldsymbol{g}_{\mu_{\mathrm{a}}}^0 \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}|$ **then**
    $\beta^* = \beta^l$
    break
  end if
  if $\boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}} \geq 0$ **then**
    determine β^* by call of zoom $\beta^{\min} = \beta^l, \beta^{\max} = \beta^{l-1}$
    break
  end if
  $\beta^{l+1} = 2 \cdot \beta^l$
end for
if Wolfe conditions are fulfilled **then**
  $\beta^* = \beta^l$
else
  print an error message
end if

7.5.2 Checkpointing

The objective function gradients with respect to the acoustical parameters require the combination of forward and adjoint solutions in every time step, as can be seen in equations (7.34) and (7.35). A naive implementation would store all vectors $\boldsymbol{V}_{t_i}, \boldsymbol{P}_{t_i}$ from the forward run to combine them with the adjoint solutions during the adjoint run. This, however, requires the storage of $n_{\mathrm{A}}^{\mathrm{step}}$ velocity and pressure solution vectors, which is not possible for a realistic image reconstruction simulation and standard working memory size. To avoid the storage of all solution vectors, a checkpointing strategy is utilized as proposed in [64] and shown in Algorithm 7. The idea is to write restarts every n^{check} steps during the forward solution and repeat the solution of the forward problem during the adjoint run. Thereby, the required storage reduces from $n_{\mathrm{A}}^{\mathrm{step}}$ solution vectors to n^{check} solution vectors, see Figure 7.1. The drawback is that the forward problem is solved twice, once in the standard solve and then again during the adjoint run, which increases computational expenses. Compared to a finite difference approach, the calculation of the gradients using the adjoint approach with checkpointing is still $\frac{n_{\mathrm{param}}+1}{3}$ times faster.

Algorithm 6 Zoom

read input parameters $\beta^{\min}$ and $\beta^{\max}$
for $l = 0$, $l < l^{\max}$ **do**
  choose $\beta^l \in (\beta^{\min}, \beta^{\max})$
  $\boldsymbol{\mu}_{\mathrm{a}}^l = \boldsymbol{\mu}_{\mathrm{a}} + \beta^l \boldsymbol{d}_{\mu_{\mathrm{a}}}$
  solution of the forward problem
  evaluation of the objective function J^l
  if $J^l > J^0 + c_1 \beta^l \boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}$ **then**
    $\beta^{\max} = \beta^l$
  else
    solution of the adjoint problem
    evaluation of the gradient $\boldsymbol{g}_{\mu_{\mathrm{a}}}^l$
    if $|\boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}| \leq |c_2 \boldsymbol{g}_{\mu_{\mathrm{a}}}^0 \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}}|$ **then**
      $\beta^* = \beta^l$
      break
    end if
    if $(\boldsymbol{g}_{\mu_{\mathrm{a}}}^l \cdot \boldsymbol{d}_{\mu_{\mathrm{a}}})(\beta^{\max} - \beta^{\min}) \geq 0$ **then**
      $\beta^{\max} = \beta^{\min}$
    end if
    $\beta^{\min} = \beta^l$
  end if
end for
$\beta^* = \beta^l$

Algorithm 7 Adjoint acoustical problem with checkpointing

for $i = 0, i \leq n_{\mathrm{A}}^{\mathrm{step}}$ **do**
  if $i \bmod n^{\mathrm{check}} = 0$ **then**
    store current adjoint solution vectors
    read solution vectors of forward problem from restart $n_{\mathrm{A}}^{\mathrm{step}} - i - n^{\mathrm{check}}$
    for $j = 0, j \leq n^{\mathrm{check}}$ **do**
      solve forward step $n_{\mathrm{A}}^{\mathrm{step}} - i - n^{\mathrm{check}} + j$
      store j-th forward solution vector
    end for
  end if
  solve $(n_{\mathrm{A}}^{\mathrm{step}} - i)$-th adjoint step
  combine forward and addjoint quantities and add to gradients
end for

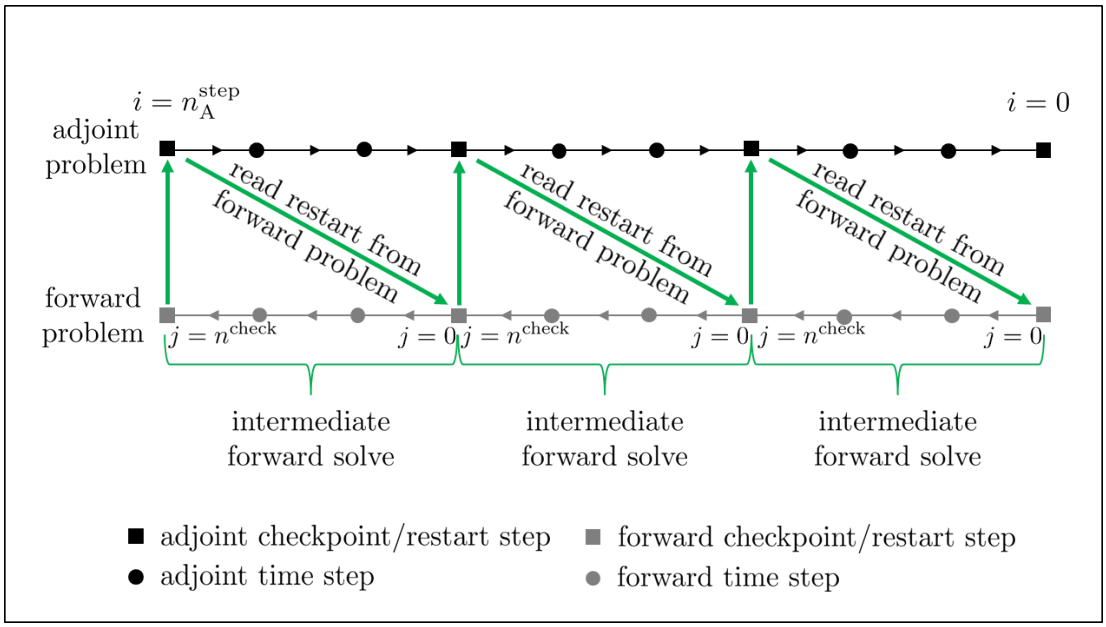


Figure 7.1: Scheme of the checkpointing approach to solve the adjoint problem and calculate the gradient contributions, in analogy to Algorithm 7.

7.6 Proof of Concept

Based on a simple example, the operating principle of the reconstruction algorithm presented in the previous sections is demonstrated. The geometry is two-dimensional with the acoustical domain $\Omega_\mathrm{A} = [-5, 5] \times [-0.1, 0.1]$ and the optical domain $\Omega_\mathrm{L} = [-1, 1] \times [-0.1, 0.1]$ as shown in Figure 7.2. The domain has detectors on the edges at $x_1 = \pm 5$ with four detectors in total. Neumann boundary conditions are applied to the edges of the acoustical domain parallel to the x_1 direction in order to get a semi-one-dimensional solution behavior. The remaining two edges are absorbing by the first order absorbing boundary condition, which is exact for one-dimensional wave propagation. The optical degrees of freedom are entirely constrained, such that the optical light flux is of the form

$$\phi = 1.$$

The optical domain is meshed with twenty equally sized elements. The acoustical domain is meshed with one hundred equally sized elements of polynomial degree $k = 2$. For acoustical time integration, the low-storage Runge–Kutta scheme LSRK3(3) of order three with three stages is chosen in combination with a time step size of $\Delta t = 0.005$ and hence a Courant number of $Cr = 0.1$. The final time is $T = 7$ such that $n_\mathrm{A}^\mathrm{step} = 1400$ time steps are solved.

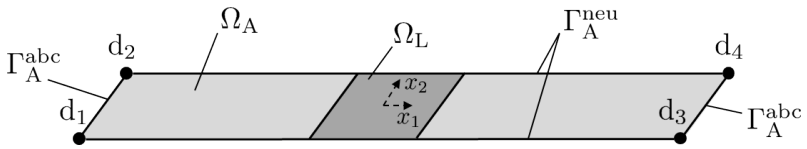


Figure 7.2: Geometrical setup for the proof of concept.

For a first forward solve, the material parameters are set to

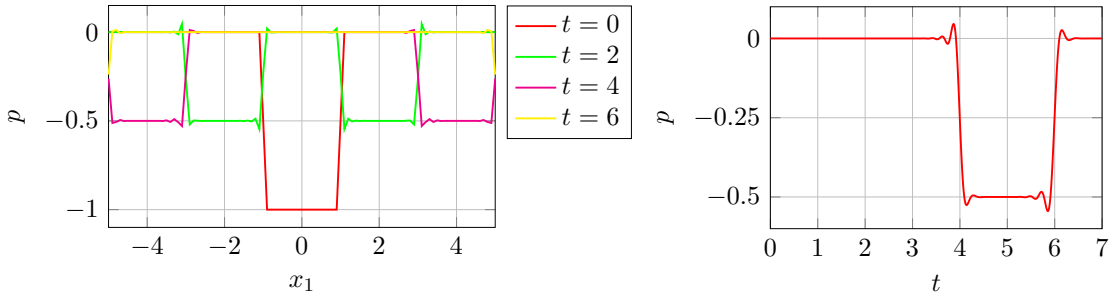
$$\begin{aligned} \mu_\mathrm{a} &= 1, & c &= 1, \\ D &= 1, & \rho &= 1. \end{aligned}$$

Unless stated otherwise, the Grüneisen parameter G is always assumed to be $G = 1$ in the remainder of this work.

The artificial measurements at the four detectors are evaluated as pressure point values. For the light flux, no evaluation of a solution is necessary because all degrees of freedom are constrained by the given Dirichlet values. The initial acoustical fields are calculated from the light flux and the material parameters in accordance with (7.5)–(7.7). The pressure along the x_1-axis for $t = 0$ and successive times is shown in Figure 7.3(a). The pressure values over time as measured at one detector are shown in Figure 7.3(b). For the four detectors, the curves are the same because the setup is symmetric.

The generated artificial measurement values are used to validate the reconstruction algorithm. For the simple setup, analytic expressions for objective function and absorption coefficient gradient are derived to validate the correctness of the methodological as well as algorithmic framework. For changes in the absorption coefficient, which is assumed to be spatially constant in the entire optical domain Ω_L, the objective function calculates as

$$J(\mu_\mathrm{a}) = \frac{1}{2} \cdot 4 \cdot 400 \cdot \left(\frac{\mu_\mathrm{a} - 1}{2} \right)^2 = 200 \left(\mu_\mathrm{a} - 1 \right)^2 .$$



(a) Pressure along x_1-axis for several times

(b) Pressure over time for one detector

Figure 7.3: Pressure visualization for artificial measurement simulation.

The factor $1/2$ stems from the definition of the objective function. The factor 4 represents the four detectors. In 400 time steps, the measurement curves and the curves for variable μ_a differ by $\frac{\mu_\mathrm{a}-1}{2}$ since the measurement curves were created with $\mu_\mathrm{a} = 1$ and the wave equation halves the initial pressure in one dimension. The gradient of the objective function with respect to the absorption coefficient calculates as

$$\frac{\mathrm{d}J(\mu_\mathrm{a})}{\mathrm{d}\mu_\mathrm{a}} = 400 \cdot (\mu_\mathrm{a} - 1)\,.$$

Running the algorithm with the same input parameters as in the simulation to create the measurement data but with $\mu_\mathrm{a} = 0.9$ yields an objective function value $J_\mathrm{sim}(\mu_\mathrm{a} = 0.9) = 1.96617$, which is close to the expected value of $J(\mu_\mathrm{a} = 0.9) = 2$. The difference results from the oscillations and smearing due to the discontinuity in the initial pressure field and correspondent discretization errors. The absorption coefficient gradient from the combination of forward and adjoint quantities evaluated by the algorithm according to (7.32) is -39.11726, which is close to the expected value of -40. Algorithmically, the absorption coefficient gradient is calculated in an element-wise manner. Figure 7.4 shows the element-wise contributions to the scalar gradient value, one received from the adjoint run, the other received by a finite difference analysis. Each element should contribute -2 to the overall gradient since they are all equally sized but deviations appear in the elements close to the boundary, which is again traced back to the discontinuity in the initial pressure field and the fact that the numerical solution procedures cannot resolve the discontinuity. A comparison between the values from the finite difference and the adjoint approach shows that both strategies result in similar values and their performance in terms of accuracy is comparable. They are not equal, not even to the discretization error, because the adjoint run relies on several approximations, i.e., values associated to the acoustical degrees of freedom are interpolated to node values of the optical domain. These operations introduce an approximation error larger than the acoustical and optical discretization error. Hence, the gradients stemming from adjoint and finite difference approach differ slightly. To clarify the mode of operation, Figure 7.5 shows the adjoint pressure plotted along the x_1-axis for several points in time. The adjoint problem is executed backwards in time starting at $t = 7$ and evolving to $t = 0$. The differences between measurement data and simulated data are the driving force for the adjoint problem and are applied on the faces associated to the detectors. The differences propagate according to the adjoint wave equation and form a structure representing the difference between the material parameters.

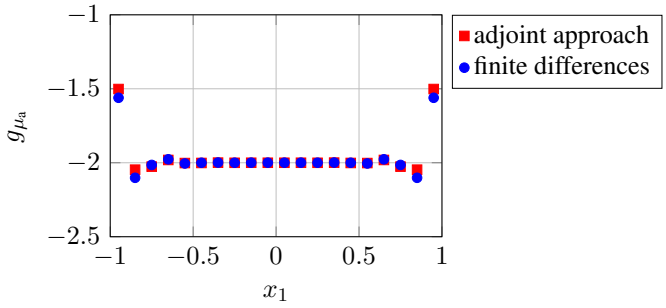


Figure 7.4: Element-wise contributions to the scalar gradient plotted over the x_1-component of the element center.

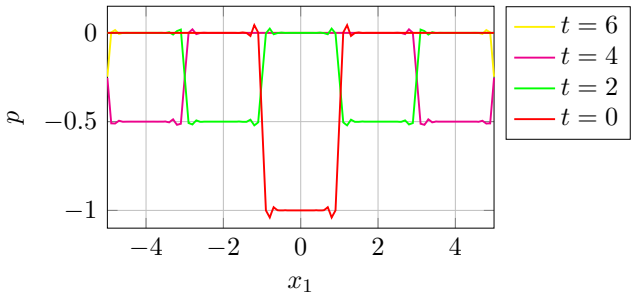
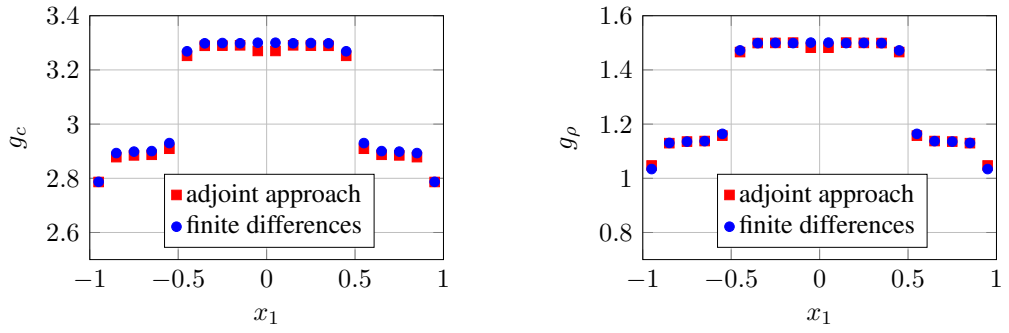


Figure 7.5: Pressure along x_1-axis for several times for gradient evaluation.

Figure 7.6 shows the element-wise gradients for speed of sound c and mass density ρ for input parameters $\mu_a = 0.9, D = 1, c = 0.9, \rho = 0.9$ where optical and acoustical domain overlap and $c = 1, \rho = 1$ in the remainder of the acoustical domain. A difference between finite difference and the adjoint approach is visible but within the expected range. Unfortunately, no analytic expressions for the objective function and gradients can be derived in the presence of acoustical heterogeneities.

To validate the correctness of the diffusion coefficient gradient, the boundary conditions must be different. If the entire optical domain is prescribed with light flux values, the diffusion coefficient gradient is always zero, because its sensitivity results purely from the optical problem. Therefore, only the two edges of the optical domain oriented in x_2 direction are prescribed with a Dirichlet value $\hat{\phi} = 1$ and measurement data is generated with all material parameters on 1. The gradients are calculated for $D = 0.9$, all other settings remain unchanged. Figure 7.7 shows the diffusion coefficient gradient evaluated with the adjoint approach and the finite difference concept. The values are very similar. Again, no analytic expression is available.

The example presented in this section serves as a validation of the derived method based on a simple analytic example. Objective function and objective function gradients are in accordance with analytically derived expressions, or (where no analytic expressions are available) with finite difference estimates.



(a) Speed of sound

(b) Mass density

Figure 7.6: Element-wise contributions to the scalar gradient plotted over the x_1-component of the element center for the speed of sound and mass density.

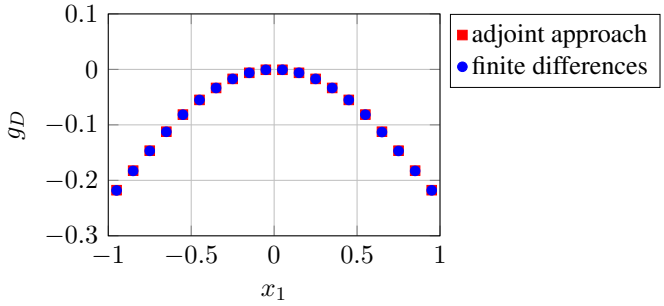


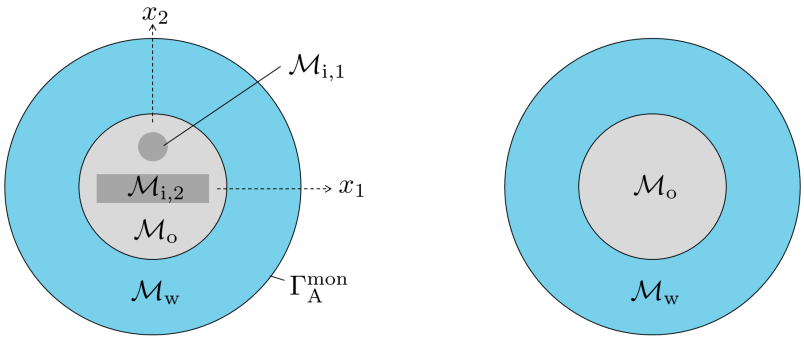
Figure 7.7: Element-wise contributions to the scalar gradient plotted over the x_1-component of the element center for the diffusion coefficient.

7.7 Numerical Examples

After validation of the algorithm by an analytic example in the preceding section, a more complex and more realistic example is examined here with dimensions and material parameters in a realistic range. The setup consists of a $20\,\mathrm{mm}$ diameter circle and an object of $10\,\mathrm{mm}$ diameter with two inclusions as displayed in Figure 7.8(a). The blue region represents the coupling medium $\mathcal{M}_\mathrm{w}$ and the background tissue $\mathcal{M}_\mathrm{o}$ is displayed in light gray. The circular inclusion $\mathcal{M}_{\mathrm{i},1}$ has a diameter of $2\,\mathrm{mm}$ and the center is located at $(x_1, x_2) = (0\,\mathrm{mm}, 3\,\mathrm{mm})$. The rectangular inclusion $\mathcal{M}_{\mathrm{i},2}$ is of size $7\,\mathrm{mm} \times 1.5\,\mathrm{mm}$ with center location at $(x_1, x_2) = (0\,\mathrm{mm}, 0\,\mathrm{mm})$. The material properties used to generate measurement data are summarized in Table 7.2. The

	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$D[\mathrm{mm}]$	$c\left[\frac{\mathrm{mm}}{\mu\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_\mathrm{w}$	0.0	0.0	1.5	1.0
$\mathcal{M}_\mathrm{o}$	0.01	0.5	1.5	1.0
$\mathcal{M}_{\mathrm{i},1}$	0.05	0.1	1.5	1.0
$\mathcal{M}_{\mathrm{i},2}$	0.1	0.5	2.0	2.0

Table 7.2: Material properties for the setup shown in Figure 7.8.



(a) Forward problem (b) Starting point for reconstruction

Figure 7.8: Setup of numerical example.

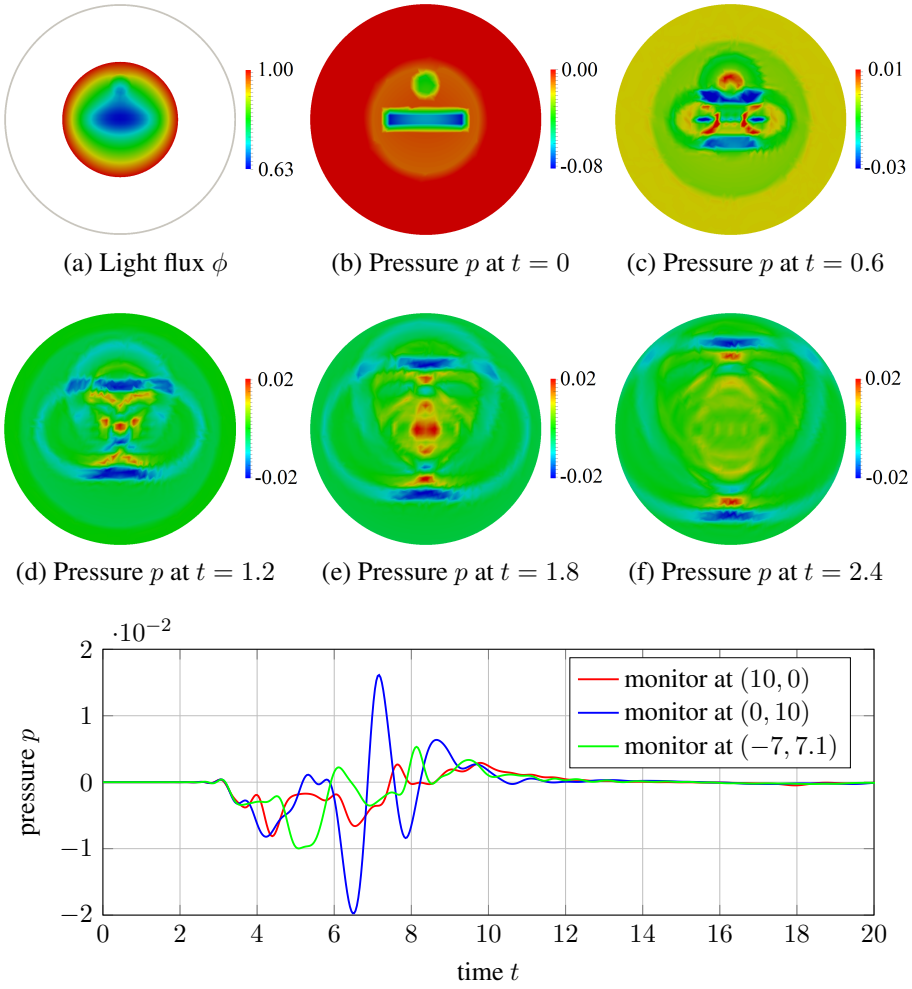
acoustical problem is solved on the entire domain while the optical problem is only solved in the object excluding the coupling medium. A Dirichlet boundary condition on the light flux is applied to the object's boundary $\partial\Omega_{\mathrm{L}}$ with $\hat{\phi} = 1\,\mathrm{J}/\mathrm{mm}^2$. The entire outer boundary $\partial\Omega_{\mathrm{A}}$ is subject to the first order ABC. The acoustical domain is discretized with 2059 quadratic elements ($k = 2$) and the optical domain with 561 elements such that the elements of optical and acoustical mesh coincide in the object. The temporal discretization of the acoustical problem uses a low-storage Runge–Kutta scheme of order three with three stages and two registers and is denoted by LSRK3(3). The time step is set to $0.006\,\mu\mathrm{s}$ to fulfill the CFL condition in all elements resulting in 5000 time steps for the entire simulation length of $30\,\mu\mathrm{s}$. With these settings, the forward problem is solved and Figure 7.9(a)–(f) visualizes the results in terms of the light flux and various pressure snapshots. Figure 7.9(g) plots three exemplary pressure monitor curves. The computational time on one processor for one forward solve is approximately $21\,\mathrm{s}$ plus $24\,\mathrm{s}$ for writing output in every 100th time step.

7.7.1 Committing the Inverse Crime

Reconstruction is run on the setup shown in Figure 7.8(b). Initially, the background material is set for the entire object. The same spatial and temporal discretization are used as for the forward solve, which means that the inverse crime is consciously committed [171]. This offers the opportunity to test the basic functionality and to determine the "optimal" convergence behavior of the inverse problem.

7.7.1.1 All Parameters in One Reconstruction

The first study concerns the reconstruction of all parameters as described in Algorithm 7 with at most 30 iterations and one sequence per parameter per iteration. The initial evaluation of the objective function yields $J = 0.8714$. Figure 7.10(a) plots the relative objective function over the iterations, whereas Figure 7.10(b) plots the relative parameter errors over the iterations. The error in the parameter fields is calculated as the sum over all elements of the square of the difference between the actual and the expected parameter values. The objective function value decreases monotonically. Every forth update, the decrease in the objective function is comparably high,



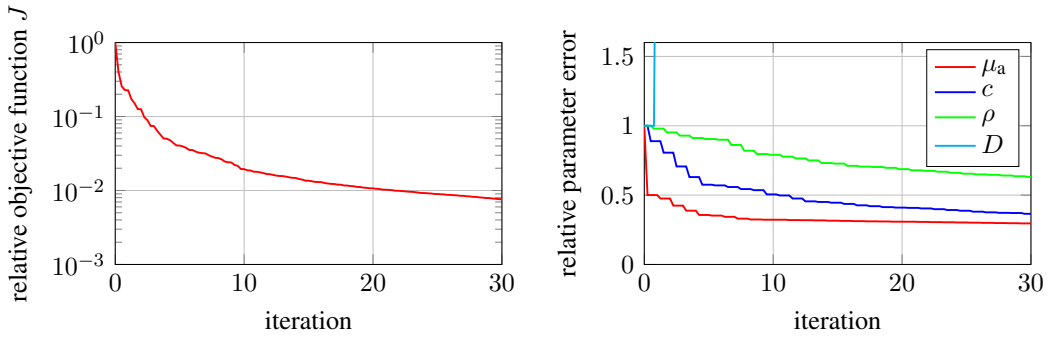
(a) Light flux ϕ (b) Pressure p at $t = 0$ (c) Pressure p at $t = 0.6$

(d) Pressure p at $t = 1.2$ (e) Pressure p at $t = 1.8$ (f) Pressure p at $t = 2.4$

(g) Exemplary pressure monitor curves

Figure 7.9: Solution of the forward problem.

which corresponds to updates of the most sensitive parameter — the absorption coefficient. The relative error in the absorption coefficient distribution converges best, followed by the speed of sound and the mass density. After 30 iterations, the reconstruction is not yet fully converged. The diffusion coefficient starts with a disadvantageous update and slowly recovers but the algorithm fails to improve the diffusion coefficient distribution. For the given setup and comparable applications, the diffusion coefficient's sensitivity is much lower than for the other parameters. Figure 7.11 shows the reconstructed parameter fields after the 30 iterations. In the image of the absorption coefficient, the two inclusions are nicely reconstructed including the magnitude of the coefficient. The speed of sound and mass density image highlight the inclusion with errors in the background material. The diffusion coefficient shows variations in the rectangular inclusion even though it should be different in the circular inclusion. In the background tissue, the diffusion coefficient is wrong by a factor of four. The overall computational time on one processor is $2.9 \cdot 10^4$ s or 8.1 h. The reconstruction includes 413 evaluations of the forward problem and 170 evaluations of the adjoint problem.



(a) The objective function

(b) The material parameters

Figure 7.10: Convergence of the objective function and the parameters.

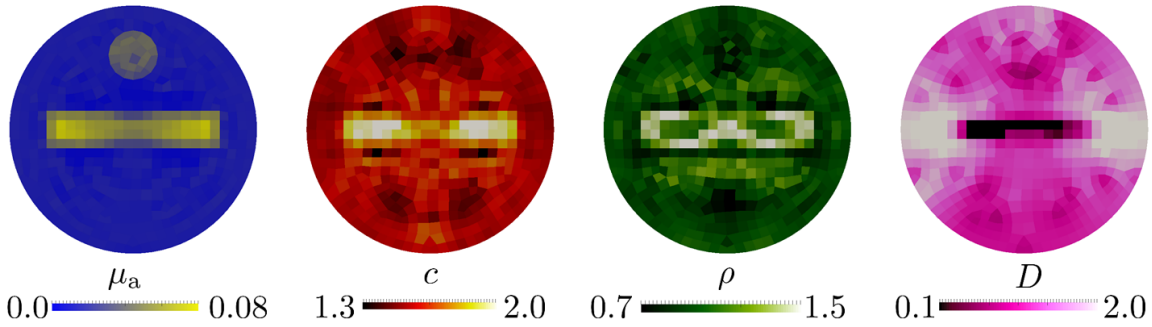
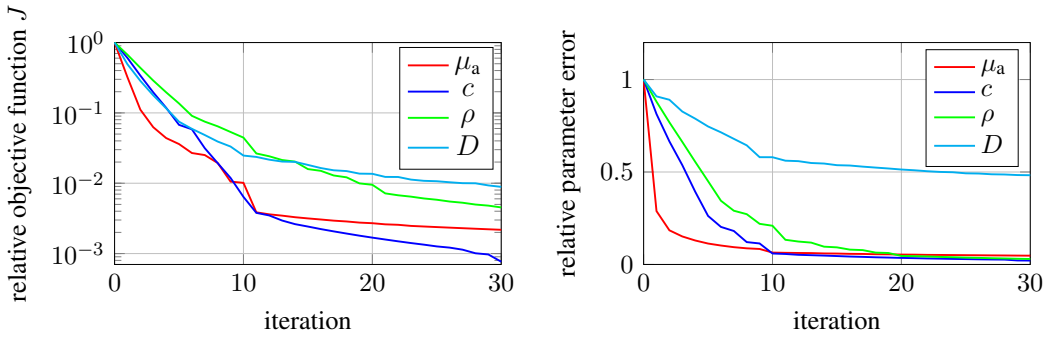


Figure 7.11: Images after 30 optimization iterations of all parameters sequentially in each iteration. From left to right: absorption coefficient, speed of sound, mass density, diffusion coefficient.

7.7.1.2 Each Parameter in its own Reconstruction

The reconstructed parameters influence each other. If an error in the absorption coefficient is present, the optimization of the other parameters tries to counteract the deviations of the pressure curves stemming from the error in the absorption coefficient. To be able to study the convergence behavior of every field without interaction in the parameter fields, reconstructions are run separately for each parameter with all other parameter set as in the generation of measurement signals. In other words, one image reconstruction is run only for the absorption coefficient, while all other parameters are set according to the sample solution. Analogous reconstructions are run for the speed of sound, the mass density, and the diffusion coefficient. The convergence in the objective function and in the parameters is given in Figure 7.12. Compared to Figure 7.10, the convergence is better in all quantities. Even the reconstruction of the diffusion coefficient shows the correct trend. The final images are shown in Figure 7.13. Compared to Figure 7.11, the images exhibit perceivably better quality. They detect the inclusion correctly with a high contrast and the absolute values are in good agreement with the expected values.

Comparison of the two considered setups highlights the interaction between the reconstructions as one parameter tries to balance errors of another parameter. The reconstruction of the absorption coefficient is similar for both setups, which relates to the fact that the absorption co-



(a) The objective function

(b) The material parameters

Figure 7.12: Convergence of the objective function and the parameters.

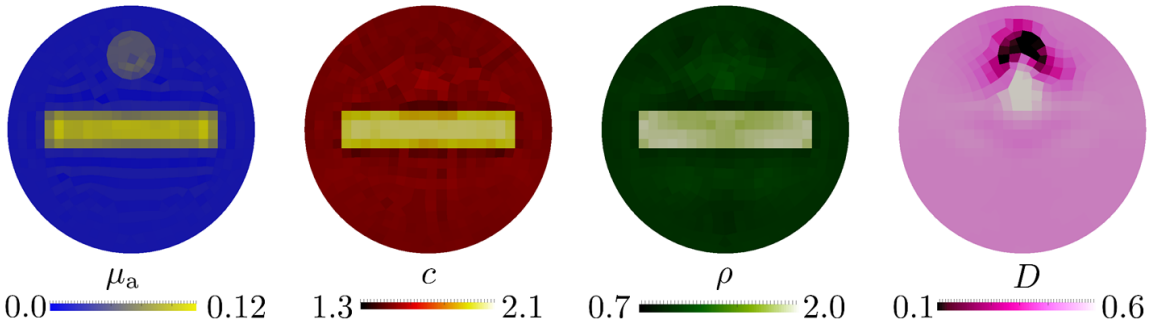


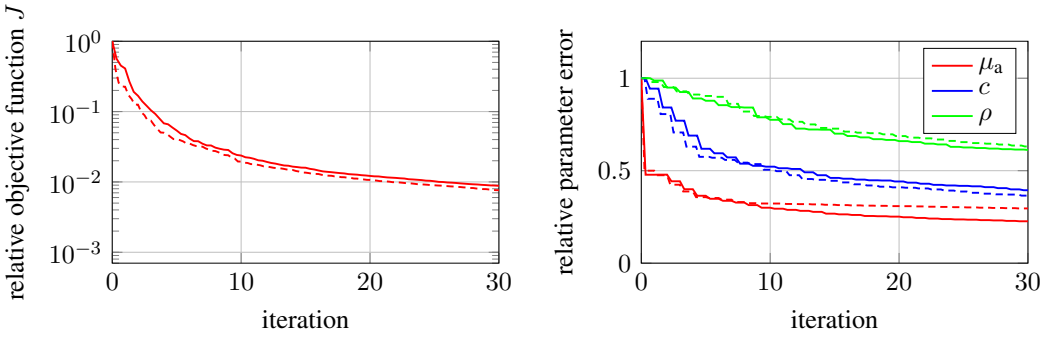
Figure 7.13: Images after 30 optimization iterations of each parameter separately while all other parameters are set correctly. From left to right: absorption coefficient, speed of sound, mass density, diffusion coefficient.

efficient is the most sensitive and the most determining parameter. Judging from the differences between the two setups and the convergence speed of the parameter fields, speed of sound and mass density are the second and the third most sensitive parameters, while the diffusion coefficient has lowest sensitivity. Generally, the diffusion coefficient has very low sensitivity in typical optoacoustic setups [143, 147, 166], which is why it will not be considered in all following studies. For the following reconstructions, the diffusion coefficient is set to 0.5 in all materials, such that it is spatially constant.

7.7.2 Differing Discretizations

The inverse crime [171] is avoided by creation of measurement data with a finer discretization and addition of noise. In this section, the effect of differing discretizations is studied. Also, the number of elements in the spatial discretizations is kept as in the preceding example but the mesh is rotated by $45°$, such that the inclusions to be reconstructed do not conform with the mesh. For the evaluation of the parameter errors, elements that lie on the edge between background and inclusion are skipped. The absorption coefficient, speed of sound, and mass density are reconstructed as in the first example of Section 7.7.1. The initial evaluation of the objective function yields $J = 0.9095$ in contrast to $J = 0.8714$ when committing the inverse

crime. This is due to two facts: First, the numerical solution is slightly different on a differing mesh and second, the evaluation of the pressure error e now requires an interpolation to the monitor positions. Figure 7.14 plots the relative objective function and the relative parameter errors over the iterations with the dashed lines repeated from Figure 7.11 for reference. The convergence behavior is very similar compared to the reconstruction with conforming meshes and the objective function converges slightly more slowly but qualitatively in the same manner. Absorption coefficient and mass density converge slightly faster and speed of sound slightly more slowly. Figure 7.15 shows the final images and the effect of the nonconforming mesh is clearly visible as zigzag contour of the inclusions. Apart from that, the images resemble those shown in Figure 7.11.



(a) The objective function

(b) The material parameters

Figure 7.14: Convergence of the objective function and the parameters for nonconforming discretizations in forward and inverse problem. The dashed lines are repeated from Figure 7.11 for reference.

The reconstruction is performed on the rotated mesh once again, but additionally, the input measurement data is obtained with a finer discretization. The finer discretization is set up by one uniform refinement step of the unrotated mesh. The time step is $\Delta t = 0.003\mu\,\mathrm{s}$. With these settings, the measurement data is obtained, which is used as input for the reconstruction on the coarse rotated mesh. Figure 7.16 plots the convergence of the objective function and the

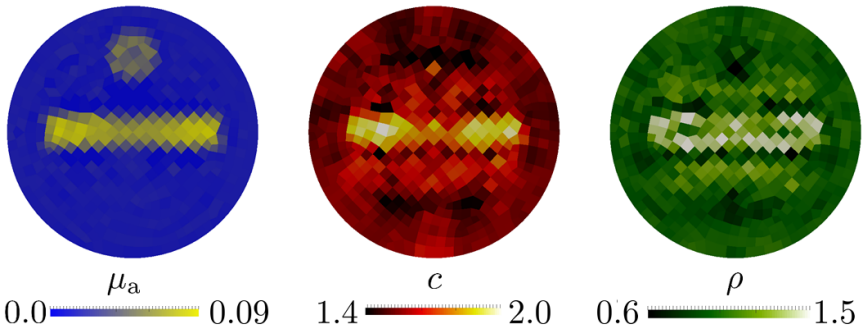
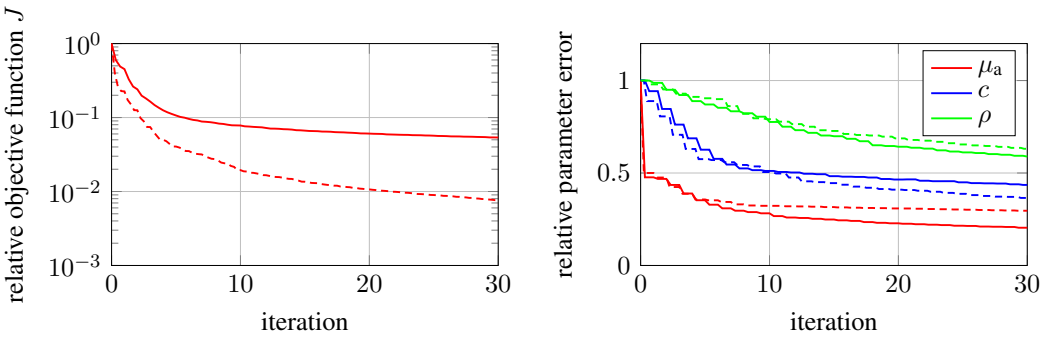


Figure 7.15: Images after 30 optimization iterations of all parameters sequentially in each iteration on a mesh not conforming with the inclusion shapes. From left to right: absorption coefficient, speed of sound, mass density.

parameters, again with the dotted lines from Figure 7.11 as reference. The convergence in the objective function is significantly slower compared to the previous setups while the convergence of the parameters is comparable to the inverse crime setup and almost the same as for the previous example. The final images are shown in Figure 7.17. The visual impression is that some spurious oscillations occur in the absorption coefficient. Also, the speed of sound shows less elements with high values. The mass density image is very similar to Figure 7.15 except for one element with a high value in the interior of the circular inclusion.



(a) The objective function

(b) The material parameters

Figure 7.16: Convergence of the objective function and the parameters for nonconforming discretizations in forward and inverse problem. The dashed lines are repeated from Figure 7.11 for reference.

7.7.3 The Influence of Noise

In this section, the conforming discretizations for generation of measurement data and reconstruction are considered and noise is added to the measurement signals before reconstruction. Gray noise according to the ISO 66-phon equal-loudness contour is generated and scaled. The maximal absolute value of all measurement signals is 0.0202. Therefore, the noise is scaled by 0.00101 and 0.00202 and added to generate measurement signals with 5% and 10% noise level, respectively. Figure 7.18 visualizes the noisy measurement signals for the detector located at $(10\,\mathrm{mm}, 0\,\mathrm{mm})$. It is very important to use gray noise. White noise generated from random numbers has a more pronounced high frequency component and is just smeared out by the spatial discretization. In Figure 7.18, the effects of low frequency noise are visible as offset for $t < 3\,\mu\mathrm{s}$ or $9\,\mu\mathrm{s} < t < 13\,\mu\mathrm{s}$. Reconstruction is run on the conforming discretization setup and the convergence results are shown in Figure 7.19. The objective function converges more slowly the higher the noise level, which conforms with the expectations considering that the optimal parameter distribution will yield a non-zero objective function. Note that a high noise level may be problematic for the line search algorithm considering fulfillment of the sufficient decrease condition. For the acoustical parameters, sound speed c and mass density ρ, the convergence is slightly slower compared to the setup without noise with a slight advantage of the 5% noise reconstruction over the 10% noise reconstruction. The absorption coefficient converges slightly better in the noisy setups. Figure 7.20 shows the resulting images for 10% noise. The 5% noise

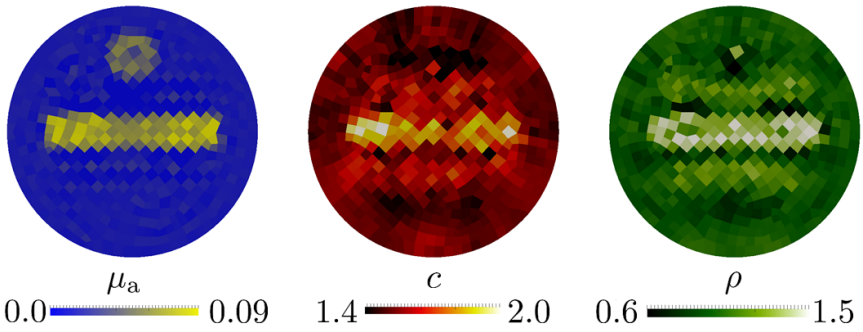


Figure 7.17: Images after 30 optimization iterations of all parameters sequentially in each iteration on a mesh not conforming with the inclusion shapes and measurement data obtained on a finer discretization. From left to right: absorption coefficient, speed of sound, mass density.

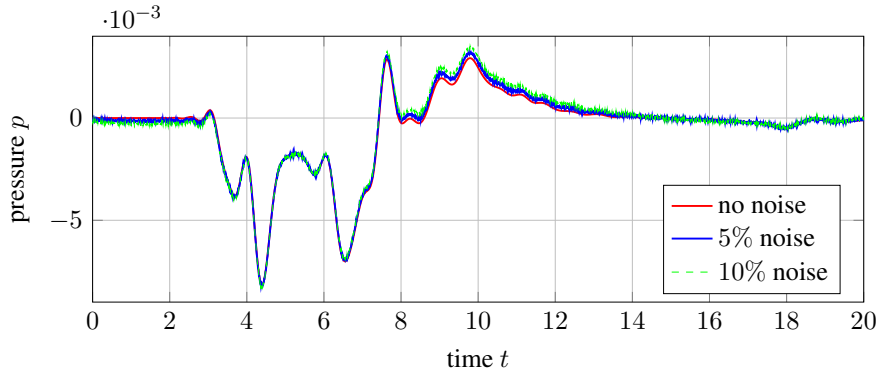
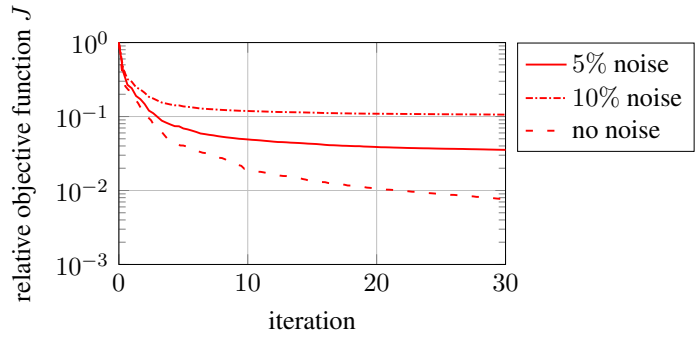


Figure 7.18: Pressure monitor curve for the detector located at $(10\,\mathrm{mm}, 0\,\mathrm{mm})$ in its original form and with 5% and 10% overlaid gray noise.

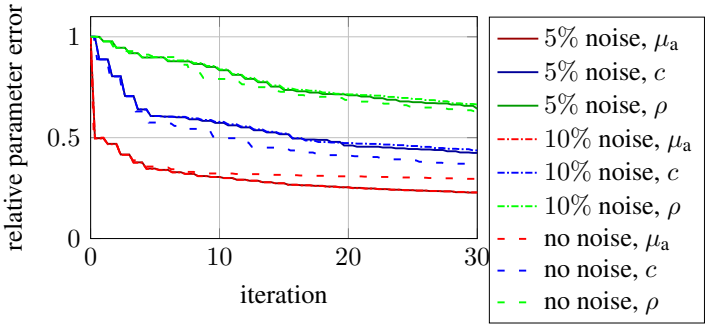
images are not shown because they are visually almost indistinguishable from the 10% noise images.

7.7.4 Avoiding the Inverse Crime

As a last example, all concepts to avoid the inverse crime are combined, i.e., the reconstruction is performed on the rotated mesh that does not conform with the inclusions using measurement data obtained with the refined discretization overlaid with 10% of gray noise. In essence, the cases of Sections 7.7.2 and 7.7.3 are combined. Figure 7.21 plots the convergence of the objective function and the parameter errors. The convergence of the objective function is slower compared to all preceding setups. The convergence of the parameters is very similar to the ones obtained in Figure 7.19. The resulting images are shown in Figure 7.22. They appear comparable to the images shown in Figure 7.17 where the discretization is rotated and measurement data is obtained on a finer mesh.



(a) The objective function



(b) The material parameters

Figure 7.19: Convergence of the objective function and the parameters.

7.7.5 Long-Term Convergence

The reconstruction as in Section 7.7.1.1 is carried out but with 300 iterations instead of 30 and omitting the reconstruction of the diffusion coefficient. Figure 7.23 plots the convergence of the relative objective function and the relative parameter errors. Figure 7.24 shows images after 100 and after 300 iterations. The images show better contrast compared to the results after 30 iterations as presented in Figure 7.11. Especially for the mass density, a significant improvement in the interior of the rectangular inclusion is visible comparing the image after 100 and after 300 iterations. The convergence of the objective function is fast in the first iterations (reaching a relative objective function value of 1% already after 26 iterations) and slows significantly down after that but continues to decrease with a final value of $7.38 \cdot 10^{-4}$ after 300 iterations. Even though the objective function reaches such a small value, the line search procedure succeeds to find a valid step length in all iterations. The parameter fields converge monotonically with a noticeable slow down after approximately 30 iterations. This example confirms once more the correctness of the implementation of the derived image reconstruction method.

The reconstruction as in Section 7.7.4 avoiding the inverse crime is carried out but with 300 iterations instead of 30. The convergence of the objective function and the parameter fields is plotted in Figure 7.25. The convergence of the objective function slows significantly down after 40 iterations. The parameter errors decrease in the first iterations. Absorption coefficient and speed of sound show an increasing error after 63 and 40 iterations, respectively. Figure 7.26 shows images at iteration 100 and 300. Comparison with Figure 7.22 shows slight differences concerning the contrast for the absorption coefficient and the mass density and a checkerboard

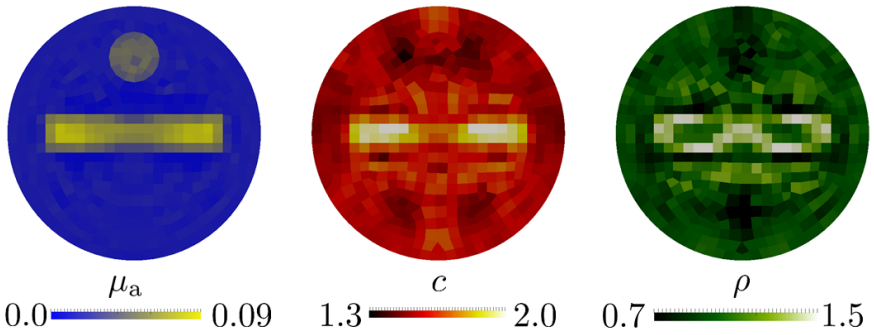
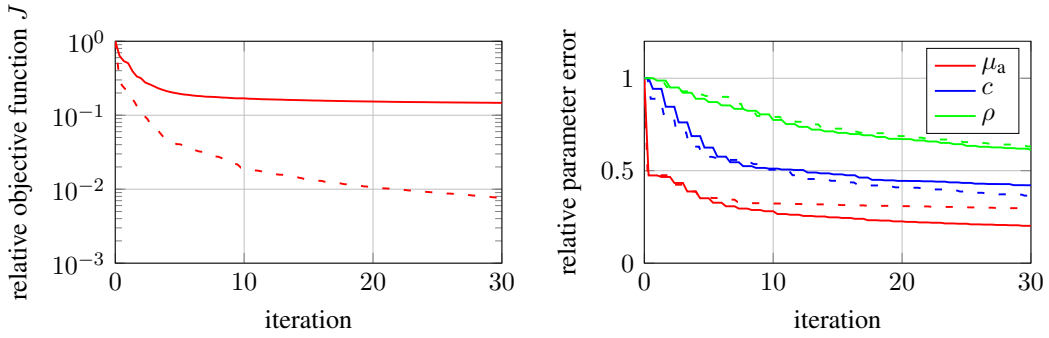


Figure 7.20: Images after 30 optimization iterations of all parameters sequentially in each iteration with 10% noise level, respectively. From left to right: absorption coefficient, speed of sound, mass density.



(a) The objective function

(b) The material parameters

Figure 7.21: Convergence of the objective function and the parameters for nonconforming discretization with 10% noise level and measurement data obtained on a finer discretization. The dashed lines are repeated from Figure 7.19 for reference.

pattern in the absorption coefficient. Also, fluctuations in the speed of sound and mass density distribution increase. This is a common error for high numbers of iterations when avoiding the inverse crime by usage of differing discretizations. The mesh rotation causes the inclusion contour to show a saw tooth contour, which in turn causes saw tooth shaped errors in the remainder of the domain. Additionally, the noise is approximated in late iterations. Note that the optoacoustic inverse problem is heavily ill-conditioned and the optimization problem generally has several local minima. Hence, the final solution depends on the first updates. The convergence behavior and the images show that the sensitivity on the material parameters is higher than on the noise and contour mismatch but that they play a role once the material parameters are reconstructed and already decreased the objective function. For reconstructions with a high number of iterations, total variation regularization should be added to avoid the increase of parameter errors during late iterations.

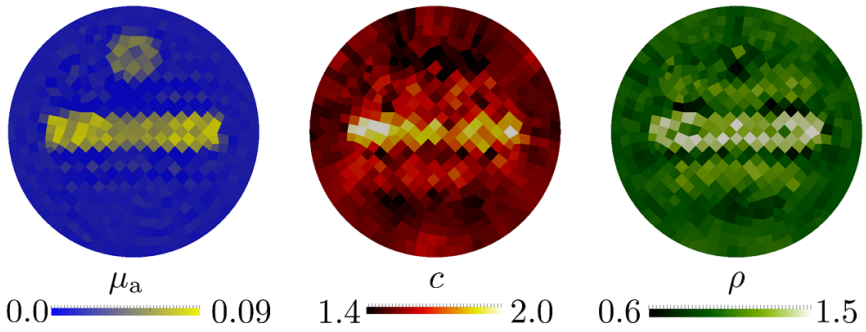
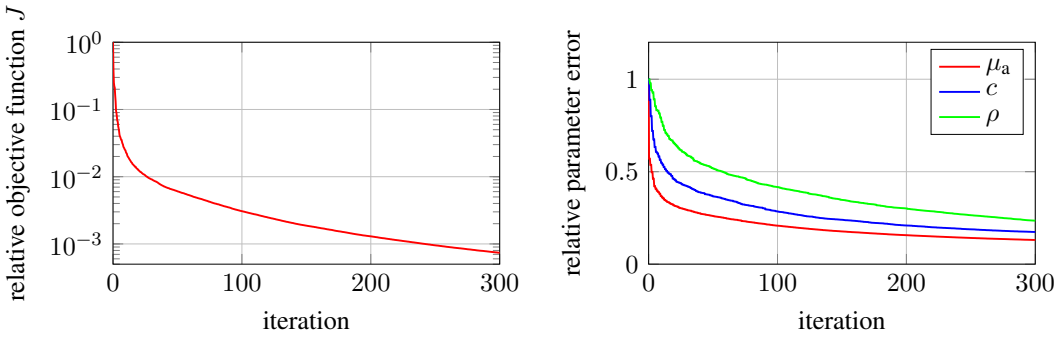


Figure 7.22: Images after 30 optimization iterations of all parameters sequentially in each iteration for nonconforming discretization with 10% noise level and measurement data obtained on a finer discretization. From left to right: absorption coefficient, speed of sound, mass density.



(a) The objective function

(b) The material parameters

Figure 7.23: Convergence of the objective function and the parameters for consciously committed inverse crime over 300 iterations.

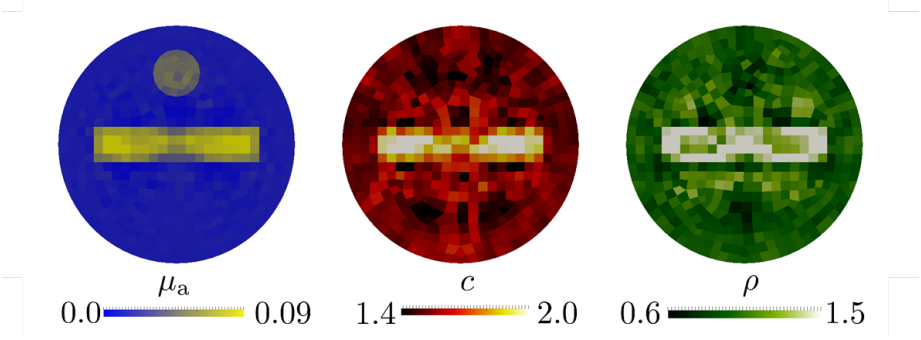
7.7.6 The Effect of Pressure Discontinuities

The absorption coefficient is discretized as element-wise constant material parameter with jumps between elements. If optical and acoustical discretization are conforming and the absorption coefficient varies over elements, the initial pressure field contains discontinuities, since it is evaluated according to

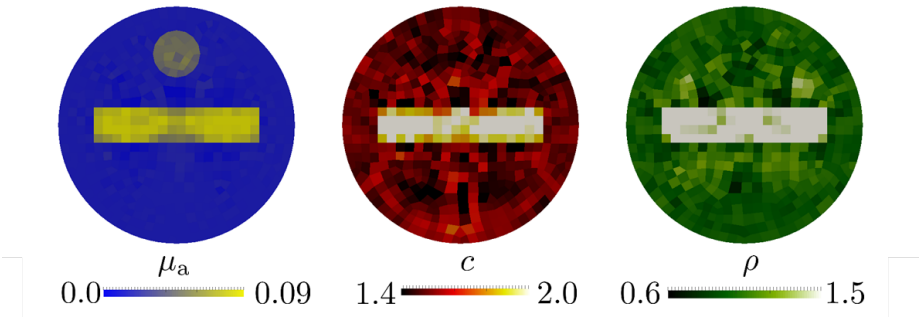
$$p_0 = -G\mu_a\phi,$$

see also equation (7.5). The acoustic wave equation transports discontinuities. Its discretized counterpart, however, smears the discontinuity potentially with slight oscillations. Naturally, the question of convergence behavior and the interaction between discretization error due to the discontinuity and image reconstruction quality arises. The following numerical example addresses this question.

The setup is similar as in the preceding sections with geometry and material as in Figure 7.8. Slightly different material parameters are used as shown in Table 7.3, i.e., the circular inclusion is omitted and the acoustical material is spatially constant. In contrast to the preceding simulations,

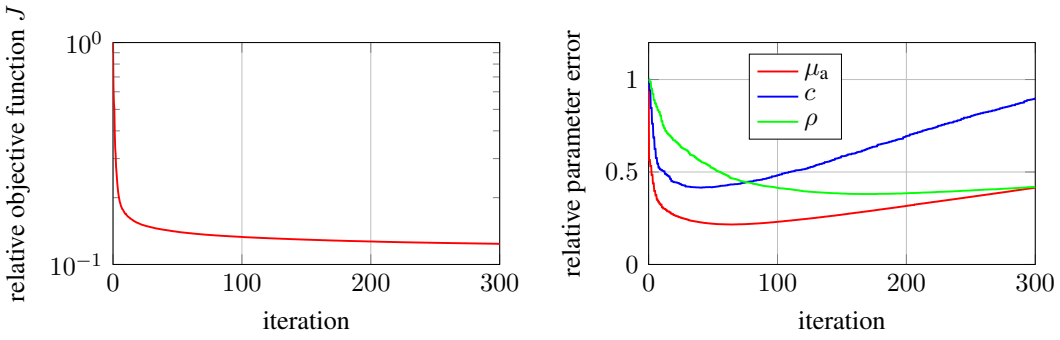


(a) 100 iterations



(b) 300 iterations

Figure 7.24: Images after 100 and 300 optimization iterations of all parameters sequentially in each iteration for consciously committed inverse crime. From left to right: absorption coefficient, speed of sound, mass density.



(a) The objective function

(b) The material parameters

Figure 7.25: Convergence of the objective function and the parameters for nonconforming discretization with 10% noise level and measurement data obtained on a finer discretization over 300 iterations.

	$\mu_a \left[\frac{1}{\text{mm}}\right]$	$D[\text{mm}]$	$c \left[\frac{\text{mm}}{\mu\text{s}}\right]$	$\rho \left[\frac{\text{mg}}{\text{mm}^3}\right]$
$\mathcal{M}_w$	0.0	0.0	1.5	1.0
$\mathcal{M}_o, \mathcal{M}_{i,1}$	0.01	0.5	1.5	1.0
$\mathcal{M}_{i,2}$	0.1	0.5	1.5	1.0

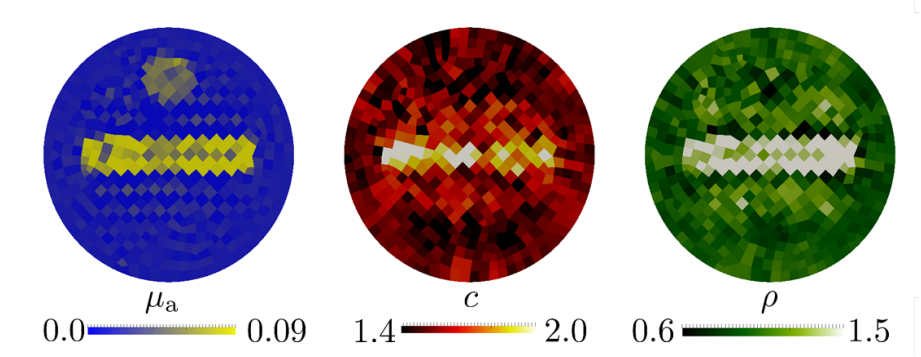
Table 7.3: Material properties for the setup shown in Figure 7.8 but for study on pressure discontinuities.

the entire optical domain is constrained with a Dirichlet condition on the light, such that the optical field is spatially constant $\phi = 1\,{}^{\text{J}}/_{\text{mm}^2}$. With $G = 1$, the initial pressure field hence calculates as $p_0 = -\mu_a$.

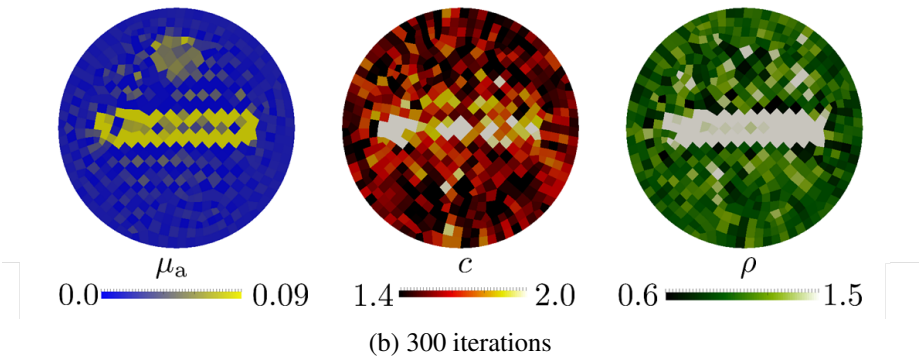
Several forward solves are run. Two meshes are used, one with 2059 elements and one with $4 \cdot 2059 = 8236$ elements as uniform refinement of the first. For time integration, LSRK3(3) is used. For linear elements $(k = 1)$, the time step sizes are $\Delta t = 0.012\,\mu\text{s}$ and $\Delta t = 0.006\,\mu\text{s}$ for the coarse and fine mesh, respectively. For refinement by adaption of the polynomial degree of the shape functions, the Courant number *Cr* is kept constant.

Figure 7.27 shows the pressure signals obtained by the detector located at $(x_1, x_2) = (10\,\text{mm}, 0\,\text{mm})$ for the coarse and fine mesh with polynomial degrees $k = 1, 2, 6$, respectively. The initial pressure has two discontinuities that arrive at the given detector at $3.33\,\mu\text{s}$ and $4.33\,\mu\text{s}$. The zoom in the right panel of Figure 7.27 shows how different the discretizations approximate the discontinuities: $k = 1$ on the coarse mesh smears both discontinuties. The other discretizations show steeper slopes and oscillations around the discontinuities. The DG discretization is expected to be of accuracy order $k + 1$. This however only holds for smooth initial fields. In the presence of discontinuities, the convergence is significantly slower, i.e., never better than order one independent of the polynomial degree of the shape functions [180].

The next question is, how the discontinuity affects the adjoint solution. Therefore, the absorption coefficient of the inclusion is set to $0.01\,{}^{1}/_{\text{mm}}$ and the first adjoint run is carried out. Figure 7.28 plots the adjoint pressure field for $t = 0$ (the final step of the adjoint run) obtained on the coarse mesh with $k = 2$ but with monitor data from all six setups considered before, i.e., the



(a) 100 iterations



(b) 300 iterations

Figure 7.26: Images after 100 and 300 optimization iterations of all parameters sequentially in each iteration for nonconforming discretization with 10% noise level and measurement data obtained on a finer discretization. From left to right: absorption coefficient, speed of sound, mass density.

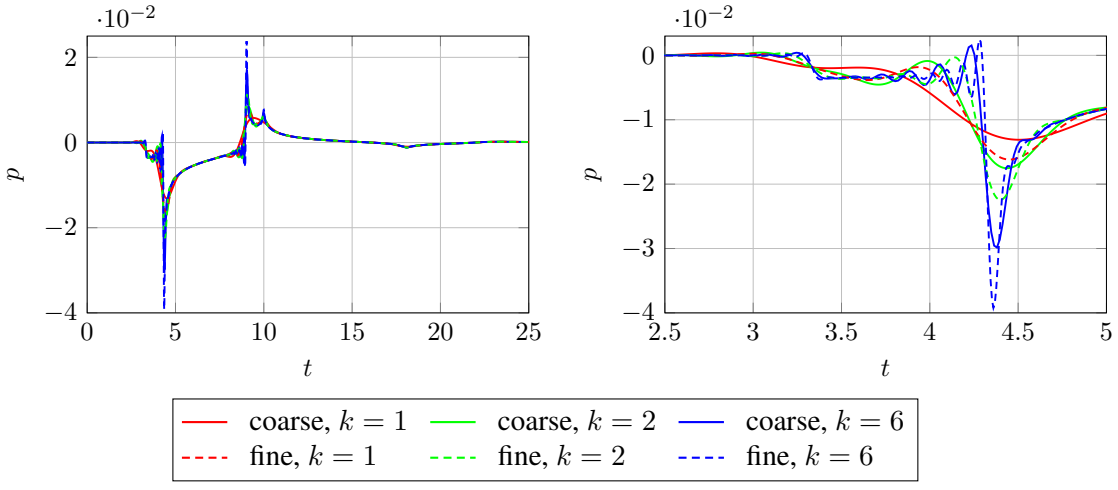


Figure 7.27: Pressure at $(x_1, x_2) = (10\,\mathrm{mm}, 0\,\mathrm{mm})$ over time for the coarse and the fine mesh with $k = 1, 2, 6$, respectively, on the time interval $t = [0\,\mu\mathrm{s}, 15\,\mu\mathrm{s}]$ and zoom to the interval $t = [2.5\,\mu\mathrm{s}, 5\,\mu\mathrm{s}]$.

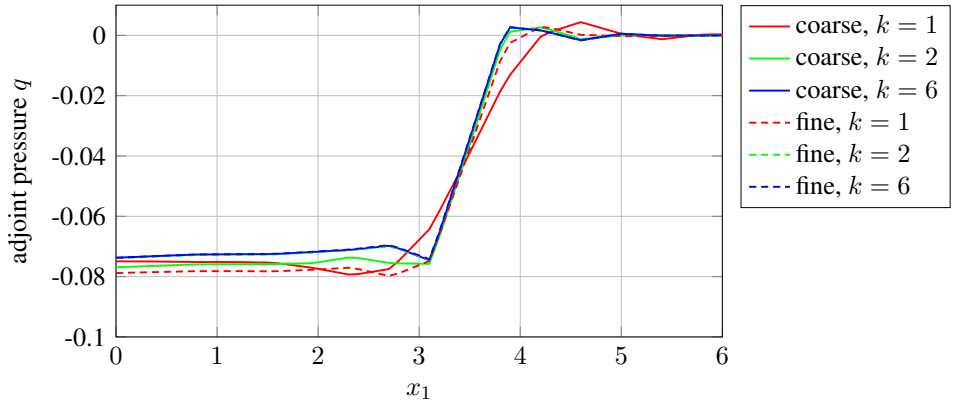


Figure 7.28: Adjoint pressure at $t = 0\,\mu\text{s}$ along a line from $(x_1, x_2) = (0\,\text{mm}, 0\,\text{mm})$ to $(x_1, x_2) = (6\,\text{mm}, 0\,\text{mm})$ obtained with the coarse mesh and $k = 2$ but monitor values from the different discretizations.

pressure curves plotted in Figure 7.27 are the input measurement data for this adjoint run. The pressure field is plotted along a line from $(x_1, x_2) = (0\,\text{mm}, 0\,\text{mm})$ to $(x_1, x_2) = (6\,\text{mm}, 0\,\text{mm})$. For $k = 1$ on the coarse and fine mesh, the curves are comparably smooth. The coarse mesh with $k = 2$ yields a slightly different value in the inclusion. All other curves are congruent, despite the noticeable differences in the input data (see Figure 7.27). That these noticeable differences do not yield a noticeable difference in the adjoint pressure field is explained by Figure 7.29. For curve (a) of Figure 7.29, a forward evaluation is run on the reconstruction setup, i.e., with $\mu_\text{a} = 0.01^1/_\text{mm}$ in the entire optical domain. The coarse mesh with $k = 2$ is used. The figure displays the difference between this simulation and the forward run to obtain the measurement data with the same discretization, both evaluated at $(x_1, x_2) = (10\,\text{mm}, 0\,\text{mm})$. Hence, curve (a) can be understood as the source applied to the detector at $(x_1, x_2) = (10\,\text{mm}, 0\,\text{mm})$ for the first adjoint run. This setup corresponds to the green curve in Figure 7.28. Curve (b) is similar but the difference is build with the forward run to obtain the measurement data on the fine mesh with $k = 6$. It hence corresponds to the dashed blue line in Figure 7.28. Comparison of curve (a) and (b) shows how different the source terms are depending on the difference of the reference discretization. The oscillations close to the discontinuities for fine discretization as shown in Figure 7.27 are reproduced in curve (b). Next, the pressure difference is applied as source term to the adjoint problem and curve (c) shows the simulated adjoint pressure at $(x_1, x_2) = (10\,\text{mm}, 0\,\text{mm})$ with (b) as source term. Curve (b) and (c) are not equal because the application of the source term involves interpolation to the detector locations and projection back onto the degrees of freedom. Also, interpolation in time is involved because the coarser discretization operates on a greater time step size. Curve (d) visualizes the effects of the signal propagation. It represents the simulated adjoint pressure for the same setup as curve (c) but evaluated at $(x_1, x_2) = (8\,\text{mm}, 0\,\text{mm})$, i.e., $2\,\text{mm}$ further inside of the domain. Apparent differences between curve (c) and (d) are an additional smoothing, which corresponds to the discretization error. A qualitative visual comparison of curves (d) and (a) shows similar slopes. In summary it can be stated that the reconstruction results do only depend negligible on the resolution of discontinuities if the discretization in the reconstruction is coarser. A coarse discretization operates as a low pass filter and high frequency contents of measurement signals are filtered out because

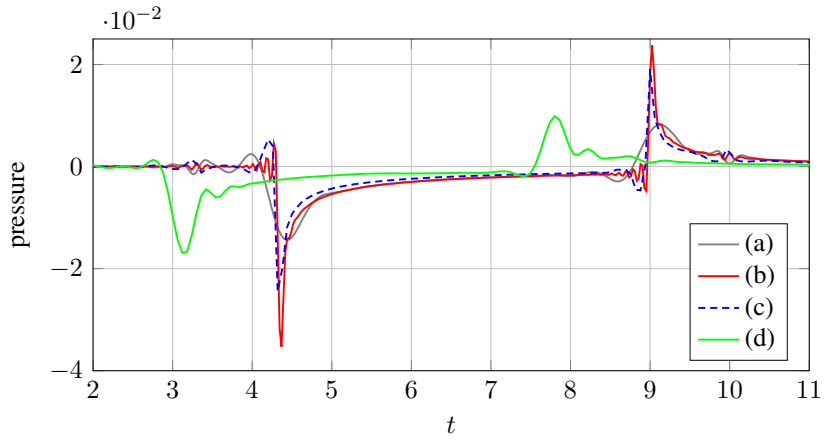


Figure 7.29: Pressure curves (a)–(c) are evaluated at $(x_1, x_2) = (10\,\mathrm{mm}, 0\,\mathrm{mm})$. Curve (a) is obtained as the difference between the pressure monitored with the coarse mesh and $k = 2$ with inclusion and without inclusion. Curve (b) is obtained as the difference between the pressure monitored with the fine mesh and $k = 6$ with inclusion and with the coarse mesh and $k = 2$ without inclusion. Curve (c) is the adjoint pressure measured during the adjoint run. Curve (d) is the adjoint pressure measured during the adjoint run but evaluated at $(x_1, x_2) = (8\,\mathrm{mm}, 0\,\mathrm{mm})$.

the approximation capabilities are limited. If the discretization in the reconstruction is finer, the results are affected as indicated by the red line in Figure 7.28. A finer discretization detects the differences in resolution. However, the reconstruction on finer meshes is an inverse crime that should always be avoided [171]. Additionally, experimental setups always represent the case of a too coarse reconstruction.

7.7.7 Conclusion

From the presented examples the following main conclusions are drawn:

- The two examples in Section 7.7.1 as well as the first example in Section 7.7.5 validate the correctness of the code.
- The comparison to the two examples in Section 7.7.1 reveal a high level of ill-conditioning already apparent in this simple full view setup. It also highlights the strong interaction between the parameter reconstructions.
- The sensitivity of the diffusion coefficient is lower compared to the other parameters and it is not considered in the remainder of this work.
- Usage of a non-conforming discretization of same characteristic element size and time step size has only a minor effect on the convergence of objective function and parameters.
- Usage of a non-conforming discretization of larger characteristic element size and time step size for reconstruction (or vice versa finer discretization for generation of measurement data) yields a slowdown in the convergence of the objective function. The convergence of the parameters is only slightly affected.
- With increasing noise level, the convergence of the objective function slows down noticeably; the acoustic parameters converge more slowly, while the absorption coefficient

converges slightly better, which however is assumed not to be reproducible over a range of imaging tests.
- The combination of noise, spatial discretization not conforming with the inclusion shapes, and finer discretization for generation of measurement data show that the differences in the discretization have the strongest influence on the resulting images rather than the noise.
- Only for long-term optimizations, the noise causes significant fluctuations, especially in the acoustic images. Total variation regularization should be applied if an optimization is run with a high number of iterations.
- The resolution of the discontinuities has an insignificant effect on the image reconstruction in case the inverse crime is avoided by reconstruction on coarser discretizations or in case experimentally obtained measurement data is used.

If measurement data is obtained not by simulation but by experiment, the parameter reconstructions will additionally try to balance modeling and discretization errors. Two major difficulties are stated for the quantitative image reconstruction in optoacoustics with the developed method: First, the computational time is high because a reconstruction requires several forward and adjoint solves and second, the inverse problem of optoacoustic image reconstruction is strongly ill-conditioned. In Chapter 8, an approach to reduce computational expense is presented and in Chapter 9, two methods opposing the ill-conditioning are derived.

8 Reduction of the Computational Domain

In this chapter, a concept to reduce the size of the computational domain in typical optoacoustic image reconstruction scenarios is presented. First, a motivation is given in Section 8.1 followed by an explanation of the functional principle in Section 8.2. Numerical evidence of the applicability is provided in Section 8.3.

8.1 Motivation

In an optoacoustic tomograph, the object of interest must be surrounded by a coupling medium to bridge the gap between the object and the transducers. Depending on the tomographic setup and the object, the gap can be comparably wide such that a large amount of the computational domain is occupied by the coupling medium. Using the example of the multispectral optoacoustic tomograph MSOT inVision256-TF (iThera Medical GmbH, München, Germany) with a transducer ring of $81\,\mathrm{mm}$ diameter (see [46] for details on the tomograph) and a mouse brain as imaging object with about $15\,\mathrm{mm}$ diameter, 96% of the reconstruction domain represent the coupling medium while only 4% are covered by the object of interest. This is a waste of computational resources in case the coupling medium is homogeneous and its parameters are not optimized. In the reconstruction algorithm presented in Chapter 7, most computational time is spent in the evaluation of the forward and adjoint acoustical problem and the computational cost is directly proportional to the number of elements in the mesh of the acoustical domain. Therefore, the potential gains of a reduction of the domain size are comparably high. To avoid the superfluous evaluation of sound propagation in the coupling medium, the computational domain is cropped and the measurement boundary is artificially moved closer to the object. Figure 8.1

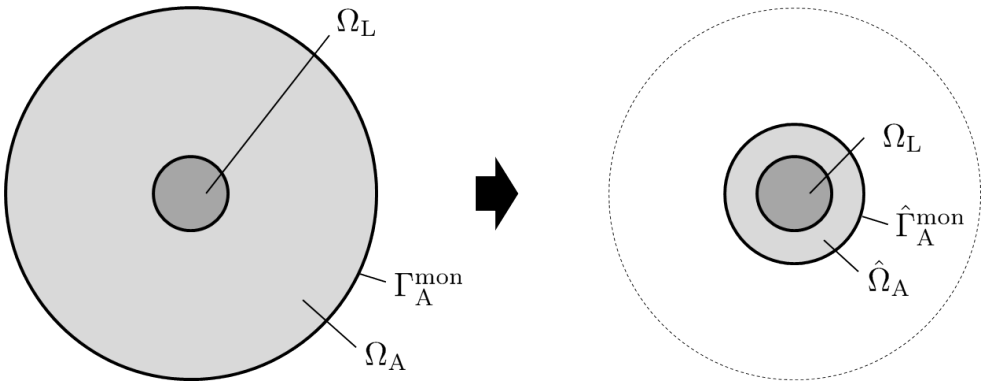


Figure 8.1: Scheme of concept to reduce the computational domain size.

shows a typical setup in the left panel: the dark gray area represents the object while the light gray area represents the coupling medium. The idea is to artificially move the measurement boundary closer to the object as shown in the right panel of Figure 8.1. In order to achieve this, the measurement data must be transferred. The approach presented in the following is an adaption of [7, 8], where it is used in the context of the wave equation and the Helmholtz equation for the localization of a scatterer. The approach is also a further development of [9].

8.2 Functional Principle

Starting point is the setup as described in Chapter 7.5 with the optical domain Ω_L and the acoustical domain Ω_A significantly larger than the optical domain. At the portion $\Gamma_\mathrm{A}^\mathrm{mon}$ of the boundary of the acoustical domain $\partial\Omega_\mathrm{A}$, the pressure is measured. For now, it is assumed that the acoustical boundary and the acoustical measurement boundary coincide $\partial\Omega_\mathrm{A} = \Gamma_\mathrm{A}^\mathrm{mon}$. The limited view scenario will be studied subsequently. The objective is to reduce the acoustical domain to $\hat{\Omega}_\mathrm{A}$, which is noticeably smaller than the original acoustical domain Ω_A as indicated in Figure 8.1. Therefore, the measurement signals have to be propagated from the original measurement boundary $\Gamma_\mathrm{A}^\mathrm{mon}$ to the boundary of the reduced domain $\hat{\Gamma}_\mathrm{A}^\mathrm{mon}$. This is achieved by a simulation on $\Omega_\mathrm{A} \setminus \hat{\Omega}_\mathrm{A}$. The measurement values are prescribed on $\Gamma_\mathrm{A}^\mathrm{mon}$ by a Dirichlet condition and the pressure is propagated back into the domain. At $\hat{\Gamma}_\mathrm{A}^\mathrm{mon}$, an absorbing boundary condition is applied. The back propagation of the measurement data relies on the reciprocity of the wave equation. Due to the fact that viscous entropy losses are not considered by the acoustic wave equation, solutions $p(\boldsymbol{x}, t)$ are time invariant, i.e., $p(\boldsymbol{x}, -t)$ are as well solutions. The problem reads

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\rho}\nabla p = 0 \qquad \text{in} \quad \Omega_\mathrm{A} \setminus \hat{\Omega}_\mathrm{A} \times [0, T], \tag{8.1}$$

$$\frac{\partial p}{\partial t} + c^2\rho\nabla \cdot \boldsymbol{v} = 0 \quad \text{in} \quad \Omega_\mathrm{A} \setminus \hat{\Omega}_\mathrm{A} \times [0, T], \tag{8.2}$$

$$p = p_\mathrm{m} \qquad \text{on} \quad \Gamma_\mathrm{A}^\mathrm{mon} \times [0, T], \tag{8.3}$$

$$\boldsymbol{v} \cdot \boldsymbol{n} - \frac{1}{c\rho}p = 0 \qquad \text{on} \quad \hat{\Gamma}_\mathrm{A}^\mathrm{mon} \times [0, T], \tag{8.4}$$

where p_m is obtained from the actual pressure measurement values $\boldsymbol{P}_{\mathrm{m},t_k}$ by interpolation and projection. Equations (8.1)–(8.4) are solved backwards in time starting from zero fields. At the inner boundary $\hat{\Gamma}_\mathrm{A}^\mathrm{mon}$, the pressure values are monitored to generate the new set of measurement values $\hat{\boldsymbol{P}}_{\mathrm{m},t_k}$. Solving the above problem is the transfer operator for

$$\boldsymbol{P}_{\mathrm{m},t_k} \text{ at } \Gamma_\mathrm{A}^\mathrm{mon} \quad \rightarrow \quad \hat{\boldsymbol{P}}_{\mathrm{m},t_k} \text{ at } \hat{\Gamma}_\mathrm{A}^\mathrm{mon}.$$

The new set of measurement data $\hat{\boldsymbol{P}}_{\mathrm{m},t_k}$ contains pressure values for all nodes on the artificial measurement boundary. The time levels t_k are determined by the time step size for the time integration. If the time step size of the measurement data is different, the pressure values are interpolated. An interesting side effect of the reduction of the geometrical size of the computational domain hence is that the number of time steps to be performed for one forward or adjoint solve of the acoustic wave equation is significantly reduced as well. This is due to the fact, that the

signals have to cross the entire distance between $\Gamma_{\mathrm{A}}^{\mathrm{mon}}$ and $\hat{\Gamma}_{\mathrm{A}}^{\mathrm{mon}}$ before non-zero pressure values arrive at $\hat{\Gamma}_{\mathrm{A}}^{\mathrm{mon}}$. All zero values are cropped out for the reconstruction of the reduced domain.

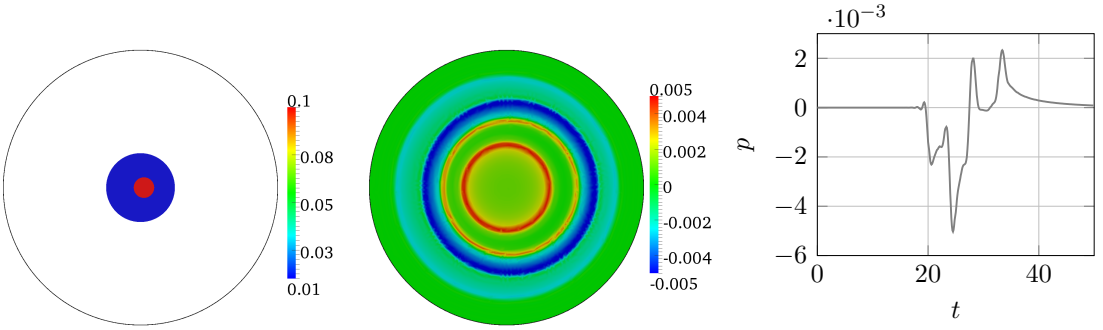
The subsequent optimization on the reduced domain $\hat{\Omega}_{\mathrm{A}}$ is performed as usual but with PMLs at its outer boundary (see Chapter 4 for a description of PMLs). The usage of PMLs is necessary because the optimization on the reduced domain $\hat{\Omega}_{\mathrm{A}}$ is significantly more sensitive to spurious reflections compared to the full domain scenario.

Note that it is not possible to carry out the reduction simulation (8.1)–(8.4) with a PML instead of an absorbing boundary condition at the inner boundary $\hat{\Gamma}_{\mathrm{A}}^{\mathrm{mon}}$. No stable configuration is found for a ring shaped PML with the interface to the actual domain at the outer boundary (i.e., no definition for $\boldsymbol{\gamma}$ such that the eigenvalues of $\boldsymbol{A}$ are equal to or larger than zero, see Chapter 4).

8.3 Numerical Evidence

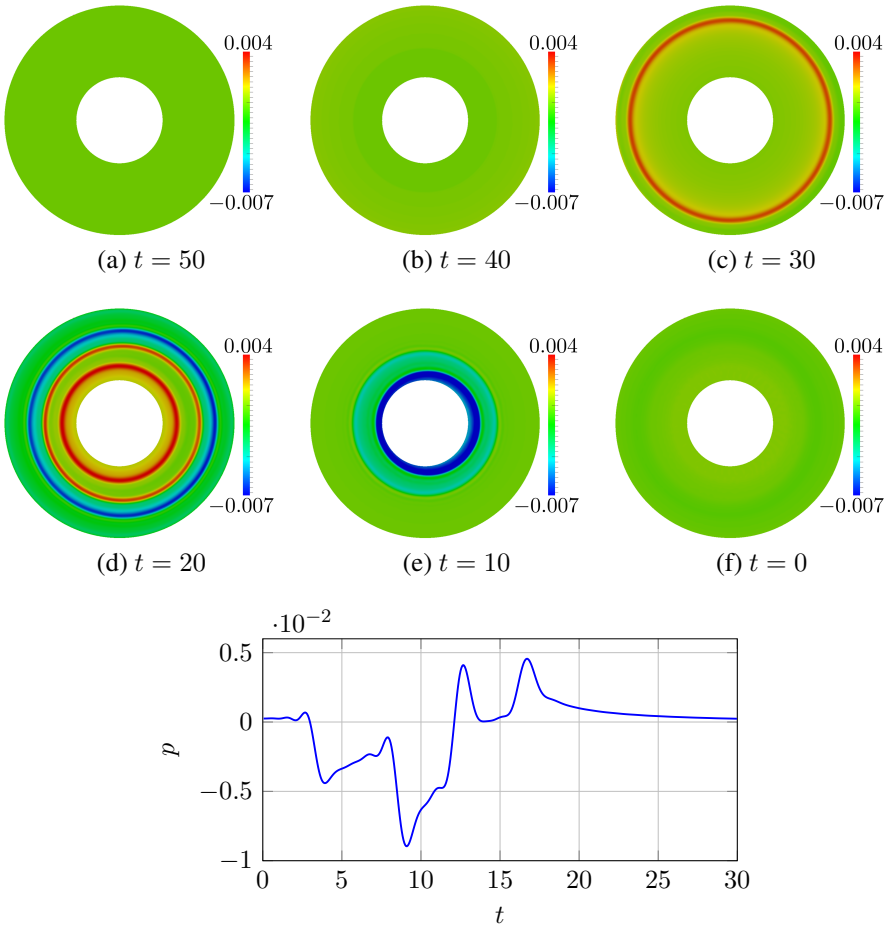
Based on a representative two-dimensional geometry, evidence for the applicability and the benefit in terms of computational cost are demonstrated in the following.

Measurement data is created with a circular geometry of radius $r = 40$ for the acoustical domain and radius $r = 10$ for the optical domain. The optical domain contains a circular inclusion of radius $r = 3$ with center at $\boldsymbol{x} = (1, 0)$. The optical properties for the generation of measurement data are $D = 0.5$ in the entire optical domain, $\mu_{\mathrm{a}} = 0.1$ within the inclusion, and $\mu_{\mathrm{a}} = 0.01$ in the rest of the optical domain. The acoustical parameters are $c = 1.5$ and $\rho = 1$ throughout the entire acoustical domain. The domains Ω_{A} and Ω_{L} consist of 8042 quadratic and 550 linear elements, respectively. For all simulations, the low-storage Runge–Kutta scheme LSRK3(3) is used with the time step $\Delta t = 0.015$ for acoustical time discretization. Figure 8.2 shows the absorption coefficient distribution in the optical domain, the pressure at $t = 15$, and the pressure over time for one of the detectors. The first signal arrives at the boundary at about $t = 20$ because it has to travel through the entire coupling medium from $r = 10$ to $r = 40$ before detection.



(a) Absorption coefficient μ_{a} (b) Pressure at $t = 15$ (c) Pressure values at $(x_1, x_2) = (40, 0)$

Figure 8.2: Generation of measurement signals.



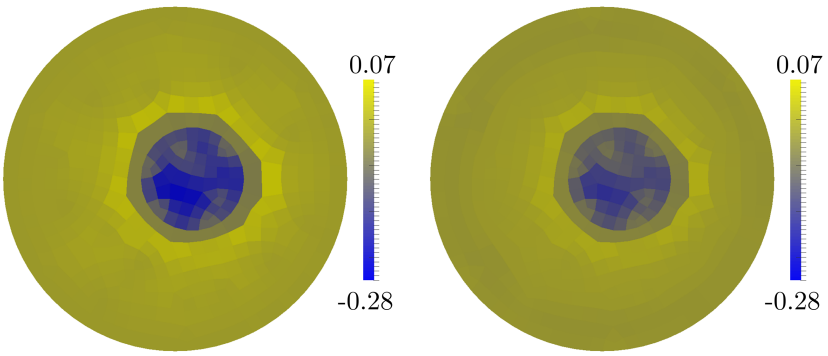
(g) Transformed measurement values resulting from reduction simulation evaluated at $(x_1, x_2) = (15, 0)$

Figure 8.3: Pressure in reduction simulation.

8.3.1 Full View

In this section, numerical evidence is given for the full view scenario, i.e., the entire acoustical boundary is monitored. The limited view scenario is addressed in Section 8.3.2 The reduction simulation is run on a ring with outer radius $r = 40$ and inner radius $r = 15$. The mesh is the same as in the simulation to generate measurement data and consists of 6806 elements. The measurement pressure values are applied on the outer ring as Dirichlet condition and new monitor values are created at the inner ring where the first order absorbing boundary condition is applied as described in equations (8.1)–(8.4). Figure 8.3 shows the backwards traveling pressure at various points in time and one of the transformed measurement signals. As can be seen by comparing Figure 8.2(c) and Figure 8.3(g), the values are qualitatively very similar. The main differences are that the transformed signal starts much earlier and that the amplitudes are higher, which is due to the fact that the signals travel inwards from a ring and accumulate.

After completion of the reduction simulation, a reconstruction on the reduced domain is carried out. The acoustical domain $\hat{\Omega}_\mathrm{A}$ consists of the same mesh as the original domain Ω_A except



(a) Full domain (b) Reduced domain

Figure 8.4: Absorption gradient in the optical domain resulting from simulations on the full and reduced acoustical domain.

that it is cropped at radius $r = 17.2$. The PML region extends from $r = 15$. The mesh in the reduced domain consists of 1626 elements. For all simulations, the low-storage Runge–Kutta scheme LSRK3(3) is used with the time step $\Delta t = 0.015$ for acoustical time discretization. The image reconstruction is started from a uniform absorption coefficient distribution $\mu = 0.01$. Figure 8.4 shows the absorption coefficient gradient in the initial run for a simulation on the full acoustical domain for reference in panel (a) and on the reduced acoustical domain in panel (b). Qualitatively, the gradients are very similar. Quantitatively, the simulation on the reduced domain yields smaller gradient values, which is due to the fact that reflections at the artificial boundary appear. Also, the influence of dissipation and dispersion errors is different in the reduced setup. Comparing the numerical values of the gradients in the inclusion shows an average deviation of 12.7%.

The computational expense for the acoustic forward and adjoint run is significantly decreased. It is proportional to the number of acoustical elements, which is reduced from 8042 to 1626 corresponding to a speedup factor of 4.95. The speedup is even higher for smaller objects, larger phantoms, or in three-dimensional simulations.

8.3.2 Limited View

The reduction procedure is basically the same for the limited view case. The setup is described in Figure 8.5. The only difference in contrast to Figure 8.1 is that the monitor boundary does not enclose the acoustical domain entirely. It is important that the reduced domain reproduces this aspect. Results would be deteriorated if monitor values are produced for the entire boundary $\partial\hat{\Omega}_\mathrm{A}$ in the reduction simulation because pressure values would be prescribed that are not representative for the measured signals but rather just side effects of the numerical concept.

Figure 8.6 shows the resulting absorption coefficient gradient in the first run in panel (c) and repeats the gradients from the full view scenario on the full and reduced domain for comparison. The gradient values are lower because the accumulated source term on the adjoint problem is smaller simply due to the fact that the monitor boundary is smaller and consists of fewer detectors. For the full view, the values outside of the inclusion seem to only depend on the distance

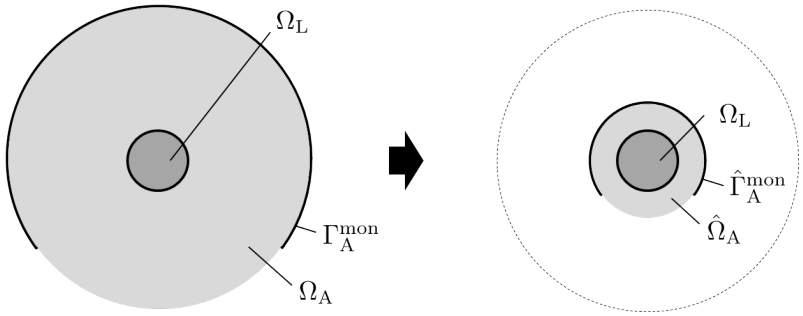
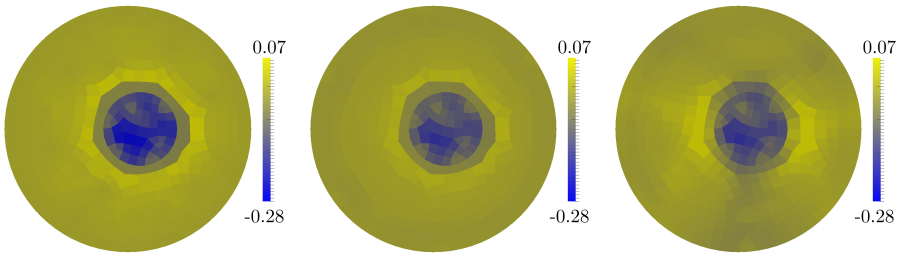


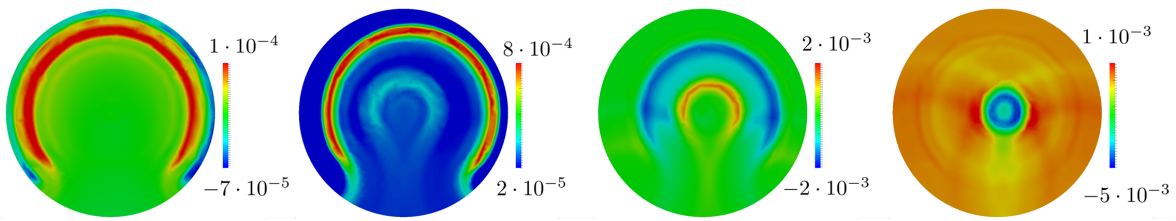
Figure 8.5: Scheme of concept to reduce the computational domain size for the limited view case.



(a) Full domain (b) Reduced domain (c) Reduced domain with limited view

Figure 8.6: Absorption gradient in the optical domain resulting from simulations on the full, the reduced acoustical domain, and the reduced acoustical domain with limited view.

to the inclusion. For the limited view, the values also show deviations depending on the orientation, which is traced back to the source term that depends on the angle as shown in Figure 8.7, where pressure snapshots for the adjoint problem are given at various points in time. The rotational symmetry is not completely restored due to the missing measurement values, see the green coloring below the inclusion in Figure 8.7(d).



(a) $t = 15$ (b) $t = 10$ (c) $t = 5$ (d) $t = 0$

Figure 8.7: Snapshots of the pressure in the adjoint run for the limited view scenario.

9 Opposing the Ill-Conditioning

Note: The following chapter is presented with reference to [143] and [147] and some parts are quoted literally.

The optoacoustic image reconstruction method as proposed in Chapter 7 is far more general than common optoacoustic image reconstruction algorithms. It allows for reconstruction of the optical absorption, as well as the acoustic material properties, i.e., speed of sound and mass density on a variety of tomographic setups. This flexibility and versatility are accompanied by an increase of the ill-conditioning of the inverse problem. Conditioning generally denotes the sensitivity of a function with respect to its input. Ill-conditioning means that small changes in the input can yield high deviations in the output. This automatically implies that small errors in the input can distort the output (i.e., the result) significantly. Also, the general well-posedness of the optoacoustic inverse problem must be questioned. A problem is said to be well-posed if a solution exists, is unique, and the output depends continuously on the input [69]. For optoacoustic imaging, the uniqueness of the solution depends on the unknown parameters, the illumination setup, completeness of measurement data, and many more, as explicated in Section 6.2.

The inverse problem (7.19) is strongly ill-conditioned. Two approaches to oppose the ill-conditioning are presented. In the first approach, the basis functions used for the parameter discretizations are optimized. Typically, voxel or pixel based basis functions are chosen. However, there are several other possibilities, e.g. local radially symmetric basis functions as proposed in [151]. The authors of [151] claim an improvement of convergence compared to local polynomial basis functions due to implicit regularization of a low-dimensional solution space if the shape parameters of the bases are chosen appropriately. In [27, 152], computationally expensive level-set functions are used to reconstruct inclusion shapes and inclusion parameter values. Their application is limited to problems with only a few inclusions of regular shape. Here, a basis approach is presented that starts from a classical voxel basis and consolidates voxels to patches, thereby introducing implicit regularization and making use of the biological partitioning into several clustered materials. Additionally, material distribution patterns are communicated from the most sensitive parameter to the less sensitive parameters, thereby improving the conditioning of the inverse problem. This approach is more flexible and robust compared to level-set reconstructions and does not require the identification of appropriate shape parameters for the bases. In this way, distinct inclusions can be reconstructed and implicit regularization is introduced while maintaining full flexibility. The approach can be thought of as a segmentation during reconstruction without strong parameter dependence. The concept of patched basis functions is derived and demonstrated in Section 9.1.

The second method to oppose the ill-conditioning is presented with reference to [143, 147]. Material identification is reasonable when the composition of an object and typical values for the material properties are known in advance. This is often the case in medical applications where only a limited number of tissue types is present, e.g., soft tissue, skin, bone, brain, muscle, organs, etc. and typical values are given in literature [29, 57]. In [143], the absorption and diffusion

coefficients are reconstructed and are used to identify materials from a user specified material catalog to update the acoustical properties accordingly. A drawback however is that the diffusion coefficient has low sensitivity in typical optoacoustic setups and its reconstruction is error prone. Here, it is proposed to reconstruct only the absorption coefficient and then utilize the acoustical gradients to find a unique assignment to materials from the catalog, see also [147]. One advantage is a reduction of computational time because only one parameter is reconstructed and the acoustical parameters are updated on the fly. The other advantage is that available prior knowledge is brought into the reconstruction scheme without restricting the generality of the approach. The method is derived and its applicability is demonstrated in Section 9.2.

9.1 Patched Parameter Basis Functions

A typical discretization of parameter fields is achieved by basis functions and scaling, i.e., exemplary for the absorption coefficient

$$\mu_{\mathrm{a}h}(\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x})\boldsymbol{\mu}_{\mathrm{a}},$$

with $\boldsymbol{b}(\boldsymbol{x})$ containing basis functions and $\boldsymbol{\mu}_{\mathrm{a}}$ containing values to scale the basis functions. A common approach to discretize the parameter distributions $\mu_{\mathrm{a}h}, D_h, c_h, \rho_h$ is as element-wise constant parameters where $\boldsymbol{b}^{\mathrm{e}}(\boldsymbol{x})$ contains basis functions that are 1 in one element of the underlying mesh $\mathcal{T}_{\mathrm{L}}^h$ and 0 in the others. This is an intuitive approach when using finite elements since parameters are commonly stored element-wise. However, this is not necessarily the best approach. Usually, the solver for the physical problem and the solver for the inverse problem have different demands on the spatial discretization. If the gradient of the objective function is calculated using finite differences, one favors to keep the number of model parameters to a minimum, which is not possible for element-wise parameter discretizations in combination with a physical solver being restricted to certain accuracy criteria. Also, inverse problems suffer from ill-conditioning, which can be counteracted by basis functions that are tailored to the respective inverse problem, e.g. by incorporating prior knowledge concerning the parameter distribution properties.

In this section, a new type of parameter basis consisting of Patched Basis Functions (PBF) is introduced. Biological bodies as well as engineering components consist of different tissue/material types with distinct differences in their properties. Within one tissue type, slight variations of the material properties are present, while discontinuities may appear from one type to the other. Also, one tissue type is spatially connected, i.e., clustered and generally not scattered. The proposed basis imitates these features by clustering several neighboring elements to a patch. Patches are built from an elementwise quantity by collecting connected elements of similar value. The patch summarizes all element-wise basis functions to one patched basis function. After setting up all patches, one does no longer have n^{ele} but n^{patch} basis functions that are 1 inside the respective patch and 0 outside the patch. Thereby, implicit regularization is introduced, while full flexibility is maintained, because the patches are rebuilt in every iteration. Additionally, the approach allows to transport information of a body's composition from the most sensitive parameter to less sensitive parameters by creating PBFs on the most sensitive parameter and reusing the created patches for the other parameters. Thus the ill-conditioning is strongly improved. Note that the method is based on the assumption that general bodies have distinct materials with sharp

boundaries over which discontinuities in the parameters appear. Therefore, the method is not restricted to optoacoustic image reconstruction and can be applied to any other inverse problem satisfying this assumption.

9.1.1 Parameter Basis Construction

Starting point is an element-wise quantity $\boldsymbol{q}^{\mathrm{e}}$, e.g. a vector containing element-wise parameter values $\boldsymbol{q}^{\mathrm{e}} = \boldsymbol{\mu}_{\mathrm{a}}$ or an element-wise parameter gradient $\boldsymbol{q}^{\mathrm{e}} = \boldsymbol{g}_{\mu_{\mathrm{a}}}$. This quantity must contain information on the material type distribution. It is then used to build patches, i.e., to cluster elements, which appear to be of similar type. Figure 9.1 shows the basic idea of the approach. Starting point is an element-wise discretization of quantity q with basis $\boldsymbol{b}^{\mathrm{e}}$ and values $\boldsymbol{q}^{\mathrm{e}}$. As can be seen from the figure, the parameter has one region on the left and one on the right where it is almost zero. In three interior elements, it has a distinct value greater than zero and the three elements have similar amplitude. It therefore can be concluded that the object consists of three distinct material patches and the correspondent patched basis function should be as in the last row of Figure 9.1 with only three basis functions associated to the three materials. In the following, a procedure is developed to derive such basis functions.

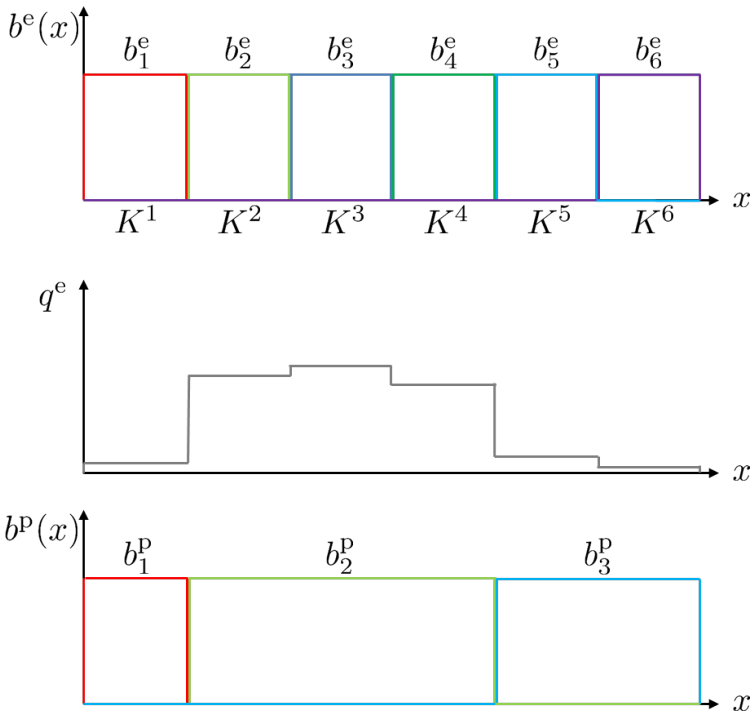


Figure 9.1: Element-wise basis functions $b^{\mathrm{e}}(x)$, distribution of quantity $\boldsymbol{q}^{\mathrm{e}}$, and patched basis functions $b^{\mathrm{p}}(x)$.

First, the range of the quantity $\boldsymbol{q}^{\mathrm{e}}$ is determined, i.e., its minimal and maximal value. The range is denoted $r = |\max(\boldsymbol{q}^{\mathrm{e}}) - \min(\boldsymbol{q}^{\mathrm{e}})|$. The element with the maximal value is used to create the first patch. With a user defined ratio α, all neighbor elements whose values of $\boldsymbol{q}^{\mathrm{e}}$ are in the interval $[\max(\boldsymbol{q}^{\mathrm{e}}) - \alpha r, \max(\boldsymbol{q}^{\mathrm{e}})]$ are associated to the original element. If a neighbor is added, its neighbors are checked as well. Hence, the criterion to build a patch is on the one hand

the similarity of value, and on the other hand the geometrical connection. For the next patch, the element with the maximum value of $\boldsymbol{q}^{\mathrm{e}}$, which is not yet associated to a patch, is used as starting point. This is repeated until every element is associated to a patch. Algorithm 8 describes the procedure schematically. It consists of two parts: a while loop terminating as soon as all elements are assigned a patch ID and a for loop averaging the elemental gradients along the patches. The algorithm makes use of a helper function 'check neighbors' to check if a neighboring element belongs to the same patch, see Algorithm 9. Special care must be taken for discretizations distributed along several processors, since the PBF does not only rely on value clustering but also on geometrical clustering. In serial, 'check neighbors' can be a simple recursive function. In parallel, the elements that need a check must be stored and communicated before the function can be called again.

Algorithm 8 Transformation to patches

determine $\max \boldsymbol{q}^{\mathrm{e}}$, $\min \boldsymbol{q}^{\mathrm{e}}$ and $r = |\max \boldsymbol{q}^{\mathrm{e}} - \min \boldsymbol{q}^{\mathrm{e}}|$
$p = 0$
while not all elements assigned to a patch **do**
  $p \leftarrow p + 1$
  find element with the maximal unassigned value and assign patch ID $P_{\mathrm{ID}} = p$
  call to function 'check neighbors'
end while
for $q = 0; q < p$ **do**
  sum the gradient values of all elements with patch ID q
  divide sum by number of elements
  write averaged value to all elements of patch q
end for

Algorithm 9 Check neighbors

for all row neighbor elements **do**
  if $q^{\mathrm{e}} \in [\max(\boldsymbol{q}^{\mathrm{e}}) - \alpha r, \max(\boldsymbol{q}^{\mathrm{e}})]$ and neighbor element not yet assigned **then**
    assign patch ID $P_{\mathrm{ID}} = p$
    store neighbors IDs for later check
  end if
end for
communicate stored neighbor IDs to all processors
for all stored neighbor IDs **do**
  call to function 'check neighbors'
end for

Several options to build patches are available and provide advantages depending on the sensitivities and parameter distributions of the problem at hand.

- PBF self: All parameters build patches according to their own gradient ($\boldsymbol{q}_{\mathrm{e}}^{\mu_{\mathrm{a}}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_{\mathrm{a}}}$, $\boldsymbol{q}_{\mathrm{e}}^{c} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{c}}$, $\boldsymbol{q}_{\mathrm{e}}^{\rho} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\rho}}$).
- PBF abs grad: The absorption coefficient builds patches according to its own gradient, the other parameters use these patches ($\boldsymbol{q}_{\mathrm{e}}^{\mu_{\mathrm{a}},c,\rho} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_{\mathrm{a}}}$).

- PBF abs vals: The absorption coefficient builds patches according to its own gradient ($\boldsymbol{q}_{\mathrm{e}}^{\mu_{\mathrm{a}}} = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{\mu}_{\mathrm{a}}}$), the other parameters build patches according to the absorption coefficient ($\boldsymbol{q}_{\mathrm{e}}^{c,\rho} = \boldsymbol{\mu}_{\mathrm{a}}$).
- PBF mixed: The absorption coefficient does not build patches, speed of sound and mass density build patches according to the absorption coefficient ($\boldsymbol{q}_{\mathrm{e}}^{c,\rho} = \boldsymbol{\mu}_{\mathrm{a}}$).

9.1.2 Numerical Example

The example from Section 7.7.4 is used to study the solution behavior for PBF, where the inverse crime is avoided by usage of a spatial discretization that does not conform with the inclusion shapes and addition of noise. The results in terms of convergence of the objective function and convergence of the error in the parameter fields are presented in Figure 9.2. As in the preceding chapters, the error in the parameter fields is calculated as the square of the difference between the actual and the expected parameter values and summed over all elements. The resulting images are summarized in Figure 9.3. In terms of the objective function, all approaches converge and none aborts early. However, the convergence is slower compared to the element-wise parameter discretization because the number of effective degrees of freedom of the parameter discretization is lower. In terms of the error in the parameter fields, all methods converge except for the error of the mass density for PBS abs grad and PBF self. In all measured quantities, PBF abs grad and PBF self deliver the slowest convergence. PBF mixed and PBF abs vals yield better convergence in the parameters compared to the element-wise discretization and the errors are significantly smaller. This also manifests as visual impression from the images given in Figure 9.3. It is important to note, that better images do not necessarily yield a lower objective function value.

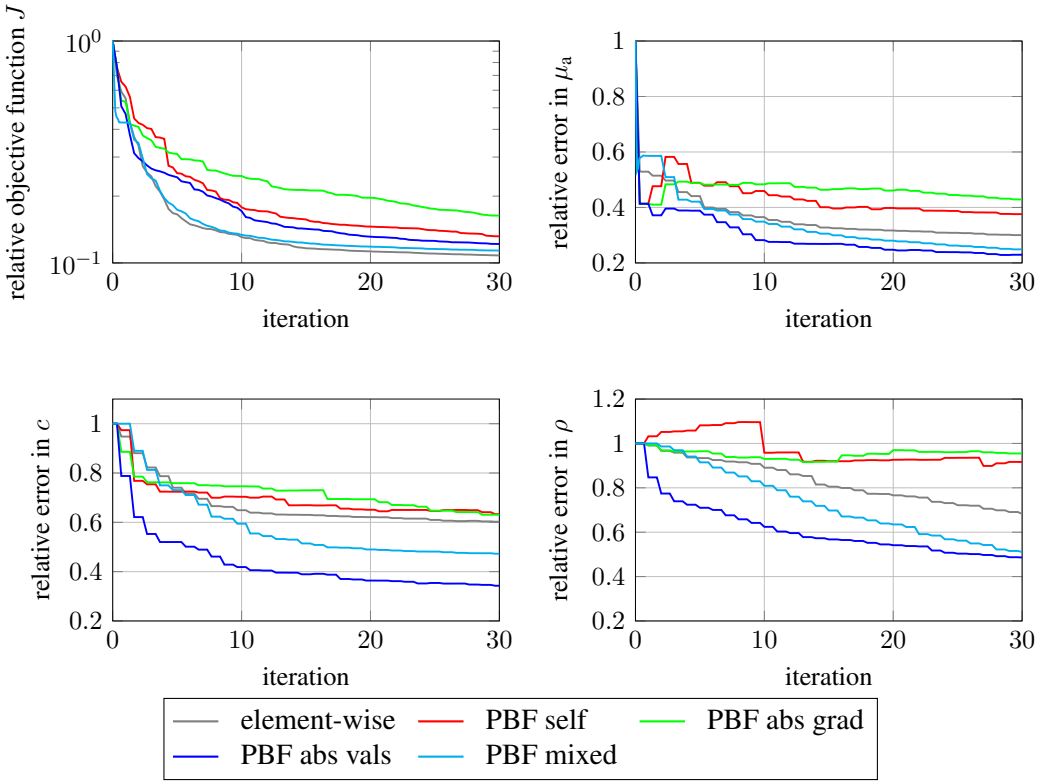


Figure 9.2: Convergence of the objective function and the parameters for nonconforming discretization with 10% noise level for several patch types. The gray lines are obtained by an element-wise parameter discretization as in the numerical examples in Section 7.7.4.

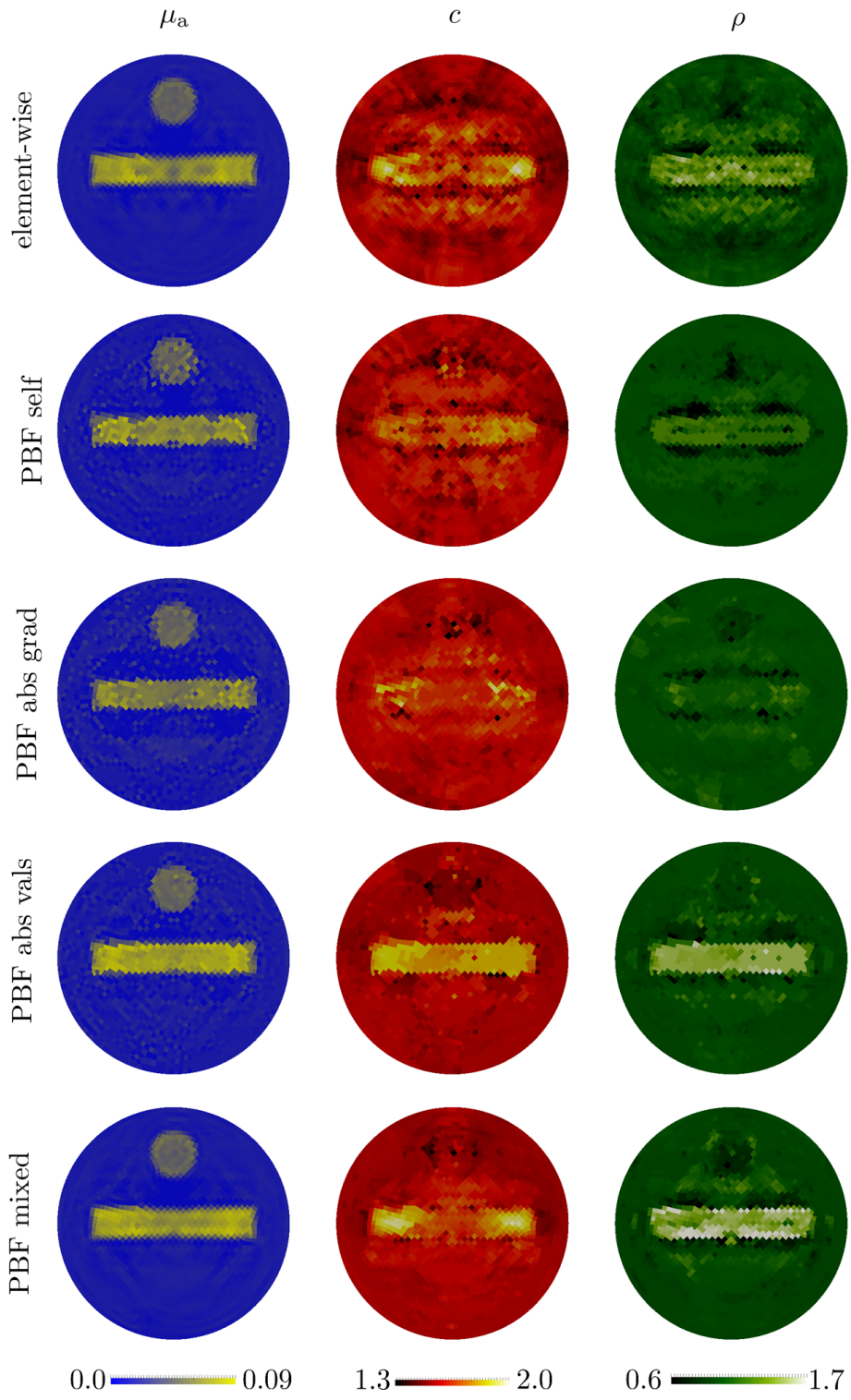


Figure 9.3: Images after 30 optimization iterations of all parameters sequentially in each iteration for nonconforming discretization with 10% noise level for several patch types. The left column, middle column, and right column show the absorption coefficient, speed of sound, and mass density, respectively. The rows represent the different patch types as labelled.

9.2 Material Identification

Generally, the composition of an examined object is known in advance. Depending on the imaging purpose and the object, different types of tissue are expected, for example muscle, brain, lung, bone, fat, and so forth. Literature supplies experimentally determined values or at least ranges of values for optical as well as acoustical material properties for these tissues, see for example [29, 57]. The basic idea is to take advantage of the fact that tabulated material properties from literature or experiments are known and specific values are expected. This expectation is made available for the reconstruction algorithm by providing a material catalog.

A biological tissue $\mathcal{M}_j$ is characterized by the parameter quadruple $\mathcal{M}_j = (\mu_j, D_j, c_j, \rho_j)$ where $j = 1, ..., n_{\text{mat}}$. Within one tissue type naturally slight variations occur. Between two tissue types, a sharp interface often exists, consider e.g. bones, organs, or individual constituents of organs. Before optimization, a set of n_{mat} materials $\mathcal{M}_j$ is defined by the user. This set contains parameters for all expected materials and is denoted as material catalog. The approach presented in [143] reconstructs absorption and diffusion coefficient to find a unique mapping between optical values and listed materials in case similar absorption coefficients are cataloged (witht first ideas developed in [59]). The diffusion coefficient however has a very low sensitivity and its reconstruction is error prone. Therefore, a new material identification technique was developed as presented in [147] that does not rely on the reconstruction of the diffusion coefficient. Instead, it uses the acoustical gradients to identify materials correctly. The absorption coefficient is reconstructed with the iterative reconstruction procedure on a pixel-based or PBF discretization and the available acoustical gradients are then utilized for the identification process.

9.2.1 Consideration of Acoustical Gradients

The detailed procedure for material identification is displayed in Algorithm 10. For input, the user has to specify a list of expected materials containing values of absorption coefficient, diffusion coefficient, speed of sound, and mass density. Only the absorption coefficient is optimized in a line search procedure and the material identification is carried out after every successful line search. First, step lengths α_c, α_ρ for c, ρ are calculated as the smallest step for which $c - \alpha_c \nabla c$ covers the range $[\min c, \max c]$ from the materials in the catalog and analogously for the mass density. Then, materials are identified that match the current element absorption coefficient. If no material is applicable, the default values for soft tissue are set. If exactly one material is applicable, the corresponding acoustical values are set. If several materials are applicable, the material for which the acoustic gradients indicate the correct trend is chosen, i.e., decrease or increase. If the acoustic gradients cannot identify any material or cannot identify one material uniquely, the one with the smallest distance to the values of the range covering distribution $c^{\text{ran}} = c - \alpha_c \nabla c$ and $\rho^{\text{ran}} = \rho - \alpha_\rho \nabla \rho$ is set.

9.2.2 Numerical Example

The considered numerical example is similar to the one from the preceding section. However, it is slightly simplified and only the rectangular inclusion is examined and the circular inclusion is not considered. For the generation of measurement data, the materials as given in Table 9.1 are used. Therein, $\mathcal{M}_{\text{w}}$ represents the coupling medium, $\mathcal{M}_{\text{o}}$ the object's background, and $\mathcal{M}_{\text{r}}$

Algorithm 10 Material identification

```
determine step length for c, ρ
for all elements e do
  find all materials M_j that contain μ_a
  if no material found then
    set default acoustical materials
  else if one material is found then
    set acoustical materials of the found material
  else
    check if the gradient directions indicate materials
    if no material direction applies then
      determine material with smallest distance and set c, ρ
    else if one material supplies indicated direction then
      set c, ρ
    else
      determine material with smallest distance and set c, ρ
    end if
  end if
end for
update speed of sound and density distribution
```

	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$D[\mathrm{mm}]$	$c\left[\frac{\mathrm{mm}}{\mu\,\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_\mathrm{w}$	0.0	0.5	1.5	1.0
$\mathcal{M}_\mathrm{o}$	0.01	0.5	1.5	1.0
$\mathcal{M}_\mathrm{r}$	0.1	0.5	2.0	2.0

Table 9.1: Material properties for the generation of measurement data.

the material of the rectangular inclusion. As in Section 7.7.2, data generation is run on a finer discretization and the reconstruction relies on the base discretization rotated by $45°$.

Four different setups I–IV are studied to demonstrate different properties of the material identification method. All reconstruction parameters are as in the examples in Section 7.7. The only input that varies between setups I–IV is the material catalog as specified in Table 9.2. Since the diffusion coefficient is not subject to reconstruction and spatially constant, it is not explicitly mentioned but always set to 0.5. Also the coupling medium is excluded from reconstruction. Setup I is conditioned best, because the material catalog perfectly represents the expected materials. Setup II has an additional material, which lies in between the two expected materials. Setup III has a third material with the same absorption coefficient but lower values for the acoustical coefficients. Therefore, it tests the material identification due to the acoustical gradients. Setup IV has one material with the absorption coefficient in between the expected values but opposite acoustical coefficients.

Figure 9.5 shows the resulting images after 30 optimization iterations. For reference, one simulation (denoted ‘standard’) is run without material identification. In Figure 9.4, the convergence of the objective function and the parameter fields is shown. Setups I–III are superior to the stan-

setup I	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$c\left[\frac{\mathrm{mm}}{\mu\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_1$	0.01	1.5	1.0
$\mathcal{M}_2$	0.1	2.0	2.0

setup II	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$c\left[\frac{\mathrm{mm}}{\mu\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_1$	0.01	1.5	1.0
$\mathcal{M}_2$	0.1	2.0	2.0
$\mathcal{M}_3$	0.05	1.75	1.5

setup III	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$c\left[\frac{\mathrm{mm}}{\mu\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_1$	0.01	1.5	1.0
$\mathcal{M}_2$	0.1	2.0	2.0
$\mathcal{M}_3$	0.1	1.0	0.6

setup IV	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$c\left[\frac{\mathrm{mm}}{\mu\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_1$	0.01	1.5	1.0
$\mathcal{M}_2$	0.1	2.0	2.0
$\mathcal{M}_3$	0.05	1.0	0.6

Table 9.2: Material properties for the generation of measurement data.

dard approach. The material identification speeds up the convergence and the images are very clear. For setup III some oscillations are observed in the convergence plots which are due to a false identification of the material with opposite acoustical values. After the 30 iterations, the correct material is identified. For setup II with an intermediate material, eight elements are identified with the additional material. They are located at the inclusion boundary and therefore yield even better results in the measured quantities compared to setup I with the matching material catalog. Setup IV does not yield satisfactory results. The objective function oscillates and several elements are identified with the wrong material. This behavior was expected to some extent because the additional wrong material has an absorption coefficient in between the default material and the expected inclusion material. If the step length in the line search algorithm for the absorption coefficient yields intermediate absorption coefficient values, this material is identified and permits convergence by updating the acoustical parameters in the opposite direction and thereby making an underestimation of the absorption coefficient more likely. For such a material catalog, an acoustical update should not be carried out after the completion of the line search for the absorption coefficient but within, i.e., as part of the standard parameter update such that the effect is tested by the Wolfe conditions (see Section 7.5.1). However, convergence might still not be obtained. Except for the very challenging material catalog, material identification helps to speed up convergence and improve the image quality significantly.

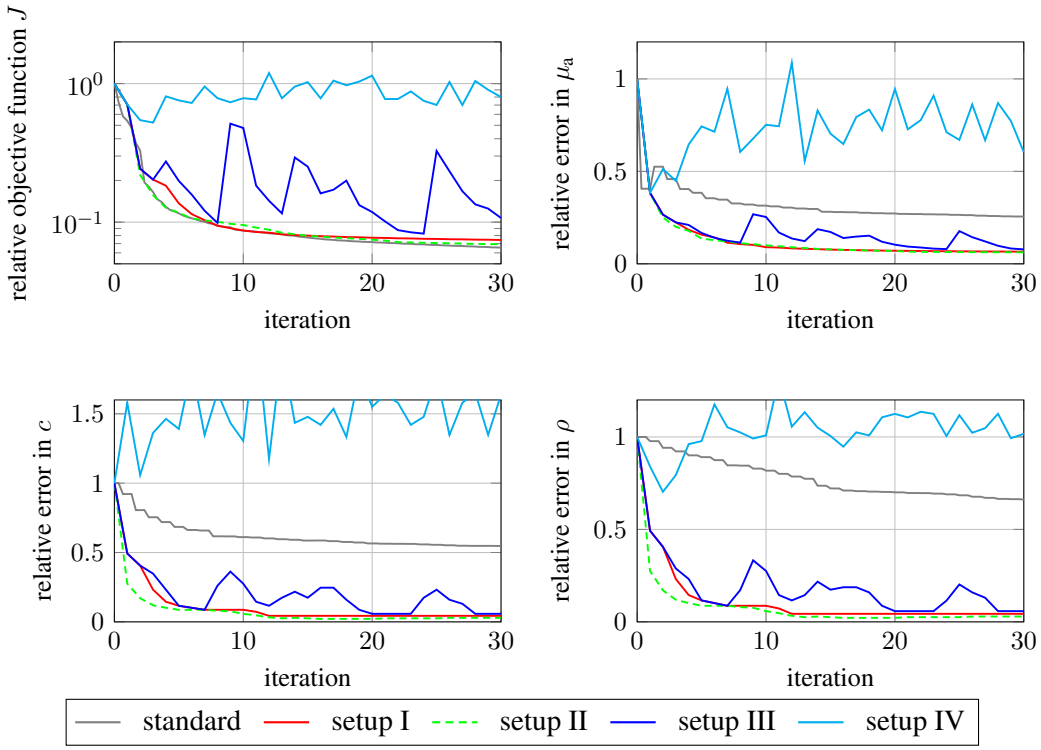


Figure 9.4: Convergence of the objective function and the parameters for different material catalogs. The gray lines are obtained by an element-wise parameter discretization without material identification.

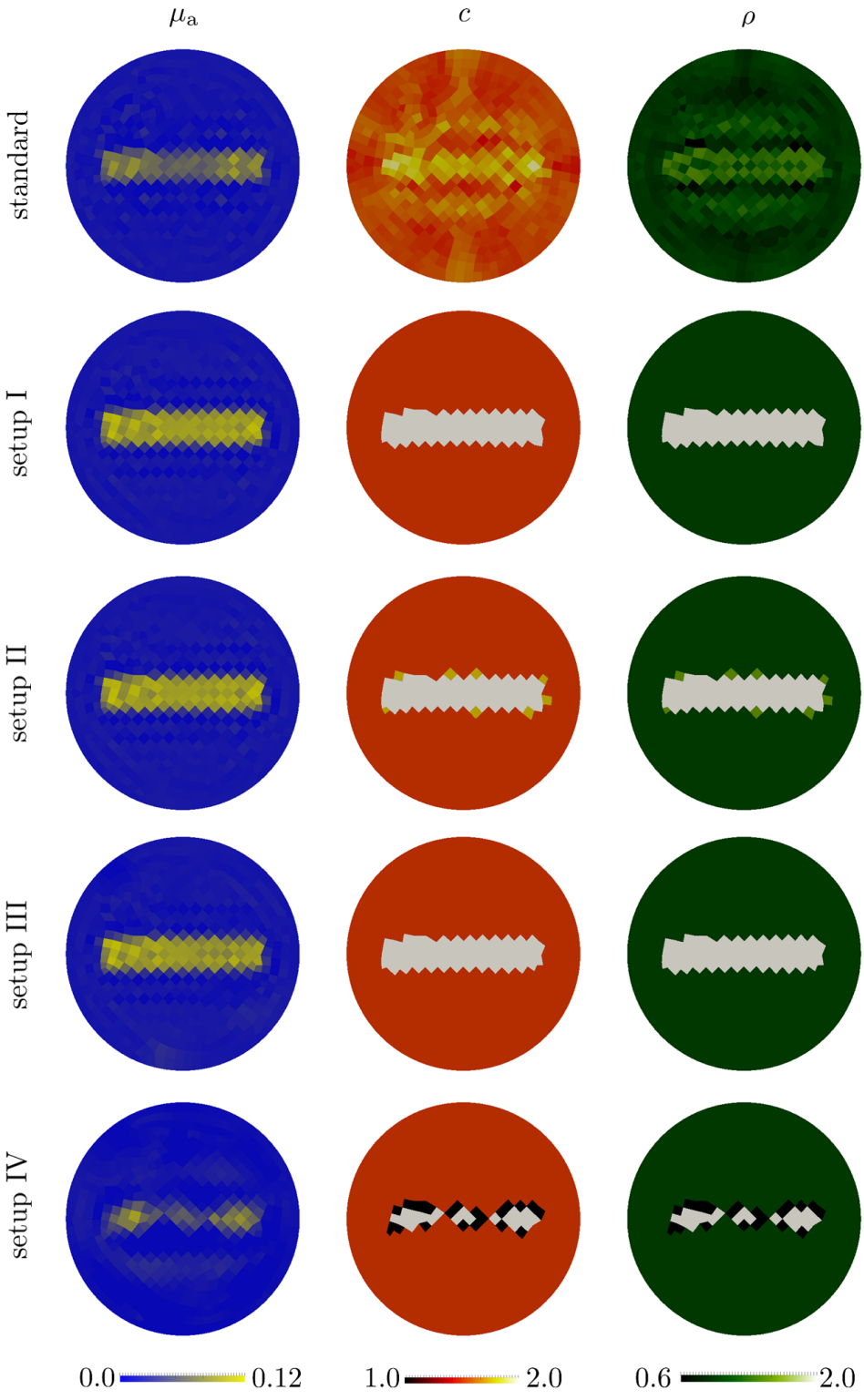


Figure 9.5: Images after 30 optimization iterations of all parameters sequentially in each iteration for different material catalogs. The left column, middle column, and right column show the absorption coefficient, speed of sound, and mass density, respectively. The rows represent the different material catalogs as labelled.

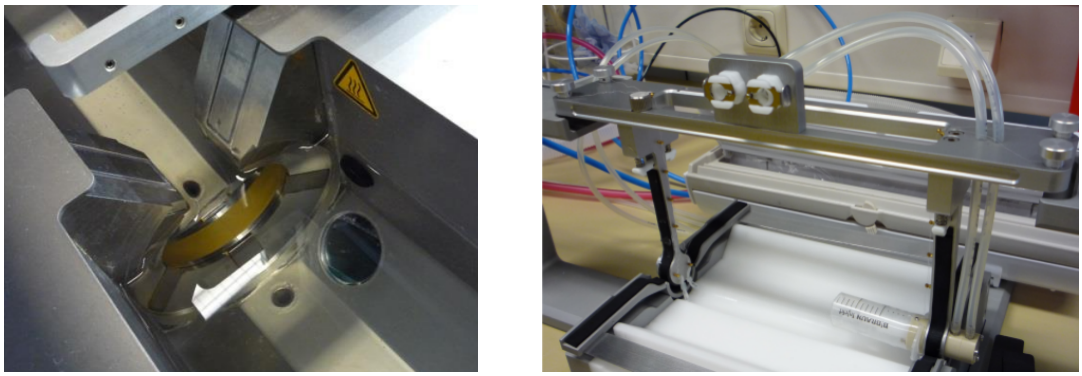
9.3 Conclusion

Optoacoustic imaging is an inherently ill-conditioned inverse problem and the ill-conditioning is worsened by the attempt to not only reconstruct the absorption coefficient but additionally the acoustical parameters. Two approaches to oppose the ill-conditioning were proposed. The adaption of the basis functions for the parameter fields does not require any additional user input and maintains the flexibility of a pure element-wise reconstruction. The results presented for the numerical example show that the convergence in terms of the objective function is slower compared to the element-wise parameter discretization. The convergence in the parameter fields, however, is improved for PBF abs vals and PBF mixed where information on the object's structure is transported from the most sensitive parameter (the absorption coefficient) to the other parameters. The material identification method requires the user to supply a list of expected material parameters, which enables to let the algorithm know what seems obvious for an experienced user. The presented results indicate that the convergence can be significantly improved. Only for very tricky material catalogs, no convergence is obtained. In general, the material identification is expected to improve results robustly if all materials approximately obey the same qualitative behavior, e.g., the higher the absorption coefficient, the higher the speed of sound. Otherwise, intermediate material identification with a false material can prevent further convergence.

Both methods to oppose ill-conditioning are applicable in the specific context of optoacoustic image reconstruction but they are applicable to a wide range of other inverse problems as well. For any kind of inverse problem involving clustered parameter distributions, a version of PBF can improve convergence and conditioning. For any inverse problem, where the knowledge of expected materials is at hand, the material identification can improve convergence. Also, both methods can be combined with traditional concepts to counter ill-conditioning, e.g. Tikhonov or total variation regularization.

10 Applications

In this chapter, the methods derived in Chapters 7, 8, and 9 are applied to experimentally obtained measurement data. In the first part of this chapter, the algorithm is applied to in-vivo mouse brain measurements. In the second part, image reconstruction of a phantom is analyzed in detail.



(a) Interior of the tomograph (b) Holder

Figure 10.1: The interior and the holder of the MSOT inVision256-T: The yellow surface covers the acoustical detectors. The dark gray rectangles on the detector ring contain the fibers to provide the laser light. The tomograph is filled with lukewarm water as coupling medium. The holder is used to place phantoms or animals in the tomograph and is supplied with an artificial ventilation system.

Both sets of measurement data were generated with the multispectral optoacoustic tomograph MSOT inVision256-TF (iThera Medical GmbH, München, Germany) with a transducer ring of 80 mm diameter covering $270°$ with 256 array elements [46]. The center frequency of the transducers is $5\,\mathrm{MHz}$ and the signals are band-pass filtered between $100\,\mathrm{kHz}$ and $6\,\mathrm{MHz}$. The acoustical signals are recorded with a sampling frequency of $40\,\mathrm{MHz}$ corresponding to a time step size of $\Delta t = 0.025\,\mu\mathrm{s}$. To decrease the influence of noise, the final signals are an average of 20 measurements. The light source is provided by a laser system with adjustable wavelength. Ten fibers provide the light from different directions to obtain a uniform illumination. For all measurements, a wavelength of $700\,\mathrm{nm}$ is chosen. Figure 10.1 shows the interior of the tomograph. A detailed sketch of the tomograph is given in Figure 1(d) of [46].

10.1 Mouse Brain Imaging

A female Hsd:Athymic Nude-Foxn1nu/nu mouse was imaged in-vivo in the specified tomograph. The animal was anesthetized with a mixture of 1.8% isoflurane in 100% O_2 at $0.8\,{}^{\mathrm{ml}}/_{\mathrm{min}}$

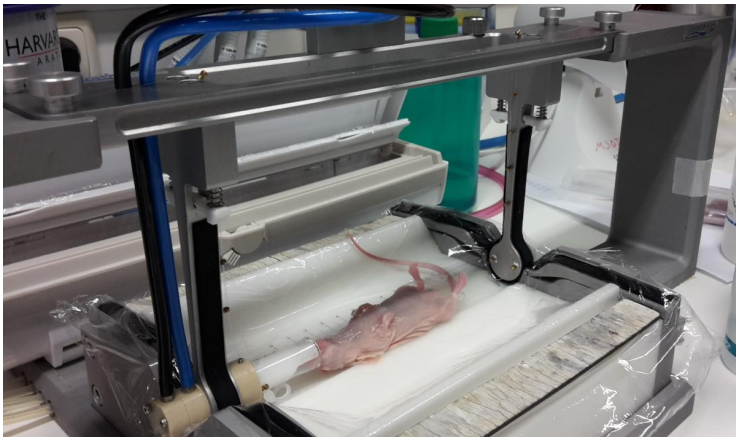


Figure 10.2: Nude mouse in animal holder.

flow rate and placed in supine position. Figure 10.2 shows the mouse in the animal holder during the experiment preparation.

10.1.1 Reduction Simulation

The first step before reconstructing images is the reduction of the computational domain as explained in Chapter 8. Therefore, a three-dimensional computational domain as displayed in Figure 10.3 is created. It consists of a cylindrical ring with outer radius $40\,\mathrm{mm}$, inner radius $12.5\,\mathrm{mm}$, and height $25\,\mathrm{mm}$. The domain is cut into five layers perpendicular to the x_3 axis. The first and last layer are PMLs of $2.5\,\mathrm{mm}$ height mimicking that the tomograph basin is larger than the computational domain. The three layers in between contain the acoustical domain where the standard wave equation is solved. All layers are divided into quarters because the detector ring covers $270°$ and three of the outer surfaces of the middle layer are the monitored surfaces where measurement values are applied. Absorbing boundary conditions are applied to all inner surfaces and new measurement data is created on three of the inner surfaces of the middle layer recreating the limited view. The middle layer is of height $8\,\mathrm{mm}$ because the measurement data is obtained in nine positions, i.e., scanning along the x_3 axis with $1\,\mathrm{mm}$ steps. The geometry is meshed with 2044800 hexahedral elements. For the shape functions, the polynomial degree $k = 2$ is chosen. The time step size is set to $\Delta t = 0.003125\,\mu\mathrm{s}$ and the time integrator LSRK3(3) is used. For the coupling medium, the acoustical material parameters are set to $\rho = 1.0\,{}^{\mathrm{mg}}\!/_{\mathrm{mm}^3}$ and $c = 1.518\,{}^{\mathrm{mm}}\!/_{\mu\mathrm{s}}$ corresponding to water at $34°\,\mathrm{C}$.

The spatial discretization yields $\approx 1.2 \cdot 10^8$ degrees of freedom for pressure and velocity and additional degrees of freedom for the auxiliary variables in the PMLs. The computational time on four nodes each with 24 cores of Intel(R) Xeon(R) CPU E5-2680 v3 operating at $2.50\,\mathrm{GHz}$ is $6909\,\mathrm{s}$ for computations and $8523\,\mathrm{s}$ for output in 14720 time steps. The result of the reduction simulation is a monitor file on the interior cylinder mantle. Figure 10.4 shows a pressure snapshot of the reduction simulation at $t = 15\,\mu\mathrm{s}$ visualizing that only $270°$ of the cylinder mantle are applied with monitor values.

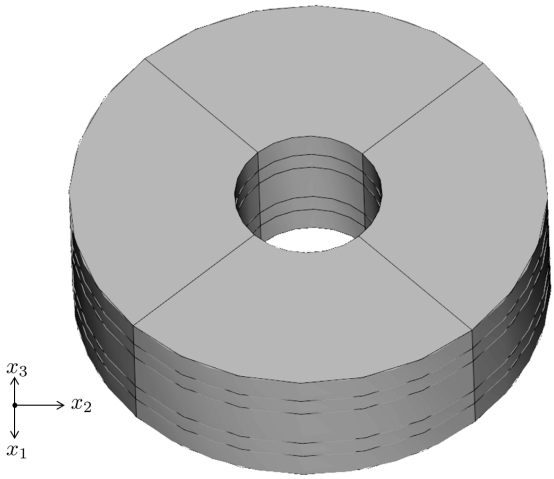


Figure 10.3: Computational domain for reduction simulation.

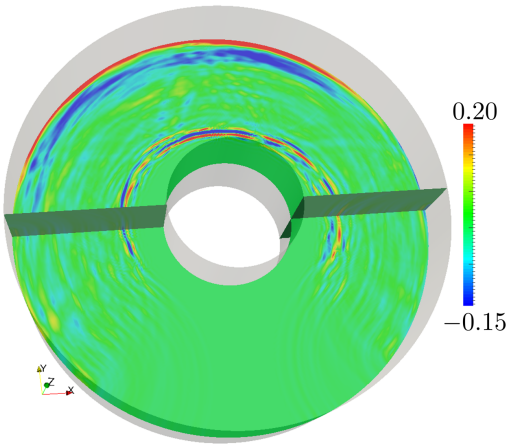


Figure 10.4: Pressure snapshot at $t = 15\,\mu\text{s}$ for the reduction simulation with the measurement data from the mouse experiment.

10.1.2 Image Reconstruction in Three Dimensions

For the image reconstruction, a cylindrical computational domain of radius $13.5\,\mathrm{mm}$ and height $12\,\mathrm{mm}$ is created. The outer $1\,\mathrm{mm}$ layers in radial and x_3 direction represent PMLs. The PMLs are terminated by absorbing boundary conditions. On $270°$ of the surface at $r = 12.5\,\mathrm{mm}$ between $x_3 = -4\,\mathrm{mm}$ and $x_3 = 4\,\mathrm{mm}$, the boundary is monitored, i.e., the objective function will be evaluated here and the source term for the adjoint problem will be applied here. The source term for the optical problem, i.e., the illumination by the laser light, is applied on the entire $360°$ of the optical cylinder mantle for $x_3 \in [-0.75\,\mathrm{mm}, 0.75\,\mathrm{mm}]$. The remainder of the optical boundary is applied with the Robin boundary condition as in equation (7.4). The optical domain is meshed with 491520 elements while the acoustical domain is meshed with 860160 linear elements ($k = 1$). The time step size is set to $\Delta t = 0.005\,\mu\mathrm{s}$ and the LSRK3(3) time integrator is used. For the PBF approach, the parameter α is set to $\alpha = 0.1$.

All reconstructions are run on four nodes each with 24 cores of Intel(R) Xeon(R) CPU E5-2680 v3 operating at $2.50\,\mathrm{GHz}$. The evaluation of one forward problem takes approximately $1000\,\mathrm{s}$ including forward optical problem, mapping, and forward acoustical problem. The acoustical part, however, is the main computational cost. The adjoint run takes approximately the same time because the additional cost for the checkpointing and omitting of writing output cancel out in terms of computational timings. Four reconstructions are run: with element-wise parameter discretization, PBF abs vals, PBF mixed, and material identification. For material identification, the following material catalog is supplied:

material	$\mu_\mathrm{a}\left[\frac{1}{\mathrm{mm}}\right]$	$c\left[\frac{\mathrm{mm}}{\mu\,\mathrm{s}}\right]$	$\rho\left[\frac{\mathrm{mg}}{\mathrm{mm}^3}\right]$
$\mathcal{M}_1$	0.01	1.53	1.0
$\mathcal{M}_2$	0.28	1.5	1.1
$\mathcal{M}_3$	0.13	1.5	1.1
$\mathcal{M}_4$	0.49	1.6	1.1
$\mathcal{M}_5$	0.13	1.6	1.1
$\mathcal{M}_6$	0.18	1.5	1.1
$\mathcal{M}_7$	0.22	1.5	1.1
$\mathcal{M}_8$	0.47	4.1	1.9

Therein, the materials $\mathcal{M}_1$ to $\mathcal{M}_8$ approximately represent the materials void tissue, skin, soft tissue, vein, artery, white matter, gray matter, and bone, respectively, with values taken from [24, 29, 108] and the table developed in [172]. Figure 10.5 shows the resulting images after three iterations and one sequence per parameter. For reference, Figure 10.6 shows a cryoslice through a mouse brain and the absorbed energy map as obtained by model based inversion. Figure 10.7 plots the convergence in terms of the relative objective function over the iterations.

The images of the absorption coefficient show the same features like the model-based image, i.e., the cortex and the temporal artery. The images are however more blurry due to the low resolution. The acoustical images obtained with element-wise parameter discretization display the skull and show fluctuations in the center of the interior. The PBF abs vals and PBF mixed reconstructions show images for the acoustical parameters that recreate the features of the absorption coefficient because these two PBF approaches transport the basis information from the absorption coefficient to the acoustical fields. For PBF abs vals, the features are more distinct

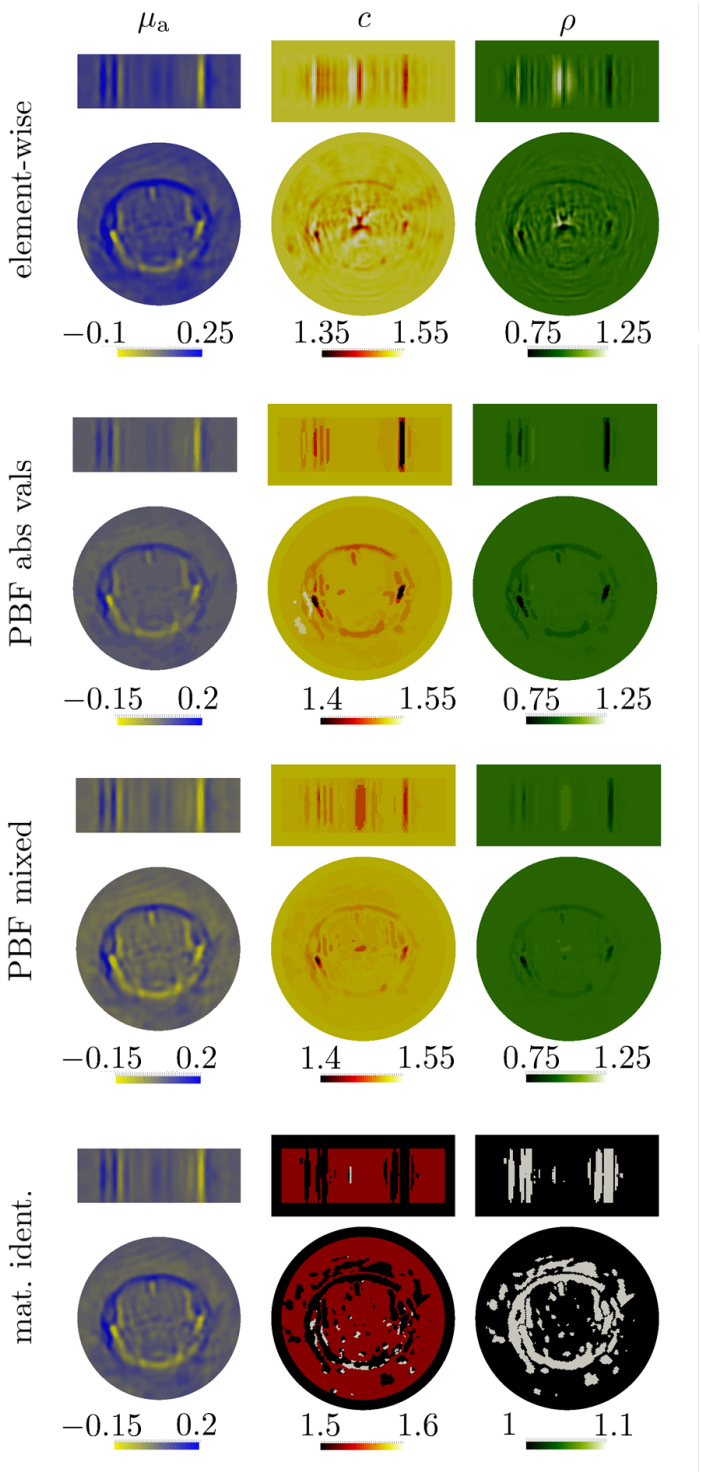
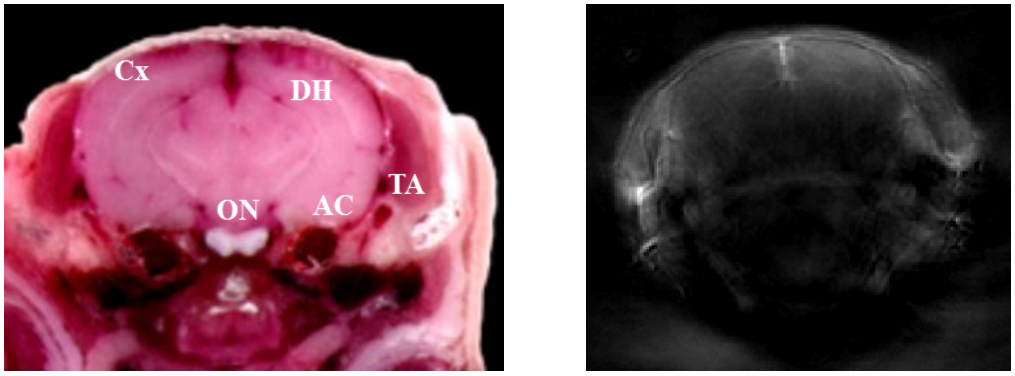


Figure 10.5: Results for image reconstruction on three-dimensional domain with element-wise parameter discretization, PBF abs vals, PBF mixed, and material identification displayed on a plane with normal vector in x_3 direction and origin $(0, 0, 0)$ and a plane with normal vector in x_2 direction and origin $(0, 0, 0)$. The rows correspond to the four setups as labeled, while the columns correspond to the absorption coefficient, the speed of sound, and the mass density (from left to right).



(a) Cryoslice through mouse brain.

(b) Absorbed energy map obtained by model based inversion.

Figure 10.6: Cryoslice through a mouse brain (Cx - cortex, DH - dorsal hippocampus, TA - temporal artery, AC - amygdala complex, ON - optic nerve) and absorbed energy map obtained by model based inversion.

compared to PBF mixed. The material identification does not identify skull because the reconstructed absorption coefficient has too low values. However, the topography is represented. In terms of convergence of the relative objective function (Figure 10.7), the element-wise reconstruction gives the lowest values. As expected, every third update for element-wise, PBF abs vals, and PBF mixed yields a high reduction because it correlates to the update of the absorption coefficient, which has the highest sensitivity. For the material identification, two marks are shown at each iteration corresponding to the objective function value before and after the acoustical update. The convergence speed is similar as compared to the other optimization schemes and the acoustical update slightly increases the objective function in the first iteration and slightly decreases the objective function in the second and third iteration.

10.1.3 Image Reconstruction in Two Dimensions

The images resulting from the three-dimensional reconstruction are blurry due to the coarse discretization with an average pixel size of $0.2125\,\mathrm{mm}$. A finer discretization cannot be used because of the computational cost. Therefore, a two-dimensional image reconstruction is set up. Due to the lower problem dimensionality, the problem complexity is reduced and higher resolution can be obtained. Reconstruction is carried out on a circular domain with an average pixel size of $0.0625\,\mathrm{mm}$, i.e., pixels 3.4 times smaller than in the three-dimensional reconstruction. The simulation settings are comparable to the three-dimensional setup as explained in Section 10.1.2. The circular acoustical domain is of radius $13.5\,\mathrm{mm}$ with the outer $1\,\mathrm{mm}$ in radial direction representing a PML and ABCs at the outer boundary. The optical domain is circular with radius $12\,\mathrm{mm}$ and consists of 124096 element. The acoustical domain is meshed with 154816 elements of polynomial degree $k = 2$. For time stepping the low-storage Runge–Kutta method of order three with three stages LSRK3(3) is used with a time step size of $0.001\,\mu\mathrm{s}$. Parameter reconstruction is carried out analogously to the three-dimensional reconstructions and Figure 10.8 presents the resulting images. Figure 10.9 displays the convergence of the objective function. In contrast to the reconstructions on the three-dimensional geometry, the images pro-

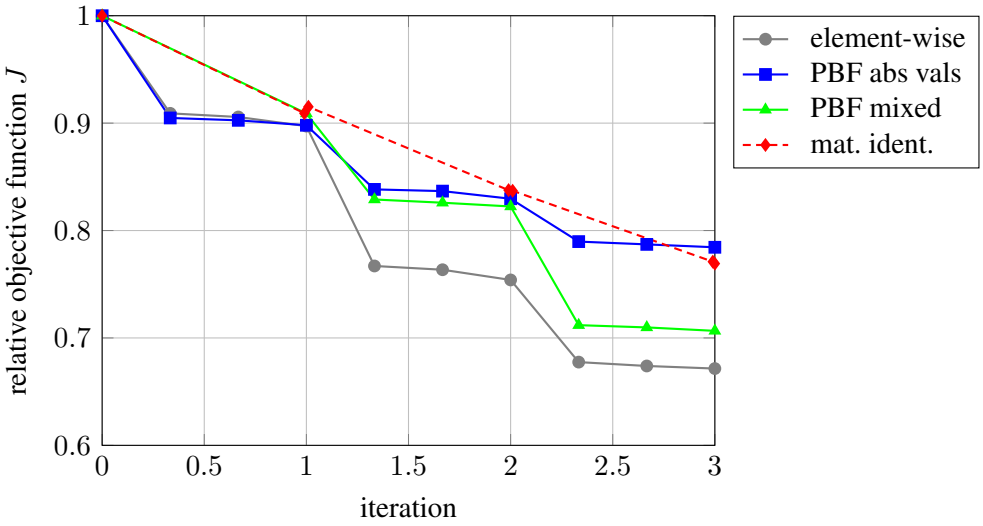


Figure 10.7: Convergence behavior in terms of the relative objective function value during image reconstruction on three-dimensional domain with element-wise parameter discretization, PBF abs vals, PBF mixed, and material identification.

vide a better resolution and more detail. The element-wise parameter discretization highlights several anatomical features as in the image obtained by model-based inversion. The acoustical images signify the skull but do not meet the expected material values for bone. The setup with PBF abs vals provides low values for the absorption coefficient and the convergence is slow compared to all other setups, see Figure 10.9. The absorption coefficient image obtained with PBF mixed is very similar to the image with element-wise parameter discretization but with slightly better contrast near the skull. The acoustical images of PBF abs vals and PBF mixed are segmented according to the absorption coefficient features and oscillations are suppressed. The material identification identifies several elements as bone tissue but not the entire skull is identified. Thereby, the absorption coefficient reconstruction provides higher contrast compared to the element-wise and the PBF mixed setup. Compared to the three-dimensional reconstruction, the impact of the acoustical parameter update is higher, which can be seen from the relative objective function value and is due to the fact that the two-dimensional reconstruction identifies bone.

10.1.4 Discussion of the Results

The reconstruction of the mouse brain images based on a three-dimensional computational domain are blurry due to the low resolution. A higher resolution can currently not be obtained because it requires corresponding mesh refinement, which increases computational cost. The evaluation of one forward problem on the given mesh takes approximately $1000\,\mathrm{s}$ on four nodes each with 24 cores of Intel(R) Xeon(R) CPU E5-2680 v3. Three reconstruction iterations with several evaluations of forward and adjoint problem take approximately 16 hours. The reconstruction on a two-dimensional computational domain allows for better resolution with an average pixel size of $0.0625\,\mathrm{mm}$ and approximately 19 hours computational time. The images obtained on the two-dimensional domain are visually better, i.e., provide more contrast and identify sev-

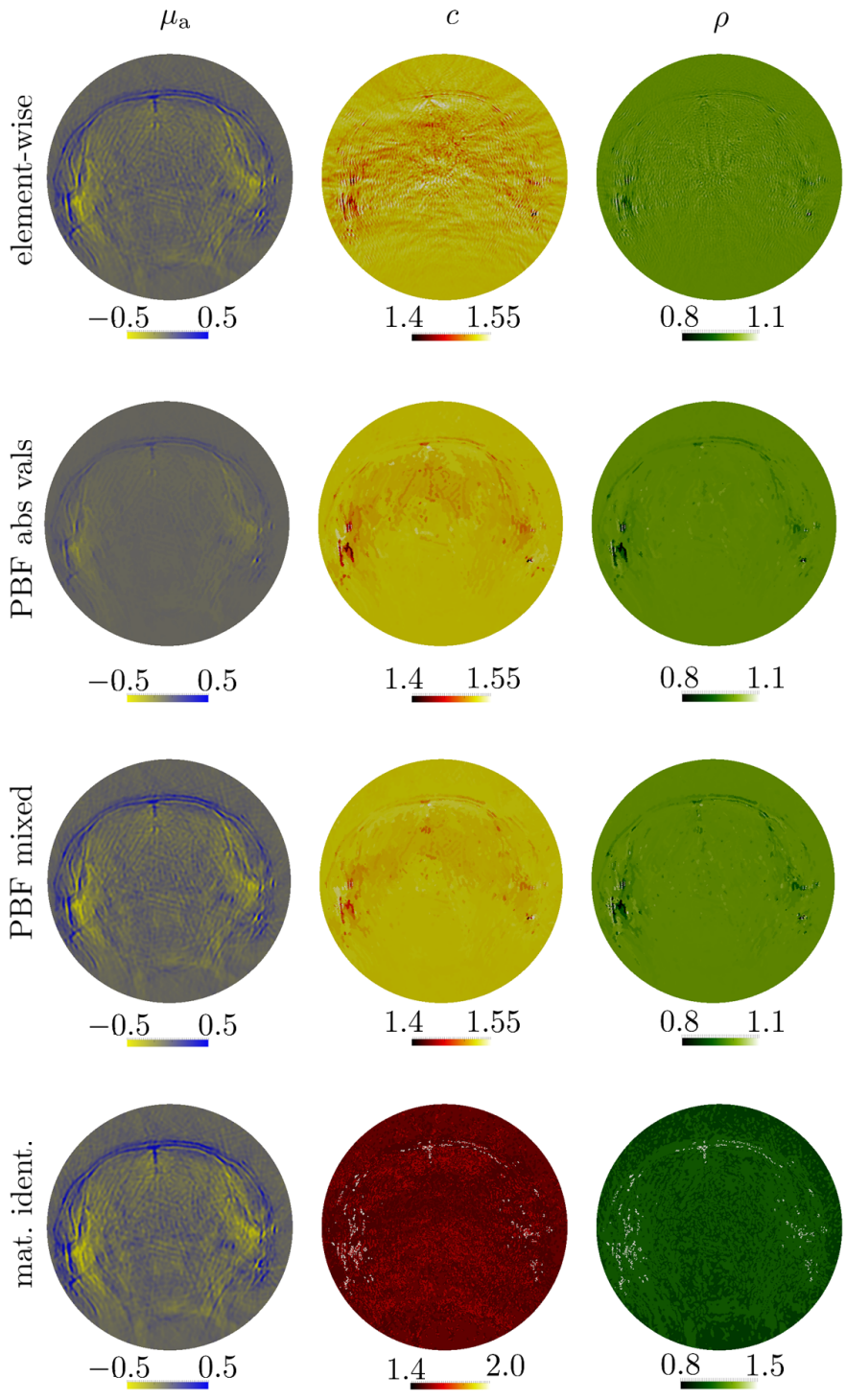


Figure 10.8: Results for image reconstruction on two-dimensional domain with element-wise parameter discretization, PBF abs vals, PBF mixed, and material identification cropped to circle of radius 9 mm. The rows correspond to the four setups as labeled while the columns correspond to the absorption coefficient, the speed of sound, and the mass density (from left to right).

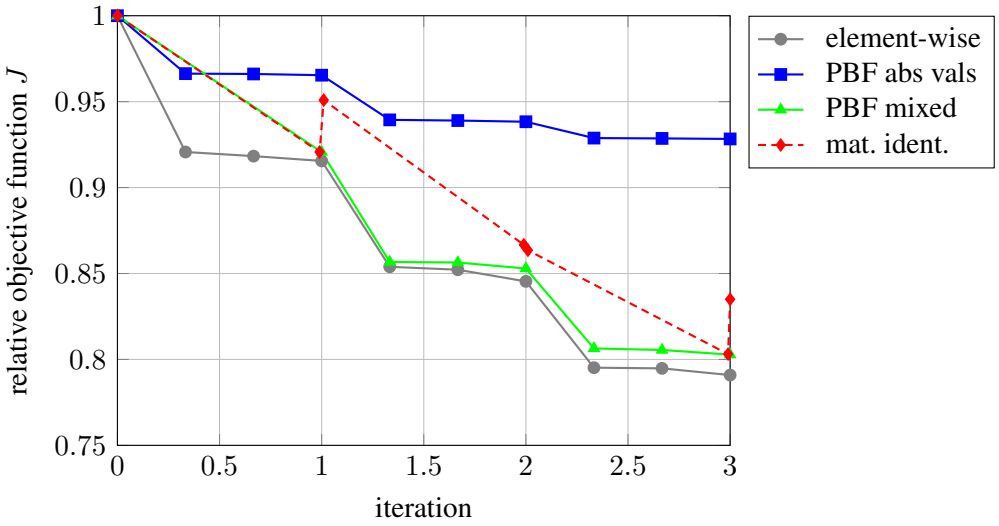


Figure 10.9: Convergence behavior in terms of the relative objective function value during image reconstruction on two-dimensional domain with element-wise parameter discretization, PBF abs vals, PBF mixed, and material identification.

eral anatomical features. If a reconstruction should be carried out in two or three dimensions is questionable and generally depends on the illumination setup. If an object is approximately cylindrical, the material properties vary slowly along the cylinder axis, and the entire mantle is illuminated, pressure waves propagate as cylindrical waves and a two-dimensional consideration is equivalent. If, however, the illumination is only along one line on the cylinder mantle, the pressure waves propagate as spherical waves and a three-dimensional simulation is physically more accurate.

In the absorption coefficient reconstructions, negative values occur, which can be due to noise but also due to modeling errors. A systematic study on negative absorption coefficient values including the effect of permitting negative values should be addressed by future work.

To the author's knowledge, this is the first time that optoacoustic images of speed of sound and mass density have been reconstructed in the context of small animal imaging following the same optimization approach and physical model as for the absorption coefficient. The acoustical images are subject to severe ill-conditioning. To gain first insights concerning the sensitivity of the acoustical parameters, a phantom study is carried out in the next section with mainly acoustical contrast and no or only slight variations in the absorption coefficient.

10.2 An Experimental Phantom Study

A phantom of diameter $20\,\mathrm{mm}$ is created. The basis is Agar (Sigma–Aldrich, St. Louis, MO, USA), a gelatin extracted from red algae used as thickening agent. Water is boiled and $1.5\,\mathrm{g}$ Agar per $100\,\mathrm{ml}$ water is added such that a jellylike consistency with acoustical properties similar to those of soft tissue will be obtained after cooling. A drop of ink is added for slight background absorption and 6% Intralipid (Sigma–Aldrich, St. Louis, MO, USA) is added for scattering.

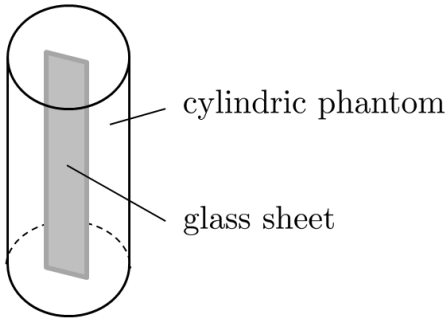


Figure 10.10: Schematic view of the phantom.

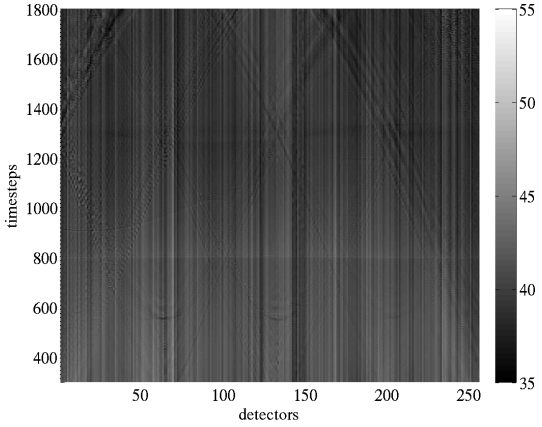


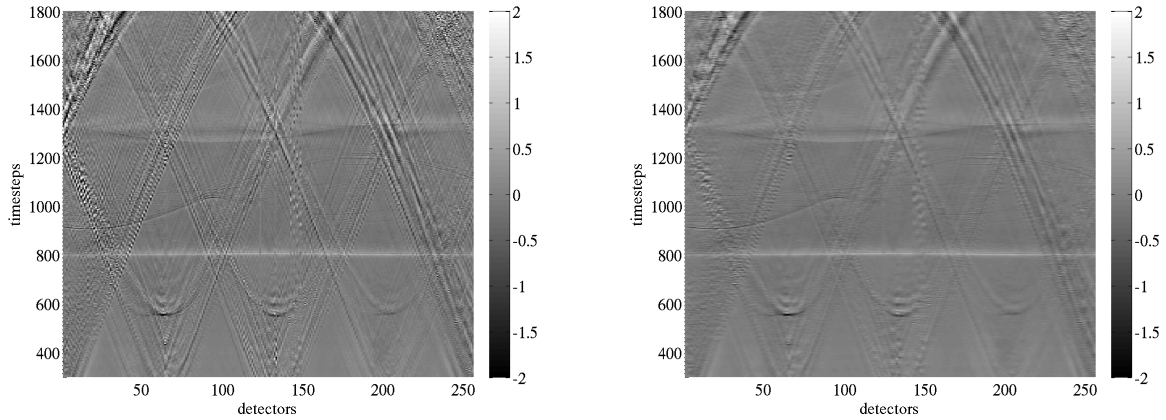
Figure 10.11: Raw data. The gray scale signifies the absolute value of the measured pressure at the detectors.

Before cooling, a thin glass sheet is inserted into the mixture to obtain a phantom as displayed in Figure 10.10. Measurement data is generated as explained in the beginning of this chapter.

To give a first impression on the character of the measurement signals, Figure 10.11 shows the raw data. Pressure curves over time are obtained for 256 detectors and 2030 time steps. Apparently, each detector is subject to a different offset causing the impression of vertical lines. Diagonal lines can be seen, which are related to signals caused close to the detectors, e.g. small dirt particles in the coupling medium. The two horizontal lines correspond to the front and back side of the phantom as seen from the detector. Between the two horizontal lines, wavy lines can be seen which are caused by an absorption mismatch and reflections at the glass sheet.

A Butterworth highpass filter[1] is applied along the time coordinate to eliminate the offset. The resulting data is shown in Figure 10.12(a). In a next step, a Butterworth low pass filter[2] is applied along the detector coordinate to diminish the effect of the dirt particles. The result is shown in Figure 10.12(b). The application of the two filters makes the characteristics of the signal related to the actual phantom and its glass inclusion slightly more visible. The following reconstructions rely on the signals to which both filters were applied.

[1] MATLAB function `butter` with arguments 1, 0.01, 'high'
[2] MATLAB function `butter` with arguments 1, 0.2, 'low'



(a) Application of a Butterworth bandpass filter along the time coordinate

(b) Application of a Butterworth lowpass filter along the detector coordinate

Figure 10.12: Filtered measurement data.

10.2.1 Reduction Simulation

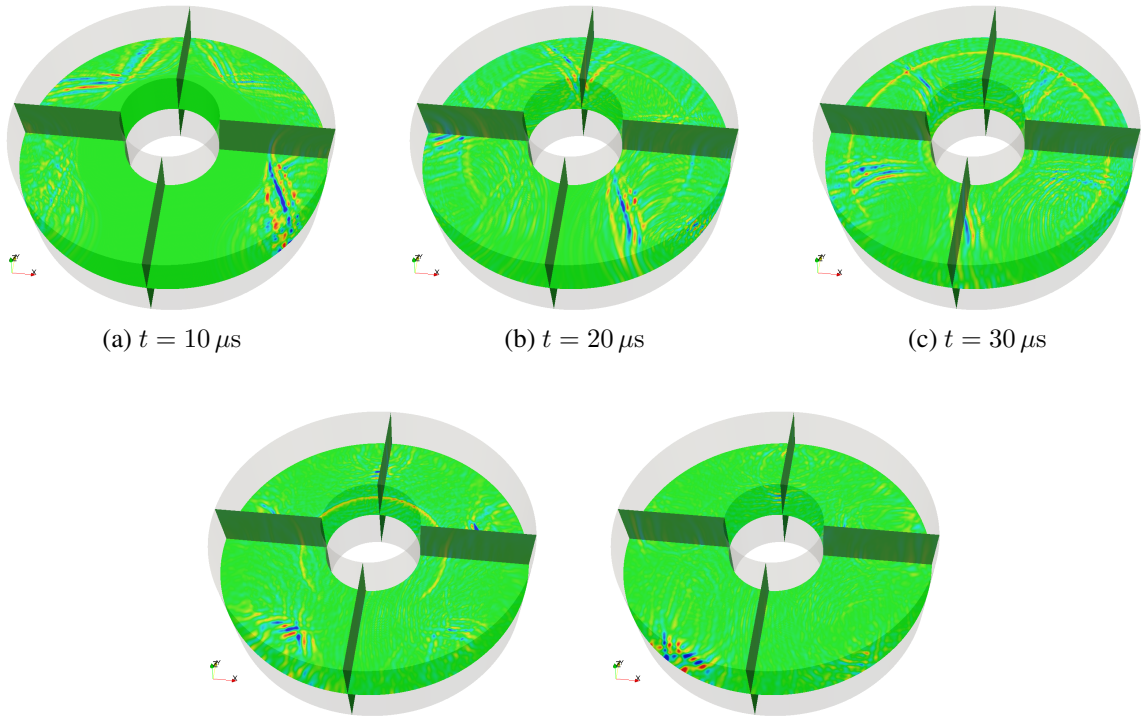
The reduction simulation for the phantom is carried out as for the mouse experiment with the same discretization and parameters as in Section 10.1.1. Figure 10.13 displays pressure snapshots at various time points to give a visual impression of the reduction simulation. The characteristic features of the monitored values are found: the curves appearing sinusoidal in Figure 10.12 are star shaped in Figure 10.13. The horizontal lines in Figure 10.12 appear circular in Figure 10.13.

10.2.2 Image Reconstruction

The image reconstruction is run on the same three-dimensional geometry and with the same parameters as for the mouse experiment presented in Section 10.1.2. Three reconstructions are run, namely the standard setup, i.e., element-wise parameter discretization without material identification, PBF self, and a reconstruction with material identification. Note that other approaches for PBF are not reasonable in this scenario, since the absorption coefficient does not offer any contrast. Figure 10.14 shows the resulting images. Apparently, the three methods fail to reconstruct the glass insertion. For the absorption coefficient, contrast appears at the phantom boundary. The acoustical parameters, show fluctuations at the phantom boundary and in the center. A slight indication of the glass inclusion can be seen in the absorption coefficient reconstruction of the element-wise discretization. The final relative objective function values after three optimization iterations are 74%, 86%, and 82%, for the element-wise discretization, PBF self, and the material identification, respectively.

10.2.3 A Representative Numerical Phantom

In order to put the results of Section 10.2.2 into perspective, the same study is carried out but with measurement data obtained from a simulation that represents the setup. Thereby, the influ-



(a) $t = 10\,\mu s$ (b) $t = 20\,\mu s$ (c) $t = 30\,\mu s$

(d) $t = 40\,\mu s$ (e) $t = 50\,\mu s$

Figure 10.13: Snapshots of the pressure at various time points during the reduction simulation.

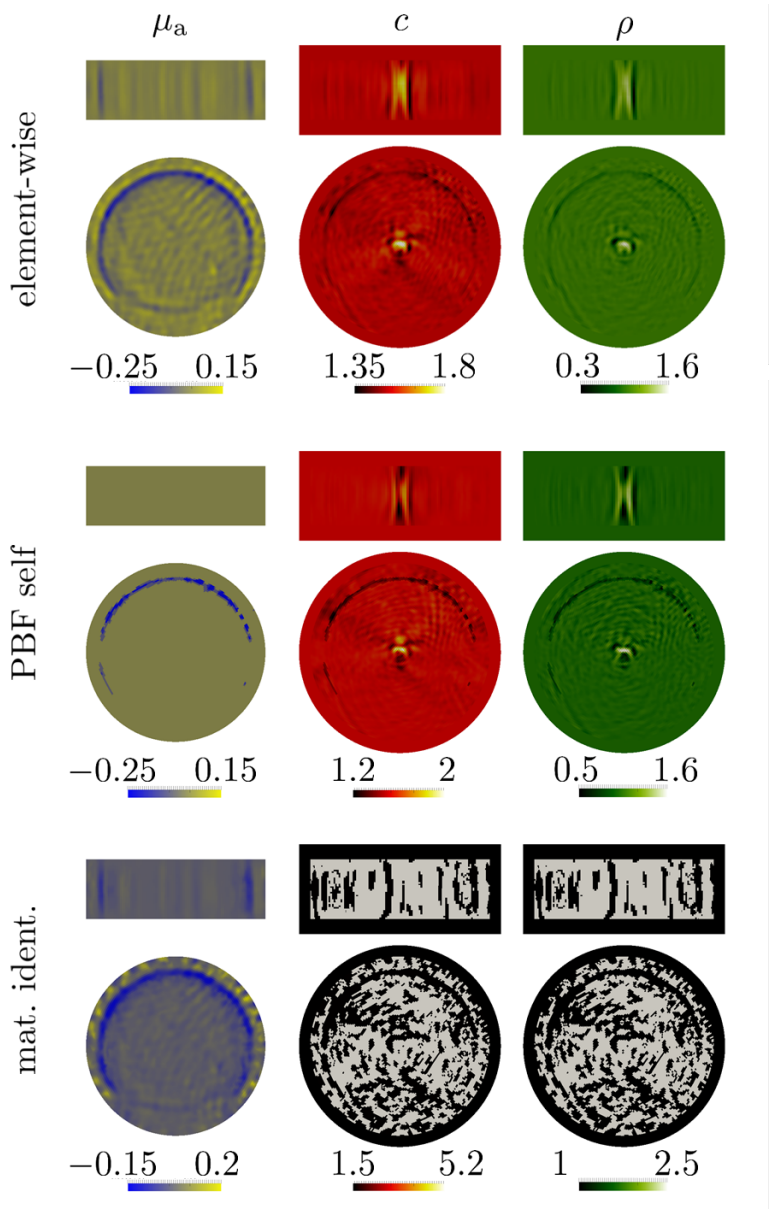


Figure 10.14: Results for image reconstruction of the phantom with element-wise parameter discretization, PBF self, and material identification displayed on a plane with normal vector in x_3 direction and origin $(0, 0, 0)$ and a plane with normal vector in x_2 direction and origin $(0, 0, 0)$. The rows correspond to the three setups as labeled while the columns correspond to the absorption coefficient, the speed of sound, and the mass density (from left to right).

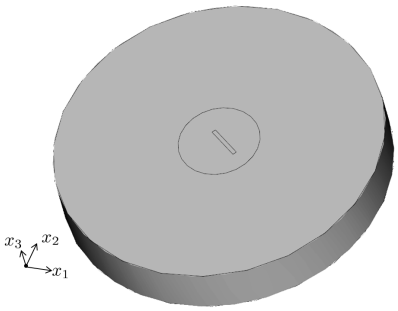
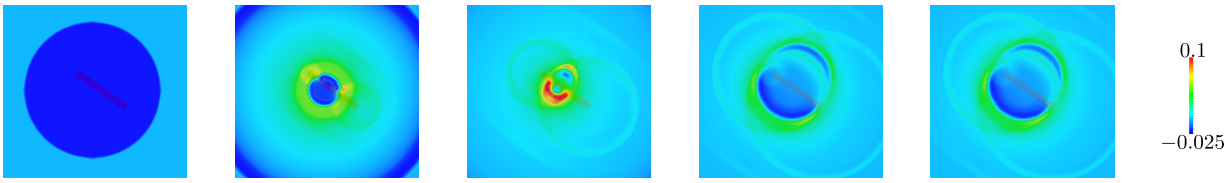


Figure 10.15: Computational domain for numerical experiment.



(a) $t = 0$ (b) $t = 5$ (c) $t = 7.5$ (d) $t = 10$ (e) $t = 10$ (f)

Figure 10.16: Pressure snapshots of the forward solve of the representative numerical phantom with a red shadow signifying the location of the glass inclusion. Panel (f) shows the legend which is the same for plots (a)–(e).

ence of noise and modeling error are investigated. A computational domain is created with the glass inclusion shaped and oriented as in the real experiment as displayed in Figure 10.15. The glass inclusion is of size $8.97\,\mathrm{mm} \times 1\,\mathrm{mm}$, rotated about $-35°$ around the x_3 axis and its center is located at $(x_1, x_2, x_3) = (1.3235\,\mathrm{mm}, 0\,\mathrm{mm}, 0\,\mathrm{mm})$. Material values of $c = 5.2\,{}^{\mathrm{mm}}/_{\mu\mathrm{s}}$ and $\rho = 2.5\,{}^{\mathrm{mg}}/_{\mathrm{mm}^3}$ are assigned to the inclusion, thereby representing glass. Absorbing boundary conditions are applied to all outer acoustic boundaries. On the top and bottom of the cylinder, $1\,\mathrm{mm}$ is assigned to be PML. The pressure is monitored at the mantle of the cylinder between $x_3 \in [-4\,\mathrm{mm}, 4\,\mathrm{mm}]$. The domain is meshed with 67496 optical elements and 1196400 quadratic acoustic elements. A forward solve is run with $\Delta t = 0.0025\,\mu\mathrm{s}$ and the LSRK3(3) time integrator. All unspecified quantities and boundary conditions are as in the preceding simulations. Figure 10.16 shows pressure snapshots of the forward solve with a red shadow signifying the position of the glass.

Figure 10.17 shows the measurement data in the same format like the Figures 10.11 and 10.12 revealing distinct differences. A reduction simulation with the measurement data is run just like for the real measurement data as described in Section 10.2.1 to obtain a data set for reconstructions. Reconstruction is also carried out as for the real phantom experiment and the resulting images are shown in Figure 10.18. The absorption coefficient shows slight fluctuations, especially at the boundary of the phantom. The acoustic coefficients show deviations at the location of the inclusion but a quantitative reconstruction of the acoustical parameters is not obtained. Also, deflections at the center of the phantom occur. In the element-wise reconstruction of the absorption coefficient, errors are seen that are related to the acoustical heterogeneity, i.e., the two smaller circles and the doubling of the contour in the bottom left part.

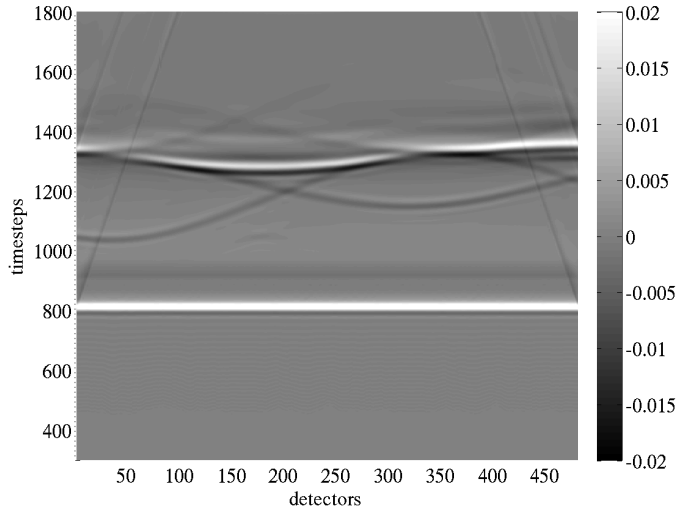


Figure 10.17: Measurement data obtained from forward simulation of the representative phantom.

10.2.4 Conclusion

The phantom is a very challenging setup for optoacoustic imaging because the glass inclusion does not introduce an optical contrast. The glass introduces an acoustical heterogeneity that causes reflections. In a standard model-based inversion, these reflections are seen as a well-known artifact: the boundary of the phantom seems duplicated as shown in Figure 10.19 at the bottom left. The reconstructions on the experimentally obtained data fail to accurately reconstruct the inclusion even though they signify that heterogeneities are present. The study based on the numerical phantom reveals that even with the perfect artificial data, a quantitatively correct image reconstruction is challenging. The inclusion is indicated in the reconstruction but not perfectly localized. Also, the material identification does not improve the image, which is mainly due to the fact that the absorption coefficient of the inclusion does not introduce a contrast and the material identification relies solely on the acoustical gradient. All in all, the sensitivity for the reconstruction of heterogeneities that are purely acoustical is very low. Additional to that, the noise in the measurement data is comparably high, which can be seen by comparing Figures 10.12 and 10.17. The measurement signals only indicate the features, which appear characteristic in the numerical signals. The temporal width of the first arriving pressure signal, i.e., the white horizontal lines in Figures 10.12 and 10.17, is different. This can be caused either by the discretization error concerning the pressure discontinuity or by the light modeling with the diffusion approximation. The diffusion approximation is known to have reduced accuracy near boundaries and sources. Last, a mismatch of the diffusion coefficient yields a different light distribution and hence a different shape of the initial pressure field. The combination of modeling errors, low sensitivity, and high noise levels prevent quantitative image reconstruction for this challenging setup of an inherently ill-conditioned inverse problem.

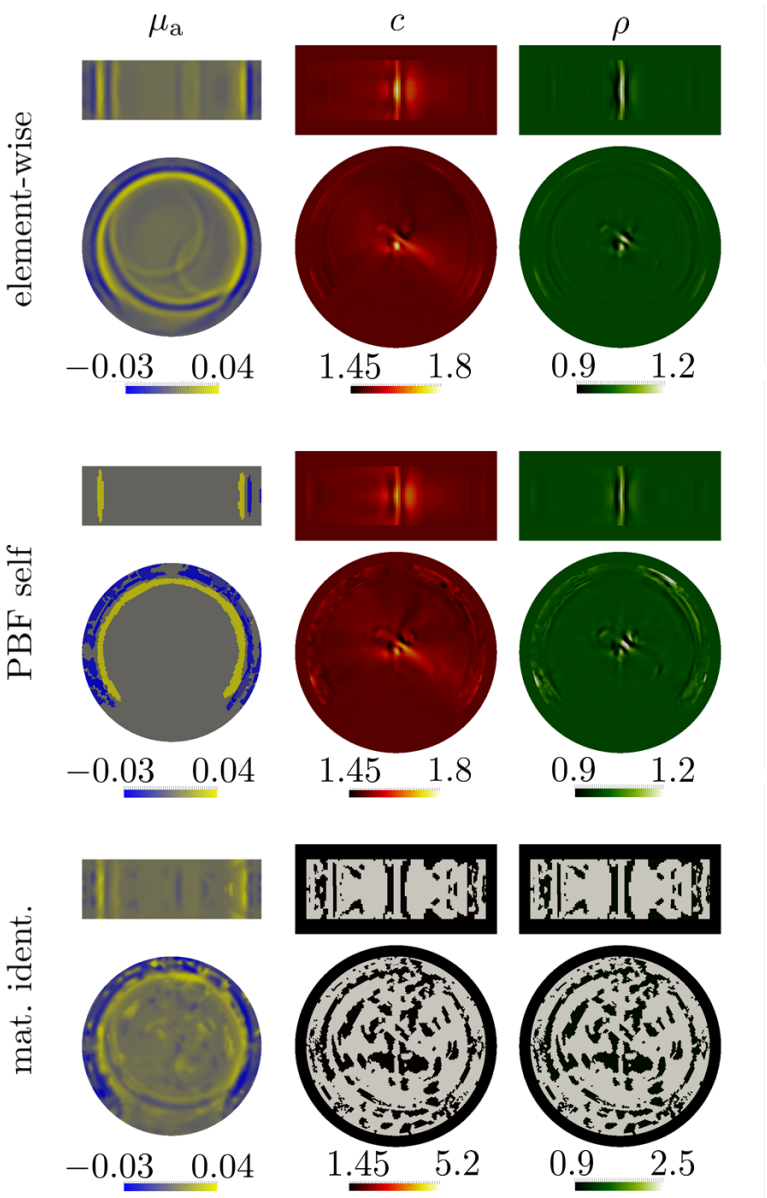


Figure 10.18: Results for image reconstruction on the representative numerical phantom with element-wise parameter discretization, PBF self, and material identification displayed on a plane with normal vector in x_3 direction and origin $(0, 0, 0)$ and a plane with normal vector in x_2 direction and origin $(0, 0, 0)$. The rows correspond to the three setups as labeled while the columns correspond to the absorption coefficient, the speed of sound, and the mass density (from left to right).

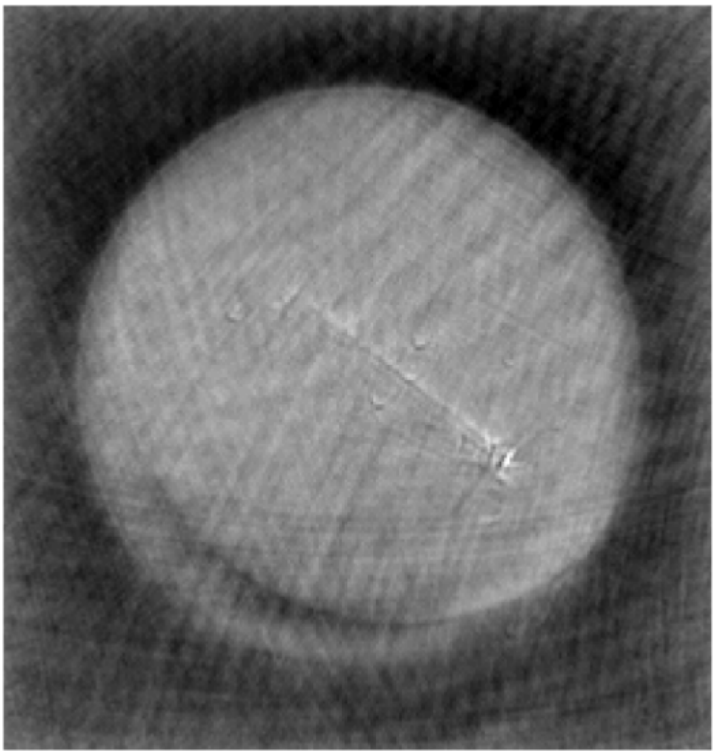


Figure 10.19: Image of the phantom with glass inclusion obtained with a standard model based reconstruction algorithm.

11 Conclusions and Outlook

In this thesis, two topics, namely the development of a high performance acoustical solver and the development of an optoacoustic image reconstruction algorithm, have been addressed. Both topics are treated separately in this concluding chapter.

11.1 High Performance Solver for Acoustics

In Chapter 2 of this work, the first building block for the high performance acoustical solver has been set: the numerical methods. HDG was introduced for the acoustic wave equation in [120] but in combination with implicit Runge–Kutta methods. In this work and in [97], a reformulation of the semi-discrete problem has been presented that allows for using explicit Runge–Kutta schemes. In contrast to HDG with implicit Runge–Kutta schemes, the interior degrees of freedom are kept while the trace variable is replaced in terms of the other variables. The update procedure does not require the solve of a global linear system but only element-local and face-local applications of inverse mass matrices, which reduces the computational cost significantly as demonstrated by performance models and relevant experiments. A second achievement is the combination of HDG with explicit ADER time stepping [145]. The method is of arbitrary high order in space and time without restrictions such as the Butcher barrier for Runge–Kutta methods. LTS with ADER time integration in combination with DG for the elastic wave equation as in [49] has been carried over to HDG for the acoustic wave equation. Thereby, every element of the triangulation of the computational domain advances with its optimal time step avoiding computational overhead due to too small time steps and keeping dispersion and dissipation errors to a minimum. HDG typically offers the option for superconvergent results if the time integration scheme is sufficiently accurate. For ADER HDG, temporal and spatial discretization are strongly interlinked and the straightforward combination of ADER and HDG does not offer superconvergent results. Therefore, a reconstruction step has been developed such that the superconvergence property is preserved. The reconstruction step is theoretically supported by an adjoint consistency analysis. With several numerical examples, it has been demonstrated that optimal convergence of order $k+1$ and superconvergence of order $k+2$ is obtained for polynomial degrees $k = 1, ..., 12$. In additional numerical examples, it has been shown that the temporal stability limit for ADER in combination with HDG is by a factor of ≈ 2 to ≈ 3 stricter compared to a low-storage Runge–Kutta scheme of order four with five stages while dispersion and dissipation behavior with the same time step size are favorable for ADER.

In Chapter 3, the second building block for the high performance acoustical solver has been presented: the implementation aspects. The implementation of the explicit Runge–Kutta schemes as well as of ADER is based on matrix-free operator evaluations with fast quadrature and sum factorization kernels that combine optimal-complexity mathematical algorithms utilizing the tensor product structure of the shape functions with a highly competitive implementation that

vectorizes over several elements and faces [95, 144]. For ADER, two additional optimizations have been proposed. An on-the-fly change between two bases has been developed to evaluate operators including face integrals with nodal basis functions with nodes on the element boundaries and to evaluate cells in the Taylor–Cauchy–Kowalevski procedure of ADER with Lagrange polynomials with nodes in the quadrature points. The second optimization concerns the efficient evaluation of high derivatives in the Taylor–Cauchy–Kowalevski procedure: the degree of the polynomial shape functions representing the higher order spatial derivatives is reduced by projection on the lower dimensional approximation space. Performance analyses have been carried out for several quantities of interest, i.e., timings, throughput, and scalability. While the theoretically derived operation counts already signify a slight benefit for ADER compared to Runge–Kutta, the actual timings show a distinct benefit, reducing the time to perform one time step by approximately a factor of four because ADER suits modern hardware architecture better: while Runge–Kutta schemes are mostly limited by the memory bandwidth, ADER performs more operations on the data that is loaded from main memory and thus reaches a higher arithmetic intensity. A detailed analysis of Runge–Kutta versus ADER integration at the CFL stability limit has shown comparable performance where the Runge–Kutta time discretization order matches the spatial discretization order. For approximations with a high order of accuracy where the Butcher barriers set in, ADER exceeds the abilities of Runge–Kutta because its computational cost does not grow overproportionally. While the findings for ADER are limited to linear hyperbolic PDEs, the optimizations regarding the basis functions and reduced vector access for the Runge–Kutta time integrators are also directly applicable to general nonlinear systems of hyperbolic PDEs.

In Chapter 4, a novel formulation for PMLs is derived, which is not only especially suitable in the context of explicit HDG but also reduces the number of required auxiliary variables. The back-transformation to time domain is based on a spectral decomposition of the gradient of the damping function, which allows the detection of unnecessary auxiliary variables. The formulation is general compared to formulations in the literature. The derived method allows to surround general prismatic bodies and even to combine straight lined with spherical or cylindrical boundaries, which has previously not been possible to the best of the author's knowledge.

Chapter 5 incorporates the methodologies and the implementation derived in the preceding chapters to solve real world problems. The solver is applied to one representative of urban acoustics, namely a training village, which has been extensively studied in the literature. Three discretization setups are chosen to recreate characteristics from [115]. For a similar number of grid points and the same number of time steps, the computational time required by the proposed acoustical solver is competitive to an adaptive rectangular decomposition method considering that adaptive rectangular decomposition relies on a semi-analytic approach using the discrete cosine transform and is formally of lower order. A cathedral like geometry is studied to demonstrate the applicability of the solver to room acoustics. Complex reflection patterns are predicted over a wide frequency range with only ≈ 20 CPU seconds per time step for $2.6 \cdot 10^8$ degrees of freedom.

Possible future research concerns the theoretical and numerical investigation of the time step limit of the fully adjoint consistent ADER scheme and the trade-off with its disadvantageous computational properties. Also, a theoretical and numerical comparison of ADER LTS with other LTS approaches or so called implicit-explicit (IMEX) methods should be carried out [86]. To enhance the applicability of the acoustical solver to a broader range of acoustical problems,

special boundary conditions are required representing e.g. walls with specific reflection properties due to windows and window sills. Especially for urban acoustics, the introduction of a convective term to represent air movement due to wind would improve the prediction accuracy in realistic scenarios. This, however, complicates the PML formulation. Considering the implementation aspects, future work should address GPUs for their higher memory bandwidth but also concepts like wavefront blocking as already used in the context of finite differences [168].

11.2 Optoacoustic Image Reconstruction Method

In Chapter 6, an introduction to optoacoustic imaging is given where the functional principal is explained and several common and less common solution approaches from the literature are reviewed.

Chapter 7 details all steps of the derivation of the image reconstruction method. First, the physical model has been developed. The light propagation is described by the diffusion approximation because biological tissue is generally strongly scattering and one can therefore assume that light loses its directionality after entering the medium. In general tomographic setups, the laser illumination only lasts for a few nanoseconds, which is a too short time period for heat or stresses to propagate. The energy conversion is hence in thermal and stress confinement and the photoacoustic effect is described by an assignment. For the description of the sound propagation, the acoustic wave equation has been chosen neglecting effects due to damping or shear waves. The physical model has been transferred to a numerical model of the optoacoustic imaging procedure by discretization with state-of-the-art methods. For the optical problem, standard continuous finite elements are used, the photoacoustic effect is discretized by a mapping between potentially non-conforming meshes and the acoustic wave equation is discretized with HDG for the spatial coordinate and Runge–Kutta or ADER for the temporal coordinate. The objective function sums differences between measured and simulated pressure time curves at the detectors. The gradients of the objective function with respect to the sought parameters, i.e., absorption coefficient, diffusion coefficient, speed of sound, and mass density have been derived using the adjoint approach. The algorithmic framework is based on a gradient-based optimization (either low-storage BFGS or steepest decent) with a line search fulfilling the strong Wolfe conditions. The derived reconstruction method is very general in terms of the physical description and the supported tomographic setups. To the best of the author's knowledge, this is the first algorithm allowing for the reconstruction of the four fields of absorption coefficient, diffusion coefficient, speed of sound, and mass density within one integral setup. A proof of concept and several numerical examples demonstrate the correctness and the properties of the derived method.

A typical optoacoustic tomograph has fixed detectors and overcomes the distance to the object by a coupling medium, which is spatially homogeneous. In Chapter 8, an approach based on the ideas of [8] has been presented to crop the computational domain such that computational overhead due to the repeated simulation of wave propagation in a homogeneous medium is avoided.

In Chapter 9, two novel methods to oppose ill-conditioning in inverse problems have been developed. The first method is based on the idea that material properties are not randomly distributed but are made up of distinct clustered constituents, e.g. the different organs in a body.

Commonly, image reconstruction determines parameter values for each pixel of the image separately or analogously, for each finite element of a mesh separately. In other words, the basis functions for the parameter fields take on the value ‘1’ in one element and ‘0’ in all others. Here, elements are clustered into patches to represent constituents of the object to be reconstructed. The patches are generated automatically from the parameter gradients or the parameter distributions. Thereby, the number of degrees of freedom in the inverse problem is significantly reduced without compromising the flexibility. The solution and the patches can evolve according to the characteristics of the object. Another gain of this concept is that information can be transported from the most sensitive parameter to the less sensitive parameters, e.g. by reusing the patched basis functions of the most sensitive parameter. The second method to oppose ill-conditioning is a material identification. Generally, an experienced user expects material parameters representing specific tissue types. This knowledge is input to the algorithm by supplying a material catalog specifying expected tissue types with representative ranges for the material properties. The algorithm only reconstructs the absorption coefficient and updates the acoustical properties by choosing best matches from the material catalog. Best matches are determined with the absorption coefficient and the gradients of the acoustical parameters. Both approaches have been validated and studied using numerical examples. In several scenarios, they speed up the convergence of the parameter errors and improve the image quality significantly. Both methods are straight forwardly carried over to other inverse problems.

In Chapter 10, all developed methods have been assembled and applied to experimentally obtained data. The presented reconstruction algorithm has successfully reconstructed the optical absorption coefficient from in-vivo mouse brain measurements and several anatomical features were clearly identified. Optoacoustic images of the speed of sound and mass density distribution have been reconstructed following the same optimization approach and physical model as for the absorption coefficient, which is (to the author’s knowledge) reported for the first time. The images suffer from low sensitivity. Therefore, a phantom study has been carried out highlighting the ill-conditioning of acoustical parameter reconstruction in optoacoustic imaging.

One very important aspect to be considered in future work is the modeling error. It is especially important to keep the modeling error low because the reconstruction procedure cannot distinguish if the deviation of measurement and simulation data is due to falsely set material parameters or due to modeling errors. Images can therefore show characteristic features that are associated to balancing or counteracting modeling errors. The first point to be addressed should be the description of light propagation. The diffusion approximation is known to yield poor results near sources and boundaries. The radiative transfer equation, from which the diffusion approximation is derived, could reduce the modeling error in the description of light propagation. The acoustical part could be extended by consideration of shear waves and damping. Extensive studies on experimentally obtained measurement data should be carried out in order to quantify modeling error and sensitivity. For the approaches opposing ill-conditioning, quantitative studies on benchmarks of general inverse problems would be interesting, especially in comparison with standard methods to improve conditioning like Tikhonov or total variation regularization. Applications should be extended concerning the tomographic setup, e.g., handheld systems.

Bibliography

[1] M. Ainsworth, Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods, *Journal of Computational Physics* **198**, 106–130, 2004.

[2] D. G. Albert and L. Liu, The effect of buildings on acoustic pulse propagation in an urban environment, *The Journal of the Acoustical Society of America* **127**, 1335–1346, 2010.

[3] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s, *SIAM Journal on Numerical Analysis* **14**, 1006–1021, 1977.

[4] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, The `deal.II` library, version 9.0, *Journal of Numerical Mathematics* **26**, 173–184, 2018.

[5] H. Ammari, E. Bossy, V. Jugnon, and H. Kang, Reconstruction of the optical absorption coefficient of a small absorber from the absorbed energy density, *SIAM Journal on Applied Mathematics* **71**, 676–693, 2011.

[6] D. Appelo, T. Hagstrom, and G. Kreiss, Perfectly matched layers for hyperbolic systems: General formulation, well-posedness, and stability, *SIAM Journal on Applied Mathematics* **67**, 1–23, 2006.

[7] F. Assous, M. Kray, F. Nataf, and E. Turkel, Time-reversed absorbing conditions, *Comptes Rendus Mathematique* **348**, 1063–1067, 2010.

[8] F. Assous, M. Kray, F. Nataf, and E. Turkel, Time-reversed absorbing condition: application to inverse problems, *Inverse Problems* **27**, 065003, 2011.

[9] N. Bähr. Verwendung von time reversed absorbing conditions zur effizienten Berechnung des photoakustischen inversen Problems. Bachelor's thesis, Technichal University of Munich, 2015.

[10] G. Bal and K. Ren, On multi-source quantitative photoacoustic tomography in a diffusive regime, *Inverse Problems* **27**, 075003, 2011.

[11] G. Bal and K. Ren, On multi-spectral quantitative photoacoustic tomography in diffusive regime, *Inverse Problems* **28**, 025010, 2012.

[12] G. Bal and G. Uhlmann, Inverse diffusion theory of photoacoustics, *Inverse Problems* **26**, 085010, 2010.

[13] A. Bell, On the production and reproduction of sound by light, *American Journal of Science* **5**, 305–324, 1880.

[14] J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *Journal of Computational Physics* **114**, 185–200, 1994.

[15] A. Bermudez, L. Hervella-Nieto, A. Prieto, and R. Rodriguez, An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems, *Journal of Computational Physics* **223**, 469–488, 2007.

[16] S. Bilbao, Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation, *IEEE Transactions on Audio, Speech, and Language Processing* **21**, 1524–1533, 2013.

[17] A. Billon, V. Valeau, A. Sakout, and J. Picaut, On the use of a diffusion model for acoustically coupled rooms, *The Journal of the Acoustical Society of America* **120**, 2043–2054, 2006.

[18] D. Botteldooren, Acoustical finite-difference time-domain simulation in a quasi Cartesian grid, *The Journal of the Acoustical Society of America* **95**, 2313–2319, 1994.

[19] D. Botteldooren, Finite-difference time-domain simulation of low-frequency room acoustic problems, *The Journal of the Acoustical Society of America* **98**, 3302–3308, 1995.

[20] S. C. Brenner and L. R. Scott, *The mathematical theory of finite element methods*, Springer-Verlag, New York, 3rd Edition, 2008.

[21] A. Breuer, A. Heinecke, S. Rettenberger, M. Bader, A.-A. Gabriel, and C. Pelties, Sustained petascale performance of seismic simulations with SeisSol on SuperMUC, In J. M. Kunkel, T. Ludwig, and H. W. Meuer (eds.), *Supercomputing*, Volume 8488 of *Lecture Notes in Computer Science*, pages 1–18, Springer, 2014.

[22] J. Brown, Efficient nonlinear solvers for nodal high-order finite elements in 3D, *Journal of Scientific Computing* **45**, 48–63, 2010.

[23] A. Buehler, M. Kacprowicz, A. Taruttis, and V. Ntziachristos, Real-time handheld multispectral optoacoustic imaging, *Optics Letters* **38**, 1404–1406, 2013.

[24] J. Bushberg, A. Seibert, E. Leidholdt, and J. Boone, *The essential physics of medical imaging*, Lippincott Williams and Wilkins, Philadelphia, PA, 3 Edition, 2012.

[25] J. C. Butcher, On the attainable order of Runge–Kutta methods, *Mathematics of Computation* **19**, 408–417, 1965.

[26] J. Cash and C. Liem, On the design of a variable order, variable step diagonally implicit Runge–Kutta algorithm, *IMA Journal of Applied Mathematics* **26**, 87–91, 1980.

[27] A. D. Cezaro, A. Leitao, and X.-C. Tai, On piecewise constant level-set (PCLS) methods for the identification of discontinuous parameters in ill-posed problems, *Inverse Problems* **29**, 015003, 2013.

[28] J. Chen and Y. Yang, Quantitative photo-acoustic tomography with partial data, *Inverse Problems* **28**, 115014, 2012.

[29] W.-F. Cheong, S. Prahl, and A. Welch, A review of the optical properties of biological tissues, *IEEE Journal of Quantum Electronics* **26**, 2166–2185, 1990.

[30] W. C. Chew and W. H. Weedon, A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates, *Microwave and Optical Technology Letters* **7**, 599–604, 1994.

[31] B. Cockburn and V. Quenneville-Belair, Uniform-in-time superconvergence of the HDG methods for the acoustic wave equation, *Mathematics of Computation* **83**, 65–85, 2013.

[32] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM Journal on Numerical Analysis* **35**, 2440–2463, 1998.

[33] B. Cockburn, J. Gopalakrishnan, and F.-J. Sayas, A projection-based error analsys of HDG methods, *Mathematics of Computation* **79**, 1351–1367, 2010.

[34] B. Cockburn, W. Qiu, and K. Shi, Conditions for superconvergence of HDG methods for second-order elliptic problems, *Mathematics of Computation* **81**, 1327–1353, 2011.

[35] B. Cockburn and C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, *Journal of Computational Physics* **141**, 199–224, 1998.

[36] B. Cockburn, Z. Fu, A. Hungria, L. Ji, M. A. Sánchez, and F.-J. Sayas, Stormer-Numerov HDG methods for acoustic waves, *Journal of Scientific Computing* **75**, 597–624, 2018.

[37] G. Cohen, P. Joly, and N. Tordjman, Higher-order finite elements with mass-lumping for the 1D wave equation, *Finite Elements in Analysis and Design* **16**, 329 – 336, 1994.

[38] F. Collino and P. Monk, The perfectly matched layer in curvilinear coordinates, *SIAM Journal on Scientific Computing* **19**, 2061–2090, 1998.

[39] M. Costabel, Principles of boundary element methods, *Computer Physics Reports* **6**, 243–274, 1987.

[40] R. Courant, K. Friedrichs, and H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Mathematische Annalen* **100**, 32–74, 1928.

[41] B. Cox, J. Laufer, S. Arridge, and P. Beard, Quantitative spectroscopic photoacoustic imaging: A review, *Journal of Biomedical Optics* **17**, 061202, 2012.

[42] J. D'Alembert, 10, *Recherches sur la courbe que forme une corde tendue mise en vibration*, A. Haude (ed.), Histoire de l'Academie royale des sciences et des belles lettres, Deutsche Akademie der Wissenschaften zu Berlin, 1747.

[43] X. Dean-Ben, M. Rui, D. Razansky, and V. Ntziachristos, Statistical approach for optoacoustic image reconstruction in the presence of strong acoustic heterogeneities, *IEEE Transactions on Medical Imaging* **30**, 401–408, 2011.

[44] M. O. Deville, P. F. Fischer, and E. H. Mund, *High-order methods for incompressible fluid flow*, Cambridge University Press, 2002.

[45] J. Diaz, *Modelling and advanced simulation of wave propagation phenomena in 3D geophysical media.*, Habilitation à diriger des recherches, Université de Pau et des Pays de l'Adour, 2016. URL `https://tel.archives-ouvertes.fr/tel-01304349`.

[46] A. Dima, N. Burton, and V. Ntziachristos, Multispectral optoacoustic tomography at 64, 128, and 256 channels, *Journal of Biomedical Optics* **19**, 36021, 2014.

[47] M. Dumbser and M. Käser, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – II. The three-dimensional isotropic case, *Geophysical Journal International* **167**, 319–336, 2006.

[48] M. Dumbser, T. Schwartzkopff, and C.-D. Munz, Arbitrary high order finite volume schemes for linear wave propagation, In E. Krause, Y. Shokin, M. Resch, and N. Shokina (eds.), *Computational Science and High Performance Computing II*, pages 129–144, Springer Berlin Heidelberg, 2006.

[49] 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, *Geophysical Journal International* **171**, 695–717, 2007.

[50] M. Dumbser, I. Peshkov, E. Romenski, and O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat-conducting fluids and elastic solids, *Journal of Computational Physics* **314**, 824–862, 2016.

[51] M. Durufle, P. Grob, and P. Joly, Influence of Gauss and Gauss–Lobatto quadrature rules on the accuracy of a quadrilateral finite element method in the time domain, *Numerical Methods for Partial Differential Equations* **25**, 526–551, 2009.

[52] P. Eller and W. Gropp, Scalable non-blocking preconditioned conjugate gradient methods, In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 18:1–18:12, Piscataway, NJ, USA, 2016, IEEE Press.

[53] B. Engquist and A. Majda, Absorbing boundary conditions for the numerical simulation of waves, *Matematics of Computation* **31**, 629–651, 1977.

[54] B. Engquist and O. Runborg, Computational high frequency wave propagation, *Acta Numerica* **12**, 181266, 2003.

[55] J. Escolano, J. Navarro, and J. Lopez, On the limitation of a diffusion equation model for acoustic predictions of rooms with homogeneous dimensions, *The Journal of the Acoustical Society of America* **128**, 1586–1589, 2010.

[56] N. Fehn, W. A. Wall, and M. Kronbichler, Efficiency of high-performance discontinuous Galerkin spectral element methods for under-resolved turbulent incompressible flows, *International Journal for Numerical Methods in Fluids* **88**, 32–54, 2018.

[57] D. Feng (ed.), *Biomedical Information Technology*, Elsevier Inc., 2008.

[58] R. Feynman, *Lectures in physics*, Basic Books, 2013.

[59] J. Fischer. Photoacoustic imaging: Detection of acoustical heterogeneities using optical material identification. Master's thesis, Technichal University of Munich, 2015.

[60] G. Gassner and D. Kopriva, A comparison of the dispersion and dissipation errors of Gauss and Gauss–Lobatto discontinuous Galerkin spectral element methods, *SIAM Journal on Scientific Computing* **33**, 2560–2579, 2011.

[61] G. Gassner, G. Hindenlang, and C.-D. Munz, A Runge–Kutta based discontinuous Galerkin method with time accurate local time stepping, pages 95–118, *Adaptive High-Order Methods in Computational Fluid Dynamics*, Z. Wang (ed.), 2011.

[62] S. Gedney, Perfectly matched layer absorbing boundary conditions, pages 273–328, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, A. Taflove (ed.), Artech House, 2005.

[63] D. Givoli, High-order local non-reflecting boundary conditions: A review, *Wave Motion* **39**, 319–326, 2004.

[64] A. Griewank, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation, *Optimization Methods and Software* **1**, 35–54, 1992.

[65] M. Grote, M. Mehlin, and T. Mitkova, Runge–Kutta-based explicit local time-stepping methods for wave propagation, *SIAM Journal on Scientific Computing* **37**, A747–A775, 2015.

[66] M. J. Grote, A. Schneebeli, and D. Schötzau, Discontinuous Galerkin finite element method for the wave equation, *SIAM Journal on Numerical Analysis* **44**, 2408–2431, 2006.

[67] H. Grün, C. Hofer, M. Haltmeier, G. Paltauf, and P. Burgholzer, Thermoacoustic imaging using time reversal, In *Proceedings of the International Congress on Ultrasonics*, 2007.

[68] W. Guo, J.-M. Qiu, and J. Qiu, A new Lax–Wendroff discontinuous Galerkin method with superconvergence, *Journal of Scientific Computing* **65**, 299–326, 2015.

[69] J. Hadamard, Sur les problemes aux derivées partielles et leur signification physique, *Princeton University Bulletin* **13**, 49–52, 1902.

[70] T. Hagstrom, Radiation boundary conditions for the numerical simulation of waves, *Acta Numerica* **8**, 47106, 1999.

[71] M. Haltmeier, L. Neumann, and S. Rabanser, Single-stage reconstruction algorithm for quantitative photoacoustic tomography, *Inverse Problems* **31**, 065005, 2015.

[72] R. Hartmann, Adjoint consistency analysis of discontinuous Galerkin discretizations, *SIAM Journal on Numerical Analysis* **45**, 2671–2696, 2007.

[73] H. He, B. Gilbert, P. Ralph, and Y. J. Yong, An adaptive filtered backprojection for photoacoustic image reconstruction, *Medical Physics* **42**, 2169–2178, 2015.

[74] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Application*, Volume 54 of *Texts in Applied Mathematics*, Springer, 2008.

[75] R. Higdon, Absorbing boundary conditions for difference approximations to the multidimensional wave equation, *Mathematics of Computation* **47**, 437–459, 1986.

[76] R. Higdon, Numerical absorbing boundary conditions for the wave equation, *Mathematics of Computation* **49**, 65–90, 1987.

[77] F. Hindenlang, G. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz, Explicit discontinuous Galerkin methods for unsteady problems, *Computers & Fluids* **61**, 86–93, 2012.

[78] M. Hornikx, Ten questions concerning computational urban acoustics, *Building and Environment* **106**, 409–421, 2016.

[79] M. Hornikx, M. Kaltenbacher, and S. Marburg, Framework for linear benchmark problems in computational acoustics, In *Forum Acusticum*, 2014.

[80] M. Hornikx, M. Kaltenbacher, and S. Marburg, A platform for benchmark cases in computational acoustics, *Acta Acoustica united with Acoustics* **101**, 811–820, 2015.

[81] Y. Hristova, P. Kuchment, and L. Nguyen, Reconstruction and time reversal in thermoacoustic tomography in acoustically homogeneous and inhomogeneous media, *Inverse Problems* **24**, 055006, 2008.

[82] C. Huang, K. Wang, L. Nie, L. Wang, and M. Anastasio, Full-wave iterative image reconstruction in photoacoustic tomography with acoustically inhomogeneous media, *IEEE Transactions on Medical Imaging* **32**, 1097–1110, 2013.

[83] A. Huerta, A. Angeloski, X. Roca, and J. Peraire, Efficiency of high-order elements for continuous and discontinuous Galerkin methods, *International Journal for Numerical Methods in Engineering* **96**, 529–560, 2013.

[84] S. Jacques and B. Pogue, Tutorial on diffuse light transport, *Journal of Biomedical Optics* **13**, 041302, 2008.

[85] H. Jiang, Z. Yuan, and X. Gu, Spatially varying optical and acoustic property reconstruction using finite-element-based photoacoustic tomography, *Journal of the Optical Society of America A* **23**, 878–888, 2006.

[86] A. Kanevsky, M. H. Carpenter, D. Gottlieb, and J. S. Hesthaven, Application of implicit-explicit high order Runge–Kutta methods to discontinuous Galerkin schemes, *Journal of Computational Physics* **225**, 1753–1781, 2007.

[87] J. Kang, *Urban Sound Enviroment*, Taylor and Francis, Milton Park, Abingdon, 2007.

[88] G. Karniadakis and S. Sherwin, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press, 2nd Edition, 2005.

[89] M. Käser and M. Dumbser, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – I. The two-dimensional isotropic case with external source terms, *Geophysical Journal International* **166**, 855–877, 2006.

[90] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations, *Applied Numerical Mathematics* **35**, 177–219, 2000.

[91] R. Kirby, S. Sherwin, and B. Cockburn, To CG or to HDG: A comparative study, *Journal of Scientific Computing* **51**, 183–212, 2012.

[92] A. Kirsch and O. Scherzer, Simultaneous reconstructions of absorption density and wave speed with photoacoustic measurements, *SIAM Journal on Applied Mathematics* **72**, 1508–1523, 2012.

[93] D. A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, Springer, 2009.

[94] M. Kronbichler and K. Kormann, A generic interface for parallel finite element operator application, *Computers & Fluids* **63**, 135–147, 2012.

[95] M. Kronbichler and K. Kormann, Fast matrix-free evaluation of discontinuous Galerkin finite element operators, *ACM Transactions on Mathematical Software* **45**, 12:1–12:37, 2019.

[96] M. Kronbichler and W. A. Wall, A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers, *SIAM Journal on Scientific Computing* **40**, A3423–A3448, 2018.

[97] M. Kronbichler, S. Schoeder, C. Müller, and W. Wall, Comparison of implicit and explicit hybridizable discontinuous Galerkin methods for the acoustic wave equation, *International Journal for Numerical Methods in Engineering* **106**, 712–739, 2016.

[98] M. Kronbichler, K. Kormann, I. Pasichnyk, and I. Allalen, Fast matrix-free discontinuous Galerkin kernels on modern computer architectures, In J. M. Kunkel, R. Yokota, P. Balaji, and D. E. Keyes (eds.), *ISC High Performance 2017, LNCS 10266*, pages 237–255, 2017.

[99] R. Kruger, L. Pingyu, Y. Fang, and C. Appledorn, Photoacoustic ultrasound (PAUS)-reconstruction tomography, *Medical Physics* **22**, 1605–1609, 1995.

[100] R. A. Kruger, Photoacoustic ultrasound, *Medical Physics* **21**, 127–131, 1994.

[101] E. J. Kubatko, B. A. Yeager, and D. I. Ketcheson, Optimal strong-stability-preserving Runge–Kutta time discretizations for discontinuous Galerkin methods, *Journal of Scientific Computing* **60**, 313–344, 2014.

[102] M. Kufner. Entwurf und Implementierung eines Perfectly Matched Layer für die lineare Akustik. Master's thesis, Technichal University of Munich, 2016.

[103] A. Ledeczi, P. Volgyesi, M. Maroti, G. Simon, G. Balogh, A. Nadas, B. Kusy, S. Dora, and G. Pap, Multiple simultaneous acoustic source localization in urban terrain, In *Fourth International Symposium on Information Processing in Sensor Networks*, pages 491–496, 2005.

[104] L. Liu and D. Albert, Acoustic pulse propagation near a right-angle wall, *The Journal of the Acoustical Society of America* **119**, 2073–2083, 2006.

[105] X. Liu, D. Peng, X. Ma, W. Guo, Z. Liu, D. Han, X. Yang, and J. Tian, Limited-view photoacoustic imaging based on an iterative adaptive weighted filtered backprojection approach, *Applied Optics* **52**, 3477–3483, 2013.

[106] R. Löhner, Error and work estimates for high-order elements, *International Journal for Numerical Methods in Fluids* **67**, 2184–2188, 2011.

[107] C. Lutzweiler, R. Meier, and D. Razansky, Optoacoustic image segmentation based on signal domain analysis, *Photoacoustics* **3**, 151–158, 2015.

[108] S. Madsen, *Optical Methods and Instrumentation in Brain Imaging and Therapy*, Springer-Verlag New York, 2013.

[109] A. Mamonov and K. Ren, Quantitative photoacoustic imaging in the radiative transport regime, *Communications in Mathematical Sciences* **12**, 201–234, 2014.

[110] S. Manohar and D. Razansky, Photoacoustics: A historical review, *Advances in Optics and Photonics* **8**, 586–617, 2016.

[111] S. Marburg, pages 309–332, *Discretization Requirements: How many Elements per Wavelength are Necessary?*, S. Marburg and B. Nolte (eds.), Springer Berlin Heidelberg, 2008.

[112] A. Modave, E. Delhez, and C. Geuzaine, Optimizing perfectly matched layers in discrete contexts, *International Journal for Numerical Methods in Engineering* **99**, 410–437, 2014.

[113] A. Modave, A. Atle, J. Chan, and T. Warburton, A GPU-accelerated nodal discontinuous Galerkin method with high order absorbing boundary conditions and corner/edge compatibility, *International Journal for Numerical Methods in Engineering* **112**, 1659–1686, 2017.

[114] P. Mohajerani, S. Tzoumas, A. Rosenthal, and V. Ntziachristos, Optical and optoacoustic model-based tomography: Theory and current challenges for deep tissue imaging of optical contrast, *IEEE Signal Processing Magazine* **32**, 88–100, 2015.

[115] N. Morales, R. Mehra, and D. Manocha, Acoustic pulse propagation in an urban environment using a three-dimensional numerical simulation, *The Journal of the Acoustical Society of America* **135**, 3231–3242, 2014.

[116] N. Morales, R. Mehra, and D. Manocha, A parallel time-domain wave simulator based on rectangular decomposition for distributed memory architectures, *Applied Acoustics* **97**, 104–114, 2015.

[117] C. Müller. Comparison between explicit and implicit hybridizable discontinuous Galerkin formulations for the wave equation. Master's thesis, Technichal University of Munich, 2014.

[118] R. P. Muoz and M. Hornikx, Hybrid Fourier pseudospectral/discontinuous Galerkin time-domain method for wave propagation, *Journal of Computational Physics* **348**, 416–432, 2017.

[119] W. Naetar and O. Scherzer, Quantitative photoacoustic tomography with piecewise constant material parameters, *SIAM Journal on Imaging Sciences* **7**, 1755–1774, 2014.

[120] N. C. Nguyen, J. Peraire, and B. Cockburn, High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics, *Journal of Computational Physics* **230**, 3695–3718, 2011.

[121] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2nd Edition, 2006.

[122] V. Ntziachristos and D. Razansky, Molecular imaging by means of multispectral optoacoustic tomography (MSOT), *Chemical Reviews* **110**, 2783–2794, 2010.

[123] T. Okuzono, T. Yoshida, K. Sakagami, and T. Otsuru, An explicit time-domain finite element method for room acoustics simulations: Comparison of the performance with implicit methods, *Applied Acoustics* **104**, 76–84, 2016.

[124] A. A. Oraevsky, S. L. Jacques, R. O. Esenaliev, and F. K. Tittel, Direct measurement of laser fluence distribution and optoacoustic imaging in heterogeneous tissues, In *Proceedings of SPIE 2323, Laser Interaction with Hard and Soft Tissue II*, 1995.

[125] S. A. Orszag, Spectral methods for problems in complex geometries, *Journal of Computational Physics* **37**, 70–92, 1980.

[126] D. Peterseim and M. Schedensack, Relaxing the CFL condition for the wave equation on adaptive meshes, *Journal of Scientific Computing* **72**, 1196–1213, 2017.

[127] S. Petersen, C. Farhat, and R. Tezaur, A space-time discontinuos Galerkin method for the solution of the wave equation in the time domain, *International Journal for Numerical Methods in Engineering* **78**, 275–295, 2008.

[128] A. Pierce, *Acoustics – An Introduction to its physical principles and applications*, Acoustical Society of America, 1991.

[129] S. Piperno, Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems, *ESAIM: Mathematical Modelling and Numerical Analysis* **40**, 815–841, 2005.

[130] A. Pompei, M. A. Sumbatyan, and N. F. Todorov, Computer models in room acoustics: The ray tracing method and the auralization algorithms, *Acoustical Physics* **55**, 821, 2009.

[131] D. Rabinovich, D. Givoli, and E. Becache, Comparison of high order absorbing boundary conditions and perfectly matched layers in the frequency domain, *International Journal for Numerical Methods in Biomedical Engineering* **26**, 1351–1369, 2010.

[132] N. Raghuvanshi, R. Narain, and M. C. Lin, Efficient and accurate sound propagation using adaptive rectangular decomposition, *IEEE Transactions on Visualization and Computer Graphics* **15**, 789–801, 2009.

[133] W. Reed and T. Hill, Triangular mesh methods for the neutron transport equation, In *Proceedings of the American Nuclear Society*, 1973.

[134] J. Reil. Entwicklung eines lokalen Zeitschrittverfahrens für die akustische Wellengleichung. Semester's thesis, Technichal University of Munich, 2016.

[135] J. Reinten, P. E. Braat-Eggen, M. Hornikx, H. S. Kort, and A. Kohlrausch, The indoor sound environment and human task performance: A literature review on the role of room acoustics, *Building and Environment* **123**, 315–332, 2017.

[136] A. Rosenthal, D. Razansky, and V. Ntziachristos, Fast semi-analytical model-based acoustic inversion for quantitative optoacoustic tomography, *IEEE Transactions on Medical Imaging* **29**, 1275–1285, 2010.

[137] A. Rosenthal, V. Ntziachristos, and D. Razansky, Acoustic inversion in optoacoustic tomography: A review, *Current Medical Imaging Reviews* **9**, 318–336, 2013.

[138] Z. S. Sacks, D. M. Kingsland, R. Lee, and J.-F. Lee, A perfectly matched anisotropic absorber for use as an absorbing boundary condition, *IEEE Transactions on Antennas and Propagation* **43**, 1460–1463, 1995.

[139] G. Sanchez, T. V. Renterghem, P. Thomas, and D. Botteldooren, The effect of street canyon design on traffic noise exposure along roads, *Building and Environment* **97**, 96–110, 2016.

[140] S. Sauter and C. Schwab, *Boundary Element Methods*, Springer Verlag Berlin-Heidelberg, 2011.

[141] L. Savioja, Real-time 3D finite-difference time-domain simulation of low- and mid-frequency room acoustics, In *Proceedings of the 13th International Conference on Digital Audio Effects, DAFx-10*, Graz, Austria, 2010.

[142] L. Savioja, T. Rinne, and T. Takala, Simulation of room acoustics with a 3-D finite-difference mesh, In *International Computer Music Conference*, Aarhus, 1994.

[143] S. Schoeder, M. Kronbichler, and W. Wall, Photoacoustic image reconstruction: material detection and acoustical heterogeneities, *Inverse Problems* **33**, 055010, 2017.

[144] S. Schoeder, K. Kormann, W. Wall, and M. Kronbichler, Efficient explicit time stepping of high order discontinuous Galerkin schemes for waves, *SIAM Journal on Scientific Computing* **40**, C803–C826, 2018.

[145] S. Schoeder, M. Kronbichler, and W. Wall, Arbitrary high-order explicit hybridizable discontinuous Galerkin methods for the acoustic wave equation, *Journal of Scientific Computing* **76**, 969–1006, 2018.

[146] S. Schoeder, M. Kronbichler, and W. Wall, ExWave: A high performance discontinuous Galerkin solver for the acoustic wave equation, *SoftwareX* **9**, 49–54, 2019.

[147] S. Schoeder, I. Olefir, M. Kronbichler, V. Ntziachristos, and W. Wall, Optoacoustic image reconstruction: The full inverse problem with variable bases, *Proceedings of the Royal Society A* **474**, 20180369, 2018.

[148] S. Schoeder, S. Sticko, G. Kreiss, and M. Kronbichler, High order cut discontinuous Galerkin methods with local time stepping for acoustics, *submitted*, 2018.

[149] T. Schwartzkopff, C. D. Munz, and E. F. Toro, ADER: A high-order approach for linear hyperbolic systems in 2D, *Journal of Scientific Computing* **17**, 231–240, 2002.

[150] T. Schwartzkopff, M. Dumbser, and C.-D. Munz, Fast high order ADER schemes for linear hyperbolic equations, *Journal of Computational Physics* **197**, 532–539, 2004.

[151] M. Schweiger and S. Arridge, Image reconstruction in optical tomography using local basis functions, *Journal of Electronic Imaging* **12**, 583–593, 2003.

[152] M. Schweiger, S. Arridge, O. Dorn, A. Zacharopoulos, and V. Kolehmainen, Reconstructing absorption and diffusion shape profiles in optical tomography by a level set technique, *Optics Letters* **31**, 471–473, 2006.

[153] S. Siltanen, T. Lokki, and L. Savioja, Rays or waves? Understanding the strengths and weaknesses of computational room acoustics modeling techniques, In *Proceedings of the International Symposium on Room Acoustics, ISRA 2010*, Melbourne, Australia, 2010.

[154] A. Sommerfeld, *Partial differential equations in physics*, Academic Press, 1949.

[155] M. Stanglmeier, N. Nguyen, J. Peraire, and B. Cockburn, An explicit hybridizable discontinuous Galerkin method for the acoustic wave equation, *Computer Methods in Applied Mechanics and Engineering* **300**, 748–769, 2016.

[156] P. Stefanov and G. Uhlmann, Instability of the linearized problem in multiwave tomography of recovery both the source and the speed, *Inverse Problems and Imaging* **7**, 1367–1377, 2013.

[157] G. G. Stokes, Volume 1 of *Cambridge Library Collection - Mathematics*, page 75129, *On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solids*, G. G. Stokes (ed.), Cambridge University Press, 2009.

[158] A. Taflove and M. E. Brodwin, Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations, *IEEE Transactions on Microwave Theory and Techniques* **23**, 623–630, 1975.

[159] A. Taruttis and V. Ntziachristos, Advances in real-time multispectral optoacoustic imaging and its applications, *Nature Photonics* **9**, 219–227, 2015.

[160] T. Tarvainen, B. Cox, J. Kaipio, and S. Arridge, Reconstructing absorption and scattering distributions in quantitative photoacoustic tomography, *Inverse Problems* **28**, 084009, 2012.

[161] S. Teukolsky, Short note on the mass matrix for Gauss–Lobatto grid points, *Journal of Computational Physics* **283**, 408–413, 2015.

[162] B. Treeby, E. Zhang, and B. Cox, Photoacoustic tomography in absorbing acoustic media using time reversal, *Inverse Problems* **26**, 115003, 2010.

[163] J. Treibig, G. Hager, and G. Wellein, LIKWID: A lightweight performance-oriented tool suite for x86 multicore environments, In *Proceedings of PSTI2010, the First International Workshop on Parallel Software Tools and Tool Infrastructures*, San Diego CA, 2010.

[164] M. Vengerov, An optical-acoustic method of gas analysis, *Nature* **158**, 28–29, 1946.

[165] L. Wang and L. Gao, Photoacoustic microscopy and computed tomography: From bench to bedside, *Annual Review of Biomedical Engineering* **16**, 155–185, 2014.

[166] L. Wang (ed.), *Photoacoustic Imaging and Spectroscopy*, CRC Press Taylor and Franics Group, 2009.

[167] L. Wang and H.-I. Wu, *Biomedical Optics: Principles and Imaging*, John Wiley and Sons, 2007.

[168] G. Wellein, G. Hager, T. Zeiser, M. Wittmann, and H. Fehske, Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization, In *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International*, Volume 1, pages 579–586. IEEE, 2009.

[169] S. Williams, A. Waterman, and D. Patterson, Roofline: An insightful visual performance model for multicore architectures, *Communications of the ACM* **52**, 65–76, 2009.

[170] A. R. Winters and D. A. Kopriva, High-order local time stepping on moving DG spectral element meshes, *Journal of Scientific Computing* **58**, 176–202, 2014.

[171] A. Wirgin, The inverse crime, *arXiv preprint* `https://arxiv.org/abs/math-ph/0401050` **v1**, 2008.

[172] M. Wünnenberg. Photoacoustic image reconstruction: A study on mouse brains. Bachelor's thesis, Technichal University of Munich, 2017.

[173] M. Xu and L. Wang, Universal back-projection algorithm for photoacoustic computed tomography, *Physical Review E* **71**, 016706, 2005.

[174] M. Xu and L. Wang, Photoacoustic imaging in biomedicine, *Review of Scientific Instruments* **77**, 041101, 2006.

[175] Y. Xu and L. Wang, Time reversal and its application to tomography with diffracting sources, *Physical Review Letters* **92**, 033902, 2004.

[176] S. Yakovlev, D. Moxey, R. Kirby, and S. Sherwin, To CG or to HDG: A comparative study in 3D, *Journal of Scientific Computing* **67**, 192–220, 2016.

[177] H. Yang, F. Li, and J. Qiu, Dispersion and dissipation errors of two fully discrete discontinuous Galerkin methods, *Journal of Scientific Computing* **55**, 552–574, 2013.

[178] L. Yao, Y. Sun, and H. Jiang, Transport-based quantitative photoacoustic tomography: Simulations and experiments, *Physics in Medicine and Biology* **55**, 1917, 2010.

[179] K. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Transactions on Antennas and Propagation* **14**, 302–307, 1966.

[180] M. Zakerzadeh and G. May, On the convergence of a shock capturing discontinuous Galerkin method for nonlinear hyperbolic systems of conservation laws, *SIAM Journal on Numerical Analysis* **54**, 874–898, 2016.

Verzeichnis der betreuten Studienarbeiten

Im Rahmen dieser Dissertation entstanden am Lehrstuhl für Numerische Mechanik in den Jahren von 2013 bis 2018 unter wesentlicher wissenschaftlicher, fachlicher und inhaltlicher Anleitung des Autors die im Folgenden aufgeführten studentischen Arbeiten. Der Autor dankt allen Studierenden für ihr Engagement bei der Unterstützung dieser wissenschaftlichen Arbeit.

Studierende(r)	**Studienarbeit**
Christoph Goering	Implicit-Explicit Runge–Kutta Methods for Acoustics with the Hybridizable Discontinuous Galerkin Method, Masterarbeit, 2017.
Maximilian Wünnenberg	Photoacoustic Image Reconstruction: A Study on Mouse Brains, Bachelorarbeit, 2017, eingeflossen in Abschnitt 10.1.
Elia Ficapal	A Cut Hybridizable Discontinuous Galerkin Method for Acoustics, Bachelorarbeit, 2017, gemeinsame Betreuung mit Martin Kronbichler.
Sebastian Buckel	Photoakustische Bildrekonstruktion mit Level-Set-Funktionen, Semesterarbeit, 2017.
Christoph Goering	Slope Limiters and Shock Capturing for Acoustics with the Hybridizable Discontinuous Galerkin Discretization, Semesterarbeit, 2017.
Johannes Reil	Entwicklung eines lokalen Zeitschrittverfahrens für die akustische Wellengleichung, Semesterarbeit, 2016, gemeinsame Betreuung mit Martin Kronbichler, eingeflossen in Abschnitt 2.3.4.
Maxime Allard und Patrick Lux	Ground Reaction Forces and Impulse for Individual Steps during Sprint, MSE Forschungspraktikum, 2016.
Matthias Kufner	Entwurf und Implementierung eines Perfectly Matched Layer für die lineare Akustik, Masterarbeit, 2016, eingeflossen in Kapitel 4.
Alexander Kalichmanow	Vergleich zwischen der k-space Methode und der Hybridisierbaren Diskontinuierlichen Galerkin Methode für die akustische Wellengleichung, Semesterarbeit, 2016.
Julia Fischer	Photoacoustic Imaging: Detection of Acoustical Heterogeneities using Optical Material Identification, Masterarbeit, 2015, eingeflossen in Kapitel 9.
Niclas Bähr	Verwendung von Time Reversed Absorbing Conditions zur effizienten Berechnung des photoakustischen inversen Problems, Bachelorarbeit, 2015, eingeflossen in Kapitel 8.
Christopher Müller	Comparison between Explicit and Implicit Hybridizable Discontinuous Galerkin Formulations for the Wave Equation, Masterarbeit, 2014, eingeflossen in Abschnitt 2.3.2.