Sparse tensor approximation for radiative transport

A dissertation submitted to
ETH Zürich

for the degree of
Doctor of Sciences

presented by
Konstantin Grella
Dipl. Phys., Universität Heidelberg
born February 13, 1983
citizen of Germany

accepted on the recommendation of
Prof. Dr. Christoph Schwab, ETH Zürich, examiner
Prof. Dr. Ralf Hiptmair, ETH Zürich, co-examiner
Prof. Dr. Helmut Harbrecht, Universität Basel, co-examiner

2013
Abstract

In this thesis we develop, analyze, implement, and compare three sparse tensor approximation methods for the stationary monochromatic radiative transfer equation. The sparse tensor methods are based on the method of spherical harmonics, the discrete ordinates method, and a randomized version of the discrete ordinates method.

The radiative transfer equation is a PDE model for the propagation of radiation inside a domain in physical space into every angular direction. Due to the five dimensions of the phase space the equation is stated on, standard discretization approaches for this PDE lead to computationally very expensive methods subject to the so-called curse of dimensionality: To double the accuracy the computational effort must be increased by a factor of 32.

We formulate the three investigated methods as specializations of a generic phase space Galerkin finite element framework. From a least squares variant of the Galerkin framework, we derive the spherical harmonics or $P_N$-approximation with nested multilevel discretizations in physical and angular space. By sparse tensorization between physical and angular approximation spaces we construct a sparse tensor spherical harmonics method for the radiative transfer equation, breaking the curse of dimensionality in the same way as sparse grids in interpolation or approximation.

As a second method, we derive the popular discrete ordinates or $S_N$-method from a variant of the Galerkin framework with streamline upwind stabilization and nonnested approximation spaces. Sparse versions are constructed by direct sparse tensorization and with the aid of the sparse grid combination technique.

For both the sparse spherical harmonics and the sparse discrete ordinates method, we prove that the convergence rate of the standard full tensor variant is maintained up to a logarithmic factor, while the number of degrees of freedom increases only as for a purely spatial problem, again up to a logarithmic factor in the number of degrees of freedom. This increase in approximation efficiency is achieved in exchange for slightly higher requirements on the regularity of the solution.

For applications with solutions which are only square-integrable over the angular domain we propose the third method developed in this thesis: a randomized sparse discrete ordinates method based on a multilevel Monte Carlo scheme. Here, we interpret the angular domain as a probability space and combine solutions of all resolutions in such a way that the solution mean over the angle converges with typical rate $1/2$ in the angular degrees of freedom, while the number of degrees of freedom again reduces essentially to that of a purely spatial problem.

A large part of this work is devoted to the implementation of the full and sparse discrete ordinates method in a modular, MPI-parallel, dimension-independent C++ code. This software allows to solve 2+1D and 3+2D radiative transfer problems in a distributed manner on HPC hardware. In a number of numerical experiments conducted with the new software, we explore the applicability as well as the limits of our methods.
Zusammenfassung

In dieser Doktorarbeit werden drei Dünngitterapproximationsmethoden für die stationäre frequenzunabhängige Strahlentransportgleichung entwickelt, untersucht, implementiert und verglichen. Die Methoden basieren auf der Kugelflächenfunktionsmethode, der Methode der diskreten Ordinaten sowie einer randomisierten Version der Diskrete-Ordinaten-Methode.

Die Strahlentransportgleichung ist ein mathematisches Modell in Form einer partiellen Differentialgleichung, das die Ausbreitung von elektromagnetischer Strahlung in jede Richtung auf einem physikalischen Gebiet beschreibt. Aufgrund der fünf Dimensionen des Phasenraums, auf dem die Gleichung gestellt ist, führen gewöhnliche Diskretisierungsansätze auf rechentechnisch äußerst aufwändige Verfahren, die dem sogenannten Fluch der Dimension unterworfen sind: Für eine Verdopplung der Genauigkeit der Lösung muss der Berechnungsaufwand verzweundreißigfacht werden.


Für dünne $P_N$- und $S_N$-Methode beweisen wir, dass die Konvergenzrate der gewöhnlichen Volltensorvariante bis auf einen logarithmischen Faktor erhalten bleibt, wohingegen die Anzahl Freiheitsgrade nur wie bei einem rein physikalischen Problem anwächst, wiederum bis auf einen logarithmischen Faktor in der Anzahl Freiheitsgrade. Diese Steigerung der Approximationseffizienz erzielen wir durch geringfügig höhere Anforderungen an die Glattheit der Lösung als in der Volltensorvariante.

Für Fälle, in denen die Lösung im Winkel lediglich quadratintegrierbar ist, stellen wir das dritte in dieser Arbeit entwickelte Verfahren vor: eine randomisierte dünne Diskrete-Ordinaten-Methode, die auf einer Multiskalen-Monte-Carlo-Technik basiert. Bei dieser Methode wird der Winkelraum als Wahrscheinlichkeitsraum aufgefasst und Lösungen verschiedener Auflösung so kombiniert, dass die für Monte-Carlo-Methoden typische Konvergenzrate von 1/2 im Winkelmittel der Lösung bezüglich der Winkelfreiheitsgrade erreicht wird, während sich die gesamte Anzahl Freiheitsgrade quasi auf jene eines rein physikalischen Problems reduziert.

Zu einem grossen Teil widmet sich diese Arbeit auch der Implementierung der gewöhnlichen und der dünnen Diskrete-Ordinaten-Methode in einem modularen, parallelen, dimensionsunabhängigen C++ Code. Dieses Programm erlaubt es, Strahlentransportprobleme in 2+1D und 3+2D auf Höchstleistungsrechnern verteilt zu lösen.
Acknowledgments

First I would like to thank my supervisor Prof. Christoph Schwab for accepting me as a PhD student and providing helpful guidance throughout the course of this project. He always took the time for explanations and discussions, which allowed me to learn and profit substantially from his exceptional knowledge. I am also indebted to my co-examiners Prof. Ralf Hiptmair for his advice and ideas and to Prof. Helmut Harbrecht for his hints at improvements of this thesis. I would like to express my gratitude also to Prof. Philipp Grohs for his help on some theoretical questions. Many thanks go to Dr. Oleg Reichmann, Dr. Roman Andreev, and Dr. Andrea Barth for their time for extended discussions, and to Dr. Julie Michaud-Bernlochner for proofreading parts of this thesis. The members of the Brutus team have been very forthcoming with help and tips on optimizing the usage of the available cluster resources. In his semester thesis, Simon Härdi invested a lot of effort into parallelizing and testing an earlier version of the code described in this work. To all colleagues and former colleagues at the Seminar for Applied Mathematics I am very thankful for making my time there such an enjoyable experience.

Financial support for the work on this thesis by Schweizerischer Nationalfonds (SNF) under project no. 121892, by Deutsche Forschungsgemeinschaft (DFG) within SPP1324, and by the European Research Council (ERC) under ERC Advanced Grant 247277 is gratefully acknowledged.

Finally I wish to say a big thanks to my parents, I owe everything to them, and especially to Helena for her continuing love and support.
Contents

Abstract iii

Acknowledgments vii

1 Introduction 1
   1.1 Equations and quantities ........................................ 1
   1.2 Applications .................................................... 2
   1.3 Methods .......................................................... 4
   1.4 Outline of the thesis ............................................. 7

2 Preliminaries 9
   2.1 Sobolev norms and spaces ....................................... 9
   2.2 Sparse grids ...................................................... 10
      2.2.1 Introduction .................................................. 10
      2.2.2 Complexity estimates ....................................... 12
   2.3 Sparse tensor combination technique .......................... 14
      2.3.1 Derivation of the combination approximation ............ 15
      2.3.2 Complexity estimates ....................................... 18

3 The phase space Galerkin method 21
   3.1 Operator formulation .......................................... 21
   3.2 The scattering operator ....................................... 22
      3.2.1 Properties from physical considerations ............... 22
      3.2.2 Models for the scattering phase function ............. 22
      3.2.3 Mathematical assumptions and properties ............. 23
   3.3 Variational formulation ....................................... 25
      3.3.1 A generic stabilized phase space variational formulation .................................................. 25
      3.3.2 Properties of the variational formulation ........... 27
   3.4 Discretization ................................................... 29
      3.4.1 Discretization in physical and angular space ........ 29
      3.4.2 Solution of the linear system ............................ 32

4 Spherical harmonics method 35
   4.1 Variational formulation ....................................... 35
   4.2 Discretization ................................................... 37
      4.2.1 Angular basis functions .................................. 37
      4.2.2 Approximation spaces without boundary conditions .... 40
      4.2.3 Approximation spaces with boundary conditions ...... 43
Contents

4.3 Approximation properties ............................................. 48
  4.3.1 Approximation on the physical domain ....................... 48
  4.3.2 Approximation on the angular domain ......................... 49
  4.3.3 Approximation properties of full tensor product space ...... 50
  4.3.4 Approximation properties of sparse tensor product space ... 51

5 Discrete ordinates method ............................................. 55
  5.1 Variational formulation .............................................. 56
  5.2 Discretization ....................................................... 59
    5.2.1 Alternative discretization by collocation in angle .......... 59
    5.2.2 SUPG stabilization parameter ................................ 60
    5.2.3 Approximation spaces ......................................... 60
    5.2.4 Quadrature in angle ........................................... 62
    5.2.5 Matrix representation of the discrete ordinates method .... 63
    5.2.6 Sparse tensor combination technique DOM .................. 64
  5.3 Error analysis ...................................................... 67
    5.3.1 Galerkin projection ............................................ 68
    5.3.2 Error estimates on the physical domain .................... 68
    5.3.3 Error estimates on the angular domain ...................... 69
    5.3.4 Error estimate for the full tensor phase space Galerkin DOM 72
    5.3.5 Consistency error due to quadrature in the scattering operator 73
    5.3.6 Error estimate for the sparse tensor phase space Galerkin DOM 76
    5.3.7 Error estimate for the combination technique DOM .......... 78

6 Randomized sparse discrete ordinates method ..................... 79
  6.1 Discretization ...................................................... 79
    6.1.1 Discretization in angle ...................................... 79
    6.1.2 Generation of uniform random points on the angular domain 80
  6.2 Approximation properties ........................................... 80
    6.2.1 Approximation on the angular domain ....................... 81
    6.2.2 Error estimate for single-level MC Galerkin method .......... 82
    6.2.3 Error estimate for multilevel MC Galerkin method .......... 84
    6.2.4 Error of incident radiation in deterministic DOM .......... 86

7 Implementation ....................................................... 89
  7.1 Requirements specification ....................................... 89
    7.1.1 Problem statement ............................................ 89
    7.1.2 Goals ......................................................... 89
    7.1.3 Use cases .................................................... 90
    7.1.4 Key functional features and nonfunctional requirements .... 90
  7.2 Software design ................................................... 92
    7.2.1 Design decisions ............................................. 92
    7.2.2 Architecture ................................................ 94
    7.2.3 Interaction .................................................. 98
Contents

7.3 Algorithms .................................................. 101
  7.3.1 Quadrature .............................................. 101
  7.3.2 Linear solver and preconditioner ................... 102
  7.3.3 Scheduling .............................................. 103
7.4 Parallel performance results ............................... 105
  7.4.1 Computing environment ............................... 105
  7.4.2 Strong scaling ......................................... 106
  7.4.3 Weak scaling ........................................... 109

8 Numerical experiments ........................................ 115
  8.1 Spherical harmonics and discrete ordinates in 2D ........ 115
    8.1.1 Algorithms .................................... 115
    8.1.2 Experiments .................................. 116
  8.2 Discrete ordinates in 3D ................................. 123
  8.3 Randomized discrete ordinates method ................... 128

9 Conclusion ................................................... 131
  9.1 Summary .................................................. 131
  9.2 Future work .............................................. 133

List of symbols .................................................. 135
References ....................................................... 137
1 Introduction

This thesis is concerned with the efficient numerical solution of a high-dimensional parametric transport equation of a particular type known as the radiative transfer equation. In this first chapter, we introduce this equation and occurring quantities, name examples of applications, describe common solution methods, and give an outline of this work.

1.1 Equations and quantities

The radiative transfer problem (RTP) (see e.g. Modest, 2003) is the task of finding the radiative intensity $u : D \times S \to \mathbb{R}$, defined on the Cartesian product of a bounded physical domain $D \subset \mathbb{R}^d$, where $d = 2, 3$, and the unit $d_S$-sphere as the angular domain $S$ with $d_S = 1, 2$, such that it satisfies

$$s \cdot \nabla_x u(x, s) + (\kappa(x) + \sigma(x))u(x, s) = \kappa(x)I_b(x) + \sigma(x) \int_S \Phi(s, s')u(x, s') \, ds', \quad (1.1a)$$

$$u(x, s) = g(x, s), \quad x \in \partial D, \; s \cdot n(x) < 0 . \quad (1.1b)$$

The partial differential equation (1.1a) is known as the stationary monochromatic radiative transfer equation (RTE), complemented by inflow boundary conditions (1.1b). Radiation may interact with a medium inside the physical domain via the two mechanisms of absorption and scattering. The strength of absorption is modeled by the absorption coefficient $\kappa \geq 0$, that of scattering by the scattering coefficient $\sigma \geq 0$, while the scattering kernel or scattering phase function $\Phi > 0$ determines the type of scattering. Radiation may be emitted by sources inside the domain represented by the blackbody intensity $I_b \geq 0$, or enter the domain from the outside or emanate from the walls, which is both described by the boundary data $g \geq 0$. The vector $n(x)$ denotes the outer unit normal on the boundary $\partial D$ of the physical domain.

The scattering phase function is normalized to $\int_S \Phi(s, s') \, ds' = 1$ for each direction $s$, meaning that numbers of photons, the mediating particles of radiation, are conserved during scattering events. Furthermore, the scattering phase function is symmetric in its arguments, $\Phi(s, s') = \Phi(s', s)$, because trajectories of photons are reversible.

The RTP without scattering results from Eq. (1.1) by setting $\sigma \equiv 0$ and $\Phi \equiv 0$.

When the terms $(\kappa(x) + \sigma(x))u(x, s)$ are moved to the right hand side of Eq. (1.1a), the motivation of the RTE as a balance equation for the density $u$ of photons at location $x$ traveling into direction $s$ becomes apparent. The change $s \cdot \nabla_x u$ of the photon density along direction $s$ is equal to the sum of negative contributions $-\kappa u$ due to absorption of the medium and $-\sigma u$ due to outscattering, and positive contributions $\kappa I_b$ from...
1 Introduction

photon emission of sources inside the domain and inscattering $\sigma \int_\mathcal{S} \Phi(s,s')u(x,s')\,ds'$ into direction $s$ from all other directions $s'$.

The zeroth angular moment of the radiative intensity is called incident radiation:

$$G(x) := \int_\mathcal{S} u(x,s)\,ds.$$  \hfill (1.2)

It is related to the first angular moment of the radiative intensity, the heat flux,

$$q(x) := \int_\mathcal{S} u(x,s)s\,ds,$$  \hfill (1.3)

via the net emission $\nabla_x \cdot q$ as follows (cf. Widmer, 2009, Sec. 1.1.2, here also respecting scattering):

$$\nabla_x \cdot q = \int_\mathcal{S} \nabla_x \cdot u(x,s)s\,ds = \int_\mathcal{S} s \cdot \nabla_x u(x,s)\,ds = \int_\mathcal{S} \left[ \kappa(x)I_b(x) - (\kappa(x) + \sigma(x))u(x,s) + \sigma(x) \int_\mathcal{S} \Phi(s,s')u(x,s')\,ds' \right]\,ds$$

$$= |\mathcal{S}| \kappa(x)I_b(x) - \kappa(x)G(x) - \sigma(x)G(x) + \sigma(x) \int_\mathcal{S} \int_\mathcal{S} \Phi(s,s')\,ds\,u(x,s')\,ds'$$

with the 1-measure of the sphere $|\mathcal{S}| = 4\pi$ for $d_S = 2$, and $|\mathcal{S}| = 2\pi$ for $d_S = 1$, respectively. For the last transformation, we exploited symmetry and normalization of the scattering phase function.

These quantities become relevant when radiation as a means of heat transfer is coupled to convection and conduction, as the radiative transfer part enters energy balance equations usually via the net emission (e.g. Larsen et al., 2002, Eq. (1.1a)). Also, the incident radiation or the heat flux lend themselves more easily to visualization than the radiative intensity with its dependence on five variables in 3D. Therefore results of numerical experiments in this thesis will be displayed in one of these variables.

1.2 Applications

The RTP appears as in Eq. (1.1) or in slightly modified form in various applications as a model for the transport of particles which interact with background media, but only negligibly with each other.

Such particles are the aforementioned photons as carriers of the electromagnetic radiation in situations when their wavelike nature does not play a role. The primary application of the RTE is therefore the simulation of the propagation of heat radiation (Modest, 2003).

However, neutrons of a particular energy, for instance, can be described by the RTE as well. To take interactions between neutrons of different energies into account, the energy variable is often discretized into a number $g_{\text{tot}}$ of fixed energies $E_g$, so-called groups. This ansatz results in a system of RTEs where the intensities of different
Applications of energies are coupled through the scattering term and a possible fission term (e.g. Hébert, 2010, Eq. (108)):

\[
s \cdot \nabla u_g(x, s) + (\kappa_g(x) + \sigma_g(x)) u_g(x, s) = Q_g(x, s) \\
+ \sigma(x) \sum_{g'=1}^{g_{\text{tot}}} \int_{S} \Phi_{g,g'}(s, s') u_{g'}(x, s') \, ds' + \frac{\chi_g}{|S|} \sum_{g'=1}^{g_{\text{tot}}} \Sigma_{f,g'} u_{g'}(x, s), \quad g = 1, \ldots, g_{\text{tot}}.
\]

Here, \( Q_g \) denotes a generic source term, \( \chi_g \) the probability of neutrons of energy \( E_g \) being created in a fission event, and \( \Sigma_{f,g'} \) the fission cross section for neutrons of energy \( E_{g'} \).

Applications of this neutron transport equation are the design and operation of nuclear reactors (Baker et al., 2012), for example. However, as the dimensionality of the equation increases to six or seven variables when time is added, billions of unknowns are required to sufficiently resolve complete reactors with standard methods. Thus, these problems are among the computationally most demanding in computational science. The simulation of temporal evolutions of complete reactors will likely only come into reach in the next years on the largest computing systems available (Kaushik et al., 2009).

Returning to Eq. (1.1a) for electromagnetic radiation, a variety of applications present themselves also for this field.

In astronomy, the transmission of infrared and visible star light through dust and interstellar gas clouds is described by the RTE (e.g. example by Kanschat, 1996).

Clouds and light from our own star also play an important role in climate modeling. A crucial detail of many climate models is the question of how much sun light and heat radiation from the earth is transmitted and how much reflected by different types of clouds in the atmosphere. Radiative transfer simulations of these clouds in 3D help to determine their properties (Evans, 1998).

In medical imaging, biological tissue is irradiated by laser light or x-rays and the reflected and scattered radiation is detected with the aim of reconstructing the inner structure of the organic material. This task corresponds to an inverse problem: The distribution of absorbers \( \kappa \) and scatterers \( \sigma \) are supposed to be determined from a quantity derived from the intensity \( u \). Such problems are usually solved iteratively by many repeated solutions of the forward problem. Starting from an initial guesses for \( \kappa \) and \( \sigma \), these functions are corrected in each iteration in order to minimize a certain discrepancy between the computed quantity and the detected one.

Obviously a fast solution of the forward problem is critical for the applicability of a method for this problem. Two ways towards this goal can be explored, the first the development of improved or new methods with shorter time to solution or better convergence properties, the second the acceleration of existing algorithms by utilizing modern computing hardware such as GPUs (e.g. Peng et al., 2011).

When radiation of higher energy or neutrons are considered, it becomes important to shield this type of radiation to limit its health-damaging effects. The RTE is therefore also employed in applications with complicated setups in medical imaging, space
exploration, or security to determine the radiation intensity and the effectiveness of absorbing materials (Shultis and Faw, 2010, Sec. 5.5 and end of Sec. 5.6).

Finally, in multiphysics simulations such as magnetohydrodynamics (e.g. simulations of plasmas and supernovae), radiative transfer contributes to energy equations as a means of energy transfer, but the radiative intensity field itself might not be the quantity of primary interest, as mass density, momentum, and energy of matter particles are to be determined as well. With such a large number of unknown quantities, the dimensionality of the application problem becomes the central issue in numerical simulations (Livne et al., 2004). Costly high-dimensional solution methods for the radiative intensity are therefore to be avoided. To reduce the dimensions, the radiative transfer problem is often assumed to be isotropic in angle so that only the radial component needs to be calculated (Livne et al., 2004; Abdikamalov et al., 2012). Methods which reduce the dimensionality and trade some accuracy for speed could be ideally suited for this application.

1.3 Methods

The comprehensive compendium on radiative transfer by Modest (2003) contains introductions to most important methods for the RTP. More recent advances and typical applications are documented in a compilation by Kanschat et al. (2008).

By a first categorization these methods can be divided into deterministic and non-deterministic methods.

Non-deterministic methods

The popular Monte-Carlo ray-tracing methods belong to the group of non-deterministic methods. In these methods packets of particles with randomly chosen properties are traced through the domain and an approximation to the intensity distribution is derived by means of stochastic analysis. The advantage of these methods is that they identify the correct solution reliably provided that the optical properties of the medium are not too extreme, that is, absorption and scattering are neither too low nor too high (Stamatellos and Whitworth, 2008). Also, since tracing individual ray packets are independent tasks, they typically lend themselves very well to parallelization (see example by Peng et al., 2011). However, convergence is slow, i.e. a large number of particle packets are required to obtain an accurate solution.

Deterministic methods

A summary of deterministic numerical methods is given by Frank (2007). Among the deterministic methods, most widely adopted are the discrete ordinates method and the method of spherical harmonics.

In the discrete ordinate method (DOM) or $S_N$-approximation, the angular domain is discretized by a number of fixed directions, which are inserted into Eq. (1.1) so that a system of spatial PDEs results. Without scattering the equations for single directions
1.3 **Methods**

are independent of each other, with scattering, however, they are coupled through the scattering integral, which is replaced by a quadrature rule over the chosen directions. After the straightforward discretization of the angular domain, the spatial PDEs are typically solved using finite differences, finite elements, or finite volume methods.

Advantageous in the DOM are the simple implementation and the fact that it can capture directed radiation relatively well as some of the ordinates can usually be chosen freely. On the downside, the method can suffer from ray effects (Lathrop, 1968): Due to the point evaluation in the angular domain, the scalar flux or incident radiation from small isotropic sources may appear star-like with rays emanating from the source into the chosen angular directions (cf. images provided by Stone, 2007, p. 2 and following). These effects occur especially pronounced in settings with low scattering and absorption, i.e. in optically thin media.

An example for truncated series expansion is the method of spherical harmonics or \( P_N \)-approximation. The solution of Eq. (1.1) is replaced by a series of spherical harmonics up to some order \( N \) with spatially dependent coefficients. Due to orthogonality relations, the scattering part often decouples or couples only few terms depending on the scattering kernel. However, the system of PDEs for the spatial coefficient functions is always coupled by the transport part \( s \cdot \nabla_x u \).

As low order series expansions in spherical harmonics do not permit a very localized resolution of the angular variable, the method performs best when the solutions are nearly isotropic in angle, which is the case in very diffusive, optically thick media. Then, very few orders might suffice for a good approximation. Indeed, the \( P_1 \) method can be formulated as a diffusion equation for the incident radiation (Modest, 2003, Sec. 15.4). For smooth solutions, the spherical harmonics method exhibits spectral convergence (Grella and Schwab, 2011a).

On the other hand, beam-like solutions require a high order to be resolved appropriately, leading to high computational complexity. In general, higher orders also lead to a sharp increase in mathematical complexity when boundary conditions are to be satisfied (Modest and Yang, 2008).

Further existing methods are moment methods with various closures (Schäfer et al., 2011), deterministic ray-tracing (Moss and Rubini, 1997), and combinations of the aforementioned methods (Evans, 1998).

**Curse of dimensionality**

The popular deterministic methods \( S_N \)- and \( P_N \)-approximation suffer from the so-called “curse of dimensionality”. By this term, we refer to the fact that the convergence rate in some error with respect to the numbers of degrees of freedom (DoF) \( M_D \) and \( M_S \) on the component domains \( D \) and \( S \) is usually divided by the dimension \( d \) and \( d_S \) of the application problem so that the error scales like \( O(M_D^{-s/d} + M_S^{-t/d_S}) \) with constants \( s \) and \( t \).

Some numbers make the problem more evident: Assumed \( s = t = 1 \) and equilibration of errors, then in 2D with \((d,d_S) = (2,1)\) we would require four times as many spatial DoFs to halve the spatial error and twice as many angular DoFs to halve the
angular error, but the overall number of DoFs $M = M_D M_S$ would increase by a factor of eight. In 3D, to cut the total error in half, a factor of 32 in the total number of DoFs would be needed, rendering accurate discretizations prohibitively expensive.

For the spherical harmonics method, there exist approaches to reduce the workload while maintaining accuracy. Schäfer et al. (2011) use a method of moments with entropy closure to modify the equations for the degrees of freedom of the highest order $N$ to apparently achieve the accuracy of a $P_{N+1}$-approximation already for their diffusive approximation of order $N$. Modest and Yang (2008) suggest a successive elimination of spherical harmonic tensors to reduce the number of simultaneous differential equations from $(N+1)^2$ to $N(N+1)/2$ in 3D. While these methods reduce the amount of computation compared to the standard full radiative transfer problem, the overall asymptotic complexity remains the same. Furthermore, the analytical derivation of formulas to ensure satisfaction of the boundary conditions becomes increasingly involved with higher order $N$.

Widmer et al. (2008) have developed a method to overcome the curse of dimensionality for radiative transfer in the context of a wavelet discretization of the angular domain. In their sparse tensor product method, they discretize physical and angular domain with hierarchical and wavelet finite elements, respectively, and then select a-priori only the most relevant product basis combinations for smooth functions to construct the trial space for the solution. Under certain regularity assumptions on the absorption coefficient $\kappa$ and the blackbody intensity $I_b$, their method achieves a log-linear complexity in the number of degrees of freedom, while convergence rates deteriorate only by a logarithmic factor. Their method is suited for the optically thick and thin regime. However, scattering had not been addressed in that work.

The contribution of this thesis is to apply the notion of sparse tensorization to the more widely used methods of $S_N$- and $P_N$-approximation for 2D and 3D applications with scattering.

We derive sparse tensor versions of both methods and prove that these sparse methods maintain the convergence rates of their standard full tensor counterparts up to logarithmic factors in the degrees of freedom of one component domain while their total number of degrees of freedom reduces essentially to that on one component domain, again up to logarithmic factors. The reduction in complexity will be achieved by slightly elevated regularity requirements on the solution.

Additionally, we propose a randomized sparse discrete ordinates method for applications with solutions of low angular regularity. As for the other methods, we prove that the benefits of sparse tensorization can be harnessed, although in a weaker sense of convergence than before.

The properties of our new methods are investigated analytically and in series of numerical experiments. These numerical experiments have become possible in part by the development of a new flexible parallel solver for radiative transfer problems in 2D and 3D. In the implementation of this software, we explore how generic programming can serve in scientific computing to create modular and efficient programs alike.
1.4 Outline of the thesis

In this thesis, we will develop, analyze, implement and test a sparse spherical harmonics method, a sparse discrete ordinates method, and a randomized sparse discrete ordinates method for the RTP.

In the next chapter, we begin by a number of preliminaries. First we introduce necessary notation of Sobolev norms and spaces. Then we review the notion of sparse grids, sparse tensorization and the related combination technique, which our methods are built upon.

In Chapter 3, we state the RTP in operator form, investigate properties of the scattering operator, and develop a generic phase space Galerkin finite element method for the RTP which will serve as theoretical framework for the analysis of the methods.

Chapter 4 is devoted to the sparse tensor spherical harmonics method. We specialize the Galerkin framework for a spectral discretization in angle with least squares stabilization, prove its well-posedness, and show that the benefits of sparse tensorization as outlined in the previous section also hold for a spectral discretization. The contents of this chapter have been published to a large extent in previous work (Grella and Schwab, 2011a).

In Chapter 5, we present a direct sparse tensor DOM and a combination technique DOM based on a streamline upwind Petrov Galerkin (SUPG) stabilized version of the Galerkin finite element framework. In this formulation we prove well-posedness of the problem statement and convergence rates of the intensity error of the direct sparse tensor DOM essentially equal to the full tensor method while reducing the complexity to that of a purely spatial problem up to logarithmic factors. Also, we consider the consistency error introduced by replacing the scattering integral by a quadrature rule according to Nyström’s method and investigate under which conditions this error can be ignored compared to the Galerkin approximation error.

We have presented the combination technique DOM for RTPs without scattering already before (Grella and Schwab, 2011b). Compared to that work, the theoretical basis for the analysis has been replaced by the phase space Galerkin method, which allows for more consistent handling of the case with scattering.

A new parallel C++ implementation of the sparse and full DOM is introduced in Chapter 7. We lead the reader through the design process of the software, outline its architecture and inner workings, highlight in some examples how numerical software can profit from the generic programming paradigm, and analyze its parallel performance by scaling measurements on the ETH cluster Brutus up to around 1000 cores.

In Chapter 8, we present the results of numerical experiments, comparing full and sparse tensor methods in 2D and 3D examples.

Chapter 9 summarizes this work, emphasizing once again advantages and disadvantages of the different methods. Finally we propose improvements to address remaining issues in our methods and give directions for further research.
2 Preliminaries

We first establish notation for function spaces used throughout this work. Then we discuss the concepts of sparse grids and the related combination technique which our methods in the following chapters build upon.

2.1 Sobolev norms and spaces

We define Lebesgue spaces of functions $f : D \rightarrow \mathbb{R}$ on nonempty bounded Lebesgue measurable domains $D \subset \mathbb{R}^d$ in the usual way (cf. Brenner and Scott, 2008, Chap. 1) via the norms

$$
\| f \|_{L^p(D)} := \left( \int_D |f(x)|^p \, dx \right)^{1/p}, \quad 1 \leq p < \infty,
$$

and for $p = \infty$

$$
\| f \|_{L^\infty(D)} := \text{ess sup}_{x \in D} \{|f(x)|\}.
$$

The sets of $L^p$-integrable functions on $D$ are then given by

$$
L^p(D) := \{ f : \| f \|_{L^p(D)} < \infty \}. \tag{2.3}
$$

The special case $L^2(D)$ forms a Hilbert space together with the inner product

$$
(u, v)_{L^2(D)} := \int_D u(x)v(x) \, dx, \quad u, v \in L^2(D). \tag{2.4}
$$

With a multi-index $\alpha \in \mathbb{N}_0^d$ with $|\alpha| = \alpha_1 + \ldots + \alpha_d$ and the notion of the weak derivative $D^\alpha f$ (Brenner and Scott, 2008, Def. 1.2.4), the definition of Sobolev spaces is based on that of the Lebesgue spaces (2.3):

$$
W^k_p(D) := \{ f \in L^p(D) : D^\alpha f \in L^p(D) \forall |\alpha| \leq k \}, \quad 1 \leq p \leq \infty. \tag{2.5}
$$

Corresponding norms are given for $k \in \mathbb{N}_0$ and $1 \leq p < \infty$ by

$$
\| f \|_{W^k_p(D)} := \left( \sum_{|\alpha| \leq k} \| D^\alpha f \|_{L^p(D)}^p \right)^{1/p}, \tag{2.6}
$$

and for $p = \infty$ by

$$
\| f \|_{W^k_\infty(D)} := \max_{|\alpha| \leq k} \| D^\alpha f \|_{L^\infty(D)}. \tag{2.7}
$$
For $p = 2$, we denote the Sobolev spaces $W^k_{2}(D)$ by $H^k(D)$, as they form Hilbert spaces together with inner product

$$
(u, v)_{H^k(D)} := \sum_{|\alpha| \leq k} (D^\alpha u, D^\alpha v)_{L^2(D)}, \quad u, v \in H^k(D).
$$

(2.7)

As the phase space of the RTP is a Cartesian product domain, we also introduce notation for Hilbert spaces over such domains. Let $\Omega$ be a Cartesian product domain $\Omega = D \times S$ with $D \subset \mathbb{R}^d$, $S \subset \mathbb{R}^d$ and $f : \Omega \rightarrow \mathbb{R}$. $L^p$-norms are extended from Eq. (2.1) to

$$
\|f\|_{L^p(\Omega)} := \left( \int_D \int_S |f(x, s)|^p \, dx \, ds \right)^{1/p}, \quad 1 \leq p < \infty,
$$

(2.8)

the maximum norm and all $L^p(\Omega)$ spaces are defined analogously by natural extension from Eqs. (2.2) and (2.3). The $L^2(\Omega)$ inner product is

$$
(u, v)_{L^2(\Omega)} := \int_S \int_D u(x, s)v(x, s) \, dx \, ds.
$$

(2.9)

If space is short we may abbreviate $(\cdot, \cdot) := (\cdot, \cdot)_{L^2(\Omega)}$ and $\|\cdot\| := \|\cdot\|_{L^2(\Omega)}$ later on.

The anisotropic Sobolev spaces $H^{s,t}(\Omega) = H^s(D) \otimes H^t(S)$ are defined by

$$
H^{s,t}(\Omega) := \{ v \in L^2(\Omega) : D^\beta_x D^\alpha_s v \in L^2(\Omega), \quad 0 \leq |\alpha| \leq s, \quad 0 \leq |\beta| \leq t \}
$$

(2.10)

with the corresponding mixed Sobolev norms $\|\cdot\|_{H^{s,t}(\Omega)}$, given by

$$
\|v\|^2_{H^{s,t}(\Omega)} := \sum_{0 \leq |\alpha| \leq s} \sum_{0 \leq |\beta| \leq t} \|D^\beta_x D^\alpha_s v\|_{L^2(\Omega)}^2.
$$

(2.11)

Here, $D^\beta_x D^\alpha_s v$ denotes the weak derivative of $v : D \times S \rightarrow \mathbb{R}$ of order $|\alpha|$ w. r. t. $x \in D$ and order $|\beta|$ w. r. t. $s \in S$, with the multi-indices $\alpha \in \mathbb{N}_0^d$ and $\beta \in \mathbb{N}_0^{d_S+1}$.

### 2.2 Sparse grids

The use of finite element spaces based on sparse grids for the solution of PDEs goes back to Zenger (1991), whose introduction we follow here while adapting the notation to our needs. A detailed account about properties, variants, and applications of sparse grids is given by Bungartz and Griebel (2004).

#### 2.2.1 Introduction

Sparse grids serve to reduce the complexity of function representations by e. g. interpolation or approximation on product domains. Let the function $u : \Omega \rightarrow \mathbb{R}$ be given on the Cartesian product domain $\Omega = D_1 \times D_2$ with component domains $D_i \subset \mathbb{R}^{d_i}$, $i = 1, 2$. For the purpose of an example, let $D_i = [0, 1] \subset \mathbb{R}$, and the approximation be
2.2 Sparse grids

interpolation, just as in the initial example by Zenger (1991). The function $u$ lives in some function space $\mathcal{V}$ isomorphic to the tensor product of component function spaces $\mathcal{V}_1 \otimes \mathcal{V}_2$, which will be specified later on.

The approximation $u_{L_1,L_2}$ to $u$, the piecewise bilinear interpolant on the tensor product mesh $\mathcal{T}^{L_1,L_2} = \mathcal{T}_1^{L_1} \times \mathcal{T}_2^{L_2}$ with $\mathcal{T}_i^{L_i} = \{x_j \in D_i : x_j = j2^{-L_i}, j = 0, \ldots, 2^{L_i}\}$, $i = 1,2$, has a finite expansion in basis functions from the tensor product function space $\mathcal{V}^{L_1,L_2} = \mathcal{V}_1^{L_1} \otimes \mathcal{V}_2^{L_2}$, which can be written as

$$u_{L_1,L_2} = \sum_{l_1=1}^{M_1} \sum_{l_2=1}^{M_2} u_{l_1,l_2} a_{l_1}(x_1) b_{l_2}(x_2), \quad u_{l_1,l_2} \in \mathbb{R}, \; a_{l_1} \in \mathcal{V}_1^{L_1}, \; b_{l_2} \in \mathcal{V}_2^{L_2}, \quad (2.12)$$

where $M_i := \dim \mathcal{V}_i^{L_i}$, here $M_i = 2^{L_i} + 1$. The basis functions $a_{l_1}, b_{l_2}$ are piecewise affine functions with $C^0$-continuity across element boundaries on their respective meshes $\mathcal{T}_i^{L_i}$. Their space is denoted by $S^{0,1}(D_i, \mathcal{T}_i^{L_i}), i = 1,2$.

In general, we will assume the finite dimensional function spaces $\mathcal{V}_i^{L_i}$ to form a nested sequence with respect to the resolution level parameter $L_i$,

$$\mathcal{V}_i^{0} \subset \mathcal{V}_i^{1} \subset \ldots \subset \mathcal{V}_i^{L_i}, \quad (2.13)$$

which is also the case in Zenger’s example.

The complexity, or number of degrees of freedom, of this full tensor representation is $M = M_1 \cdot M_2$, in the example $M = (2^{L_1} + 1)^{d_1}(2^{L_2} + 1)^{d_2} = O(2^{d_1L_1+d_2L_2})$, where we included the dimensions $d_i$ of the domains $D_i$ for generality, here $d_i = 1$.

Instead of the standard nodal basis functions, one could also choose a hierarchic basis so that the approximation function spaces additionally satisfy a hierarchy relation

$$\mathcal{V}_i^{L_i} = \mathcal{V}_i^{L_i-1} \oplus \mathcal{W}_i^{L_i}, \quad L_i = 1, \ldots, L_i, \quad i = 1,2. \quad (2.14)$$

The basis functions in a so-called detail space $\mathcal{W}_i^{L_i}$ can then be associated with the resolution level $L_i$, $\mathcal{V}_i^{L_i}$ now contains basis functions of all resolution levels up to $L_i$.

The full tensor product approximation space $\mathcal{V}^{L_1,L_2}$ consists of the sum of all tensor product detail spaces:

$$\mathcal{V}^{L_1,L_2} = \bigoplus_{l_1=0}^{L_1} \bigoplus_{l_2=0}^{L_2} \mathcal{W}_1^{l_1} \otimes \mathcal{W}_2^{l_2} \quad (2.15)$$

Therefore, the full tensor approximation can be expressed as a sum of contributions from the detail spaces:

$$u_{L_1,L_2} = \sum_{l_1=0}^{\infty} \sum_{l_2=0}^{\infty} w_{l_1,l_2}, \quad w_{l_1,l_2} \in \mathcal{W}_1^{l_1} \otimes \mathcal{W}_2^{l_2}. \quad (2.16)$$

Requiring $u \in C^4(\Omega)$ we can show for the example considered here that these contributions decay for increasing resolution levels (Zenger, 1991, Eq. (2.5)):

$$\|w_{l_1,l_2}\|_{\infty} \leq 4^{-l_1-l_2-1} \left\| \frac{\partial^4 u}{\partial x_1^4 \partial x_2^2} \right\|_{\infty}. \quad (2.17)$$
Preliminaries

The norm $\| \cdot \|_\infty$ represents the maximum norm for continuous functions, $\| u \|_\infty := \max_{x \in \Omega} |u(x)|$. Using this decay estimate, the error of uniform interpolation with the same resolution in both component domains can be bounded by

$$
\| u - u_{L,L} \|_\infty \leq c 2^{-2L} \left\| \frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} \right\|_\infty,
$$

(2.18)

where $c$ is a generic constant that may depend on the domain $\Omega$ and its dimension, but is independent of $L$. At the same time, the complexity of the tensor product detail spaces is

$$
| W_{l_1}^{l_1} \otimes W_{l_2}^{l_2} | = | W_{l_1}^{l_1} | \cdot | W_{l_2}^{l_2} | = 2^{l_1-1}2^{l_2-1} = 2^{l_1+l_2-2}.
$$

When comparing the benefit ratios $\| u_{l_1,l_2} \|_\infty / | W_{l_1}^{l_1} \otimes W_{l_2}^{l_2} |$ of the tensor product detail spaces, we find that especially those spaces in which both $l_1$ and $l_2$ are large contribute relatively little to the solution while containing large numbers of degrees of freedom. This observation leads to the idea of including only contributions from those detail spaces into the approximation for which the benefit ratio is above a certain tolerance. Transferred to the sum of contributions this means we construct a sparse tensor product approximation space by

$$
\hat{V}_{L_1,L_2} := \bigoplus_{0 \leq f(l_1,l_2) \leq L_1} W_{l_1}^{l_1} \otimes W_{l_2}^{l_2},
$$

(2.19)

and correspondingly a sparse tensor approximation

$$
\hat{u}_{L_1,L_2} := \sum_{0 \leq f(l_1,l_2) \leq L_1} w_{l_1,l_2},
$$

(2.20)

where the sparsity profile $f : [0, L_1] \times [0, L_2] \to \mathbb{R}$, which implicitly depends on $L_1$ and $L_2$, determines which of the tensor product detail spaces are to be included in the approximation.

In Zenger’s example where the approximation spaces over the component domains are identical so that component resolutions are coupled as $L_1 = L_2 = L$ and the benefit ratio is required to be greater than a fixed tolerance, the resulting optimal sparsity profile is simply $f(l_1, l_2) = l_1 + l_2$ (see Bungartz and Griebel, 2004, Sec. 3.2, for detailed derivation).

For such a sparsity profile the error of the sparse tensor interpolant can be estimated as

$$
\| u - \hat{u}_{L,L} \| \leq cL2^{-2L} \left\| \frac{\partial^4 u}{\partial x_1^2 \partial x_2^2} \right\|_\infty,
$$

(2.21)

in which the convergence rate is worse than in Eq. (2.18) only by a factor logarithmic in the degrees of freedom of a component domain.

2.2.2 Complexity estimates

With the simple linear sparsity profile $f(l_1, l_2) = l_1 + l_2$, the complexity of the approximation reduces to $\hat{M} = (L - 1)2^L + 1$, basically the number of degrees of freedom $M_i$ of one component domain up to a logarithmic factor, since $L = \log_2(M_i - 1)$. 

12
For a more general setting in which the resolutions can differ between the component domains, the next lemmata result in a complexity estimate, making use of the following definition.

**Definition 2.1.** The relation \( a_l \lesssim b_l \) with quantities \( a \) and \( b \) depending on a scaling parameter or resolution level \( l \) occurring on both sides of the relation is defined as

\[
\exists c > 0 \text{ s. t. } a_l \leq cb_l, \text{ with } c \text{ independent of } l.
\]

We first prove an intermediate result which will be used again later on.

**Lemma 2.2.** Let \( b_1 > 1 \), \( L_i \geq 0 \), \( i = 1, 2 \), and \( b_1^{L_1} \leq b_2^{L_2} \), then it holds

\[
\sum_{l=0}^{L_1} b_1^l b_2^{L_2 - L_1 l / L_1} \lesssim L_1^0 \max \{ b_1^{L_1}, b_2^{L_2} \}
\]

where \( \theta = 1 \) if \( b_1^{L_1} = b_2^{L_2} \) and zero otherwise.

**Proof.** We have

\[
\sum_{l=0}^{L_1} b_1^l b_2^{L_2 - L_1 l / L_1} = b_2^{L_2} \sum_{l=0}^{L_1} \left( \frac{b_1}{b_2^{L_2 / L_1}} \right)^l.
\]

Two cases can now be distinguished:

1. \( b_1 = b_2^{L_2 / L_1} \), or equivalently \( b_1^{L_1} = b_2^{L_2} \). Then

\[
b_2^{L_2} \sum_{l=0}^{L_1} \left( \frac{b_1}{b_2^{L_2 / L_1}} \right)^l = (L_1 + 1) b_2^{L_2} = (L_1 + 1) b_1^{L_1}.
\]

2. \( b_1 < b_2^{L_2 / L_1} \), then consequently \( L_2 / L_1 > \log b_1 / \log b_2 \) as \( L_1 \) and \( L_2 \) increase, and thus

\[
b_2^{L_2} \sum_{l=0}^{L_1} \left( \frac{b_1}{b_2^{L_2 / L_1}} \right)^l = b_2^{L_2} \left[ \frac{b_1^{L_2 / L_1}}{b_2^{L_2 / L_1} - b_1} - \frac{b_1^{L_1+1} b_2^{L_2 - L_2 / L_1}}{b_2^{L_2 / L_1} - b_1} \right] \leq c b_2^{L_2},
\]

with a generic constant \( c > 0 \) independent of \( L_i \). The last step follows because the first fraction inside the brackets is in \( O(1) \) for large \( L_i \), and the second fraction can be dropped due to its sign.

These cases can be combined into the statement of the lemma. \( \square \)

With the aid of the previous lemma we quickly arrive at an estimate for the complexity of the sparse tensor product space.

**Lemma 2.3.** Assuming the dimensions of the detail spaces \( W_i^{l_i} \) scale as \( \dim(W_i^{l_i}) \leq c_i b_i^{l_i} \) with constants \( c_i > 0 \) and bases \( b_i > 1 \), \( i = 1, 2 \), and a sparsity profile \( f(l_1, l_2) = l_1 + l_1^{l_2} l_2 \), the
total complexity of the sparse tensor approximation on the sparse tensor product of component approximation spaces \(V^l_i\) with \(\dim(V^l_i) = M_i\) is

\[
\hat{M}_{L_1,L_2} \lesssim L_1^\theta \max\{b_1^{L_1}, b_2^{L_2}\} \lesssim (\log M_1)^\theta \max\{M_1, M_2\},
\]

where \(\theta = 1\) if \(b_1^{L_1} = b_2^{L_2}\) and \(\theta = 0\) otherwise.

**Proof.** Griebel and Harbrecht (2013a, Thm. 4.1) estimate the complexity of sparse tensor product spaces with very similar parametrization. Here, we simply sum up the contributions from the tensor product detail spaces:

\[
\hat{M}_{L_1,L_2} \leq c_1 c_2 \sum_{0 \leq l_1 + l_2 \leq L_1} b_1^{l_1} b_2^{l_2} = c_1 c_2 \sum_{l_1 = 0}^{l_1^{\text{max}}(L_1)} \sum_{l_2 = 0}^{b_1^{L_2} - l_2^{L_1}} b_1^{l_1} b_2^{l_2} = c_1 c_2 \sum_{l_1 = 0}^{L_1} b_1^{l_1} \sum_{l_2 = 0}^{L_2 - l_2^{L_1}} b_2^{l_2},
\]

with \(l_2^{\text{max}}(L_1)\) being the maximum feasible \(l_2\) index value resulting from solving \(f(l_1, l_2) \leq L_1\) w.r.t. \(l_2\). The generic constant \(c\) hides factors independent of \(L_1\) or \(L_2\) from now on:

\[
\hat{M}_{L_1,L_2} \leq c \sum_{l_1 = 0}^{L_1} b_1^{l_1} \left( b_2^{L_2 - L_1 l_2^L_1 / L_2} - 1 \right) = c b_2 \sum_{l_1 = 0}^{L_1} b_1^{l_1} b_2^{L_2 - L_1 l_2^L_1 / L_2} - c \sum_{l_1 = 0}^{L_1} b_1^{l_1}.
\]

We drop the second sum as it would only reduce the complexity due to its negative sign. By applying Lemma 2.2 (\(b_1^{L_1} \leq b_2^{L_2}\) can be achieved by exchanging the indices if necessary) to the first sum we already obtain the first estimate of Eq. (2.22). The second follows by recalling that the dimensions of the component approximation spaces satisfy

\[
b_1^{L_1} \leq \sum_{l_1 = 0}^{L_1} b_1^{l_1} = M_1. \tag*{\qed}
\]

Approximation on sparse tensor product spaces therefore offers a significant complexity reduction while essentially maintaining the convergence rate of full tensor approximation as long as the function to be approximated satisfies the necessary regularity requirements.

### 2.3 Sparse tensor combination technique

The sparse grid combination technique on a tensor product mesh with mesh width \(h\) as developed by Griebel et al. (1992) for function approximation combines the approximations obtained on \(O(\log(h^{-1}))\) different full grids to generate an approximation equivalent to the one on a sparse grid. The theoretical advantages of sparse grid approximations can also be harnessed for the combination technique: As seen in the previous section, for sufficiently smooth solutions, the number of degrees of freedom reduces from \(O(h^{-2})\) to \(O(h^{-1} \log(h^{-1}))\) in 2D, which is essentially the complexity of
a one-dimensional problem. At the same time, the accuracy deteriorates only by a logarithmic factor from $O(h^2)$ to $O(h^2 \log(h^{-1}))$.

The practical advantages of the combination technique over a direct sparse grid approach from the point of view of implementation are twofold: First, only full grid problems have to be solved for which standard full grid solvers can be applied. No programming of direct sparse grid solution approaches is required. Second, as all the single full grid problems are independent, the technique apparently lends itself well to parallelization: one assigns each subproblem to a computing node. During the solution procedure for the full subproblems no communication between the nodes is necessary, only in the end the solutions have to be collected and combined.

In the next section, we present this technique with regard to tensor products of two component spaces, following Hegland et al. (2007).

2.3 Sparse tensor combination technique

To begin with this exposition, we set out from the same setup as in the beginning of the sparse grid section so that $u, \Omega, D_i, \mathcal{V}, \mathcal{V}_i, V_i^{l_i}, i = 1, 2$, retain their meaning. The component domains may again stand for general subsets $D_i \subset \mathbb{R}^{d_i}$. Our component approximation spaces $V_i^{l_i}$ form nested sequences as in (2.13).

Additionally, orthogonal projectors $P_i^{l_i} : \mathcal{V}_i \rightarrow V_i^{l_i}, i = 1, 2$, provide mappings to the component approximation spaces, and the projector to a tensor product approximation space $V_1^{l_1} \otimes V_2^{l_2}$ is accordingly defined for a function $u \in \mathcal{V}$ by $P^{l_1,l_2} : \mathcal{V} \rightarrow V_1^{l_1} \otimes V_2^{l_2}$. The expression $P^{l_1,l_2}u$ represents a full tensor approximation to $u$ in $V_1^{l_1} \otimes V_2^{l_2}$.

Suppose we would like to approximate $u$ on a space smaller than the full approximation space $V_1^{l_1} \otimes V_2^{l_2}$ in order to save degrees of freedom, but without sacrificing too much accuracy. In the previous section on sparse grids, we showed that the sparse tensor approximation $\hat{u}_{l_1,l_2}$ achieves this, but requires that the approximation spaces are hierarchic, i.e. a direct sum of detail spaces $\mathcal{W}_i^{l_i}$ (cf. Eq. (2.14)). However, this decomposition of function spaces might not exist under all circumstances, or would require the use of a specific basis entailing undesired disadvantages.

Then we could formally define an approximation $\tilde{u}$ on a space $(V_1^{l_1} \otimes V_2^{l_2})$ with $l_1 < L_1$ and $l_2 < L_2$ by

$$\tilde{u} := (P^{l_1,l_2} + P^{l_1,l_2} - P_{(V_1^{l_1} \otimes V_2^{l_2}) \cap (V_1^{l_1} \otimes V_2^{l_2})}u,$$

in which we have to subtract once the contributions from the intersection of the two full tensor subspaces of smaller size according to the inclusion/exclusion principle.

If a decomposition $V_i^{l_i} = V_i^{l_i} \oplus \mathcal{W}_i^{l_i}$ for $i = 1, 2$ was available, we would obtain

$$(V_1^{l_1} \otimes V_2^{l_2}) \cap (V_1^{l_1} \otimes V_2^{l_2}) = ((V_1^{l_1} \oplus \mathcal{W}_1^{l_1}) \otimes V_2^{l_2}) \cap (V_1^{l_1} \otimes (V_2^{l_2} \oplus \mathcal{W}_2^{l_2}))$$

$$= (V_1^{l_1} \otimes V_2^{l_2}) \cap (V_1^{l_1} \otimes V_2^{l_2}) \cap (V_1^{l_1} \otimes V_2^{l_2} \oplus V_1^{l_1} \otimes \mathcal{W}_2^{l_2})$$

$$= V_1^{l_1} \otimes V_2^{l_2}.$$
Hence we see that the approximation \( \tilde{u} := (P_{L^1} + P_{L^2} - P_{L^1} - P_{L^2}) u \) (2.23)

coincides with \( \bar{u} \) if the approximation spaces \( V_{L^i} \) are nested with a decomposition into detail spaces. In any other case, \( \tilde{u} \) is a different approximation, but one which can be computed in a straightforward way (in contrast to \( \bar{u} \)) as a combination of three full tensor approximations of smaller resolution levels.

Now let us suppose we seek an approximation to \( u \) from a more generally defined sparse approximation space

\[
\hat{V}_{L^1, L^2} = \bigcup_{0 \leq f(l_1, l_2) \leq L_1} (V_{1}^{l_1} \otimes V_{2}^{l_2})
\]

with a sparsity profile \( f \) as before in Eq. (2.19) so that \( \hat{V}_{L^1, L^2} \) has a smaller dimension than the full tensor product space \( V_{L^1, L^2} = V_{1}^{L^1} \otimes V_{2}^{L^2} \).

Generalizing Eq. (2.23) a sparse approximation can be defined as the linear combination of smaller full tensor product approximations according to this combination formula:

\[
\hat{u}_{L^1, L^2} := \sum_{l_1=0}^{L_1} P_{l_1}^{l_2 \max} (l_1) u - \sum_{l_1=0}^{L_1-1} P_{l_1}^{l_2 \max} (l_1+1) u.
\] (2.24)

Here, the sparse maximum \( l_2 \) level \( l_2^{\max} : [0, L_1] \to \{0, 1, \ldots, L_2\} \) is obtained by solving \( f(l_1, l_2) = L_1 \) for \( l_2 \) and rounding the result down to the nearest integer. Instead of defining the approximation directly via its details as in the sparse tensor case in Eq. (2.20), we combine \( L_1 + 1 \) full tensor approximations from full tensor subspaces of smaller refinement and add the approximations. According to the inclusion/exclusion principle, we then still have to subtract \( L_1 \) full tensor approximations of smaller dimension than the positively weighted ones to arrive at the sparse tensor combination technique approximation \( \hat{u}_{L^1, L^2} \).

In cases where the approximation spaces can be decomposed into detail spaces with the property (2.14), the question arises whether the direct sparse tensor approximation \( \hat{u}_{L^1, L^2} \) in \( \hat{V}_{L^1, L^2} \) to \( u \) from Eq. (2.20) is actually identical to the sparse tensor combination technique approximation \( \hat{u}_{L^1, L^2} \).

In general, it turns out that the combination approximation \( \hat{u}_{L^1, L^2} \) is not the same as the direct sparse approximation \( \hat{u}_{L^1, L^2} \), it is only an approximation to \( \hat{u}_{L^1, L^2} \). However, it has the same asymptotic complexity and is equivalent in its convergence towards the original function \( u \) (Griebel et al. (1992, Sec. 3) and Griebel and Harbrecht (2013b)).

By the following lemma we show that the combination technique approximation as given by the combination formula (2.24) can always be restated as a sum of differences of projections of consecutive resolution level (also compare Lemma 7 by Garcke (2013) for equivalence of sparse grid and combination technique interpolant). In practice, the projectors \( P_{l_1}^{l_2} \) in the lemma could denote \( L^2 \)-projection, Galerkin projection, or also interpolation.
Lemma 2.4 (Equivalence of combination formula and sum of differences). For the combination technique approximation \( \tilde{u}_{L_1,L_2} \) given by the combination formula (2.24) it holds
\[
\tilde{u}_{L_1,L_2} = \sum_{0 \leq f(l_1,l_2) \leq L_1} \Delta^{l_1,l_2} u,
\]
where \( \Delta^{l_1,l_2} := p^{l_1,l_2} - p^{l_1-1,l_2} - p^{l_1,l_2-1} + p^{l_1-1,l_2-1} \) with the convention \( P^{-1} = 0 = P^{-1'} \).

Proof. The full tensor subspace approximations in the combination formula (2.24) can be represented by the sum of their details as
\[
P^{l_1,l_2} u = \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \Delta^{l_1,l_2} u.
\]
Inserting into the combination formula, we obtain
\[
\tilde{u}_{L_1,L_2} = \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \Delta^{l_1,l_2} u - \sum_{l_1=0}^{L_1-1} \sum_{l_2=0}^{L_2} \Delta^{l_1+1,l_2} u - \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \Delta^{l_1,l_2} u
\]
\[
+ \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \Delta^{l_1,l_2} u
\]
\[
= \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{L_2} \left( \sum_{i=0}^{\max(l_1)} \Delta^{l_1,l_2} u - \sum_{i=0}^{\max(l_1)+1} \Delta^{l_1,l_2} u \right) + \sum_{i=0}^{\max(L_1+1)} \sum_{l_2=0}^{L_2} \Delta^{l_1,l_2} u.
\]
With the convention \( \max(L_1+1) = -1 \), the last term always evaluates to zero. We obtain further
\[
\tilde{u}_{L_1,L_2} = \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{\max(l_1)} \sum_{i=0}^{\max(l_1)+1} \Delta^{l_1,l_2} u.
\]
This sum is equivalent to (which can be seen best with help of Fig. 2.1)
\[
\tilde{u}_{L_1,L_2} = \sum_{l_1=0}^{L_1} \sum_{l_2=0}^{\max(l_1)} \Delta^{l_1,l_2} u.
\]
\( \square \)

Note that for the reformulation asserted by Lemma 2.4 a tensor product representation of the projector \( P^{l_1,l_2} = P^{l_1}_1 \otimes P^{l_2}_2 \) is not required, the reformulation merely relies on the rearrangement of terms in the telescopic sum.
As a further development, the full tensor subspaces can be selected adaptively based on an a-posteriori error indicator. This idea has been investigated by Hegland (2003) and Garcke (2007), for instance.

Furthermore, in the latter reference, the author also introduces combination coefficients chosen as to minimize a certain difference between the combination approximation and measurements in the framework of parameter estimation. Nowadays, this method is often referred to as the “opticom” method and allows to find the combination approximation optimally close to the direct sparse approximation with respect to a suitable minimization functional.

Here, we restrict ourselves to the classical combination technique with combination coefficients in \{-1, 1\} according to the inclusion/exclusion principle.

### 2.3.2 Complexity estimates

As before in the section about sparse grids, we assume a scaling of the component complexities of \( M_i = \dim(V_i^{L_i}) \leq c_i b_i^{L_i}, i = 1, 2 \), which leads to the dimension of the full tensor approximation space

\[
M_{L_1, L_2} \lesssim b_1^{L_1} b_2^{L_2} \lesssim M_1 M_2. \tag{2.26}
\]

Each of the full subspaces \( V_1^{L_1} \otimes V_2^{L_2} \) of reduced resolution, from which we seek the subspace approximations in the combination technique, thus has a complexity of \( M_{l_1, l_2} \leq c_1 c_2 b_1^{l_1} b_2^{l_2} \). The overall complexity \( \tilde{M}_{l_1, l_2} \) of the sparse tensor combination technique approximation for the case of a linear sparsity profile \( f(l_1, l_2) = l_1 + L_1 l_2 / L_2 \)
2.3 Sparse tensor combination technique

is given in the following lemma, which extends Lemma 4.2.1 by Bieri (2009) to the case of resolution levels differing between the two component domains.

**Lemma 2.5.** Given the dimension of the component spaces \( M_i := \dim(V_i^{l_i}) \leq c_i b_i^{l_i} \), \( c_i > 0 \), \( b_i > 1 \), \( L_i > 1 \), \( i = 1, 2 \), and \( l_2^{\text{max}} = L_2 - L_2 l_1 / L_1 \), the total complexity of the combination technique approximation of a function is

\[
\tilde{M}_{L_1,L_2} \lesssim L_1^\theta \max\{b_1^{l_1}, b_2^{l_2}\} \lesssim (\log M_1)^\theta \max\{M_1, M_2\},
\]

where \( \theta = 1 \) if \( b_1^{l_1} = b_2^{l_2} \) and zero otherwise.

**Proof.** The sum of subproblem complexities in the combination technique is according to the combination formula (2.24)

\[
\tilde{M}_{L_1,L_2} = \sum_{l_1=0}^{L_1} M_{l_1} r_2^{\max}(l_1) + \sum_{l_1=0}^{L_1-1} M_{l_1} r_2^{\max}(l_1+1) \lesssim \sum_{l_1=0}^{L_1} b_1^{l_1} b_2^{l_2 \max(l_1)}.
\]

Since the negatively weighted approximations (the second sum) always contribute fewer degrees of freedom than the positive ones, they have been estimated by a constant multiplied by the positive approximations. To the last expression, we can directly apply Lemma 2.2 and arrive at the assertion.

We conclude from a comparison of Lemmata 2.3 and 2.5 that the combination technique realizes the same reductions in degrees of freedom as the direct sparse tensor approximation, at least asymptotically, and that both approximations are identical in the tensor product setting with nested sequences of component subspaces as given by relation (2.13).
3 The phase space Galerkin method

We focus again on our problem of interest, the radiative transfer problem, and derive a variational framework based on a phase space Galerkin method. Before we begin with this, however, we introduce an operator formulation which allows for more compact notation in the next section and discuss properties of the scattering operator which will be relevant in the upcoming chapters.

3.1 Operator formulation

Problem (1.1) reads in operator form: Find the intensity $u(x, s) : D \times S \rightarrow \mathbb{R}$ such that

$$Au = f, \quad u|_{\partial \Omega_-} = g.$$  

(3.1)

In this, $\partial \Omega_-$ denotes the inflow part of the boundary $\partial \Omega = \partial D \times S$ of the computational domain or phase space $\Omega = D \times S$. The inflow boundary is defined by

$$\partial \Omega_- := \{(x, s) \in \Omega : x \in \Gamma_-(s)\}.$$  

(3.2)

with the physical part of the inflow boundary

$$\Gamma_-(s) := \{x \in \partial D : s \cdot n(x) < 0\}.$$  

(3.3)

Correspondingly we define the physical part of the outflow boundary as

$$\Gamma_+(s) := \{x \in \partial D : s \cdot n(x) > 0\}.$$  

(3.4)

The radiative transfer operator $A = T + Q$ consists of the transport operator $T$,

$$Tu := (s \cdot \nabla_x + \kappa)u,$$  

(3.5)

and the scattering operator $Q$,

$$Qu := \sigma Q_1 u := \sigma(\text{Id} - \Sigma)u := \sigma(x)u(x, s) - \sigma(x) \int_S \Phi(s, s')u(x, s') \, ds'.$$  

(3.6)

Here, $Q_1 = \text{Id} - \Sigma$ is the unity scattering operator, and $\Sigma$ is the scattering integral operator, the integral of $\Phi$ and $u$. The source function $f$ contains the sources of radiation in the domain,

$$f := \kappa I_b,$$  

(3.7)

and $g$ is the incoming radiation on the boundary $\partial \Omega_-$, as in Sec. 1.1.
3.2 The scattering operator

3.2.1 Properties from physical considerations

The infinitesimal quantity $\Phi(s, s') ds'$ can be interpreted as the probability of scattering of radiation coming from the area element $ds'$ into the direction $s$. In order for this interpretation to be meaningful and also to make sense physically, we require the phase function to be nonnegative, i.e. $\Phi(s, s') \geq 0$ for all $s, s' \in S$.

The interpretation as probability distribution already explains the postulation for normalization of the phase function:

$$\int_S \Phi(s, s') \, ds' = 1 \quad \forall s \in S.$$ 

This normalization, however, can also be interpreted physically as conservation of the number of photons during each scattering event: The radiative intensity at a point in space is only redistributed in angle by scattering, not increased or decreased. As each photon carries a specific amount of energy this normalization is sometimes also referred to as conservation of scattered energy.

We further assume an isotropic medium so that the scattering phase function depends only on the incoming and outgoing directions $s', s$ of the radiation, but not on the spatial position $x$.

Often the scattering particles of the medium can be well approximated by spheres. Then the scattering phase function does not vary with the azimuthal angle, it only depends on the inner product of $s$ and $s'$. From this it also follows that the scattering phase function is symmetric in $s$ and $s'$.

3.2.2 Models for the scattering phase function

In radiative transfer, common scattering phase functions $\Phi$ follow the previous physical considerations. Typically they are also smooth in angle. However, phase functions from Mie scattering theory in particular can oscillate strongly or possess large peaks in the forward or backward direction. These three models are frequently found in radiative transfer applications:

1. The simplest model, isotropic scattering, takes the phase function to be constant with respect to the angle. With normalization we obtain

$$\Phi(s, s') = \frac{1}{|S|},$$

with $|S|$ the measure of the $d_S$-sphere.

2. If the scattering particles are spherical and small compared to the wavelength of the radiation, the scattering is of Rayleigh type. The corresponding phase function is the dipole phase function (Modest, 2003, Chap. 11)

$$\Phi(s, s') = c(1 + (s \cdot s')^2), \begin{cases} c = \frac{1}{5\pi} & \text{for } d_S = 1, \\ c = \frac{3}{16\pi} & \text{for } d_S = 2, \end{cases}$$

(3.9)
3.2 The scattering operator

with $c$ computed from normalization.

3. Another common model is the Henyey-Greenstein phase function. It is often used as an approximation to more complicated phase functions from Mie scattering theory with a strong forward scattering peak (Modest, 2003, p. 385). The normalized phase function reads

$$\Phi(s, s') = \begin{cases} 
\frac{1 - \gamma^2}{2\pi} \left( \frac{1 + \gamma^2 - 2\gamma s \cdot s'}{1 - \gamma^2} \right) & \text{for } d_S = 1, \\
\frac{1}{4\pi} \left( \frac{1 - \gamma^2 - 2\gamma s \cdot s'}{1 + \gamma^2 - 2\gamma s \cdot s'} \right)^{3/2} & \text{for } d_S = 2.
\end{cases} \quad (3.10)$$

The asymmetry factor $-1 < \gamma < 1$ determines if the scattering is strongly backward peaked ($\gamma \to -1$), isotropic ($\gamma = 0$), or forward peaked ($\gamma \to 1$). In applications, it is usually tuned to data from experiments or taken from the literature.

3.2.3 Mathematical assumptions and properties

We call scattering forward dominant (cf. Kanschat, 2008, Def. 1) if $\Phi(s, s') = \sum_{k=0}^{\infty} a_k \cos(k \arccos(s \cdot s'))$ with all $a_k \geq 0$, which is the case for isotropic, Rayleigh, and also for Henyey-Greenstein scattering as long as the asymmetry factor $\gamma \geq 0$.

A number of useful properties of the scattering operator can be derived from the following theorem, which we have adapted to our setting.

**Theorem 3.1** (Hilbert-Schmidt theorem for integral operators (e.g. Knapp, 2005, Thm. 2.4)). Let $S$ be the angular domain and let $\Phi(\cdot, \cdot) \in L^2(S \times S)$ be symmetric in its arguments, i.e. $\Phi(s, s') = \Phi(s', s)$ for all $s, s'$ in $S$. Then the linear operator $\Sigma$ defined by

$$\Sigma v := \int_S \Phi(s, s') v(s') \, ds',$$

$v \in L^2(S)$,

is a self-adjoint compact operator on the Hilbert space $L^2(S)$, and it holds:

1. $\|\Sigma\|_{L^2(S) \to L^2(S)} \leq \|\Phi\|_{L^2(S \times S)} = \left( \int_S \int_S |\Phi(s, s')|^2 \, ds \, ds' \right)^{1/2}$.

2. Let $\lambda_n$ be the eigenvalues of $\Sigma$ so that $\Sigma v_n = \lambda_n v_n$. Then there are only countably many eigenvalues $\lambda_n$, and they are all real. For any $\epsilon > 0$, there are only finitely many eigenvalues with $|\lambda_n| \geq \epsilon$. The largest value of $|\lambda_n|$ is $\|\Sigma\|_{L^2(S) \to L^2(S)}$.

3. The eigenspaces $V^{(\Sigma)}_{\lambda_n} = \{v \in L^2(S) : \Sigma v = \lambda_n v\}$ are mutually orthogonal with respect to the $L^2(S)$ inner product, and the vector subspace orthogonal to all $V^{(\Sigma)}_{\lambda_n}$ is the kernel of $\Sigma$. If $v_1, v_2, \ldots$ is an enumeration of the union of orthonormal bases of the spaces $V^{(\Sigma)}_{\lambda_n}$, then for any $w \in L^2(S)$ there is an expansion $\Sigma w = \sum_{n=1}^{\infty} (\Sigma w, v_n)_{L^2(S)} v_n$ with the series converging in $L^2(S)$. 

23
Under the assumption that scattering by $\Phi$ is forward dominant, Kanschat (2008, Lemmata 2 and 3) proves that $\Sigma$ is positive semi-definite, i.e.

$$ (v, \Sigma v)_{L^2(S)} \geq 0 \quad \forall v \in L^2(S). $$

(3.11)

With symmetry and normalization of $\Phi$, he shows that $\|\Sigma\|_{L^2(S) \to L^2(S)} = 1$ (Kanschat, 2008, Lemma 5).

From these facts and Thm. 3.1 we can infer that the spectrum of $\Sigma$ is restricted to $[0, 1]$ with the smaller eigenvalues bounded away from the largest eigenvalue of 1.

Then, $Q_1 = \text{Id} - \Sigma$ can be considered as a perturbation of the identity operator with eigenvalues in $[0, 1]$ with an isolated eigenvalue $\lambda_0 = 0$, from which the next largest eigenvalue $\lambda_1$ differs by a positive constant. This is due to the fact that the eigenvalues $\lambda_n$ of $Q_1$ and $\lambda_n^{(\Sigma)}$ of $\Sigma$ are related by

$$ Q_1 v_n = (\text{Id} - \Sigma)v_n = (1 - \lambda_n^{(\Sigma)})v_n = \lambda_n v_n. $$

Consequently, the eigenspaces $V_{\lambda_n}$ of $Q_1$ are related to those of $\Sigma$ by $V_{\lambda_n} = V_{(1 - \lambda_n)^{\Sigma}}$.

Specifically, it holds $V_1 = \ker \Sigma$ and $\ker Q_1 = V_1^{(\Sigma)}$. Since the eigenspaces of $\Sigma$ and its kernel are all mutually orthogonal (Thm. 3.1 point 3), we also have a splitting of $L^2(S)$ into the eigenspaces $V_{\lambda_n}$ of $Q_1$ and its kernel: $L^2(S) = \bigcup_{n=1}^{\infty} (V_{\lambda_n}) \oplus \ker Q_1$ with $\ker Q_1 \perp \bigcup_{n=1}^{\infty} (V_{\lambda_n})$. We can therefore represent any $w \in L^2(S)$ as

$$ w = \sum_{n=1}^{\infty} (w, v_n)_{L^2(S)} v_n + w_0 $$

(3.12)

with a $w_0 \in \ker Q_1$. Here, the $v_1, v_2, \ldots$ now constitute a basis of eigenfunctions of $\bigcup_{n=1}^{\infty} V_{\lambda_n}$, which are mutually orthogonal and orthogonal to $w_0$.

We are going to denote the union of the $Q_1$ eigenspaces by $(\ker Q_1)_{\perp} := \bigcup_{n=1}^{\infty} (V_{\lambda_n})$, and the $L^2(S)$ projector onto $(\ker Q_1)_{\perp}$ by $P_{\perp}$.

The previous considerations and definitions then lead to the following properties of $Q_1$:

**Lemma 3.2.** For any $u \in L^2(\Omega)$, the scattering operator $Q$ as defined by Eq. (3.6) satisfies (cf. Ávila et al., 2011, Eq. (11))

$$ \lambda_1 \|\sigma P_{\perp} u\|_{L^2(\Omega)} \leq \|Qu\|_{L^2(\Omega)} \leq \|\sigma\|_{L^\infty(\Omega)} \|u\|_{L^2(\Omega)}, $$

(3.13)

$$ (u, Qu)_{L^2(\Omega)} \geq \|Qu\|_{L^2(\Omega)}^2 \geq 0, $$

(3.14)

in which the projector $P_{\perp}$ maps $u(x, \cdot)$ to $(\ker Q)^{\perp}$, the space orthogonal to the kernel of $Q$, and $\lambda_1 \in (0, 1]$ is the smallest nonzero eigenvalue of $Q_1$.

**Proof.** The relations are a simple consequence of the decomposition (3.12) of any function into eigenfunctions and a kernel element of $Q_1$. As an example, we show the second property (3.14).
In our model of the RTE, we assume $\sigma$ to be a function of $x$ only, therefore we can set $\tilde{u} = \sigma^{1/2}u$, and it suffices to show $(\tilde{u}, Q_1 \tilde{u}) \geq \|Q_1 \tilde{u}\|^2$. However, we will simply write $u$ instead of $\tilde{u}$ here.

We apply expansion (3.12) to $u \in L^2(\Omega)$ yielding

$$u(x, s) = \sum_{n=1} (u(x, s), v_n(s))_{L^2(S)} v_n(s) + u_0(x, s) = \sum_{n=1} \alpha_n(x) v_n(s) + u_0(x, s)$$

with coefficient functions $\alpha_n(x) \in L^2(D)$. Now $v_n$ are eigenfunctions of $Q_1$ and $u_0(x, \cdot)$ is in the kernel of $Q_1$, thus we obtain

$$(u, Q_1 u)_{L^2(\Omega)} = \left( \sum_{n=1} \alpha_n(x) v_n(s) + u_0(x, s), \sum_{n=1} \lambda_n \alpha_n(x) v_n(s) \right)_{L^2(\Omega)}$$

$$= \left( \sum_{n=1} \alpha_n(x) v_n(s), \sum_{n=1} \lambda_n \alpha_n(x) v_n(s) \right)_{L^2(\Omega)}$$

because $u_0(x, \cdot)$ is $L^2(S)$-orthogonal to all $v_n$. As we have $\lambda_n \leq 1$ and mutual $L^2(S)$-orthogonality of all $v_n$, we get further

$$(u, Q_1 u)_{L^2(\Omega)} = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \lambda_n (\alpha_m(x) v_m(s), \alpha_n(x) v_n(s))_{L^2(\Omega)}$$

$$= \sum_{n=1}^{\infty} \lambda_n (\alpha_n(x) v_n(s), \alpha_n(x) v_n(s))_{L^2(\Omega)}$$

$$\geq \sum_{n=1}^{\infty} \lambda_n^2 (\alpha_n(x) v_n(s), \alpha_n(x) v_n(s))_{L^2(\Omega)}$$

$$= \left( \sum_{n=1}^{\infty} \lambda_n \alpha_n(x) v_n(s), \sum_{n=1}^{\infty} \lambda_n \alpha_n(x) v_n(s) \right)_{L^2(\Omega)}$$

$$= \|Q_1 u\|_{L^2(\Omega)}^2 \geq 0. \quad \square$$

### 3.3 Variational formulation

In this section we derive the variational framework for the radiative transfer problem which will later be used in the analysis of our methods. We base the framework on a phase space Galerkin FEM with stabilization applied to the operator RTP (3.1). For the purpose of the analysis, we restrict ourselves to zero inflow boundary conditions here, that is, $g = 0$ in Eq. (3.1).

#### 3.3.1 A generic stabilized phase space variational formulation

Following the standard Galerkin approach we would multiply problem (3.1) from the left by a test function $v : \Omega \to \mathbb{R}$ from a suitable test space $V_{\text{test}}$ over the phase space $\Omega$
The phase space Galerkin method

and demand that the resulting equations be satisfied for all such test functions from the test space. This variational formulation of the standard Galerkin FEM would then read: Find \( u \in \mathcal{V}_0 \), where \( \mathcal{V}_0 \) is the trial space over the phase space \( \Omega \) of all admissible solutions, such that

\[
(v, (T + Q)u)_{L^2(\Omega)} = (v, f)_{L^2(\Omega)} \quad \forall v \in \mathcal{V}_{\text{test}}.
\]  \tag{3.15}

In the Bubnov-Galerkin approach, both test and trial space are chosen as the Hilbert space \( \mathcal{V}_0 = \{ u \in \mathcal{V} : u|_{\partial \Omega} = 0 \} \), with \( \mathcal{V} = \{ u \in L^2(\Omega) : s \cdot \nabla_x u \in L^2(\Omega) \} \).  \tag{3.16} \tag{3.17}

The discretization consists in restricting a basis of the trial and test space to a finite subset.

However, it is known that for convection-reaction equations of hyperbolic character, for which problem (3.1) with \( \sigma \equiv 0 \) is an example, this standard Bubnov-Galerkin approach results in an unstable scheme (Johnson, 1987; Ávila et al., 2011).

Different ways of stabilizing the scheme have been proposed and used in the literature. We present three approaches that all can be cast into the form

\[
(Rv, Au)_{L^2(\Omega)} = (Rv, f)_{L^2(\Omega)} \quad \forall v \in \mathcal{V}_0.
\]  \tag{3.18}

with a stabilization operator \( R \) that involves the derivative in transport direction and \( \mathcal{V}_{\text{test}} = \mathcal{V}_0 \).

1. **Least squares.** In this approach (due to Manteuffel et al., 2000), the complete radiative transfer operator is also applied to the test function, i.e. \( R = \varepsilon A \), with a stabilization parameter \( \varepsilon \) that may depend on the physical quantities \( \kappa \) and \( \sigma \):

\[
(\varepsilon Au, Au)_{L^2(\Omega)} = (\varepsilon Au, f)_{L^2(\Omega)} \quad \forall v \in \mathcal{V}_0.
\]  \tag{3.19}

2. **SUPG.** In the Streamline Upwind Petrov Galerkin approach (Brooks and Hughes, 1982), the derivative in transport direction is added to the test function, weighted by a factor \( \delta > 0 \) depending on the local mesh width and coefficients \( \kappa \) and \( \sigma \) so that \( R = R_{\text{supg}} = \text{Id} + \delta s \cdot \nabla_x \):

\[
(v + \delta s \cdot \nabla_x v, Au)_{L^2(\Omega)} = (v + \delta s \cdot \nabla_x v, f)_{L^2(\Omega)} \quad \forall v \in \mathcal{V}_0.
\]  \tag{3.20}

3. **T-stabilized form.** A combination of the least squares form and the SUPG method is the T-stabilized form (Grella and Schwab, 2011a) with \( R = \varepsilon T \), which avoids mesh-dependent quantities and squaring the scattering operator:

\[
(\varepsilon T v, Au)_{L^2(\Omega)} = (\varepsilon T v, f)_{L^2(\Omega)} \quad \forall v \in \mathcal{V}_0.
\]  \tag{3.21}

The stabilization parameter \( \varepsilon \) is chosen similarly to the least squares approach.
Remark 1. If the stabilization parameters \( \varepsilon \) of the T-stabilized form and the least squares form are defined identically, these two formulations coincide in applications without scattering. With the choice \( \varepsilon = 1/\kappa \), the T-stabilized form would be identical to the SUPG form if we set \( \delta = 1/\kappa \). However, in the SUPG form, \( \delta \) is usually proportional to the size of the mesh width in a FEM formulation.

We are now in the position to state the variational formulation of problem (3.1): Find \( u \in V_0 \) such that

\[
(Rv, Au)_{L^2(\Omega)} = (Rv, f)_{L^2(\Omega)} \quad \forall v \in V_0.
\]

(3.22)

Also, we define here for later reference the physical part of \( V \) as

\[
V_D := \{ u \in L^2(D) : s \cdot \nabla_x u \in L^2(D) \text{ for almost every } s \in S \},
\]

(3.23)

where “almost every” is to be understood with respect to the Lebesgue measure on the angular domain.

### 3.3.2 Properties of the variational formulation

The following lemma collects auxiliary results which will become helpful later.

**Lemma 3.3 (Auxiliary results).**

1. Let \( v \in V \). Then \( (v, s \cdot \nabla_x v)_{L^2(\Omega)} \geq \frac{1}{2} \int_S \int_{\Gamma-(s)} v^2 s \cdot n(x) \, dx \, ds \). If furthermore \( v \in V_0 \), then \( (v, s \cdot \nabla_x v)_{L^2(\Omega)} \geq 0 \).

2. For \( v \in H^{1,0}(\Omega) \), it holds \( \| s \cdot \nabla_x v \| \leq \sqrt{d} \| v \|_{H^{1,0}(\Omega)} \).

**Proof.**

1. A proof is given by Manteuffel et al. (2000, Thm. 2.1). It uses the divergence theorem and exploits the fact that \( v|_{\partial \Omega_-} = 0 \) for \( s \cdot n(x) < 0 \) if \( v \in V_0 \), where \( n(x) \) is the outward unit normal on the boundary \( \partial D \):

\[
(v, s \cdot \nabla_x v)_{L^2(\Omega)} = \frac{1}{2} \int_S \int_D \nabla_x \cdot (sv^2) \, dx \, ds = \frac{1}{2} \int_S \int_{\partial D} v^2 s \cdot n(x) \, dx \, ds
\]

\[
= \frac{1}{2} \int_S \int_{\Gamma-(s)} v^2 s \cdot n(x) \, dx \, ds + \frac{1}{2} \int_S \int_{\Gamma+(s)} v^2 s \cdot n(x) \, dx \, ds.
\]

As \( s \cdot n \geq 0 \) in the second integral, we obtain the first assertion. If additionally \( v \in V_0 \), then the first integral vanishes, and the second assertion follows.

2. We again quote Manteuffel et al. (2000, Lemma 4.1 (i)):

\[
\| s \cdot \nabla_x v \|_{L^2(\Omega)}^2 \leq \int_D \int_S \left( \sum_{i=1}^d s_i D_{x_i} v \right)^2 \, ds \, dx \leq d \int_D \int_S \sum_{i=1}^d (s_i D_{x_i} v)^2 \, ds \, dx
\]

\[
\leq d \sum_{i=1}^d \| D_{x_i} v \|^2 \leq d \| v \|^2_{H^{1,0}(\Omega)}.
\]

\( \square \)
At this point, we introduce the problem related \textit{triple bar norm}

\[
\|||v|||^2 := \|v\|^2_{L^2(\Omega)} + \|s \cdot \nabla_x v\|^2_{L^2(\Omega)} + \|Q_1 v\|^2_{L^2(\Omega)}, \quad v \in \mathcal{V}.
\]  

(3.24)

This norm can be estimated from above by the $H^{1,0}(\Omega)$ norm, as the following lemma states.

\textbf{Lemma 3.4.} For any $v \in H^{1,0}(\Omega)$ it holds $|||v||| \leq c_n \|v\|_{H^{1,0}(\Omega)}$ with a constant $c_n > 0$.

\textit{Proof.} With Lemma 3.3 item 2 and property (3.13) of the scattering operator, the statement ensues directly:

\[
|||v|||^2 = \|v\|^2_{L^2(\Omega)} + \|s \cdot \nabla_x v\|^2_{L^2(\Omega)} + \|Q_1 v\|^2_{L^2(\Omega)}
\leq 2 \|u\|^2_{L^2(\Omega)} + d \|v\|^2_{H^{1,0}(\Omega)} \leq \max\{2, d\} \|v\|^2_{H^{1,0}(\Omega)},
\]

so that $c_n = \max\{\sqrt{2}, \sqrt{d}\}$. \hfill \Box

For the proof of existence and uniqueness of solutions to Eq. (3.22), boundedness of the bilinear form and linear form on the right hand side of the variational formulations are crucial ingredients. We show these properties here for a generic version of the SUPG and T-stabilized formulation with respect to the \textit{triple bar norm}.

\textbf{Lemma 3.5 (Continuity of the bilinear form).} Assume $\sigma, \kappa \in \mathcal{L}^\infty(\mathcal{D})$ with $\|\sigma\|_{\mathcal{L}^\infty(\mathcal{D})} =: \sigma_{\text{max}}, \|\kappa\|_{\mathcal{L}^\infty(\mathcal{D})} =: \kappa_{\text{max}}$. Let

\[
a_{\epsilon_1, \epsilon_2} (u,v) := (\epsilon_1 v + \epsilon_2 s \cdot \nabla_x v, Au)_{L^2(\Omega)}, \quad u, v \in \mathcal{V},
\]  

(3.25)

with stabilization coefficients $\epsilon_i : \mathcal{D} \rightarrow \mathbb{R}, i = 1, 2$, with $\epsilon_i > 0$ and $\epsilon_{i,\text{max}} := \|\epsilon_i\|_{\mathcal{L}^\infty(\mathcal{D})} < \infty$. Then there exists a constant $0 < c_c < \infty$ such that for all $u, v$ in $\mathcal{V}$

\[
|a_{\epsilon_1, \epsilon_2} (u,v)| \leq c_c \|u\| \|||v|||.
\]  

(3.26)

\textit{Proof.} To begin with, we estimate for all $u, v \in \mathcal{V}$

\[
\|Rv\| = \|\epsilon_1 v + \epsilon_2 s \cdot \nabla_x v\| \leq \epsilon_{1,\text{max}} \|v\| + \epsilon_{2,\text{max}} \|s \cdot \nabla_x v\|
\leq \max\{\epsilon_{1,\text{max}}, \epsilon_{2,\text{max}}, 1\} \|v\|, \quad \|Au\| \leq \kappa_{\text{max}} \|u\| + \|s \cdot \nabla_x u\| + \sigma_{\text{max}} \|Q_1 u\| \leq \max\{\kappa_{\text{max}}, 1, \sigma_{\text{max}}\} \|u\|.
\]  

(3.27)

(3.28)

Thus, we have by Cauchy-Schwarz inequality

\[
|a_{\epsilon_1, \epsilon_2} (u,v)| \leq \|Rv\| \|||v||| \leq \max\{\epsilon_{1,\text{max}}, \epsilon_{2,\text{max}}, 1\} \max\{\kappa_{\text{max}}, 1, \sigma_{\text{max}}\} \|||v||| \|||u|||.
\]  

\hfill \Box

After that, continuity of the right hand side linear form follows quickly.

\textbf{Lemma 3.6 (Continuity of the linear form).} Given the assumptions of Lemma 3.5 about $\kappa, \sigma, \epsilon_1$, and $\epsilon_2$, and additionally $f \in \mathcal{L}^2(\Omega)$, there is a constant $0 < c_l < \infty$ such that for the linear form

\[
l_{\epsilon_1, \epsilon_2} (v) := (\epsilon_1 v + \epsilon_2 s \cdot \nabla_x v, f)
\]  

(3.29)

and all $v \in \mathcal{V}$ it holds

\[
|l_{\epsilon_1, \epsilon_2} (v)| = |(\epsilon_1 v + \epsilon_2 s \cdot \nabla_x v, f)| \leq c_l |||v|||.
\]
3.4 Discretization

Proof. With estimate (3.27) and Cauchy-Schwarz inequality, we obtain directly

$$|l_{\varepsilon_1,\varepsilon_2}(v)| = |(\varepsilon_1 v + \varepsilon_2 s \cdot \nabla_x v, f) | \leq \max\{\varepsilon_{1,\text{max}},\varepsilon_{2,\text{max}}, 1\} ||v|| ||f||.$$  

Existence and uniqueness of the solution to (3.22) are guaranteed by the following theorem.

**Theorem 3.7** (Existence and uniqueness of solution). **Assuming the bilinear form of (3.22) is coercive, i.e.** $(Rv, Av) \geq c_{\varepsilon} ||v||^2$ for all $v \in V_0$ and a $c_{\varepsilon} > 0$, and bounded, and the linear form of (3.22) is bounded, then for any $f \in L^2(\Omega)$ there exists a unique solution $u \in V_0$ to problem (3.22).

Proof. As $(V_0, (\cdot, \cdot))$ is a Hilbert space, the Lax-Milgram theorem (see e.g. Brenner and Scott, 2008, Thm. 2.7.7) ensures existence and uniqueness of the solution.

### 3.4 Discretization

#### 3.4.1 Discretization in physical and angular space

In this section, we restrict the infinite dimensional trial and test spaces to finite dimensional subsets, first in the physical domain, then in angle.

**Discretization in physical space**

Above, we introduced in (3.16) and (3.17) the Hilbert spaces

$$V = \{u \in L^2(\Omega) : s \cdot \nabla_x u \in L^2(\Omega)\}$$

and $V_0 \subset V$, additionally satisfying zero inflow boundary conditions, as test and trial spaces for the variational formulation (3.22).

To discretize $V_D$, the physical part of $V$, we choose the space

$$V_D^l := S^{0,1}(D, \mathcal{T}_D^l) \subset H^1(D)$$

of continuous piecewise affine functions on a dyadically refined mesh $\mathcal{T}_D^l$ over $D$. Here, the parameter $l$ stands for the physical resolution. It is related to the mesh width $h$ in $\mathcal{T}_D^l$ by $h = O(2^{-l})$. With respect to the resolution $l = 0, \ldots, L$, the spaces $V_D^l$ form a nested sequence as defined in Eq. (2.13).

By defining the trial space as a subset of $H^1(D)$ we effectively assume a higher regularity on the solution than what is guaranteed by the definition (3.17) of $V$. For instance, solutions with line discontinuities due to the transport of discontinuous boundary data into the domain are not included in $V_D^l \otimes L^2(S)$. However, in order to leverage the advantages of a sparse tensor approximation, a higher regularity of the solution will be required in any case. Furthermore, since $V_D^l \otimes L^2(S)$ is dense in $V$, even discontinuous solutions would be approximated with increasing resolution.
Let $M_D := \dim \mathcal{V}_D^L$ be the number of physical degrees of freedom, then

$$M_D = O(2^{dL})$$  \hspace{1cm} (3.31)

with the dimension $d$ of the physical domain. The exact number will depend on the geometry of the domain. For a square or cube $D = [0,1]^d$, respectively, we obtain

$$M_D = (2^L + 1)^d.$$  \hspace{1cm} (3.32)

At this stage $\mathcal{V}$ is approximated by the semi-discrete tensor product space

$$\mathcal{V}_D^L \otimes L^2(S) \subset H^1(D) \otimes L^2(S) = H^{1,0}(D \times S) \subset \mathcal{V}.$$  \hspace{1cm} (3.33)

The space $\mathcal{V}_0$ is approximated correspondingly by $(\mathcal{V}_D^L \otimes L^2(S)) \cap \mathcal{V}_0$.

The $x$-discrete variational formulation of the radiative transfer problem then reads: Find $u_L \in (\mathcal{V}_D^L \otimes L^2(S)) \cap \mathcal{V}_0$ such that

$$(Ru_L, Au_L)_{L^2(\Omega)} = (Ru_L, f)_{L^2(\Omega)} \forall v_L \in \mathcal{V}_D^L \otimes L^2(S).$$  \hspace{1cm} (3.34)

**Discretization in angular space**

In angle, we restrict $\mathcal{V}$ in a similar manner to a finite dimensional subspace $\mathcal{V}_S^N$ with

an angular resolution parameter $N$, dimension $M_S$, and a basis $\{\beta_j\}_{j=1}^{M_S}$, for which the exact choice will again only be given in the discussion of each particular method. In any case the angular resolution and the dimension will be related by

$$M_S = \begin{cases} 2N + 1 & \text{if } d_S = 1, \\ (N + 1)^2 & \text{if } d_S = 2 \end{cases} = O(N^{d_S}).$$  \hspace{1cm} (3.35)

The fully discretized full tensor approximation space of $\mathcal{V}$ is then defined as

$$\mathcal{V}^{L,N} := \mathcal{V}_D^L \otimes \mathcal{V}_S^N \subset H^{1,0}(D \times S) \subset \mathcal{V},$$  \hspace{1cm} (3.36)

and the discretized approximation space with boundary conditions as $\mathcal{V}_{0,L,N} := \mathcal{V}^{L,N} \cap \mathcal{V}_0$. In these spaces, the full tensor approximate solution is represented by

$$u_{L,N}(x,s) := \sum_{i=1}^{M_D} \sum_{j=1}^{M_S} u_{ij} \alpha_i(x) \beta_j(s)$$  \hspace{1cm} (3.37)

with solution coefficients $u_{ij} \in \mathbb{R}$. The full tensor approximation space has the dimension

$$M = M_{L,N} = M_D \cdot M_S.$$  \hspace{1cm} (3.38)

The discrete variational formulation finally reads: Find $u_{L,N} \in \mathcal{V}_{0,L,N}^{L,N}$ such that

$$(Ru_{L,N}, Au_{L,N})_{L^2(\Omega)} = (Ru_{L,N}, f)_{L^2(\Omega)} \forall v_{L,N} \in \mathcal{V}_{L,N}.$$  \hspace{1cm} (3.39)
3.4 Discretization

Matrix representation

If we insert the basis expansion (3.37) of the approximation solution into the discrete variational formulation (3.39), we can derive a linear system of equations for the solution coefficients $u_{ij}$.

Let us first order the solution coefficients into a column vector $u \in \mathbb{R}^{M_S \cdot M_D}$ such that the enumeration of the angular index $j$ is outermost, i.e.

$$(u)_{(j-1)M_D+i} = u_{ij}$$  \hspace{1cm} (3.40)

In the same way, we order products of basis functions into a row vector $\psi \in (V^{L,N})^{M_S \cdot M_D}$ of basis functions $\psi_{(j-1)M_D+i}(x,s) = \alpha_i(x)\beta_j(s)$. With this vector, the radiative transfer matrix is defined by

$$A := (R\psi, A\psi)_{L^2(\Omega)}.$$  \hspace{1cm} (3.41)

Elementwise, this definition is to be read as

$$(A)_{(l-1)M_D+k, (j-1)M_D+i} = (R\psi_{(l-1)M_D+k, A\psi_{(j-1)M_D+i}})_{L^2(\Omega)}.$$  \hspace{1cm} (3.42)

Similarly, we define the transport matrix

$$T := (R\psi, T\psi)_{L^2(\Omega)}$$  \hspace{1cm} (3.43)

and the scattering matrix

$$Q := (R\psi, Q\psi)_{L^2(\Omega)}.$$  \hspace{1cm} (3.43)

Finally the right hand side vector $f \in \mathbb{R}^{M_S \cdot M_D}$ is defined as

$$f := (R\psi, f)_{L^2(\Omega)}.$$  \hspace{1cm} (3.44)

Using these definitions, the discrete variational formulation permits an algebraic representation: Find $u \in \mathbb{R}^{M_S \cdot M_D}$ as the solution to

$$Au = f \quad \Leftrightarrow \quad (T + Q)u = f.$$  \hspace{1cm} (3.45)

So far, all integrals have been assumed to be evaluated exactly. For numerical evaluation, however, we will have to choose certain quadrature rules depending on the choice of bases to compute the integrals of inner products and the scattering operator.

Even without having selected bases, we can inspect the structure of the matrix elements further. To this end, we analyze the generic formulation as in Eq. (3.25) with $R = \varepsilon_1 v + \varepsilon_2 s \cdot \nabla v$. An entry of the transport matrix can be divided into the following...
components:

\[(T)_{kl,ij} = (R\alpha_k\beta_l, Ta_i\beta_j)_{L^2(\Omega)} = (\varepsilon_1\alpha_k\beta_l + \varepsilon_2 s \cdot \nabla x\alpha_k\beta_l, s \cdot \nabla x\alpha_i\beta_j + \kappa\alpha_i\beta_j)_{L^2(\Omega)}\]

\[
= \sum_{n=1}^{d} \sum_{m=1}^{d} \left( \varepsilon_2 \frac{\partial \alpha_k}{\partial x_n} \cdot \frac{\partial \alpha_i}{\partial x_m} \right)_{L^2(D)} (s_n \beta_l, s_m \beta_j)_{L^2(S)}
+ \sum_{n=1}^{d} \left( \varepsilon_2 \frac{\partial \alpha_k}{\partial x_n}, s \alpha_i \right)_{L^2(D)} (s_n \beta_l, \beta_j)_{L^2(S)}
+ \sum_{m=1}^{d} \left( \varepsilon_1 \alpha_k, \frac{\partial \alpha_i}{\partial x_m} \right)_{L^2(D)} (\beta_l, s_m \beta_j)_{L^2(S)} + (\varepsilon_1 \alpha_k, s \alpha_i)_{L^2(D)} (\beta_l, \beta_j)_{L^2(S)}.
\]

(3.46)

We see that the transport matrix will be block diagonal if the angular basis functions are completely localized, i.e. \(\text{supp} \beta_j \cap \text{supp} \beta_l \neq \emptyset\) if and only if \(j = l\).

An element of the scattering matrix has the structure

\[(Q)_{kl,ij} = (R\alpha_k\beta_l, Qa_i\beta_j)_{L^2(\Omega)} = (\varepsilon_1\alpha_k\beta_l + \varepsilon_2 s \cdot \nabla x\alpha_k\beta_l, s \alpha_i \beta_j - s \sigma(\alpha_i \beta_j))_{L^2(\Omega)}\]

\[
= (\varepsilon_1 \alpha_k, s \alpha_i)_{L^2(D)} (\beta_l, \beta_j)_{L^2(S)} + \sum_{n=1}^{d} \left( \varepsilon_2 \frac{\partial \alpha_k}{\partial x_n}, s \alpha_i \right)_{L^2(D)} (s_n \beta_l, \beta_j)_{L^2(S)}
- (\varepsilon_1 \alpha_k, s \alpha_i)_{L^2(D)} (\beta_l, \Sigma \beta_j)_{L^2(S)} - \sum_{n=1}^{d} \left( \varepsilon_2 \frac{\partial \alpha_k}{\partial x_n}, s \alpha_i \right)_{L^2(D)} (s_n \beta_l, \Sigma \beta_j)_{L^2(S)}.
\]

(3.49)

This matrix will not necessarily be block diagonal with localized angular basis functions because the scattering integral operator \(\Sigma\) may “smear out” its argument. However, if the angular basis functions satisfy an orthogonality relation, only few block diagonals might actually be nonzero.

### 3.4.2 Solution of the linear system

We compare two iterative solution methods for the system of linear equations (3.45). The presentation here summarizes a corresponding section by Kanschat et al. (2008, Sec. 6.2).

**Picard iteration**

In the Picard iteration or \(\Lambda\) iteration, the scattering part is moved to the right hand side of Eq. (3.45) and considered as a perturbation:

\[Tu^{(k+1)} = -Qu^{(k)} + f.\]
If the perturbation assumption is justified, that is, if $\sigma$ is small, the method works satisfactorily, otherwise convergence may become very slow.

An improvement over the Picard iteration is obtained by subtracting $Tu^{(k)}$ and applying the preconditioner $P = T^{-1}$ on both sides:

$$u^{(k+1)} = (I - PA)u^{(k)} + Pf.$$  \tag{3.50}

If $PA$ is symmetric positive definite, the Richardson method achieves a convergence towards the exact solution $u^*$ of Eq. (3.50) as

$$\|u^{(k+1)} - u^*\|_2 \leq \frac{\text{cond}_2(\frac{PA}{PA}) - 1}{\text{cond}_2(\frac{PA}{PA}) + 1} \|u^{(k)} - u^*\|_2,$$

where $\text{cond}_2(PA)$ is the 2-norm condition number of the matrix $PA$. The appeal of both iteration schemes is that only the transport part has to be inverted. However, even the improved scheme requires a large number of steps to solve scattering dominated problems.

**Krylov space solver**

The alternative to the Picard iteration is to apply a Krylov space solver such as GMRES or BiCGStab directly to Eq. (3.45).

This approach also works well in applications where scattering and transport part are of equal magnitude. However, the case $\sigma \gg \kappa$ may require sophisticated preconditioners to limit the number of iterations.

If all eigenvalues of the preconditioned system matrix $PA$ are real and positive, the error of Krylov space solvers can be estimated by (e.g. Golub and Loan, 1989, Thm. 10.2.5)

$$\|u^{(k+1)} - u^*\|_{PA} \leq \frac{\sqrt{\text{cond}_2(\frac{PA}{PA})} - 1}{\sqrt{\text{cond}_2(\frac{PA}{PA})} + 1} \|u^{(k)} - u^*\|_{PA},$$

with the $PA$-norm defined by $\|u\|_{PA} := \sqrt{u^\top PAu}$. In practice, this can mean a significant reduction in iterations compared to the Picard iteration scheme.
4 Spherical harmonics method

In this chapter, we develop a sparse version of the $P_N$-method for radiative transfer. This sparse tensor $P_N$-method will be based on the T-stabilized form of the variational formulations introduced in Sec. 3.3. In the first section, we prove coercivity of this formulation to ensure its well-posedness.

The gist of our sparse tensor spherical harmonics method is presented in Sec. 4.2: Based on the ideas of sparse grids, we introduce sparse tensor product spaces containing only conforming, relevant products of spatial and angular basis functions in order to reduce the complexity of the discretization.

In Sec. 4.3, we show that the sparse tensor spherical harmonics method delivers the benefits of sparse grids of essentially undeteriorated convergence while reducing the number of degrees of freedom to that of a problem in the physical domain alone.

A numerical comparison between the standard full tensor and sparse tensor method in a series of 2+1D examples can be found in Chap. 8.

Advantages and disadvantages of the $P_N$-method have been discussed in Sec. 1.3 already.

We refer to the $P_N$-method also as “spherical harmonics method” even though the spectral functions of the 1-sphere, the circle, are rather known as trigonometric functions. However, in the literature, the spectral approach in angle is usually denominated by spherical harmonics method, therefore we make use of this term in both the 3+2D and 2+1D case.

4.1 Variational formulation

As we intend to approximate the solution of the RTP in a direct sparse way according to the notion of sparse grids outlined in Sec. 2.2, we will choose a hierarchic basis for the finite function space over the physical domain later on. In such a setting, the mesh dependent quantity $\delta$ of the SUPG variational formulation (3.20) may be difficult to interpret since the finite physical approximation space contains basis functions of all resolutions up to a finest level. On the other hand, the least squares variational formulation (3.19) contains a term $(\Sigma v, \Sigma u)_{L^2(\mathcal{S})}$ which represents a six-dimensional integral over the angular domain in the 3D case. Even though the inner integrals of the two scattering integral operators do not couple, this term might be costly to evaluate.

For these reasons we decide upon the T-stabilized version of the variational formulation (3.21) for the spherical harmonics method with the bilinear form

$$a_T(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R},$$

$$a_T(u, v) := (\varepsilon Tv, Au)_{L^2(\Omega)}. \quad (4.1)$$

$$a_T(u, v) := (\varepsilon Tv, Au)_{L^2(\Omega)}. \quad (4.2)$$
The corresponding linear form on the right hand side is
\[ l_T(v) := (\varepsilon T v, f)_{L_2(\Omega)}. \tag{4.4} \]

The choice \( \varepsilon = 1/\kappa \) in (4.2) and (4.4) enforces \( \varepsilon_1 = 1 \) and \( \varepsilon_2 = 1/\kappa \) in the continuity Lemmata 3.5 and 3.6. If \( \kappa_{\min} := \min_{x \in D} \kappa(x) > 0 \) is satisfied, boundedness of the T-stabilized bilinear form and linear form follows. It remains to prove coercivity of the bilinear form.

**Lemma 4.1 (Coercivity of T-stabilized bilinear form).** Let \( \varepsilon = 1/\kappa, \min_{x \in D} \kappa(x) =: \kappa_{\min} > 0, \min_{x \in D} \sigma(x) =: \sigma_{\min} > 0 \). Assume further that
\[ \kappa_{\max} \left( \max_{x \in D} \left\{ \frac{\sigma}{\kappa} \right\} \right)^2 \leq 4\sigma_{\min}, \tag{4.5} \]
then there exists a constant \( c_\varepsilon > 0 \) so that for all \( v \in V_0 \)
\[ a_T(v, v) \geq c_\varepsilon \|v\|^2. \]

**Proof.** For the case without scattering when least squares variational formulation and T-stabilized formulation coincide, Widmer (2009, Thm. 2.2) proved well-posedness in the norm \( \|v\|_S := \|v\| + \|s \cdot \nabla_x v\| \). This proof here proceeds similarly, taking scattering into account.

We begin by splitting the bilinear form into its constituent inner products:
\[
a_T(v, v) = (\varepsilon \kappa v + s \cdot \nabla_x v, A v) = \left( v + \frac{1}{\kappa} s \cdot \nabla_x v, s \cdot \nabla_x v + \kappa v + Q v \right)
= (v, s \cdot \nabla_x v) + (v, \kappa v) + (v, Q v) + \left( \frac{1}{\kappa} s \cdot \nabla_x v, s \cdot \nabla_x v \right)
+ \left( \frac{1}{\kappa} s \cdot \nabla_x v, \kappa v \right) + \left( \frac{1}{\kappa} s \cdot \nabla_x v, Q v \right).
\]
The first inner product can be omitted due to statement 1 of Lemma 3.3, the same applies to the fifth term. The second is bounded by
\[ (v, \kappa v) \geq \kappa_{\min} \|v\|^2. \]
To estimate the third inner product, we use property (3.14) of the scattering operator.

The fourth inner product is bounded from below by
\[
\left( \frac{1}{\kappa} s \cdot \nabla_x v, s \cdot \nabla_x v \right) \geq \frac{1}{\kappa_{\max}} \|s \cdot \nabla_x v\|^2.
\]

For the sixth inner product we apply Cauchy-Schwarz inequality and Young’s inequality \( ab \leq a a^2/2 + b^2/(2a) \) with a parameter \( \alpha > 0 \):
\[
\left( \frac{1}{\kappa} s \cdot \nabla_x v, Q v \right) \geq -\max_{x \in D} \left\{ \frac{\sigma}{\kappa} \right\} \|s \cdot \nabla_x v\| \|Q v\|
\geq -\frac{\alpha}{2} \max_{x \in D} \left\{ \frac{\sigma}{\kappa} \right\} \|s \cdot \nabla_x v\|^2 - \frac{1}{2\alpha} \max_{x \in D} \left\{ \frac{\sigma}{\kappa} \right\} \|Q v\|^2
\]
Combining these previous estimates yields
\[
a_T(v, v) \geq \kappa_{\min} \|v\|^2 + \left( \frac{1}{\kappa_{\max}} - \frac{\alpha}{2} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \right) \|s \cdot \nabla_x v\|^2 \\
+ \left( \sigma_{\min} - \frac{1}{2\alpha} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \right) \|Q_1 v\|^2 \\
\geq \min \left\{ \kappa_{\min} \frac{1}{\kappa_{\max}} - \frac{\alpha}{2} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\}, \sigma_{\min} - \frac{1}{2\alpha} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \right\} \||v||^2.
\]

We obtain condition (4.5) by eliminating \(\alpha\):
\[
\frac{1}{\kappa_{\max}} \geq \frac{\alpha}{2} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \Rightarrow \kappa_{\max} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \leq \frac{2}{\alpha} \\
\sigma_{\min} \geq \frac{1}{2\alpha} \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \Rightarrow 4\sigma_{\min} \left( \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \right)^{-1} \geq \frac{2}{\alpha} \\
\Rightarrow \kappa_{\max} \left( \max_{x \in D} \left\{ \frac{\sigma}{K} \right\} \right)^2 \leq 4\sigma_{\min}.
\]

If we take \(\kappa\) and \(\sigma\) to be constant in the physical domain, condition (4.5) simplifies to \(\sigma \leq 4\kappa\) so that we see that it is not too restrictive.

Thus, together with continuity well-posedness of the T-stabilized variational problem (3.21) follows from Thm. 3.7.

4.2 Discretization

We first introduce spherical harmonics of the angular domain with \(d_S = 1\) and 2 and discuss relevant properties as these functions will constitute the basis of angular approximation spaces in this chapter. Then we specify the full tensor approximation space without boundary conditions and derive its sparse version. Finally we show how conforming full and sparse tensor product spaces can be constructed in order to implement zero inflow boundary conditions in a strong sense.

The discretization is conceptually similar to that by Widmer et al. (2008). Here, however, we use spherical harmonics instead of wavelets as angular basis.

4.2.1 Angular basis functions

Definition of spherical harmonics

In the case \(d_S = 1\), we choose the real Fourier basis of sine and cosine functions to expand functions on the circle. These basis functions \(S_{n,m}^{(1)} : [0, 2\pi] \rightarrow \mathbb{R}\) are defined as
\[
S_{n,m}^{(1)}(\varphi) := \begin{cases} 
\frac{1}{\sqrt{2\pi}} & \text{if } n = 0, \\
\frac{\sin(n\varphi)}{\sqrt{\pi}} & \text{if } n > 0 \text{ and } m = 1, \\
\frac{\cos(n\varphi)}{\sqrt{\pi}} & \text{if } n > 0 \text{ and } m = 2,
\end{cases}
\]
\[
(n, m) \in \mathbb{N}_0 \times \{1, 2\} \text{ for } n > 0,
\]
unifying the notation for \(d_S = 1\) and \(d_S = 2\).
4 Spherical harmonics method

For basis functions for $d_S = 2$ on the 2-sphere $S^2$, we select real-valued spherical harmonics as e.g. Blanco et al. (1997) to avoid complex arithmetics. These real-valued spherical harmonics $S_{n,m}^{(2)} : [0, \pi] \times [0, 2\pi] \to \mathbb{R}$ are obtained from a linear combination of the complex-valued spherical harmonics $Y_{n,\tilde{m}}$ (Sabaka et al., 2010, Eq. (22)) of the same order $n$:

$$S_{n,m}^{(2)} := \begin{cases} \frac{(-1)^{\tilde{m}}}{\sqrt{2}} (Y_{n,\tilde{m}} + (-1)^{\tilde{m}} Y_{n,-\tilde{m}}) & \text{if } \tilde{m} > 0, \\ Y_{n,0} & \text{if } \tilde{m} = 0, \\ \frac{(-1)^{\tilde{m}}}{\sqrt{2}} (Y_{n,-\tilde{m}} - (-1)^{\tilde{m}} Y_{n,\tilde{m}}) & \text{if } \tilde{m} < 0, \end{cases}$$

in which $-n \leq \tilde{m} \leq n$ and $\tilde{m} = m - n - 1$ so that $m = 1, \ldots, 2n + 1$.

The index $m$ for basis functions of the same order $n$ has the maximum value $m_{n,d_S}$, which is

$$m_{n,1} = 2 \quad \text{for } d_S = 1,$$

$$m_{n,2} = 2n + 1 \quad \text{for } d_S = 2.$$  \hspace{1cm} (4.8)

Properties of spherical harmonics

Spherical harmonics form an orthonormal basis of $L^2(S)$ (Sabaka et al., 2010, Eq. (28)). Any function $f \in L^2(S)$ can be represented in spherical harmonics as

$$f(s) = \sum_{n=0}^{\infty} \sum_{m=1}^{m_{n,d_S}} a_{n,m} S_{n,m}^{(d_S)}(s),$$

with coefficients $a_{n,m}$ determined from

$$a_{n,m} = (f, S_{n,m}^{(d_S)})_{L^2(S)}.$$

The expansion of a function in spherical basis functions has a number of useful relations to its derivatives on the sphere. In spherical coordinates $(\rho, \vartheta_1, \ldots, \vartheta_{d_S})$, the Laplacian is given by

$$\Delta = \frac{\partial^2}{\partial \rho^2} + \frac{d_S}{\rho} \frac{\partial}{\partial \rho} - \frac{1}{\rho^2} \delta,$$

where the Beltrami operator $\delta$ contains the angular part of the Laplacian of the form

$$\delta = - \sum_{j=1}^{d_S} \frac{1}{q_j} \frac{\partial}{\sin^{d_S-j} \vartheta_j} \left( \frac{\partial}{\vartheta_j} \sin^{d_S-j} \vartheta_j \frac{\partial}{\vartheta_j} \right),$$

$q_1 = 1$, $q_j = (\sin \vartheta_1 \sin \vartheta_2 \ldots \sin \vartheta_{j-1})^2$, $j > 1$.

The expansion coefficients $b_{n,m}$ in terms of spherical harmonics of the $l$-th power of the Beltrami operator can now be expressed by the expansion coefficients $a_{n,m}$ of the original function $f \in \text{Dom}(\delta^l)$, the domain of $\delta^l$, by (see Mikhlin and Prössdorf, 1986, Ch. 8, §4, Eq. 8)

$$b_{n,m} = n^l(n + d_S - 1)^l a_{n,m}.$$
For \( t \in \mathbb{N}_0 \), the \( t \)-th power of the Beltrami operator therefore has the spherical expansion

\[
\delta^t f(s) = \sum_{n=1}^{\infty} \sum_{m=1}^{m_n, d_S} n^t (n + d_S - 1)^t a_{n, m} S_{n, m}^{(d_S)}(s),
\]

and for its \( L^2 \)-norm, it holds the Parseval equation

\[
\| \delta^t f \|^2_{L^2(S)} = \sum_{n=1}^{\infty} \sum_{m=1}^{m_n, d_S} n^{2t} (n + d_S - 1)^{2t} |a_{n, m}|^2.
\] (4.10)

From this equation, we see that the domain of \( \delta^t \) consists of those functions for which

\[
\sum_{n=1}^{\infty} \sum_{m=1}^{m_n, d_S} n^{4t} |a_{n, m}|^2 < \infty
\]

is satisfied. We can also rewrite (4.10) in the form

\[
\| \delta^{t/2} f \|^2_{L^2(S)} = \sum_{n=1}^{\infty} \sum_{m=1}^{m_n, d_S} n^t (n + d_S - 1)^t |a_{n, m}|^2.
\] (4.11)

For odd \( t \in \mathbb{N}_0 \), \( \delta^{t/2} \) is understood in the sense of interpolation of linear operators (see e.g. Triebel, 1995, Sec. 1.15.1, for the definition). We arrive at the following theorem:

**Theorem 4.2.** For a function \( f \in H^t(S) \cap \text{Dom}(\delta^{t/2} f) \cap L^2(S) \), where \( L^2(S) \) is the subspace of \( L^2(S) \) orthogonal to \( 1 \), \( t > 0 \), we obtain the estimate for its \( H^t \)-norm

\[
\| f \|_{H^t(S)} \simeq \| \delta^{t/2} f \|_{L^2(S)},
\]

where the symbol \( \simeq \) is the abbreviation for a two-sided estimate with positive constants independent of \( f \).

The proof is given by e. g. Mikhlin and Prössdorf (1986, Ch. 8, §4, Thm. 4.1).

**Definition of Legendre polynomials on angular regions**

In the \( (d, d_S) = (2, 1) \) case, we choose Legendre polynomials as basis functions on the angular regions \( S_q = [\varphi_1, \varphi_2] \subset S^1 \).

Let \( L_n(x) \) be the normalized Legendre polynomials which are \( L^2 \)-orthonormal on \([-1, 1]\). We map the polynomials to \( S_q \) via an affine transformation \( X_q(\varphi) \) to obtain basis functions \( S_{n, m}^{(1,q)} \) on an angular region:

\[
S_{n, m}^{(1,q)}(\varphi) := \begin{cases} 
L_0(X_q(\varphi)) & \text{if } n = 0, \\
L_{2n+m-2}(X_q(\varphi)) & \text{if } n > 0,
\end{cases} \quad n = 0, \ldots, N; m = 1, 2.
\] (4.12)

They conform to the notation of the spherical harmonics on the circle with an additional index \( q \) for the angular region.
4.2.2 Approximation spaces without boundary conditions

We have already discretized the generic variational formulation in Sec. 3.4 resulting in the discrete variational formulation (3.39). All that remained to arrive at a concrete algebraic representation of the RTP was the choice of finite function spaces $V^L_D$ and $V^N_S$ with appropriate bases.

Full tensor approximation

In Sec. 2.2 about sparse grids, we argued that to realize a direct sparse approximation the function spaces over the component domains should provide a hierarchic decomposition into a direct sum of detail spaces (cf. Eq. (2.14)).

On the physical domain, the space of piecewise affine nodal functions $S^{0,1}(D, T^l_D)$ on a dyadically refined mesh $T^l_D$, $l = 0, \ldots, L$, satisfies such a hierarchy relation if the hierarchic nodal basis is chosen. The physical finite function spaces are hence $V^L_D = S^{0,1}(D, T^l_D)$ as before in Eq. (3.30). Their dimension varies with the mesh width $2^{-L}$ as $M_D = O(2^{dL})$, likewise the dimension of the detail spaces as $\dim(W^l_D) = O(2^{dL})$.

In the angular domain $S$, we use a spectral discretization. Spherical harmonics $S^{(dS)}_{n,m} \in L^2(S)$ up to order $N$ as defined in Sec. 4.2.1 span the function spaces

$$V^N_S := \mathbb{P}^{dS}_N = \text{span}\{S^{(dS)}_{n,m} : n = 0, \ldots, N; m = 1, \ldots, m_{n,dS}\}$$ (4.13)

with $m_{n,dS}$ from (4.8). Their dimension $M_S$ is related to the order $N$ as in Eq. (3.35). A hierarchy is intrinsic to the space of spectral functions. The detail spaces are simply given by

$$W^m_S = \text{span}\{S^{(dS)}_{n,m} : m = 1, \ldots, m_{n,dS}\}.$$ with the dimension $\dim(W^m_S) = O(n_{dS}^{dS-1})$.

Based on the discretizations of the component domains, we construct the finite full tensor product approximation space as in Eq. (3.36),

$$V^{L,N} = V^L_D \otimes V^N_S = S^{0,1}(D, T^l_D) \otimes \mathbb{P}^{dS}_N,$$ (4.14)

and formally $V^{L,N}_0 = V^{L,N} \cap V_0$. However, in general, the boundary domain $\partial\Omega_-$ does not possess tensor product structure because its part in physical space depends on the direction $s$. Therefore the subspaces $V^{L,N}_0$ will generally not be of tensor product type once the boundary condition in Eq. (3.1) with $g = 0$ is imposed. To deal with this problem and to retain a tensor-product-like structure of the approximation space for the case with boundary conditions we will split up the physical function space in Section 4.2.3 and tensorize subspaces so that only conforming product combinations are created. For now, we continue by “sparsifying” $V^{L,N}$ instead of $V^{L,N}_0$, which exposes the construction of the sparse approximation space more clearly.
4.2 Discretization

Figure 4.1: Illustrating example of the structure of a full (left) and a sparse (right) tensor product space for $V_D^{=3} = S^{0,1}([0,1]^2, T_D^3)$ and $V_S^{N=3} = \mathbb{P}_3^1$.

Sparse tensor approximation

As the total number of degrees of freedom of the full tensor approximation space $V^{L,N}$ is $M_D \cdot M_S$, the curse of dimensionality renders approximation in this function space very expensive when the resolution levels $L$ and $N$ are increased.

Here, we follow the approach presented by Widmer et al. (2008) and perform an a-priori selection of degrees of freedom that are likely to be relevant for smooth solutions. Often only few of the product basis functions $\alpha_i(x)\beta_j(s)$, $i = 1, \ldots, M_D$, $j = 1, \ldots, M_S$, actually contribute significantly to the representation of the final solution. Especially for smooth solutions it can be observed that the contribution from highly resolving functions in physical space combined with high-frequent angular functions is usually low. Therefore we will form efficient trial spaces by including only product basis functions $\alpha_i(x)\beta_j(s)$ of limited combined variation. Even though the component spaces may be large, even infinite, the product spaces are sparse tensor product function spaces of much smaller dimension. With this approach, we can reduce the number of degrees of freedom significantly while committing only a moderate error in the representation of the solution.

By construction, the approximation spaces over physical and angular domain exhibit a hierarchic structure (2.14). Therefore we can compose these spaces from sequences of detail spaces as follows:

$$V^{l_D}_D = \bigoplus_{i=0}^{l_D} W^{i}_D, \quad l_D = 0, \ldots, L, \quad \text{and} \quad V^{l_S}_S = \bigoplus_{i=0}^{l_S} W^{i}_S, \quad l_S = 0, \ldots, N,$$

where we set $W^0_D := V^0_D$ and $W^0_S := V^0_S$, respectively. On the angular domain, the decomposition is $L^2$-orthogonal due to the same property of the spherical harmonics. The full tensor product space $V^{L,N}$ at level $L$ and order $N$ is given by the sum of tensor
product detail spaces analogously to Eq. (2.15):
\[ V_{L,N} = V_D^L \otimes V_S^N = \bigoplus_{0 \leq l_D \leq L \atop 0 \leq l_S \leq N} W_{D}^{l_D} \otimes W_{S}^{l_S}. \]  
(4.16)

Also in analogy to the definition (2.19) in the sparse grid introduction, we obtain the sparse tensor product space \( \hat{V}_{L,N} \subset V_{L,N} \) by restricting the sum to a smaller set of resolution indices:
\[ \hat{V}_{L,N} := \bigoplus_{0 \leq l_D \leq L \atop 0 \leq l_S \leq N} W_{D}^{l_D} \otimes W_{S}^{l_S}, \]  
(4.17)

with the sparsity profile \( f : [0, L] \times [0, N] \to \mathbb{R} \) to be specified. An example of such a sparse tensor product space is visualized in Fig. 4.1.

In contrast to the usual practice in the literature of the \( P_N \)-method (e. g. Modest and Yang, 2008; Schäfer et al., 2011), where a discretization of the same spatial resolution is employed for every angular degree of freedom, we adapt the spatial resolution to the angular order. Effectively the equations for every “mode” are solved on a different physical mesh, in general with coarser meshes for higher modes and finer meshes for lower modes, suited to their importance in the approximation of functions of higher mixed regularity. In this way we achieve a significant reduction of the number of degrees of freedom, which is estimated in the following lemma.

**Lemma 4.3.** Given a sparsity profile \( f(l_D, l_S) = l_D + \frac{L \log(l_S + 1)}{\log(N + 1)} \), the dimension of the sparse tensor product space \( \hat{V}_{L,N} \) can be estimated by
\[ \hat{M} = \dim(\hat{V}_{L,N}) \lesssim L^\theta \max\{2^{d_L}, N^{d_S}\} \lesssim (\log M_D)^\theta \max\{M_D, M_S\}, \]
where \( \theta = 1 \) if \( 2^{d_L} = (N + 1)^{d_S - 1} \) and zero otherwise. The relation \( \lesssim \) is given by Def. 2.1.

**Proof.** We have with Eq. (3.31) and some constant \( c_D > 0 \)
\[ \dim(W_{D}^{l_D}) = \dim(V_{D}^{l_D}) - \dim(V_{D}^{l_D - 1}) \leq c_D 2^{d_D}, \]
and on the angular domain from Eq. (3.35)
\[ \dim(W_{S}^{l_S}) = \begin{cases} 2 & \text{if } d_S = 1, \\ 2N + 1 & \text{if } d_S = 2 \end{cases} \leq c_S (l_S + 1)^{d_S - 1}, \]
with another constant \( c_S > 0 \). By introducing a logarithmic index \( \lambda = (d_S - 1)/d_S \log_2(l_S + 1) \) which varies between 0 and \( \Lambda := (d_S - 1)/d_S \log_2(N + 1) \), Lemma 4.3 becomes applicable and we obtain
\[ \hat{M}_{L,N} \lesssim \hat{M}_{L,\Lambda} \lesssim L^\theta \max\{2^{d_L}, 2^{d_S}\} \lesssim L^\theta \max\{2^{d_L}, N^{d_S}\}, \]
which can also be written in terms of degrees of freedom as
\[ \hat{M}_{L,N} \lesssim (\log M_D)^\theta \max\{M_D, M_S\} \]
since \( 2^{d_L} \leq M_D \) and \( N^{d_S} \leq M_S \). As \( M_D \) and \( M_S \) depend implicitly on \( L, N \), we use the relation \( \lesssim \) here as well.
Discretization

Remark 2. Of course, since $L$ and $N$ take integer values it will be difficult to increase them in such a way that condition $2^{dL} = (N + 1)^{dS-1}$ always holds exactly. However, if $L$ is incremented and then $N$ chosen as to minimize the deviation from the condition, the factor of $L$ or $\log M_D$ respectively in the complexity estimate of $\dim(\hat{V}^{L,N})$ will become noticeable. This is approximately the situation when the numbers of degrees of freedom in the extremal detail spaces $W_L^{D}$ and $W_N^{S}$ are equilibrated (“approximately” because we only bounded the dimension of $W_L^{D}$ from above).

4.2.3 Approximation spaces with boundary conditions

From definition (3.2) of the inflow boundary domain $\partial D$ it becomes obvious that this boundary is not a tensor product domain. At every point $x$ on $\partial D$, the range of inflow directions depends on the orientation of the outward normal of the domain. Thus, the approximation space $V_{0}^{L,N}$ respecting boundary conditions cannot be represented by a simple tensor product function space either.

Additionally, Dirichlet boundary conditions on the sphere can in general only be satisfied approximately by truncated series of spherical harmonics. The degrees of freedom associated with product basis functions involving spherical harmonics are no pointwise degrees of freedom on the angular domain, therefore simply setting some of them to zero will not satisfy this kind of boundary conditions.

Two common formulations of boundary conditions for the $P_N$-method used in the literature are:

- Marshak’s formulation (see e.g. Modest, 2003, Sec. 15.3), in which additional conditions for the degrees of freedom are derived from integral equations over the inflow hemispheres at points on the boundary.
- The inclusion of a boundary functional in the variational formulation which suppresses the deviation of the solution from the boundary conditions (e.g. Manteuffel et al., 2000).

However, both approaches only lead to weakly satisfied boundary conditions. Still worse is the fact that terms of the complexity $O(M_D M_S)$ ensue whose avoidance is the purpose of the sparse approximation.

Therefore we propose a different way here. We will split the physical function spaces into interior domain functions and functions on a part of the physical boundary with common normal. Interior domain functions will be tensorized as before with spherical harmonics of the angular domain, while boundary functions will be tensorized only with newly added spectral basis functions of the outflow part of their boundary. On the inflow part of their boundary, the resulting product basis functions satisfy zero inflow boundary conditions by construction.

Construction

Assume that the physical domain $D$ is a polyhedral domain with nonzero volume $\text{Vol}(D) > 0$. For the treatment of curved boundaries see Remark 4 at the end of this
4 Spherical harmonics method

Figure 4.2: To obtain conforming product basis functions, the depicted physical basis functions over $D = [0, 1]^2$ may only be tensorized with angular functions which are nonzero only on the marked outflow regions on $S^1$.

Figure 4.3: Boundary faces $\Gamma_k$ and outflow hemispheres $\Sigma_k$ for the example of an irregular hexagon as physical domain $D$ and corresponding spherical regions $S_\eta$ on the angular domain $S$. 
subsection. The boundary $\partial D$ of $D$ then consists of $K$ closed planar faces $F_k$:

$$\partial D = \bigcup_{k=1}^{K} F_k.$$ 

We denote the interior of a face $F_k$ by $\Gamma_k := F_k \setminus \partial F_k$ so that $\Gamma_k$ is open. As all points $x$ on a face $\Gamma_k$ share the same outward unit normal $n_k$, their outflow directions lie in the same open hemisphere $\Sigma_k$:

$$\Sigma_k := \{s \in S : s \cdot n_k > 0\}.$$ 

**Remark 3.** We are going to ignore the limit case $s \cdot n(x) = 0$ and the boundaries of the faces $F_k$ since they constitute sets of measure zero that are not relevant for the variational formulation.

If we consider all the points $x$ on several different faces $\Gamma_{kj}$, $j = 1, \ldots, \nu$, then their common outflow directions are contained in the intersection of the outflow hemispheres of the faces. There are $n_S$ unique common outflow regions denoted by

$$S_q := \bigcap_{k=k_1, \ldots, k_\nu} \Sigma_k, \; q = 1, \ldots, n_S,$$

such that $S = \bigcup_{q=1}^{n_S} S_q$ and $S_q \neq \emptyset \land (1 \leq q \neq r \leq n_S \Rightarrow S_q \cap S_r = \emptyset)$.

These angular regions constitute a partition of the angular domain. For an illustration of these quantities see Fig. 4.3. As the regions are bounded by great circles on the 2-sphere $S^2$, they are spherical polygons. For $d_S = 1$, the regions represent intervals of the circle. On the $S_q$, we define additional angular basis functions $S_{n,m}^{d_S,q}$, which for our examples with $d_S = 1$ are dilated Legendre polynomials as defined in (4.12).

For $d_S = 2$, the construction of spectral functions on polygonal regions of the sphere is possibly best accomplished on a spherical mesh. By dividing the spherical regions $S_q$ into triangular elements, spectral basis functions that are precisely aligned with the borders of inflow and outflow directions of the physical domain can be defined (cf. e.g. Giraldo and Warburton, 2005, on a spectral element method over a triangular mesh on $S^2$).

The span of these functions is denoted by

$$V_{S,q}^l := \mathcal{P}_l^{d_S,q} = \text{span} \{ S_{n,m}^{d_S,q} : n = 0, \ldots, l; \; m = 1, \ldots, m_{n,d_S} \}.$$ 

Detail spaces on the angular regions are defined as

$$W_{S,q}^l := \text{span} \{ S_{n,m}^{d_S,q} : n = l; \; m = 1, \ldots, m_{n,d_S} \}.$$ 

When the angular region $S_q$ is given, we denote the region of the physical domain boundary from which we obtain outflow into $S_q$ by

$$\Gamma_+(S_q) := \{ x \in \partial D : n(x) \cdot s > 0 \; \forall s \in S_q \}.$$
4 Spherical harmonics method

For the setup of the sparse approximation space $\tilde{V}_0^{L,N}$ with boundary conditions, we first split the hierarchic physical approximation space into several subspaces:

$$V_D^l = V_{D,0}^l \oplus \sum_{q=1}^{n_S} V_{\partial D,q}^l,$$

in which $V_{D,0}^l$ contains the functions which vanish on the boundary,

$$V_{D,0}^l := \{ v \in V_D^l : v|_{\partial D} = 0 \},$$

and $V_{\partial D,q}^l$ contains the functions whose nonzero boundary part is completely contained in $\Gamma_+(D_q)$:

$$V_{\partial D,q}^l := \{ v \in V_D^l \setminus V_{D,0}^l : v(x) = 0 \ \forall x \in \partial D \setminus \Gamma_+(S_q) \}.$$

This means that some physical basis functions can be contained in several $V_{\partial D,q}^l$ with different $q$.

The same separation is possible for the detail spaces $W_D^l$ into $W_{D,0}^l$ and $W_{\partial D,q}^l$. With these splittings we can define the full approximation space with boundary conditions $V_0^{L,N}$ as

$$V_0^{L,N} := \left( V_{D,0}^L \otimes V_S^N \right) \oplus \bigoplus_{q=1}^{n_S} \left( V_{\partial D,q}^L \otimes V_{S,q}^N \right)$$

$$= \bigoplus_{0 \leq l_D \leq L} \bigoplus_{0 \leq l_S \leq N} W_{D,0}^{l_D} \otimes W_S^{l_S} \otimes \bigoplus_{q=1}^{n_S} \bigoplus_{0 \leq l_D \leq L} \bigoplus_{0 \leq l_S \leq N} W_{\partial D,q}^{l_D} \otimes W_{S,q}^{l_S}.$$

The sparse version of $V_0^{L,N}$ is then defined based on the “sparse tensor product” $\hat{\otimes}$ as

$$\hat{V}_0^{L,N} := \left( V_{D,0}^L \otimes V_S^N \right) \oplus \bigoplus_{q=1}^{n_S} \left( V_{\partial D,q}^L \otimes V_{S,q}^N \right)$$

$$= \bigoplus_{0 \leq f(l_D,l_S) \leq L} W_{D,0}^{l_D} \otimes W_S^{l_S} \otimes \bigoplus_{q=1}^{n_S} \bigoplus_{0 \leq f(l_D,l_S) \leq L} W_{\partial D,q}^{l_D} \otimes W_{S,q}^{l_S}.$$

To illustrate these constructions, we describe and distinguish between several cases depending on the position of the physical basis function $\alpha_i$ (see also Fig. 4.2):

1. If $\alpha_i(x)$ is nonzero only on the interior of $D$ and zero on all $\Gamma_k$, the only product combinations need to be those with spherical harmonics of the full angular domain. Combinations with angular region functions are not required (Fig. 4.2 left).

2. If $\alpha_i(x)$ is nonzero on several faces $\Gamma_k$, only combinations with angular region functions $S_{n,m}^{(d,q)}(s)$ which are nonzero on the angular regions $S_q$ in the intersection of all outflow hemispheres $\Sigma_k$, $\Sigma_k$, are conforming (Fig. 4.2 center and right). Special cases are:
4.2 Discretization

a) $\alpha_i(x)$ is nonzero only on one $\Gamma_k$. Then only combinations with angular region functions that are nonzero on the hemisphere $\Sigma_k$ are conforming (Fig. 4.2 center).

b) $\alpha_i(x)$ is nonzero on several faces $\Gamma_{k_1}, \ldots, \Gamma_{k_v}$ whose unit normals $\mathbf{n}_{k_1}, \ldots, \mathbf{n}_{k_v}$ do not point into a common hemisphere, i.e. $\bigcap_{k=k_1,\ldots,k_v} \Sigma_k = \emptyset$. Then any combination involving $\alpha_i(x)$ cannot be contained in the tensor product spaces since there is no common outflow direction of the faces.

The number of angular basis functions to choose from for a product basis function increases from $M_S$ to $M_{S,\text{tot}} := (n_S + 1)M_S$, the number of physical basis functions remains at $M_D$. The total number of degrees of freedom of $\hat{V}_0^L,N$ or $V_0^L,N$, respectively, increases depending on the dimensionality and geometry of the problem, but the increase is subasymptotic in $L$, i.e. the ratio of boundary functions to interior functions goes to zero if the physical resolution is increased (for the structure of the function spaces and contained degrees of freedom see also Fig. 4.4).

For complicated geometries of the physical domain (see also Fig. 4.3), the construction will become rather involved, but the mathematical complexity does not increase with the order $N$ of the angular approximation. In the test case of the unit square as physical domain, which we will use for our experiments in Chapter 8, the outflow regions coincide to only four spherical regions with our construction so that the dictionary of angular functions remains pleasantly small.

Remark 4. In this approach, curved boundaries would have to be approximated by planar faces with sufficient resolution. Alternatively, if elements with curved faces are to be handled, each curved nodal boundary basis function on level $l_D$ could define a space $V_{l_D}^{l_D,D,q}$ with itself as the only contained element. This space would then have to be tensorized with a space of spectral functions over the common outflow region of all points on the intersection of the boundary and the support of the basis function. A loss in approximation power would appear on the boundary towards the corners of the support of the basis function. We refer to the construction of a graded sparse tensor product space by Widmer (2009), which addresses this problem. At the expense of an additional logarithmic factor in the numbers of degrees of freedom, the approximation rate on this graded space can still be bounded by the rate on the sparse tensor product approximation space without boundary conditions.

Complexity

Crude estimates of the dimensions of the approximation spaces with boundary conditions can simply be derived by assuming $\dim(V_S^N)$ to be $(n_S + 1)$ times as large as in the case without boundary conditions. Then the assertions of Lemma 4.3 still hold asymptotically as

$$\dim(\hat{V}_0^L,N) \lesssim (\log M_D) \theta \max\{M_D, M_S\}$$

since $n_S$ is independent of the resolution parameters $L$, $N$ and can be absorbed by the constant $c_S$ in the proof of the lemma.
4 Spherical harmonics method

Figure 4.4: Marked degrees of freedom are conforming and included in the full (left, 1853 DoFs) or sparse (right, 275 DoFs) approximation space, respectively. Leftmost column contains combinations with full spherical harmonics, following columns those with Legendre polynomials over angular regions $S_q$.

4.3 Approximation properties

We continue to follow Widmer (2009) for the analysis of the method, but deviate whenever necessary to accommodate the spectral discretization in angle.

As the sparse tensor product space contains fewer elements than the full tensor product space, we have to expect a loss of accuracy in general. In the next sections, we therefore investigate the impact of the reduction of the approximation space on the rate of approximation in the full and sparse tensor product space. We will find that, at least for solutions of a certain smoothness, both full and sparse tensor product method achieve the same asymptotic convergence rate up to logarithmic factors.

4.3.1 Approximation on the physical domain

For approximation results on the physical domain we require an estimate for the $H^1$-error of the approximation of functions in $H^{s+1}(D)$, $s \in [0, 1]$, by piecewise affine functions on the triangulation $T_D^L$. As $H^{s+1}(D) \subset C^0(D)$ by Sobolev’s inequality only if $s + 1 > d/2$, which yields $s > 0$ for $d = 2$ and $s > 1/2$ for $d = 3$, standard nodal interpolation is not necessarily defined. Hence we will derive approximation results on the physical domain via a Clément-type quasi-interpolation operator satisfying the following properties (Scott and Zhang, 1990, Thm. 4.1 and Cor. 4.1):

**Lemma 4.4 (Approximation of quasi-interpolation).** For polyhedral $D \subset \mathbb{R}^d$ and a shape-
regular triangulation $T_D^L$ on $D$ with mesh width $h = 2^{-L}$, the quasi-interpolation $P^L_D v$ of a function $v \in H^{s+1}(D)$, $s \in [0, 1]$, to the space $V^L_D = S^{0,1}(D, T_D^L)$ of piecewise affine functions on $T_D^L$ satisfies the error estimate

$$\|v - P^L_D v\|_{H^1(D)} \leq c_H 2^{-sL} \|v\|_{H^{s+1}(D)},$$

48
where \( c_H > 0 \) is a constant independent of \( L \).

**Lemma 4.5** (Stability of quasi-interpolation). Under the assumptions of Lemma 4.4, quasi-interpolation is \( H^1 \)-stable, i.e. there exists a constant \( c_B > 0 \) independent of \( L \) such that for all \( v \in H^1(D) \) it holds

\[
\| P^L_1 v \|_{H^1(D)} \leq c_B \| v \|_{H^1(D)}.
\]

### 4.3.2 Approximation on the angular domain

The following lemma shows an \( N^{-t} \) convergence rate in angle for the \( L^2 \)-projection \( P^N_{L^2(S)} : L^2(S) \rightarrow V^N_S \), where \( V^N_S \) is the space of full spherical harmonics defined in Eq. (4.13).

**Lemma 4.6.** For \( w \in H^1(S), t \in \mathbb{N}_0 \), the approximation error of the expansion of \( w \) in spherical harmonics up to order \( N \) by \( L^2(S) \)-projection is given by

\[
\left\| w - P^N_{L^2(S)} w \right\|_{L^2(S)} \leq c_1 N^{-t} \| w \|_{H^1(S)},
\]

where the constant \( c_1 > 0 \) is independent of \( N \).

**Proof.** We use the spherical harmonics series representation from Eq. (4.9) for \( w \) and Parseval’s theorem to express the square of the approximation error as an infinite series:

\[
\left\| w - P^N_{L^2(S)} w \right\|^2_{L^2(S)} = \sum_{n=N+1}^{\infty} \sum_{m=1}^{m_n d_S} |a_{n,m}|^2 \cdot \frac{n^{-t}(n + d_S - 1)^{-t}n^t(n + d_S - 1)^t}{1}
\]

by factoring out the maximum of the first term. With Eq. (4.11) and Thm. 4.2

\[
\left\| w - P^N_{L^2(S)} w \right\|^2_{L^2(S)} \leq ((N + 1)(N + d_S))^{-t} \sum_{n=1}^{m_n d_S} |a_{n,m}|^2 n^t(n + d_S - 1)^t
\]

\[
= ((N + 1)(N + d_S))^{-t} \| \delta^{t/2}w \|_{L^2(S)}^2
\]

\[
\leq c_1 ((N + 1)(N + d_S))^{-t} \| w \|_{H^t(S)}^2
\]

\[
\leq c_1 c_2 N^{-2t} \| w \|_{H^t(S)}^2
\]

where \( c_1 > 0 \) is a constant from the proof of Thm. 4.2, and \( c_2 > 0 \) is chosen such that \( ((N + 1)(N + d_S))^{-t} \leq c_2 N^{-2t} \). Taking the square root of both sides yields the statement of the lemma with \( c_1 = \sqrt{c_1 c_2} \).

**Remark 5.** Note that the approximation rate is only limited by the smoothness \( t \) of the function to be approximated, as usual in spectral approximations.
4.3.3 Approximation properties of full tensor product space

As the approximation spaces satisfying boundary conditions do not possess ordinary tensor product structure, the analysis of their approximation properties cannot be reduced in a straightforward way to the properties of approximation over the component domains. Widmer (2009) proved the same approximation rate as for the spaces without boundary conditions on her graded sparse tensor product spaces. These spaces are refined towards the transition zone between inflow and outflow boundary at the cost of an additional log-factor in the total number of degrees of freedom. However, the construction and analysis is rather technical and lengthy. Here, we will therefore content ourselves with the approximation properties of solutions quasi-interpolated and $L^2$-projected into the full and sparse tensor product spaces without boundary conditions. The validity of the approximation rates for approximation with boundary conditions will be verified in numerical experiments later on.

Based on the quasi-interpolation operator $P_l^T$ and the $L^2$-projector in angle $P_{l_2}^N$, we define the difference operators $\Delta_l^D := P_l^D - P_{l-1}^D$ with $P_{l-1}^D = 0$ and $\Delta_l^S := P_l^S - P_{l-1}^S$ with $P_{l-1}^S = 0$.

**Theorem 4.7 (Approximation on full tensor product space).** The approximation $u_{l,N} = P_l^T \otimes P_{l_2}^N u$ on the full tensor product space $V_{l,N}$ of a function $u \in H^{s+1,0}(\Omega) \cap H^{1,l}(\Omega)$, $s \in \{0, 1\}$, $l \in \mathbb{N}_0$, satisfies the asymptotic error estimate

$$\|u - u_{l,N}\|_{H^{1,0}(\Omega)} \lesssim 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)} + N^{-l} \|u\|_{H^{1,l}(\Omega)},$$

with the relation $\lesssim$ as in Def. 2.1.

**Proof.** We begin by splitting the error in two parts and writing the contributions as telescope sums:

$$\|u - u_{l,N}\|_{H^{1,0}(\Omega)} \leq \sum_{l_D=0}^{L} \sum_{l_S=0}^{N} \Delta_l^D \otimes \Delta_l^S u_{l_D, l_S} \|_{H^{1,0}(\Omega)} + \sum_{l_D=0}^{L} \sum_{l_S=0}^{N} \Delta_l^D \otimes \Delta_l^S u_{l_D, l_S} \|_{H^{1,0}(\Omega)}$$

Using Lemma 4.5 we remove $P_l^T$ from the first term and then apply the error estimate for the angular $L^2$-projector (Lemma 4.6) to it. The second term can directly be estimated by the error estimate of Lemma 4.4 for the quasi-interpolation operator:

$$\|u - u_{l,N}\|_{H^{1,0}(\Omega)} \leq c_B \|\text{Id} \otimes (\text{Id} - P_{l_2}^N) u\|_{H^{1,0}(\Omega)} + \|\text{Id} - P_l^T \otimes \text{Id} u\|_{H^{1,0}(\Omega)}$$

An increase of the number of degrees of freedom $M_D$ and $M_S$ is most effective in terms of error reduction if the error contributions from the discretizations in physical and angular space are of the same order of magnitude. Optimally with $N = 2^{L} \cdot 2^{L_2}/1$ (up to constants), the order $N$ of the spherical harmonics must be increased reciprocally.
4.3 Approximation properties

proportionally to the \( s/t \)-th power of the mesh width to reduce physical and angular contribution to the error at the same rate. As \( s \in \{0, 1\} \), the increase in \( N \) can be all the slower the larger the angular regularity \( t \) of the solution is.

4.3.4 Approximation properties of sparse tensor product space

The sparsity profile will be chosen slightly more general than in Lemma 4.3 such that the maximum angular resolution index never falls below a certain minimum angular resolution level \( N_0 \). If, for a given physical resolution, the sparsity profile yields a maximum angular resolution less than \( N_0 \), the returned angular resolution will be set to \( N_0 \) instead. We introduce this minimum angular resolution here even though it will only become relevant at the end of Sec. 5.3.5 for the discrete ordinates method because this will allow us to reuse parts of the proof to the following theorem. With \( N_0 = 0 \), this sparsity profile again coincides with the choice of the sparsity profile in Lemma 4.3.

**Theorem 4.8** (Approximation on sparse tensor product space). Given a sparsity profile with

\[
I^\text{max}_S(I_D) = \begin{cases} 2^\left\lfloor \log_2(N+1)/L(I_D) \right\rfloor & \text{if } 2^\left\lfloor \log_2(N+1)/L(I_D) \right\rfloor > N_0, \\ N_0 & \text{else}, \end{cases}
\]

and assuming that \( L \) and \( N \) vary such that \( -s + t \left\lfloor \log_2(N + 1) \right\rfloor / L = \zeta = \text{const} \), the approximation \( \hat{u}_{L,N} \) of a function \( u \in H^{s+1,f}(\Omega) \), \( s \in \{0, 1\} \), \( t \in \mathbb{N}_0 \), on the sparse tensor product approximation space without boundary conditions \( \hat{V}^{L,N} \) satisfies the error estimate

\[
\|u - \hat{u}_{L,N}\|_{H^1,0(\Omega)} \lesssim L(N^{-t} + 2^{-sL}) \|u\|_{H^{s+1,0}(\Omega)},
\]

with relation “\( \lesssim \)” from Def. 2.1.

**Proof.** This proof follows along the lines of that of the corresponding Theorem 2.6 by Widmer (2009), while allowing for the more general sparsity profile.

Recall that \( P^L_1 \), \( P^N_2 \), \( \Delta^L_1 \), and \( \Delta^N_2 \) are defined as in Sec. 4.3.3. We split the error into two terms:

\[
\|u - \hat{u}_{L,N}\|_{H^1,0(\Omega)} \leq \left\| \sum_{I_D=0}^{L} \sum_{I_S=I^\text{max}_S(I_D)+1}^{\infty} \Delta^L_1 \otimes \Delta^L_2 u \right\|_{H^1,0(\Omega)} + \left\| \sum_{I_D=L+1}^{\infty} \sum_{I_S=0}^{\infty} \Delta^L_1 \otimes \Delta^S_2 u \right\|_{H^1,0(\Omega)}.
\]

The second term on the right hand side can be estimated by Lemma 4.4:

\[
II = \left\| (\text{Id} - P^L_1) \otimes \text{Id} u \right\|_{H^1,0(\Omega)} \leq c_H 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]
4 Spherical harmonics method

This term will not contribute to the asymptotic terms.

The first term on the right hand side of (4.21) is split up further:

\[
I = \left\| \sum_{l_D=0}^{L} \left( P_{l_D}^I - P_{l_D-1}^I \right) \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{1,0}(\Omega)}
\]

\[
= \left\| \sum_{l_D=0}^{L} \left( P_{l_D}^I - \text{Id} + P_{l_D-1}^I \right) \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{1,0}(\Omega)}
\]

\[
\leq \left( \left\| (\text{Id} - P_{l_D}^I) \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{1,0}(\Omega)}
+ \left\| (\text{Id} - P_{l_D-1}^I) \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{1,0}(\Omega)} \right).
\]

Both norms on the right hand side of (4.23) can be estimated by Lemma 4.4 and Lemma 4.6:

\[
\left\| (\text{Id} - P_{l_D}^I) \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{1,0}(\Omega)} \leq c_H 2^{-sl_D} \left\| \text{Id} \otimes (\text{Id} - P_{L^2}^{\max(l_D)}) u \right\|_{H^{s+1,0}(\Omega)}
\]

\[
\leq c_H 2^{-sl_D} c_l l_S^{\max(l_D)} - t \| u \|_{H^{s+1,0}(\Omega)}.
\]

Inserting back into (4.23) yields

\[
I \leq 2c_H c_l \| u \|_{H^{s+1,0}(\Omega)} \sum_{l_D=0}^{L} 2^{-sl_D} l_S^{\max(l_D)} - t.
\]

(4.24)

The maximum angular resolution index \( l_S^{\max(l_D)} \) from (4.20) returns a value of \( N_0 \) for \( l_D \) greater than a certain full resolution level \( l_D^f \). This level can be obtained from the condition

\[
2^{\frac{\log_2(N+1)}{L(l_D^f)}} = N_0 \quad \Rightarrow \quad l_D^f = \left\lfloor L \left( 1 - \frac{\log_2(N_0)}{\log_2(N+1)} \right) \right\rfloor.
\]

We split the sum on the right hand side of (4.24) into a sparse part for all \( l_D \leq l_D^f \) and a full part for all \( l_D > l_D^f \):

\[
\sum_{l_D=0}^{L} 2^{-sl_D} l_S^{\max(l_D)} - t = \sum_{l_D=0}^{l_D^f} 2^{-sl_D} 2^{-t\left( \frac{\log_2(N+1)}{L(l_D^f)} \right)} + \sum_{l_D=l_D^f+1}^{L} 2^{-sl_D} N_0^{-t}.
\]

(4.25)

We transform the sparse part, using the assumption \( \zeta = -s + t \left( \frac{\log_2(N+1)}{L} \right) \):

\[
\sum_{l_D=0}^{l_D^f} 2^{-sl_D} 2^{-t\left( \frac{\log_2(N+1)}{L(l_D^f)} \right)} = 2^{-t\left( \frac{\log_2(N+1)}{L} \right)} \sum_{l_D=0}^{l_D^f} 2^{-s+tl_D}.
\]

(4.26)
4.3 Approximation properties

Since we are only interested in the order of convergence, we simplify from here on at some points by omitting the floor operation \( \lfloor \cdot \rfloor \) and setting \( \log_2(N + 1) \approx \log_2 N \). We estimate the sum on the right hand side of (4.26) by its largest summand. Two cases can be distinguished here:

1. If \( \zeta \leq 0 \), the largest summand occurs for \( l_D = 0 \). Using \( 2^{-t[\log_2(N+1)]} \leq c_I N^{-t} \):

\[
2^{-t[\log_2(N+1)]} \sum_{l_D=0}^{l_D^f} 2^{\xi l_D} \leq c_I N^{-t} \left( \left\lfloor L(1 - \frac{\log_2(N_0)}{\log_2(N+1)}) \right\rfloor + 1 \right) \leq c_I L N^{-t},
\]

with a constant \( c_I > 0 \), as \( L/\log_2(N+1) = t/(\zeta + s) \) is also assumed to be constant.

2. If \( \zeta > 0 \), the largest summand occurs for \( l_D = l_D^f \):

\[
2^{-t[\log_2(N+1)]} \sum_{l_D=0}^{l_D^f} 2^{\xi l_D} \leq c_K \left( 2^{L_s \left( -1 + \frac{\log_2(N_0)}{\log_2(N)} \right)} N_0^{-t} + 2^{L_s \left( -1 + \frac{\log_2(N_0)}{\log_2(N)} \right)} LN_0^{-t} \right)
- 2^{L_s \left( -1 + \frac{\log_2(N_0)}{\log_2(N)} \right)} N_0^{-t} \log_2(N_0) \frac{L}{\log_2(N)}
\]

\[
\leq c_K K L 2^{-sL},
\]

with constants \( c_K, c_{KK} > 0 \), and using \( L/\log_2(N) \approx t/(\zeta + s) = \text{const} \).

The full part in (4.25) can be estimated by

\[
\sum_{l_D=l_D^f+1}^{L} 2^{-sl_D} N_0^{-t} \leq c_F \left( 2^{L_s \log_2(N_0)} - 1 \right) N_0^{-t} 2^{-sL},
\]

with a new constant \( c_F > 0 \). It exhibits only subasymptotic growth.

In summary, we may write

\[
\sum_{l_D=0}^{L} 2^{-sl_D} l_S^{\max}(l_D) = cL(N^{-t} + 2^{-sL}),
\]

with another constant \( c > 0 \). By combining this estimate with Eqs. (4.24) and (4.22), we arrive at

\[
\| u - \hat{u}_{L,N} \|_{H^{0,0}(\Omega)} \leq cL(N^{-t} + 2^{-sL}) \| u \|_{H^{s+1,0}(\Omega)}.
\]
5 Discrete ordinates method

In this chapter, we introduce sparse versions of the discrete ordinates method (DOM). Construction and analysis will once again be based on a specialization of the phase space Galerkin framework from Chap. 3. We discuss both a direct sparse approach as in the spherical harmonics method from Chap. 4 and an approach using the combination technique (Sec. 2.3) for sparsification.

Ordinarily, the DOM discretizes the angular space by selecting a number of discrete directions. Then a purely physical transport subproblem is solved for each of the fixed directions and the subproblem solutions are e.g. interpolated in angle to form a solution to the RTP.

In the combination technique DOM, not all the subproblems are solved with the same physical resolution. Instead, we compute high resolution solutions only for few directions, while the resolution is lowered for many other directions in a way to achieve our complexity and accuracy goal: provided that the exact solution to the RTP is sufficiently smooth, the approximation rates with respect to the number of degrees of freedom deteriorate only by a logarithmic factor, while the problem size is reduced to that of a purely physical problem (also up to logarithmic factors).

The combination technique has been used before to solve problems other than the standard Laplace problem: the Navier-Stokes equations (Griebel et al., 1996), general time-dependent advection problems (Lastdrager et al., 2001), and the linear gyrokinetics equations (Kowitz et al., 2013), to name a few examples. Frequently cited advantages of the combination technique over direct sparse approaches include the greater flexibility with respect to the choice of bases in the component domains, compatibility with existing full tensor solvers, and straightforward parallelization due to independent subproblems.

Also here, we will exploit the flexibility in the selection of the component approximation spaces and their bases. On the physical domain, standard affine nodal basis functions on a globally refined mesh will serve as the basis of the finite element space. Hence, the mesh-dependent parameter of the SUPG variational formulation (3.20) is well-defined, and we can choose this formulation of the stabilized Galerkin framework.

As we achieve the reduction in numbers of degrees of freedom by combination of full tensor subproblems, the complexity of additional terms without tensor product structure for the weak satisfaction of boundary conditions in the variational formulation is not of concern here. Augmenting the variational formulation by these terms will allow us to prove error estimates directly for the discrete Galerkin solution and simplify the treatment of arbitrary inflow conditions.

The following first section of this chapter introduces the augmented SUPG variational formulation and proves its continuity and coercivity so that well-posedness of the problem is established.
In Sec. 5.2, we briefly present the standard discretization of the discrete ordinates method by collocation in angle, but then follow the phase space Galerkin approach as introduced before. We define angular basis and the discrete formulation, investigate its complexity, and elucidate the resulting structure of the linear system.

Sec. 5.3 derives the approximation properties of the direct sparse approach and the combination technique approach with special consideration to the consequences of quadrature for the evaluation of the scattering integral.

Numerical results from 2D and 3D experiments are reported in Chap. 8.

### 5.1 Variational formulation

We supplement the SUPG variational formulation (3.20) with the following bilinear form introduced by Manteuffel et al. (2000, Eq. (2.16)) to satisfy inflow boundary conditions in a weak sense:

\[
b(u,v) := (v, s \cdot nu)_{L^2(\partial\Omega_-)} = \int_S \int_{\Gamma(s)} s \cdot nu v \, dx \, ds,
\]

in which we have omitted the dependence of the outward unit normal \(n\) on the position \(x\). This bilinear form is defined for functions \(u,v : \partial D \times S \to \mathbb{R}\) with finite value of the inflow norm

\[
\|v\|_\perp := (b(v,v))^{1/2}.
\]

For functions \(v \in V\), we define the trace operator as the restriction to the boundary

\[
\gamma(v) := v|_{\partial \Omega_-}.
\]

By means of this operator, the boundary bilinear form \(b(\cdot, \cdot)\) and the norm \(\|\cdot\|_\perp\) are also defined for functions \(u,v \in V\) via

\[
b(u,v) = (\gamma(v), s \cdot n \gamma(u))_{L^2(\partial \Omega_-)}.
\]

We combine the triple bar norm and the inflow norm to a new norm

\[
\|v\|_1^2 := \|v\|^2 + \|v\|_\perp^2,
\]

which gives rise to the function space

\[
V_1 := \{v \in V : \|v\|_1 < \infty\}.
\]

The bilinear form of our variational ansatz for the RTP is then defined as

\[
a_{\text{supg}}(\cdot, \cdot) : V_1 \times V_1 \to \mathbb{R},
\]

\[
a_{\text{supg}}(u,v) := (v + \delta s \cdot \nabla x v, Au)_{L^2(\Omega)} - 2b(u,v),
\]

and the linear form as

\[
l_{\text{supg}}(\cdot) : V_1 \to \mathbb{R},
\]

\[
l_{\text{supg}}(v) := (v + \delta s \cdot \nabla x v, f)_{L^2(\Omega)} - 2b(g,v).
\]
5.1 Variational formulation

In the following, we will abbreviate \((\cdot, \cdot) := (\cdot, \cdot)_{L^2(\Omega)}\) and \(\|\cdot\| := \|\cdot\|_{L^2(\Omega)}\) again.

To establish well-posedness of the formulation, we prove continuity and coercivity in the \(\|\cdot\|_1\) norm. Even though the Galerkin framework of Chap. 3 has to be extended, we will benefit from the preparatory work there.

**Lemma 5.1 (Continuity of augmented SUPG bilinear form).** Given the assumptions of Lemma 3.5 about coefficients \(\kappa\) and \(\sigma\) and choosing \(\varepsilon_1 = 1\) and \(\varepsilon_2 = \delta\) with \(\delta_{\text{max}} := \|\delta\|_{L^\infty(D)} < \infty\), there is a constant \(0 < c_c < \infty\) such that for all \(u, v \in \mathcal{V}_1\)

\[
|a_{\text{supg}}(u, v)| \leq c_c \|u\|_1 \|v\|_1.
\]

**Proof.** We proceed analogously to Manteuffel et al. (2000, Thm. 3.3). Using the Cauchy-Schwarz inequality as well as estimates (3.27) and (3.28) it holds

\[
|a_{\text{supg}}(u, v)| \leq \|Rv\| \|Au\| + 2 \|v\| \|u\| \leq 2 \left(\|Rv\|^2 + \|v\|^2\right)^{1/2} \left(\|Au\|^2 + \|u\|^2\right)^{1/2} \leq 2 \max\{1, \delta_{\text{max}}\} \max\{\kappa_{\text{max}}, 1, \sigma_{\text{max}}\} \|u\|_1 \|v\|_1.
\]

**Lemma 5.2 (Continuity of augmented SUPG linear form).** Given the assumptions of Lemma 5.1 about \(\kappa, \sigma, \varepsilon_1, \varepsilon_2, \delta, \) and additionally \(f \in L^2(\Omega), \) \(g : \partial\Omega_\pm \rightarrow \mathbb{R}\) with \(\|g\|_\pm < \infty,\) there is a constant \(0 < c_l < \infty\) such that for \(v \in \mathcal{V}_1\) it holds

\[
|l_{\text{supg}}(v)| \leq c_l \|v\|_1.
\]

**Proof.** The proof relies on the same transformations as the one of Lemma 5.1:

\[
|l_{\text{supg}}(v)| \leq \|Rv\| \|f\| + 2 \|v\| \|g\| \leq 2 \left(\|Rv\|^2 + \|v\|^2\right)^{1/2} \left(\|f\|^2 + \|g\|^2\right)^{1/2} \leq 2 \max\{1, \delta_{\text{max}}\} (\|f\| + \|g\|_\pm) \|v\|_1.
\]

We continue by proving coercivity of the bilinear form. For simplicity we shall assume \(\delta\) and \(\kappa\) to be constant on the physical domain. Coercivity of the SUPG variational formulation for the RTP is also proved by Ávila et al. (2011, Lemma 2), although in a different norm.

**Lemma 5.3 (Coercivity of the augmented SUPG bilinear form).** Let \(\kappa, \delta\) be positive functions which are constant on the physical domain \(D.\) Assume \(\min_{\chi \in D} \sigma =: \sigma_{\text{min}} > 0\) and \(\sigma_{\text{max}} = \|\sigma\|_{L^\infty(D)},\) and additionally that

\[
\delta \sigma_{\text{max}}^2 = 4\sigma_{\text{min}}, \quad \delta \kappa < 3.
\]

Then the bilinear form \(a_{\text{supg}}(\cdot, \cdot)\) from (5.6) is coercive on \(\mathcal{V}_1 \times \mathcal{V}_1:\) there is a constant \(c_c > 0\) such that for all \(v \in \mathcal{V}_1\) it holds

\[
a_{\text{supg}}(v, v) \geq c_c \|v\|_1^2.
\]
5 Discrete ordinates method

Proof. The approach to the proof is basically the same as for the T-stabilized bilinear form in Lemma 4.1, therefore we focus on the differences here. For an overview of the involved terms we again split the bilinear form into separate inner products:

\[ a_{\text{supg}}(v, v) = (v + \delta s \cdot \nabla x v, Av) - 2b(v, v) \]

\[ = (v + \delta s \cdot \nabla x v, s \cdot \nabla x v + \kappa v + Q v) - 2(\gamma(v), s \cdot n \gamma(v))_{L^2(\partial \Omega_\gamma)} \]

\[ = (v, s \cdot \nabla x v) + (v, \kappa v) + (v, Q v) + (\delta s \cdot \nabla x v, s \cdot \nabla x v) \]

\[ + (\delta s \cdot \nabla x v, \kappa v) + (\delta s \cdot \nabla x v, Q v) - 2(\gamma(v), s \cdot n \gamma(v))_{L^2(\partial \Omega_\gamma)} \quad (5.9) \]

As we assumed \( \delta \) and \( \kappa \) to be constant, we can factor these coefficients out of the fifth inner product and combine it with the first. Applying statement 1 of Lemma 3.3 yields

\[ (1 + \delta \kappa) (v, s \cdot \nabla x v) \geq -(1 + \delta \kappa) \frac{1}{2} \|v\|_2^2. \]

Together with the boundary term, we obtain

\[ (v, s \cdot \nabla x v) + (\delta s \cdot \nabla x v, \kappa v) - 2b(v, v) \geq \left( \frac{3}{2} - \frac{1}{2} \delta \kappa \right) \|v\|_2^2. \quad (5.10) \]

The second and third term in Eq. (5.9) are estimated as in Lemma 4.1.

The fourth inner product in Eq. (5.9) is now

\[ (\delta s \cdot \nabla x v, s \cdot \nabla x v) = \delta \|s \cdot \nabla x v\|^2. \]

For the sixth inner product we apply Cauchy-Schwarz inequality and Young’s inequality with a parameter \( \alpha > 0 \):

\[ (\delta s \cdot \nabla x v, Q v) \geq -\delta \sigma_{\text{max}} \|s \cdot \nabla x v\| \|Q_1 v\| \geq -\delta \sigma_{\text{max}} \left( \frac{\alpha}{2} \|s \cdot \nabla x v\|^2 + \frac{1}{2 \alpha} \|Q_1 v\|^2 \right) \]

Combining all estimates yields the result:

\[ a_{\text{supg}}(v, v) \geq \kappa \|v\|^2 + \delta (1 - \frac{\alpha}{2} \sigma_{\text{max}}) \|s \cdot \nabla x v\|^2 + (\sigma_{\text{min}} - \frac{1}{2 \alpha} \delta \sigma_{\text{max}}) \|Q_1 v\|^2 \]

\[ + \frac{1}{2} (3 - \delta \kappa) \|v\|_2^2 \]

\[ \geq \min\{\kappa, \delta (1 - \frac{\alpha}{2} \sigma_{\text{max}}), \sigma_{\text{min}} - \frac{1}{2 \alpha} \delta \sigma_{\text{max}}, \frac{1}{2} (3 - \delta \kappa)\} \|v\|_1^2. \]

By eliminating \( \alpha \) analogously to the proof of Lemma 4.1, we obtain \( \delta \sigma_{\text{max}}^2 < 4 \sigma_{\text{min}} \).

Well-posedness of the variational formulation follows from a variation of Thm. 3.7.

Theorem 5.4 (Existence and uniqueness of solution to augmented SUPG variational formulation). Provided that \( f \in L^2(\Omega) \) and \( \|g\|_{-} < \infty \) there exists a unique solution \( u \in \mathcal{V}_1 \) to the variational formulation

\[ a_{\text{supg}}(u, v) = l_{\text{supg}}(v) \quad \forall v \in \mathcal{V}_1. \quad (5.11) \]

Proof. Since \( (\mathcal{V}_1, \| \cdot \|_1) \) is a Hilbert space and Lemmata 5.1 - 5.3 guarantee continuity of the augmented SUPG bilinear form and linear form as well as coercivity of the bilinear form, the Lax-Milgram theorem (Brenner and Scott, 2008, Thm. 2.7.7) ensures existence and uniqueness of the solution to (5.11).
5.2 Discretization

In the literature, the discrete ordinates method is often derived by direct collocation of the angular variable. We first present this approach briefly in the next subsection in order to demonstrate that it leads to the same linear system as the phase space Galerkin framework with quadrature in angle, which is the way of discretization we will follow herein afterwards. This equivalence of the linear systems justifies the name “discrete ordinates method” also for the phase space Galerkin approach in the presence of quadrature.

5.2.1 Alternative discretization by collocation in angle

For the discretization of the RTP (1.1) by collocation (e.g. Modest, 2003, Sec. 16.2), one chooses a number of directions $s_j \in S, j = 1, \ldots, M_S$, which form a quadrature rule

$$\int_S f(s) \, ds \approx \sum_{j=1}^{M_S} w_j f(s_j)$$

(5.12)

on the angular domain together with weights $w_j$ for functions $f \in C^0(S)$. From the collocation and substitution of the quadrature rule for the scattering integral a system of purely spatial PDEs for the directional intensities $u_j(x) := u(x, s_j)$ results, which is coupled in the case with scattering:

$$s_j \cdot \nabla_x u_j(x) + (\kappa(x) + \sigma(x)) u_j(x) = \kappa(x) I_b(x) + \sigma(x) \sum_{m=1}^{M_S} w_m \Phi(s_j, s_m) u_m(x),$$

$$j = 1, \ldots, M_S.$$  

Boundary conditions are replaced correspondingly by $u_j(x)\big|_{\partial D} = g_j(x)$ for $s_j \cdot n(x) < 0$. Each of the spatial PDEs is then discretized on the physical domain by e.g. a stabilized Galerkin method leading to a coupled system of variational formulations for $u_j \in V_D^L$ and $j = 1, \ldots, M_S$

$$\left( R_j v, T_j u_j + \sigma u_j - \sum_{m=1}^{M_S} w_m \Phi(s_j, s_m) u_m \right)_{L^2(D)} - 2 \left( v, s_j \cdot n u_j \right)_{L^2(\Gamma_+)} = \left( R_j v, f \right)_{L^2(D)} - 2 \left( v, s_j \cdot n g_j \right)_{L^2(\Gamma_-)} \quad \forall v \in V_D^L$$

(5.13)

with directional stabilization and transport operators

$$R_j := R|_{s=s_j}, \quad T_j := T|_{s=s_j}, \quad j = 1, \ldots, M_S.$$  

(5.14)

The angular directions $s_j$ can in principle be chosen arbitrarily, but are often selected in practice to be invariant under rotations of 90 degrees around any coordinate axis because then only directions into the positive quadrant or octant of the angular domain
need to be specified. Furthermore, depending on the type of scattering kernel \( \Phi \) it might be beneficial for the accuracy of the quadrature if the sets of directions satisfy certain moment equations (Modest, 2003, Eq. (16.12)), that is, constants, \( s \), and \( ss^\top \) are integrated exactly.

In the sequel, however, we shall derive Eq. (5.13) or the linear system resulting from (5.13) as a phase space Galerkin discretization with piecewise constant basis functions on the angular domain and a midpoint quadrature rule. The treatment of physical and angular discretization within the same Galerkin framework will be crucial to the development of error estimates. The one-point quadrature rule also offers a convenient positivity preserving discretization of the scattering operator. Despite the different theoretical derivations, the resulting linear systems will be exactly the same, and so will be the numerical solutions which are obtained from either of the two approaches.

### 5.2.2 SUPG stabilization parameter

Ávila et al. (2011) derive the value of the SUPG stabilization parameter \( \delta \) for constant \( \kappa \) and \( \sigma \) as an algebraic approximation to the radiative transfer operator as

\[
\delta = \left( c_1^2 h^{-2} + (\kappa + \sigma)^2 + 2 c_1 \sigma h^{-1} M^{-1} S \right)^{-1/2},
\]

where \( h \) is the mesh width of the physical triangulation \( T_{DL} \). They use \( c_1 = 2 \) based on numerical experiments. If we develop (5.15) into a series in \( h \) around 0, we obtain

\[
\delta = c_1^{-1} h - \frac{1}{2} c_1^{-2} \sigma M^{-1} S h^2 + O(h^3).
\]

In our implementation, we only take the first term of this series into account as the second will be small already, and set \( c_1 = 10/3 \) after numerical testing.

### 5.2.3 Approximation spaces

We consider two variants for the definition of the approximation spaces. The first variant of nested approximation spaces lends itself more easily to analysis, the second of nonnested spaces provides more flexibility in the implementation.

**Nested approximation spaces**

On the physical domain, we employ the same discretization as in Sec. 4.2.2 for the spherical harmonics method. A dyadically refined family of triangulations \( T_{DL}^l, l = 0, \ldots, L \), together with hierarchical hat functions as basis gives rise to the nested sequence of approximation spaces \( V_{DL}^l = S^{0,1}(D, T_{DL}^l) \) with detail spaces \( W_{DL}^l = V_{DL}^l \ominus V_{DL}^{l-1} \).

In angle we proceed alike: On the family of dyadically refined triangulations \( \bar{T}_{SN}^\bar{n}, \bar{n} = 0, \ldots, \bar{N} \), with partition regions \( \bar{S}_j, j = 1, \ldots, O(2^{d\bar{n}}) \), characteristic functions span
the approximation spaces $\tilde{V}_S^N = S^{-1,0}(S, \tilde{T}_S^N)$. Corresponding detail spaces are given by $\tilde{W}_S^N = \tilde{V}_S^N \ominus \tilde{V}_S^{N-1}$. The number of angular degrees of freedom relates to the resolution level as $M_S = O(2^{d_SN})$.

The full tensor approximation space is then defined as in Eq. (2.15) by

$$V^{L,N}_{nest} = V_D^L \otimes \tilde{V}_S^N = \bigoplus_{0 \leq l_D \leq L} W_D^{l_D} \otimes \tilde{W}_S^{l_S},$$

a nested sparse tensor product approximation space can be defined as in (2.19) by

$$\tilde{V}^{L,N}_{nest} = \bigoplus_{0 \leq f(l_D,l_S) \leq L} W_D^{l_D} \otimes \tilde{W}_S^{l_S}. \quad (5.16)$$

In this case the sparsity profile is given by $f(l_D,l_S) = l_D + Ll_S/\tilde{N}$. This construction is suited for a direct sparse approach as introduced in Sec. 2.2.

A dyadically refined triangulation of the angular domain with characteristic functions as basis is also employed by Widmer (2009).

**Nonnested approximation spaces**

Dropping the requirement that the subspaces be nested permits us to span the physical approximation spaces $V_D^l$ with standard nonhierarchical hat functions as physical basis. Note that the detail spaces $W_D^l$ can still be defined as before, but they are not spanned any more by a simple subset of the physical basis functions.

Likewise, the family of angular partitions $T_S^n$, $n = 0, \ldots, N$ is not required to be nested. We only assume that the partition splits the angular domain into regions $S_j$ with disjoint interiors such that $S = \bigcup_{j=1}^{M_S} S_j$. The relation between the angular resolution $N$ and the angular number of degrees of freedom $M_S$ shall hold as in (3.35): $M_S = O(N^{d_S})$.

In the case $d_S = 1$, we subdivide the circle into intervals of equal size, i.e. $|S_j| = 1/M_S$. For $d_S = 2$, partitions on the sphere could be constructed as the Voronoi tessellation of any family of point sets, e.g. the maximum determinant points by Womersley and Sloan (2001). However, for the analysis of the quadrature error in Sec. 5.3.3, we shall assume that the points lie in the barycenters of the partition regions, which is satisfied by spherical centroidal Voronoi tessellations (Du et al., 2003, Sec. 5). Points of a spherical centroidal Voronoi tessellation are asymptotically uniformly distributed and suited for interpolation and quadrature alike. Some examples of such partitions for different resolutions are visualized in Fig. 5.1.

The full tensor approximation space is now given by

$$V^{L,N} = V_D^L \otimes V_S^N,$$

the sparse approximation space by

$$\tilde{V}^{L,N} = \sum_{l_D=0}^{L} V_D^{l_D} \otimes V_S^{\max(l_D)}.$$
Figure 5.1: Nonnested partitions of the sphere for angular resolution levels $N = 6, 16, 33$ with 49, 289, and 1156 elements, respectively (images generated with code by Burkardt, 2002).

with a sparsity profile that we assume to result in a maximum angular index of $l_{\text{max}} = 2^{\lfloor \log_2(N+1)/L(L-l_D) \rfloor}$ for the moment.

**Remark 6.** A nested sequence of angular approximation spaces could also be constructed by clustering regions of an arbitrary partition of high resolution. The nesting would then be achieved by coarsening of a fine partition instead of refinement of a coarse mesh.

### 5.2.4 Quadrature in angle

To complete the discretization, all integrals of inner products and the scattering integral operator in bilinear form (5.6) and linear form (5.7) are replaced by quadrature rules.

We will focus on the integrals over the angle here. For the theoretical construction of the method, we will use the simplest quadrature rule, the midpoint rule. Therefore, we choose the directions $s_j$ as the barycenters of the angular regions $S_j$, i.e.

$$s_j = \frac{1}{|S_j|} \int_{S_j} s \, ds, \quad j = 1, \ldots, M_S.$$

**Definition 5.5 (Composite midpoint rule on angular domain).** Let $\mathcal{T}_S^N = \{S_j\}_{j=1}^{M_S}$ be a partition of the sphere, $N$ and $M_S$ be related by (3.35), $s_j$ the barycenters of the elements $S_j$ of the partition, and the weights $w_j = |S_j|$ identical to the area of the partition elements. The composite midpoint rule on the angular domain for functions $f \in C^0(S)$ is then defined as the quadrature rule

$$Q^{(M_S)}[f] = \sum_{j=1}^{M_S} w_j f(s_j).$$

It is exact for piecewise affine functions on $\mathcal{T}_S^N$.

**Remark 7.** In principle, one could choose different quadrature rules to replace the integrals of inner products and those of the scattering integral operator. However, for the exposition here, we take both quadrature rules to be the same.
5.2 Discretization

5.2.5 Matrix representation of the discrete ordinates method

We shall see how the choices of basis and quadrature affect the evaluation of the elements of the transport and scattering matrices \( T \) and \( Q \). Kanschat (2008, Sec. 6.1) derives the matrix representation of the SUPG DOM in a similar way.

Revisiting formula (3.49) for elements of the scattering matrix first, we find that the angular inner products are transformed in the following ways:

\[
(\beta_l, \Sigma \beta_j)_{L^2(S)} = \left( \beta_l, \int_S \Phi(s, s') \beta_j(s') \, ds' \right)_{L^2(S)} \approx \left( \beta_l, \sum_{m=1}^{M_S} w_m \Phi(s, s_m) \beta_j(s_m) \right)_{L^2(S)}
\]

Only the term for \( m = j \) remains of the quadrature sum because \( \beta_j \) is the characteristic function on \( S_j \):

\[
(\beta_l, \Sigma \beta_j)_{L^2(S)} \approx (\beta_l, w_j \Phi(s, s_j))_{L^2(S)} \approx w_l w_j \Phi(s_l, s_j)
\]

The other terms are treated in the same way:

\[
(s_n \beta_l, \Sigma \beta_j)_{L^2(S)} \approx w_l(s_l) n w_j \Phi(s_l, s_j) \tag{5.17}
\]

\[
(s_n \beta_l, \beta_j)_{L^2(S)} = w_l(s_l) \tag{5.18}
\]

\[
(\beta_l, \beta_j)_{L^2(S)} = w_l \tag{5.19}
\]

In this expression, \((s_l)_n\) stands for the \( n \)-th component of the vector \( s_l \).

An entry of the transport matrix can therefore be written as

\[
(Q)_{kl,ij} = w_l \delta_{ij} \left( R_l a_k, \sigma a_i \right)_{L^2(D)} - w_l w_j \Phi(s_l, s_j) \left( R_l a_k, \sigma a_i \right)_{L^2(D)}, \tag{5.20}
\]

where \( \delta_{ij} \) is the Kronecker delta. The directional stabilization operator \( R_l \) was defined in Eq. (5.14). If we apply substitutions (5.18), (5.19), and a corresponding one for the \((s_n \beta_l, s_m \beta_j)_{L^2(S)}\) term to an entry of the transport matrix (3.47), we get

\[
(T)_{kl,ij} = w_l \delta_{ij} \left( R_l a_k, T_j a_i \right)_{L^2(D)}, \tag{5.21}
\]

where \( T_j \) is the directional transport operator from Eq. (5.14).

The boundary form with basis functions inserted becomes

\[
(\alpha_k \beta_l, s \cdot n a_i \beta_j)_{L^2(\partial \Omega_\text{in})} = \int_S \beta_l \beta_j \int_{\Gamma_\text{in}}(s) s \cdot n a_k a_i \, dx \, ds \approx \sum_{m=1}^{M_S} w_m \beta_l(s_m) \beta_j(s_m) \int_{\Gamma_\text{in}}(s_m) s_m \cdot n a_k a_i \, dx = w_l \delta_{ij} \int_{\Gamma_\text{in}}(s_j) s_j \cdot n a_k a_i \, dx =: \frac{1}{2} (B)_{kl,ij}, \tag{5.22}
\]

by which we have defined the boundary matrix \( B \).

With the physical basis functions arranged in a row vector \( \alpha \in (V_D^\text{i})^{M_D} \) and solution coefficients as degrees of freedom in a column vector \( u \in \mathbb{R}^{M_S \cdot M_D} \) as before in (3.40), we
can introduce more compact definitions for the block matrices occurring in Eqs. (5.20)-(5.22):

\[ T_j := (R_j \alpha, T_j \alpha)_{L^2(D)} \]
\[ M_j(\sigma) := (R_j \alpha, \sigma \alpha)_{L^2(D)} \]
\[ B_j := 2 (\alpha, s_j \cdot n_\alpha)_{L^2(\Gamma \setminus \{s_j\})}. \]

Componentwise, these definitions are to be read as:

\[ (T_j)_{ki} := (R_j \alpha_k, T_j \alpha_i)_{L^2(D)}, k, i \in \{1, \ldots, M_D\}. \]

We also combine \( w_j \Phi(s_i, s_j) =: \omega_{ij} \) into the matrix of scattering factors \( \Omega \in \mathbb{R}^{M_S \times M_S} \).

The right hand side load vector for direction \( s_j \) is given by:

\[ f_j := (R_j \alpha, f)_{L^2(D)}, \]

the boundary vector on the right hand side by:

\[ g_j := 2 (\alpha, s_j \cdot n_\alpha)_{L^2(\Gamma \setminus \{s_j\})}. \]

If we divide matrices (5.21)-(5.22) row-wise by \( w_j \), we arrive at the algebraic DOM problem in matrix representation: Find \( u \in \mathbb{R}^{M_S \cdot M_D}, u = (u_1, \ldots, u_{M_S})^\top \), as the solution to the linear system of equations

\[
\begin{bmatrix}
T_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & T_{M_S}
\end{bmatrix}
+ \begin{bmatrix}
M_1(\sigma) \\
\vdots \\
M_{M_S}(\sigma)
\end{bmatrix}
- \begin{bmatrix}
\omega_{11}M_1(\sigma) & \cdots & \omega_{1,M_S}M_1(\sigma) \\
\vdots & \ddots & \vdots \\
\omega_{M_S,1}M_{M_S}(\sigma) & \cdots & \omega_{M_S,M_S}M_{M_S}(\sigma)
\end{bmatrix}
- \begin{bmatrix}
B_1 \\
\vdots \\
B_{M_S}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
\vdots \\
u_{M_S}
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
\vdots \\
f_{M_S}
\end{bmatrix}
- \begin{bmatrix}
g_1 \\
\vdots \\
g_{M_S}
\end{bmatrix}. \tag{5.23}
\]

We solve this system iteratively with a Krylov subspace solver as described in Sec. 3.4.2.

Remark 8. Note that this linear system is identical to the one we would obtain by inserting physical basis functions into the collocated variational formulation (5.13) and writing that formulation in block matrix form as long as the quadrature rule is also chosen identically.

5.2.6 Sparse tensor combination technique DOM

We construct the approximate sparse DOM solution to the RTE with the sparse tensor combination technique (Sec. 2.3) according to the combination formula (2.24) as

\[
\hat{u}_{L,N} := \sum_{i=0}^{L} u_{i,S}^{\max}(i) - \sum_{i=0}^{L-1} u_{i,S}^{\max}(i+1), \tag{5.24}
\]
where \( u_{i, l_{\text{max}}} \in V^{L,N} \) is the approximate full tensor solution to the RTE obtained by solving the discrete variational formulation (5.37) with \( L = i \) and \( N = l_{\text{max}}^{S(i+1)} \). The sparsity profile will be chosen as introduced in Eq. (4.20) such that

\[
l_{S}^{\text{max}}(l_{D}) = \begin{cases} 
2^\left\lceil \log_2(N + 1)/L - l_{D} \right\rceil & \text{if } 2^\left\lceil \log_2(N + 1)/L - l_{D} \right\rceil > N_0, \\
N_0 & \text{else.}
\end{cases}
\]

We will now analyze the complexity of our sparse method with regard to this sparsity profile.

**Complexity**

The first lemma in this subsection bounds the total number of degrees of freedom in the sparse discrete ordinates method.

**Lemma 5.6.** Let the maximum angular index \( l_{S}^{\text{max}} \) be given as in (5.25) and the number of degrees of freedom of the component approximation spaces be given by \( M_{D} = \dim(V_{L}^{D}) \leq c_{D}2^{dL} \) and \( M_{S} = \dim(V_{N}^{S}) \leq c_{S}N^{dS} \). Then the total number of degrees of freedom of the sparse discrete ordinates method is bounded by

\[
\tilde{M}_{L,N,N_0} \lesssim L^{\theta} \max\{2^{dL}, N^{dS}\} + 2^{dL} N_0^{dS},
\]

with \( \theta = 1 \) if \( 2^{dL} = N^{dS} \) and \( \theta = 0 \) otherwise.

**Proof.** The prescribed sparsity profile effectively combines a full subproblem with resolutions \( L \) and \( N_0 \) with the full subproblems from the combination formula. For the estimate of this lemma we simply add the degrees of freedom of the full subproblem, \( M_{\text{full}} \lesssim 2^{dL} N_0^{dS} \), to those of a combination technique solution with \( N_0 = 0 \) denoted by \( M_{\text{sp}} \), ignoring that the full subproblem might actually replace several subproblems from the combination technique.

A logarithmic angular resolution index \( \lambda := \log_2 l_{S}, \Lambda := \log_2 N \), permits us to write \( \dim(V_{S}^{l_{S}(\lambda)}) \leq c_{S}2^{dS} \) so that we have by Lemma 2.5

\[
M_{\text{sp}} \lesssim L^{\theta} \max\{2^{dL}, 2^{dS}\Lambda\} = L^{\theta} \max\{2^{dL}, N^{dS}\},
\]

with \( \theta = 1 \) if \( 2^{dL} = N^{dS} \) and 0 otherwise. The sum of \( M_{\text{full}} \) and \( M_{\text{sp}} \) is the assertion of the lemma.

**Remark 9.** We could even afford to increase \( N_0 \) proportionally to \( L^{1/dS} \) without compromising the goal of reducing the number of degrees of freedom essentially to that of a problem stated on one of the component domains alone.

Next we estimate the maximum memory consumption of the sparse discrete ordinates method. Using a sparse matrix data structure, the maximum memory consumption \( \tilde{m}_{L,N,N_0} \) of the method will be bounded by a multiple of the largest number of nonzero entries \( \text{nnz}(A_{i,l_{S}^{\text{max}}(i)}) \) occurring for any of the matrices \( A_{i,l_{S}^{\text{max}}(i)}, i = 0, \ldots, L \), of the
subproblems in the combination technique since the matrices constitute the objects with the highest memory requirements and since we can solve the subproblems sequentially:

\[ \hat{m}_{L,N,N_0} = c_{\text{mem}} \max_{i=0,...,L} \text{nnz}(A_{i,l_S^{\text{max}}(i)}). \]

The constant \( c_{\text{mem}} \) represents the cost of storing one matrix entry.

**Lemma 5.7.** Given the maximum angular index \( l_S^{\text{max}} \) as in (5.25), the maximum memory consumption by the sparse discrete ordinates method is bounded asymptotically by

\[ \hat{m}_{L,N,N_0} \lesssim \max\{N^{2d_S}, 2^{dL}, 2^{dL}N_0^{2d_S}\}. \]

**Proof.** The number of nonzero entries of the matrix \( A_{l_D,l_S^{\text{max}}(l_D)} \) can be deduced from the left hand side of Eq. (5.23):

\[ \text{nnz}(A_{l_D,l_S^{\text{max}}(l_D)}) \lesssim 2^{d_Dl_D^{\text{max}}(l_D)}2^{dS}, \]

because the single blocks of physical submatrices \( M_{(\sigma)} \) are sparse, but the system matrix will in general be full of such blocks as the scattering operator is nonlocal. Hidden constants depend on the dimension \( d \) and the quality of the mesh, i.e. the maximum number of overlapping supports of physical basis functions.

We first consider the case \( N_0 = 0 \). In this case we have to find the maximum

\[ \max_{l_D=0,...,L} 2^{d_Dl_D^{\text{max}}(l_D)}2^{dS} \frac{\log_2(N+1)}{L(L-1_D)}, \]

which occurs for \( l_D = 0 \) or \( l_D = L \) because the exponent is linear in \( l_D \). Together with the case \( N_0 > 0 \) we obtain

\[ \max_{i=0,...,L} \text{nnz}(A_{i,l_S^{\text{max}}(i)}) \lesssim \max\{N^{2d_S}, 2^{dL}, 2^{dL}N_0^{2d_S}\} \lesssim \max\{M_S^2, M_D, M_DN_0^{2d_S}\}. \]

The maximum memory costs are therefore reduced to those of a purely spatial or angular problem, whereas the full method requires \( m_{L,N} \lesssim 2^{dL}N^{2d_S} \lesssim M_DM_S^2 \).

**Remark 10.** Without scattering, we lose the factor of 2 in the exponent of the angular resolution. However, since problems for different directions decouple, memory requirements can be further reduced to multiples of \( M_D \) for both full and sparse method by solving for different directions independently.

Finally we estimate the amount of computational work required for the evaluation of the discrete operator of the sparse DOM.

**Lemma 5.8.** The number of floating point operations for one evaluation of the discrete operator in the sparse discrete ordinates method is bounded by

\[ \text{nflops}_{L,N,N_0} \lesssim L^\theta \max\{2^{dL}, N^{2d_S}\} + 2^{dL}N_0^{2d_S} \]

with \( \theta = 1 \) if \( 2^{dL} = N^{2d_S} \) and \( \theta = 0 \) otherwise.
5.3 Error analysis

Proof. In a full subproblem of the combination technique, the number of operations is proportional to the number of nonzero entries of the matrix:

\[ \text{nfllops}_{l_D,l_S}^\max(I_D) \lesssim \text{nnz} (A_{l_D,l_S}^\max(I_D)) \lesssim 2^{dl_D} l_S^\max(I_D)^{2d_S}. \]

We first consider the case \( N_0 = 0 \). Again, by introducing a logarithmic angular index \( \lambda := \log_2 l_S, \Lambda := \log_2 N \), the maximum angular index becomes \( \lambda_{\max} = \Lambda - \Lambda l_D / L \). Summation of the numbers of operations of one matrix-vector multiplication over all subproblems yields

\[ \text{nfllops}_{l_D,l_S}^\max(I_D) = \text{nfllops}_{l_D,\lambda_{\max}}(I_D) \lesssim \sum_{l_D=0}^L 2^{dl_D} 2^{2d_S(\Lambda - \Lambda l_D / L)}. \]

This sum can directly be estimated by Lemma 2.2. We therefore obtain

\[ \text{nfllops}_{l_D,l_S}^\max(I_D) \lesssim L^\theta \max\{2^{dl_L}, 2^{2d_S\Lambda}\} \lesssim L^\theta \max\{2^{dl_L}, N^{2d_S}\}, \]

with \( \theta = 1 \) if \( 2^{dl_L} = N^{2d_S} \) and zero otherwise.

If \( N_0 > 0 \), the full subproblem of resolution \( L \) and \( N_0 \) requires \( \text{nfllops}_{\text{full}} = 2^{dl_L} N_0^{2d_S} \) operations. In total, we get the bound

\[ \text{nfllops}_{L,N_0,N_0} \lesssim L^\theta \max\{2^{dl_L}, N^{2d_S}\} + 2^{dl_L} N_0^{2d_S} \lesssim \log(M_D)^\theta \max\{M_D, M_S^2\} + M_D N_0^{2d_S}. \]

Thus, together with an ideal preconditioner so that the number of iterations in the linear solver remains independent of the resolution, the total amount of computational work reduces essentially to that of a problem on a component domain.

Remark 11. Again, without scattering, or if we applied the combination technique to a method for the RTP which results in a block-banded scattering matrix (e.g. \( P_N \)-method), we could also lose the factor of 2 in the exponent of the angular resolution.

5.3 Error analysis

In this section we analyze the convergence properties of the DOM. After stating properties of approximation on the component domains, we estimate the error of the standard full tensor DOM. Then we show how the error introduced by quadrature for the scattering integral modifies the error estimate for the full DOM. This motivates the adapted sparsity profile with minimum angular resolution \( N_0 \) from (4.20), which we will respect in the final proof of convergence of the sparse tensor DOM.

The relevant bilinear form of this section is \( a_{\supg}(\cdot, \cdot) \) from (5.6), we may therefore refer to it by \( a(\cdot, \cdot) \) in short.
5 Discrete ordinates method

5.3.1 Galerkin projection

First, we define the Galerkin projector into the full tensor product approximation space \( V^{L,N} = V^L_D \otimes V^N_S \) in the usual way via orthogonality of its residual with respect to the bilinear form \( a(\cdot, \cdot) \).

Definition 5.9 (Galerkin projector). The Galerkin projector \( P^{L,N} : V_1 \to V^{L,N} \) is defined by
\[
a(P^{L,N}u, v) = a(u, v) \quad \forall v \in V^{L,N}.
\]

Stability of the Galerkin projector is asserted by the following lemma.

Lemma 5.10 (Stability of the Galerkin projector). Let \( v \in V_1 \). Then there is a constant \( c_P > 0 \) independent of \( L \) and \( N \) so that
\[
\| P^{L,N}v \|_1 \leq c_P \| v \|_1.
\]

Proof. With continuity (Lemma 5.1) of the bilinear form we obtain
\[
|a(P^{L,N}v, v_{L,N})| = |a(v, v_{L,N})| \leq c_e \| v \|_1 \| v_{L,N} \|_1 \quad \forall v_{L,N} \in V^{L,N}.
\]
Since this holds for all \( v_{L,N} \in V^{L,N} \), we can set \( v_{L,N} = P^{L,N}v \) and exploit coercivity of the bilinear form (Lemma 5.3):
\[
c_e \| P^{L,N}v \|_1^2 \leq |a(P^{L,N}v, P^{L,N}v)| = |a(v, P^{L,N}v)| \leq c_e \| v \|_1 \| P^{L,N}v \|_1.
\]
If \( P^{L,N}v \neq 0 \) we obtain the result with \( c_P = c_e / c_e \) by dividing by \( c_e \) and \( \| P^{L,N}v \|_1 \). \( \square \)

In complete analogy to Def. 5.9 we can define the Galerkin projector \( \tilde{P}^{L,\tilde{N}} \) onto the nested full tensor approximation space \( V^{L,\tilde{N}}_{\text{nest}} \). Its stability with respect to the \( \| \cdot \|_1 \)-norm follows exactly in the same way as for the Galerkin projector \( P^{L,\tilde{N}} \).

5.3.2 Error estimates on the physical domain

Galerkin projection error

Based on the Galerkin projector from Def. 5.9, a semidiscrete Galerkin projector on the physical domain can be defined as follows.

Definition 5.11 (Galerkin projector on the physical domain). The semidiscrete Galerkin projector on the physical domain \( P^L_D : V_1 \to V^L_D \otimes L^2(S) \) is defined by
\[
a(P^L_D u, v) = a(u, v) \quad \forall v \in V^L_D \otimes L^2(S).
\]

It enjoys the same stability properties as the Galerkin projector \( P^{L,N} \).

Lemma 5.12 (Stability of the Galerkin projector on the physical domain). There is a constant \( c_P > 0 \) independent of \( L \) so that for all \( v \in V_1 \)
\[
\| P^L_D v \|_1 \leq c_P \| v \|_1.
\]

68
Next we derive an error estimate for the Galerkin approximation on the physical domain. At this point, as the approximation is semidiscrete, the angular part of the Galerkin projected solution is exact.

**Lemma 5.13** (Error estimate for Galerkin projection on physical domain). Let \( u \in H^{s+1,0}(\Omega), s \in \{0, 1\} \), be the exact solution to problem (5.11) and \( u_L := P^L_D u \in V^L_D \otimes L^2(S) \) the Galerkin projected solution to

\[
a_{\text{supg}}(u_L, v_L) = l_{\text{supg}}(v_L) \quad \forall v_L \in V^L_D \otimes L^2(S)
\]

with \( a_{\text{supg}}(\cdot, \cdot) \) from (5.6) and \( l_{\text{supg}}(\cdot) \) from (5.7). Then, there is a constant \( c_p > 0 \) independent of \( L \) such that

\[
\|u - u_L\|_1 \leq c_p 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]

**Proof.** We follow Ávila et al. (2011, Lemma 3 and Theorem 1). After inserting the quasi-interpolated solution \( \hat{u}_L := (P^L_D \otimes \text{Id}_S)u \) with \( P^L_D \) from Sec. 4.3.1 the triangle inequality permits us to write

\[
\|u - u_L\|_1 \leq \|u - \hat{u}_L\|_1 + \|\hat{u}_L - u_L\|_1.
\]

For the first part, we use the fact that there is a constant \( c_n > 0 \) for all \( v \in H^1(D) \otimes L^2(S) \) such that

\[
\|v\|_1 \leq c_n \|v\|_{H^{1,0}(\Omega)}.
\]

Thus, we can apply Lemma 4.4:

\[
\|u - \hat{u}_L\|_1 \leq c_n \|u - \hat{u}_L\|_{H^{1,0}(\Omega)} \leq c_n c_H 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]

For the second part in (5.27), we use coercivity of the bilinear form, then in a second step Galerkin orthogonality, and finally continuity of the bilinear form to write

\[
\|u_L - \hat{u}_L\|_1^2 \leq c_c^{-1} a(u_L - \hat{u}_L, u_L - \hat{u}_L) \leq c_c^{-1} a(u - \hat{u}_L, u_L - \hat{u}_L)
\]

\[
\leq c_c c_c^{-1} \|u - \hat{u}_L\|_1 \|u_L - \hat{u}_L\|_1
\]

\[
\leq c_c c_c^{-1} c_n \|u - \hat{u}_L\|_{H^{1,0}(\Omega)} \|u_L - \hat{u}_L\|_1,
\]

and therefore with Lemma 4.4

\[
\|u_L - \hat{u}_L\|_1 \leq c_c c_c^{-1} c_n c_H 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]

By inserting into (5.27) we arrive at the result

\[
\|u - u_L\|_1 \leq c_n c_H (1 + c_c c_c^{-1}) 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]

\[
\Box
\]

### 5.3.3 Error estimates on the angular domain

#### \( L^2 \)-projection error

Let \( \mathcal{T}^N_S := \{ S_j \}_{j=1}^{M_S} \) the family of possibly nonnested (cp. Remark 6) partitions of the angular domain from Sec. 5.2.3, for which the number of elements \( M_S \) is related to
the angular resolution parameter $N$ by $M_S = O(N^{d_S})$, as given by relation (3.35). For the purpose of further analysis, we assume that the family of partitions contains only elements of similar size, i.e. it is quasi-uniform. For quasi-uniform partitions the ratio of the radius of the largest inscribed circle $r_{\text{max}}$ to the radius of the smallest inscribed circle $r_{\text{min}}$ must be bounded:

$$1 \leq \frac{r_{\text{max}}}{r_{\text{min}}} \leq c_{\text{qu}} < \infty.$$ 

As the number of elements $M_S$ is proportional to $N^{d_S}$, we obtain the estimates

$$r_{\text{max}} \leq \bar{c}_{\text{qu}} N^{-1}, \quad r_{\text{min}} \geq c_{\text{qu}} N^{-1}. \quad \text{(5.28)}$$

The occurring constants $c_{\text{qu}}, \bar{c}_{\text{qu}},$ and $\tilde{c}_{\text{qu}}$ are all greater than zero and depend only on the geometry of the partition, but not on $N$.

Also, we require the partition to be shape-regular, which means that no element can degenerate to zero volume. Our definition relates the radius of the circumscribed circle $R_j$ of each element to the radius of its inscribed circle $r_j$: For every $N \in \mathbb{N}$, there exists a constant $c_{sr} > 0$ depending only on the geometry on the partition, but not on $N$ such that for all elements $S_j \in T^N_S$ it holds

$$0 < c_{sr}^{-1} \leq \frac{R_j}{r_j} \leq c_{sr} < \infty. \quad \text{(5.29)}$$

If we define the mesh width of the partition as $h_j := \text{diam}(S_j)$, then $r_j \leq h_j \leq R_j$. With $j^* := \arg\max_{j \in \{1, \ldots, M_S\}} R_j$ and inequalities (5.28) and (5.29), we can estimate the maximum mesh width by

$$h_{\text{max}} \leq R_{\text{max}} \leq c_{sr} r_{j^*} \leq c_{sr} \bar{c}_{\text{qu}} N^{-1}. \quad \text{(5.30)}$$

This estimate permits to express the approximation rate of $L^2$-projection on the angular domain in terms of the angular resolution $N$.

**Lemma 5.14.** For functions $v \in H^t(S)$, $t \in \{0, 1\}$, the $L^2$-projection to the space of characteristic functions on the partition $T^N_S$ with maximum mesh width $h_{\text{max}} \lesssim N^{-1}$ satisfies the error estimate

$$\|v - P^N_{L^2(S)} v\|_{L^2(S)} \leq c_l N^{-1} \|v\|_{H^t(S)}, \quad \text{(5.31)}$$

where the constant $c_l > 0$ depends on the partition geometry, but is independent of $N$.

For nested partitions $T^N_S$, the relation between the angular number of degrees of freedom and the resolution is $M_S = O(2^{d_S N})$ so that $h_{\text{max}} \lesssim 2^{-N}$. The corresponding statement to (5.31) for the $L^2$-projection $\tilde{P}^N_{L^2}$ onto the nested angular approximation space $\tilde{V}^N_S$ therefore reads

$$\|v - \tilde{P}^N_{L^2(S)} v\|_{L^2(S)} \leq c_l 2^{-tN} \|v\|_{H^t(S)}. \quad \text{(5.32)}$$
5.3 Error analysis

Galerkin projection error

The semidiscrete angular Galerkin projector projects the angular part of the solution $u \in V_1$ to the finite dimensional space $V^N_S$.

**Definition 5.15 (Angular Galerkin projector).** The semidiscrete angular Galerkin projector $P^N_S : V_1 \to V_D \otimes V^N_S$ is defined by

$$a(P^N_S u, v) = a(u, v) \quad \forall v \in V_D \otimes V^N_S.$$ 

Stability with respect to the $\| \cdot \|_1$-norm is obtained as in Lemma 5.10. The next lemma asserts approximation properties of the angular Galerkin projector.

**Lemma 5.16 (Error estimate for angular Galerkin projection).** Let $u \in H^{1,t}(\Omega)$, $t \in \{0,1\}$, be the exact solution to problem (5.11) and $u_N := P^N_S u \in V_D \otimes V^N_S$ the Galerkin projected solution with angular part from the subspace $V^N_S$ of $L^2(S)$. Then there is a constant $c_a > 0$ independent of $N$ such that

$$\| u - u_N \|_1 \leq c_a N^{-t} \| u \|_{H^{1,t}(\Omega)}.$$ 

**Proof.** The proof proceeds analogously to the one of Lemma 5.13 while substituting the $L^2$-projected solution with Lemma 5.31 for the quasi-interpolated solution, the details are therefore omitted here. 

Quadrature error

We estimate the quadrature error of the composite midpoint rule (Def. 5.5) on the angular domain. The following lemma is stated in a dimension independent form to cover both circle and sphere, but note that on the circle exponential convergence can be obtained for periodic functions.

**Lemma 5.17 (Quadrature error of composite midpoint rule).** If $f \in C^2(S)$ and $\mathcal{T}^N_S$ is a family of quasi-uniform shape-regular partitions of the angular domain with angular resolution $N$ related to the number $M_S$ of partition elements by (3.35), then there exists a constant $c_{quad} > 0$ depending on the geometry of the partition and the angular domain, but independent of $M_S$ or $N$ so that for all $N \in \mathbb{N}$

$$| \int_S f(s) \, ds - Q^{(M_S)}[f] | \leq c_{quad} M_S^{-1} \max_{s \in S} \{ \sigma(D^2_s f(s)) \},$$

where $D^2_s f(s)$ is the Hessian of $f$ evaluated at $s$ and $\sigma$ is the maximum singular value of this Hessian.

**Proof.** We can estimate the global error by the error on the elements:

$$| \int_S f(s) \, ds - \sum_{j=1}^{M_S} w_j f(s_j) | \leq \sum_{j=1}^{M_S} | \int_{S_j} f(s) \, ds - w_j f(s_j) |.$$
5 Discrete ordinates method

For each element $S_j$, we expand $f$ by a Taylor series around $s_j$:

$$
\int_{S_j} f(s) \, ds = \int_{S_j} \left( f(s_j) + \nabla_s f|_{s=s_j}(s - s_j) + \frac{1}{2}(s - s_j)^T D_s^2 f|_{s=s_j}(s - s_j) \right) \, ds,
$$

where $s^* \in S_j$ is some intermediate angle. For the middle term, we obtain

$$
\int_{S_j} \nabla_s f|_{s=s_j}(s - s_j) \, ds = \nabla_s f|_{s=s_j} \left( \int_{S_j} s \, ds - \int_{S_j} s_j \, ds \right) = \nabla_s f|_{s=s_j} |S_j| = 0
$$
due to the definition of the barycenter. The last term can be estimated by the radius $R_j$ of the circumcircle of $S_j$:

$$
\int_{S_j} \frac{1}{2}(s - s_j)^T D_s^2 f|_{s=s_j} (s - s_j) \, ds \leq \frac{1}{2} \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} \int_{S_j} (s - s_j)^2 \, ds \\
\leq 2 \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} |S_j| R_j^2.
$$

If we subtract $|S_j| f(s_j)$ from expansion (5.33), the first term is canceled. This means for the error we retain only (5.34). From quasi-uniformity and shape-regularity of the partition we obtain $R_j \leq c_{sr} \varepsilon_0 \varepsilon_d^3 M_S^{-1/2}$ for every $j$, since $M_S \leq c_d N^{d_s}$. Summing up over the elements $S_j$ yields

$$
| \int_{S} f(s) \, ds - \sum_{j=1}^{M_S} w_j f(s_j) | \leq \sum_{j=1}^{M_S} \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} |S_j| R_j^2 \\
\leq \sum_{j=1}^{M_S} 2 \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} \pi R_j^4 \\
\leq 2 \pi \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} M_S c_{sr} \varepsilon_0 \varepsilon_d^3 c_{sr} \varepsilon_d^2 M_S^{-2} \\
\leq 2 \pi \max_{s \in S_j} \{ \bar{\sigma}(D_s^2 f) \} c_{sr} \varepsilon_0 \varepsilon_d^2 c_{sr} \varepsilon_d M_S^{-1}. \quad \square
$$

5.3.4 Error estimate for the full tensor phase space Galerkin DOM

The following theorem gives an error estimate for the full tensor approximation on nonnested approximation spaces. Possible consistency errors from the Nyström method for the scattering operator will be taken into account in the next section.

**Theorem 5.18** (Error estimate full tensor exact variational formulation). The full tensor discrete ordinates approximation $u_{L,N} = P_{L,N} u$ of a solution $u \in H^{s+1,0}(\Omega) \cap H^{1,t}(\Omega)$, $s \in \{0,1\}$, $t \in \{0,1\}$, to the variational problem (5.11) satisfies the asymptotic error estimate

$$
\| u - u_{L,N} \|_1 \lesssim 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + N^{-t} \| u \|_{H^{1,t}(\Omega)},
$$

with relation “$\lesssim$” as in Def. 2.1.
5.3 Error analysis

Proof. By Céa’s Lemma (Brenner and Scott, 2008, Thm. 2.8.1) the Galerkin approximation is quasi-optimal in $V_{L^2}$, its error can therefore be bounded (up to constants) by the error of any other approximation to $u$ in $V_{L^2}$, for example the quasi-interpolated and $L^2$-projected approximation $P^l \otimes P^N_{L^2} u$:

$$\left\| u - P^l \otimes P^N_{L^2} u \right\|_{1} \leq \left\| u - P^l \otimes \text{Id} u \right\|_{1} + \left\| P^l \otimes \text{Id} u - P^l \otimes P^N_{L^2} u \right\|_{1} \leq 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + \left\| \text{Id} \otimes P^N_{L^2} P^l \otimes \text{Id} u \right\|_{1} \leq 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + N^{-t} \left\| P^l \otimes \text{Id} u \right\|_{H^{1,1}(\Omega)} \leq 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + N^{-t} \| u \|_{H^{1,1}(\Omega)}.$$  

Here, we used the approximation properties of the quasi-interpolant from Lemma 4.4 and of the angular $L^2$-projection from Lemma 5.14. The last step follows by stability of the quasi-interpolation in the $H^1(D)$-norm (Lemma 4.5).

The corresponding result to (5.35) for the nested approximation spaces reads

$$\left\| u - u_{L,N} \right\|_{1} \leq 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + 2^{-tN} \| u \|_{H^{1,1}(\Omega)}.$$  

5.3.5 Consistency error due to quadrature in the scattering operator

In the previous sections, we analyzed the error of the discrete solution $u_{L,N}$ of the exact bilinear form $a_{\sup}(\cdot, \cdot)$. However, for the implementation all integrals have to be replaced by quadrature rules. Here we investigate the effect of the Nyström method for the scattering operator, in which the scattering integral is approximated by a quadrature rule. The errors due to quadrature for the integrals of the inner product $(\cdot, \cdot)_{L^2(\Omega)}$ in the bilinear form are neglected. These errors will be small if quadrature rules which are exact for the chosen basis functions are used and the coefficient functions $\kappa$ and $\sigma$ do not vary too much over an element of the spatial triangulation or angular partition.

Introducing a quadrature rule for the scattering integral can be regarded as replacing the scattering integral operator $\Sigma$ by a discretized scattering integral operator

$$\tilde{\Sigma} v := \sum_{m=1}^{M_S} w_m \Phi(s, s_m) v(s, s_m)$$  

for a function $v \in V_1$ with weights $w_m$ and nodes $s_m$ of a quadrature rule. Consequently, instead of solving the discrete variational problem with exact bilinear form for $u_{L,N}$,

$$a_{\sup}(u_{L,N}, v_{L,N}) = l_{\sup}(v_{L,N}) \quad \forall v_{L,N} \in V_{L^2},$$  

we solve a discrete variational formulation with inexact bilinear form: Find $\tilde{u}_{L,N} \in V_{L^2}$ such that

$$a_N(\tilde{u}_{L,N}, v_{L,N}) = l_{\sup}(v_{L,N}) \quad \forall v_{L,N} \in V_{L^2}.$$  

73
The inexact bilinear form in this expression is given for \( u, v \in V_1 \) by

\[
a_N(u, v) := (R_{supg} v, s \cdot \nabla_x u + (\kappa + \sigma) u + \sigma \Sigma u)_{L^2(\Omega)} - 2b(u, v).
\]

(5.39)

The following lemma after Strang’s first lemma (e.g. Braess, 2007, Ch. III, Lemma 1.1) bounds the error to the exact formulation incurred by this inconsistency.

**Lemma 5.19 (Error estimate for inexact variational formulation).** Let \( u \in V_1 \) be the exact solution to problem (5.11), \( u_{L,N} \in V^{L,N} \) the solution to problem (5.37), and \( u_{L,N} \in V^{L,N} \) the solution to problem (5.38). Then there is a constant \( c > 0 \) independent of \( L, N \) such that

\[
\| u - u_{L,N} \|_1 \leq c (\| u - u_{L,N} \|_1 + \| E^{(N)}[\Phi(\cdot, s') u_{L,N}(\cdot, s')] \|_{L^2(\Omega)}).
\]

where

\[
E^{(N)}[f(s')] := \left| \int_S f(s') \, ds' - \sum_{m=1}^{M_S} w_m f(s_m) \right|
\]

is the quadrature error of the quadrature rule (5.12) employed for the scattering integral.

**Proof.** We start by triangle inequality:

\[
\| u - u_{L,N} \|_1 \leq \| u - u_{L,N} \|_1 + \| u_{L,N} - u_{L,N} \|_1
\]

The second term is estimated using coercivity of the exact bilinear form:

\[
\| u_{L,N} - u_{L,N} \|_1^2 \leq \frac{1}{c_e} a_{supg}(u_{L,N} - u_{L,N}, u_{L,N} - u_{L,N})
\]

We introduce the abbreviation \( v_{L,N} = u_{L,N} - u_{L,N} \) and insert a zero:

\[
\| v_{L,N} \|_1^2 \leq \frac{1}{c_e} a_{supg}(u_{L,N} - u_{L,N}, v_{L,N}) - l_{supg}(v_{L,N}) + l_{supg}(v_{L,N})
\]

We substitute the first occurrence of the right hand side linear functional according to (5.37) and the second according to (5.38):

\[
\| v_{L,N} \|_1^2 \leq \frac{1}{c_e} a_{supg}(u_{L,N} - u_{L,N}, v_{L,N}) - a_{supg}(u_{L,N}, v_{L,N}) + a_N(u_{L,N}, v_{L,N})
\]

\[
= \frac{1}{c_e} a_N(u_{L,N}, v_{L,N}) - a_{supg}(u_{L,N}, v_{L,N})
\]

This expression is the consistency error between the exact and inexact bilinear form. It can be estimated further:

\[
\| v_{L,N} \|_1^2 \leq \frac{1}{c_e} \left| (R_{supg} v_{L,N}, \sigma(\Sigma - \Sigma)u_{L,N})_{L^2(\Omega)} \right|
\]

\[
= \frac{1}{c_e} \left| (R_{supg} v_{L,N}, \sigma E^{(N)}[\Phi(s, s') u_{L,N}(x, s')] \right|_{L^2(\Omega)}
\]

\[
\leq \frac{1}{c_e} \max\{2, \delta_{max}\} \sigma \max \| v_{L,N} \|_1 \left| E^{(N)}[\Phi(s, s') u_{L,N}(x, s')] \right|_{L^2(\Omega)}.
\]
5.3 Error analysis

In conclusion we obtain

\[ \|u - \overline{u}_{L,N}\|_1 \leq c(\|u - u_{L,N}\|_1 + \left\| E^{(N)}[\Phi(s,s')\overline{u}_{L,N}(x,s')] \right\|_{L^2(\Omega)}) , \]

where \( c = \max\{1, 2\sigma_{\text{max}}/c_{\varepsilon}, \delta_{\text{max}} \sigma_{\text{max}}/c_{\varepsilon}\} . \)

\[ \square \]

For the choice of the composite midpoint rule we arrive at the following estimate.

**Theorem 5.20** (Error estimate for inexact variational formulation with midpoint rule). Let \( \overline{u}_{L,N} \in V^{L,N} \) be the solution to problem (5.38) with the composite midpoint rule from Def. 5.5 and \( u \in H^{s+1,0}(\Omega) \cap H^{1,1}(\Omega), s \in \{0, 1\}, t \in \{0, 1\} \), the solution to problem (5.11). Then it holds

\[ \|u - \overline{u}_{L,N}\|_1 \lesssim 2^{-sL}\|u\|_{H^{s+1,0}(\Omega)} + N^{-t}\|u\|_{H^{1,1}(\Omega)} + N^{-2}\max\{\sigma(D^2_s(\Phi(\cdot, s')\overline{u}_{L,N}(\cdot, s'))\} \right\|_{L^2(\Omega)} . \]

The relation “\( \lesssim \)” is as in Def. 2.1.

**Proof.** The statement follows by estimating \( \|u - u_{L,N}\|_1 \) in Lemma 5.19 by Thm. 5.18 and inserting the error estimate for the midpoint rule from Lemma 5.17 as the quadrature error in Lemma 5.19.

\[ \square \]

For the choice of characteristic functions on the angular partition \( T^N_s \) as angular basis, we can reformulate the last term in the estimate of Thm. 5.20 slightly.

**Corollary 5.21** (Error estimate for inexact variational formulation with midpoint rule and characteristic functions in angle). Let \( \overline{u}_{L,N} \in V^{L,N} = (V^D_s \otimes S^{-1,0}(S, T^N_s)) \) be the solution to problem (5.38) with the composite midpoint rule from Def. 5.5 and \( u \in H^{s+1,0}(\Omega) \cap H^{1,1}(\Omega), s \in \{0, 1\}, t \in \{0, 1\} \), the solution to problem (5.11). Then it holds

\[ \|u - \overline{u}_{L,N}\|_1 \lesssim 2^{-sL}\|u\|_{H^{s+1,0}(\Omega)} + N^{-t}\|u\|_{H^{1,1}(\Omega)} + N^{-2}\max\{\sigma(D^2_s(\Phi(\cdot, s')\overline{u}_{L,N}(\cdot, s'))\} \right\|_{L^2(\Omega)} \|\overline{u}_{L,N}\|_{L^2(D; L^\infty(S))} . \]

**Proof.** We revisit the proof of Lemma 5.17 for a bound on the quadrature error of the composite midpoint rule. There, we estimated in (5.34)

\[ \int_{S_j} \frac{1}{2}(s - s_j)^T D^2_s f|_{s=s_j}(s - s_j) ds \]

with \( f \) being the integrand function. Inserting \( f(s') = \Phi(s, s')\overline{u}_{L,N}(x, s') \) we continue from there. Then, as \( \overline{u}_{L,N} \) is constant over \( S_j \), we have

\[ D^2_s(\Phi(s, s')\overline{u}_{L,N}(x, s'))|_{x'=s'} = D^2_s(\Phi(s, s'))|_{x'=s'} \overline{u}_{L,N}(x, s'), \quad s^* \in S_j . \]
Hence, we can pull $\pi_{L,N}$ out of the integral:
\[
\int_{S_j} \frac{1}{2} (s' - s_j) \top D^2_{s'}(\Phi(s,s')\pi_{L,N}(x,s'))|_{s'=s_j} (s' - s_j) \, ds'
\]
\[
= \pi_{L,N}(x,s^*) \int_{S_j} \frac{1}{2} (s' - s_j) \top D^2_{s'}(\Phi(s,s'))|_{s'=s_j} (s' - s_j) \, ds'
\]
\[
\leq \pi_{L,N}(x,s^*) \int_{S_j} \frac{1}{2} (s' - s_j) \top \tilde{\sigma}(D^2_{s'}(\Phi(s,s')))|_{s'=s_j} (s' - s_j) \, ds'
\]
\[
\leq \pi_{L,N}(x,s_j) \frac{1}{2} \max_{s' \in S_j} \{ \tilde{\sigma}(D^2_{s'}(\Phi(s,s'))) \} |S_j|h^2_{\max}.
\]

Summing over the elements of the partition and applying (5.30) yields for the quadrature error
\[
E^{(N)}[\Phi(s,s')\pi_{L,N}(x,s)] \leq \sum_{j=1}^{M_S} \pi_{L,N}(x,s_j) \frac{1}{2} \max_{s' \in S_j} \{ \tilde{\sigma}(D^2_{s'}(\Phi(s,s'))) \} |S_j|h^2_{\max}
\]
\[
\leq \frac{1}{2} c_{sr}^2 s^2 N^{-2} \max_{s' \in S} \{ \tilde{\sigma}(D^2_{s'}(\Phi(s,s'))) \} \| \pi_{L,N}(x,\cdot) \|_{L^\infty(S)}.
\]

By inserting this result into Lemma 5.19 and using Thm. 5.18, we obtain
\[
\| u - \pi_{L,N} \|_1 \leq 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + N^{-l} \| u \|_{H^{1,0}(\Omega)}
\]
\[
+ N^{-2} \left\| \max_{s' \in S} \{ \tilde{\sigma}(D^2_{s'}(\Phi(s,s'))) \} \| \pi_{L,N}(x,\cdot) \|_{L^\infty(S)} \right\|_{L^2(\Omega)}
\]
\[
= 2^{-sL} \| u \|_{H^{s+1,0}(\Omega)} + N^{-l} \| u \|_{H^{1,0}(\Omega)}
\]
\[
+ N^{-2} \left\| \max_{s' \in S} \{ \tilde{\sigma}(D^2_{s'}(\Phi(\cdot,s'))) \} \right\|_{L^2(S)} \| \pi_{L,N} \|_{L^\infty(S)} \right\|_{L^2(D)}.
\]

As long as $\pi_{L,N}$ remains bounded with respect to $L$ and $N$, we can deduce that if we choose the composite midpoint rule on a partition of the angular domain as quadrature rule for the scattering integral, the consistency error converges faster than the angular contribution to the approximation error. However, for scattering kernels with large second derivatives (e.g. highly oscillating), the consistency error may be the dominant error up to large values of $N$. This is due to the fact that integrating oscillating functions by the composite midpoint rule requires at least a few quadrature nodes per wavelength for reasonable results. To ensure accurate integration of such functions it might be necessary to maintain a minimum angular resolution $N_0$ when solving full tensor subproblems in the combination technique. In the next section, we therefore modify the sparsity profile accordingly.

### 5.3.6 Error estimate for the sparse tensor phase space Galerkin DOM

Here we analyze the convergence properties of a direct sparse tensor approximation on the sparse tensor product space $\tilde{V}_{nest}^{L,N}$ from (5.16) constructed from tensor products
of nested sequences of approximation spaces over the component domains. Then, the
direct sparse tensor solution can be represented as the sparse Galerkin projection of
the exact solution \( \hat{u}_{L,N} = \hat{P}_{nest}^{L,N} u \), with the sparse Galerkin projector defined by

\[
a(\hat{P}_{nest}^{L,N} u, v) = a(u, v) \quad \forall v \in \hat{V}_{nest}^{L,N}.
\]

The following theorem estimates the error of such a direct sparse solution. The
sparsity profile will include a minimum angular resolution \( \bar{N}_0 \) to ensure that the
consistency error due to quadrature in the angular domain is small. As the boundary
conditions are included in the bilinear form, they are respected in this estimate, which
is in contrast to the corresponding Thm. 4.8 for the sparse spherical harmonics method.
We also refer to Griebel and Harbrecht (2013a, Thms. 4.3 and 7.1) for similar sparse
approximation results.

**Theorem 5.22 (Error estimate of direct sparse tensor solution).** Assume that a sparsity
profile with

\[
l_S^{max}(l_D) = \begin{cases} \bar{N}/L(L - l_D) & \text{if } \bar{N}/L(L - l_D) > \bar{N}_0, \\ \bar{N}_0 & \text{else,} \end{cases}
\]

is given and \( \bar{N}_0 \) is chosen such that the consistency error due to quadrature for the scattering
integral is negligible. Assume further that \( L \) and \( \bar{N} \) vary such that \( -sL + t\bar{N} = \bar{z} = \text{const} \),
then the direct sparse tensor discrete ordinates approximation \( \hat{u}_{L,N} \) of a function \( u \in H^{s+1,t}(\Omega) \),
\( s, t \in \{0, 1\} \), satisfies the error estimate

\[
\| u - \hat{u}_{L,N} \|_1 \lesssim L(2^{-sL} + 2^{-t\bar{N}}) \| u \|_{H^{s+1,t}(\Omega)}.
\]

**Proof.** The proof is virtually identical to that of Thm. 4.8, therefore we only outline the
differences here.

As a first step, we exploit the quasi-optimality of the Galerkin approximation on the
sparse tensor product space to replace the Galerkin approximation error by the error
of the quasi-interpolated and \( L^2 \)-projected approximation. Additionally applying the
norm estimate \( \| \varphi \|_1 \approx \| \varphi \|_{H^{1,0}(\Omega)} \) yields

\[
\| u - \hat{u}_{L,N} \|_1 \lesssim \left\| u - \sum_{l_D=0}^{L} \sum_{l_S=0}^{l_S^{max}(l_D)} \Delta^{l_D}_1 \otimes \tilde{\Delta}^{l_S}_S \right\|_{H^{1,0}(\Omega)}.
\]

Here, \( \Delta^{l_D}_1 := P^{l_D}_1 - P^{l_D-1}_1 \) and \( \tilde{\Delta}^{l_S}_S := \tilde{P}^{l_S}_S - \tilde{P}^{l_S-1}_S \) are the difference operators between
two consecutive resolution levels with the convention \( P^{0}_1 = 0 = \tilde{P}^{0}_S \). They project to
the detail spaces \( \tilde{W}^{l_D}_D \) and \( \tilde{W}^{l_S}_S \), respectively. From (5.41), we proceed exactly as in the
proof of Thm. 4.8 simply by substituting \( \bar{N} \) in all occurrences of \( \log_2(N) \) and \( \bar{N}_0 \) in all
occurrences of \( \log_2(N_0) \) in the proof of Thm. 4.8. Instead of Lemma 4.6, Lemma 5.14 in
the form of estimate (5.32) is now used in the argumentation.
5.3.7 Error estimate for the combination technique DOM

For the proof of the convergence of the combination technique solution we refer to the recent work by Griebel and Harbrecht (2013b) for elliptic differential operators. The applicability of their proof to the variational formulation (5.11) of the radiative transfer problem is to date open. The pivotal point in the proof is the following assumption about the semidiscrete Galerkin projectors.

Assumption 5.23. The semidiscrete Galerkin projectors satisfy the extended stability estimates

\[ \| P^L_D u \|_{H^{1,l}(\Omega)} \lesssim \| u \|_{H^{1,l}(\Omega)}, \quad \| P^N_S u \|_{H^{1+s,0}(\Omega)} \lesssim \| u \|_{H^{1+s,0}(\Omega)} \]

for all \( L, N \in \mathbb{N} \) with \( s, t \in \{0,1\} \) for functions \( u \in H^{1,l}(\Omega) \) or \( H^{1+s,0}(\Omega) \), respectively.

If this assumption is satisfied, we have the following theorem.

Theorem 5.24 (Error estimate of combination technique solution). Assume that a sparsity profile with

\[ I^\text{max}_S(l_D) = \begin{cases} 2^{\lfloor \log_2(N+1) \rfloor / L(L-l_D)} & \text{if } 2^{\lfloor \log_2(N+1) \rfloor / L(L-l_D)} > N_0, \\ N_0 & \text{else,} \end{cases} \]

is given and \( N_0 \) is chosen such that the consistency error due to quadrature for the scattering integral is negligible. Assume further that Assumption 5.23 holds for \( P^L_D \) and \( P^N_S \). Let \( L \) and \( N \) vary such that \(-s+t \lfloor \log_2(N+1) \rfloor / L = \xi = \text{const}\), then the combination technique discrete ordinates approximation \( \tilde{u}_{L,N} \) of a function \( u \in H^{s+1,l}(\Omega) \), \( s, t \in \{0,1\} \), satisfies the error estimate

\[ \| u - \tilde{u}_{L,N} \|_1 \lesssim L(N^{-t} + 2^{-sL}) \| u \|_{H^{s+1,l}(\Omega)}. \]

Proof. Our energy space is the function space \( \mathcal{V}_1 \), the spaces of higher regularity are accordingly given by

\[ \mathcal{H}^{s,t}_{\text{mix}} := \{ v \in \mathcal{V}_1 : \sum_{0 \leq |\alpha| \leq s} \sum_{0 \leq |\beta| \leq t} \| D^\alpha \mathcal{D}_x^\beta v \|_1^2 < \infty \}. \]

It holds \( H^{s+1,l}(\Omega) \subset H^{s,t}_{\text{mix}} \subset H^{s,l}(\Omega) \).

Even though nested approximation spaces on the component domains are assumed in the work by Griebel and Harbrecht (2013b), this is not required in the proof. The identities \( P^{l_D}_D = P^{l_D}_D P^D_D \) and \( P^{l_D}_D = P^{l_D}_D P^D_S \) used in Lemma 1 of the reference can also be exploited in the nonnested case. With Assumption 5.23, Lemma 1 holds. Lemma 2 and Theorem 2 by Griebel and Harbrecht (2013b) then follow in a straightforward way also for the radiative transfer problem with the \( L^2 \)-projectors of the reference replaced by quasi-interpolation operator \( P^L_D \) and \( L^2 \)-projector \( P^S_L \), respectively.

To support the statement of this theorem we refer to the numerical experiments of Chapter 8.
6 Randomized sparse discrete ordinates method

As a third sparse method for radiative transfer, we present a randomized version of the discrete ordinates method.

By interpreting the angular domain as a probability space with uniform measure and sampling the directions $s_j$ randomly from this probability space, we are able to reduce the regularity requirements on the exact solution for provable convergence to $L^2$ in angle only. In exchange for the lower regularity, the notion of convergence has to be weakened to a convergence of the intensity mean over the angular domain in a mean square sense, which corresponds to mean square convergence in the incident radiation. With a multilevel Monte Carlo technique, the randomized sparse discrete ordinates method achieves this convergence already for complexities of essentially purely physical problems, and is thus in the spirit of the other sparse methods in this work.

As we will start out from the same variational formulation as for the standard deterministic DOM, we proceed directly to the discretization of the formulation in Sec. 6.1.

In Sec. 6.2, we prove complexity and convergence estimates for the incident radiation in the deterministic and the randomized sparse DOM.

6.1 Discretization

We set out from the augmented SUPG variational formulation (5.11) and discretize on the physical domain as before in Sec. 3.4.1 by restricting the physical function space to $V^L_D$, consisting of piecewise affine functions on a triangulation $\mathcal{T}_D^L$.

6.1.1 Discretization in angle

As quadrature rule on the angular domain, we now choose Monte Carlo integration

$$\int_S f(s) \, ds \approx \sum_{m=1}^{M_S} w_m f(s_m),$$

(6.1)

with uniform weights $w_m = \lvert S \rvert / M_S$ and randomly chosen directions $s_j$, independent and uniformly distributed over the angular domain.

By the construction of an angular partition $\mathcal{T}_S^N = \{S_j\}_{j=1}^{M_S}$, in which each element contains exactly one of the directions $s_j$, and the associated angular basis $\{\beta_j\}_{j=1}^{M_S}$
of characteristic functions on the partition, we arrive at the same algebraic DOM problem (5.23) as in the deterministic DOM, only that quadrature weights and directions have changed.

6.1.2 Generation of uniform random points on the angular domain

On the circle, we sample the directions \( s_j \) uniformly distributed from the interval \([0, 2\pi)\). On the sphere \( S^2 \), one of the simplest and fastest ways to obtain uniformly distributed points is to draw the azimuthal angle \( \phi \) uniformly from \([0, 2\pi)\) and as polar variable \( u = \cos \theta \) uniformly from \([-1, 1]\). Then the resulting direction \( s = (\sqrt{1 - u^2} \cos \phi, \sqrt{1 - u^2} \sin \phi, u)^\top \) is uniformly distributed on the surface of \( S^2 \), and both spherical and Cartesian coordinates of \( s \) are readily available (Weisstein, 2013).

6.2 Approximation properties

The discussion here follows the analysis by Barth et al. (2011) to a large extent, who derived and investigated the Multilevel Monte Carlo method for a standard elliptic model problem with stochastic coefficients.

For our method, we regard the probability space \((S, \mathcal{A}, P_S)\): the angular domain \( S \) serves as the set of elementary events, \( \mathcal{A} \subset 2^S \) is the \( \sigma \)-algebra of all possible events, and the measure \( P_S : \mathcal{A} \to [0, 1] \) is defined as the uniform measure \( dP_S = ds / |S| \) on the angular domain.

By this interpretation the theory of stochastic differential equations becomes available to us for further exploitation, even though the radiative transfer problem itself is deterministic.

The analysis of our method will rely on the definition of Bochner spaces of strongly measurable, square-summable functions \( v : S \to V \) with a Banach space \((V, \|\cdot\|_V)\) as range:

\[
L^2(S; V) := \{v : S \to V : v \text{ strongly measurable, } \|v\|_{L^2(S; V)} < \infty\},
\]

with the corresponding norm

\[
\|v\|_{L^2(S; V)} := \left( \int_S \|v\|^2_V \, dP_S \right)^{1/2} = \left( \frac{1}{|S|} \int_S \|v\|^2_V \, ds \right)^{1/2}.
\]

We observe here that the anisotropic Hilbert spaces \( H^{i,0}(\Omega) \), \( i = 0, 1 \), from (2.10) over the phase space \( \Omega = D \times S \) are isomorphic to the Bochner spaces \( L^2(S; H^i(D)) \), and their norms are equivalent:

\[
\|v\|^2_{H^{i,0}(\Omega)} = \int_S \|v\|^2_{H^i(D)} \, ds = |S| \int_S \|v\|^2_{H^i(D)} \frac{1}{|S|} \, ds = |S| \|v\|^2_{L^2(S; H^i(D))}. \tag{6.2}
\]
6.2 Approximation properties

Additionally, we will later make use of the relation

\[ \|v(x)\|_{H^i(\Omega)} = \left( \int_S \|v(x)\|_{H^i(D)}^2 \, ds \right)^{1/2} = |S|^{1/2} \|v(x)\|_{H^i(D)}, \quad i = 0, 1, \quad (6.3) \]

which holds for \( v \in H^i(D) \) that do not depend on \( s \).

Convergence results in this section will be of the form

\[ \lim_{n \to \infty} \|w - w_n\|_{L^2(S;V)} = 0, \quad (6.4) \]

with \( w \in V \) a function which is constant on \( S \), and a sequence of \( V \)-valued random variables \( w_n \in L^2(S;V) \). By the way the norm on \( L^2(S;V) \) is defined, (6.4) also permits to be written as

\[ \lim_{n \to \infty} (E[\|w - w_n\|_V^2])^{1/2} = 0, \]

with the expectation \( E[X] := \int_S X \, dP_S \) of a summable random variable \( X : S \to \mathbb{R} \). The convergence results here therefore state that the squared distance between \( w \) and \( w_n \) measured in the \( \|\cdot\|_V \) norm converges in expectation, or in other words, in a mean square sense. In particular, there may be “unlucky” sequences \( w_n \) for which the squared distance does not decrease at all. In numerical computations, convergence might become apparent only after a certain number of realizations of sequences \( w_n \) have been taken into account to approximate the expectation by a sample mean sufficiently well. In the next section, we will estimate how fast the expectation can be approximated by a Monte Carlo sample mean.

6.2.1 Approximation on the angular domain

We will now interpret the solution \( u(x,s) \) of the RTP as a random variable in \( L^2(S;L^2(D)) \), where \( s \in S \) is a random variable and \( x \in D \) the usual position in the physical domain. If we sample \( s \) from \( S \) and denote the obtained realization by \( s_j \), we obtain a sampled solution \( u_j(x) := u(x,s_j) \in L^2(D) \), i.e. one realization of the random variable \( u(x,s) \). In the following, we will denote a sequence of \( M_S \) realizations of \( s \) as \( S_{M_S} = \{s_j\}_{j=1}^{M_S} \).

By its definition in (1.2), the incident radiation \( G(x) \) is related to the expectation of \( u \) over \( S \) as

\[ G(x) = \int_S u(x,s) \, ds = |S| \int_S u(x,s) \frac{1}{|S|} \, ds = |S| E[u]. \quad (6.5) \]

An approximation to the incident radiation is obtained by applying the Monte Carlo quadrature rule (6.1) to the integral over \( S \):

\[ G_{M_S}(x;S_{M_S}) := \frac{|S|}{M_S} \sum_{j=1}^{M_S} u(x,s_j) \in L^2(S;L^2(D)) \quad (6.6) \]

with \( G_{M_S}(x;S_{M_S}) \) being itself a random variable due to its dependence on the set \( S_{M_S} \) of independent, randomly chosen directions.
6 Randomized sparse discrete ordinates method

As we identify the sample \( u(x, s_j) \) with the solution for the direction \( s_j \), the requirement for independent samples in (6.6) precludes the analysis of the radiative transfer equation with scattering in the Monte Carlo framework since the scattering term necessarily couples solutions for different directions, i.e. the samples are not independent any more. Therefore, we regard \( u \) and \( u_j \) as solutions to the radiative transfer equation without scattering for the purpose of this analysis.

Before we estimate the error of the approximate incident radiation, we cite Lemma 4.1 by Barth et al. (2011) on the mean square convergence rate of the sample mean of a function \( u \in L^2(S; L^2(D)) \).

**Lemma 6.1.** For \( M \in \mathbb{N} \) many independent and identically distributed (iid.) samples of directions \( s_j \) and \( u(x, s) \in L^2(S; L^2(D)) \) it holds

\[
\|\mathbb{E}[u] - \frac{1}{M} \sum_{j=1}^{M} u(x, s_j)\|_{L^2(S; L^2(D))} \leq M^{-1/2} \|u\|_{L^2(S; L^2(D))},
\]

where \( u(x, s_j) \) is the sample solution corresponding to direction \( s_j \).

The rate of mean square convergence of \( G_{M_S}(x; S_{M_S}) \) to \( G \) is then a direct result.

**Corollary 6.2.** For \( M_S \in \mathbb{N} \) many independent and identically distributed (iid.) samples of directions \( s_j \) and \( u(x, s) \in L^2(\Omega) \) it holds

\[
\|G(x) - G_{M_S}(x; S_{M_S})\|_{L^2(S; L^2(D))} \leq M_S^{-1/2} \|u\|_{L^2(\Omega)}.
\]

**Proof.** We abbreviate \( u(x, s_j) = u_j(x) \). By (6.5), we have \( G(x) = |S| \mathbb{E}[u] \). Using this relation we transform the left hand side of (6.7) so that Lemma 6.1 becomes applicable:

\[
\|G(x) - G_{M_S}(x; S_{M_S})\|_{L^2(S; L^2(D))} = |S|^{-1/2}\|\mathbb{E}[u] - \frac{1}{M_S} \sum_{j=1}^{M_S} u_j(x)\|_{L^2(S; L^2(D))}.
\]

With the norm equivalence (6.2) we obtain the assertion. \( \square \)

### 6.2.2 Error estimate for single-level MC Galerkin method

In reality the solutions \( u_j \) of the RTP for given directions are not known exactly. Instead, we solve the discrete variational problem (5.37) with angular partition \( T^N_S \) as constructed above in Sec. 6.1.1 by means of the linear system (5.23) and obtain a solution vector \( (u_1, \ldots, u_{M_S})^\top = u \in \mathbb{R}^{M_SM_D} \). As we have shown in Sec. 5.2, this solution vector is related to the solution \( u_{L,j} \) of the purely physical variational formulation (5.13) by \( u_{L,j}(x) = u_j^\top \alpha(x) \), where \( \alpha(x) \in (L^2(D))^{M_D} \) is the vector of physical basis functions. Also, it holds \( u_{L,j} = P^j_D u_j \) with the Galerkin projector on the physical domain from Def. 5.11. Since we consider the case without scattering here, the approximations \( u_{L,j} \) do not depend on the angular resolution \( N \). The approximation of the incident radiation that is available to us is thus in fact the sample mean of approximate solutions:

\[
G_{L,M_S}(x; S_{M_S}) := \frac{|S|}{M_S} \sum_{j=1}^{M_S} u_{L,j}(x) \in L^2(S; L^2(D)),
\]

82
6.2 Approximation properties

again a random variable as \( u_{L,j} \) depends on \( s_j \). The error of this single-level incident radiation is estimated by the following theorem.

**Theorem 6.3** (Mean square error of single-level incident radiation). Let \( u \in H^{s+1,0}(\Omega) \), \( s \in \{0,1\} \), be the solution to the exact variational formulation (5.11), then the root mean square error of the sampled approximate incident radiation from (6.8) is bounded by

\[
\|G - G_{L,M_S}\|_{L^2(S;L^2(D))} \lesssim (2^{-sL} + M_S^{-1/2}) \|u\|_{H^{s+1,0}(\Omega)},
\]

with relation “\( \lesssim \)” from Def. 2.1.

**Proof.** The theorem corresponds to Thm. 4.3 by Barth et al. (2011). By triangle inequality, we have

\[
\|G - G_{L,M_S}\|_{L^2(S;L^2(D))} \leq \left\| G - \int_S u_L \, ds \right\|_{L^2(S;L^2(D))} + \left\| \int_S u_L \, ds - G_{L,M_S} \right\|_{L^2(S;L^2(D))},
\]

where \( u_L \) is the solution to the spatially discrete variational problem (5.26) so that \( u_L \) is \( \|\cdot\|_1 \)-stable as asserted by Lemma 5.12 and converges to \( u \) as asserted by Lemma 5.13.

We transform the first part using (6.5), then bound it from above by Jensen’s inequality, which states that \( \phi(\mathbb{E}[f]) \leq \mathbb{E}[\phi(f)] \) for convex \( \phi : \mathbb{R} \to \mathbb{R} \) and integrable real-valued \( f \), apply (6.2), and estimate by Lemma 5.13:

\[
\left\| G - \int_S u_L \, ds \right\|_{L^2(S;L^2(D))} = |S|^{1/2} \left\| \mathbb{E}[u - u_L] \right\|_{L^2(S;L^2(D))} \leq |S|^{1/2} \left( \int_S \left( \mathbb{E}[u - u_L] \right)^2 \, ds \right)^{1/2} \leq |S|^{1/2} \left( \int_S \left( u - u_L \right)^2 \, ds \right)^{1/2} \leq \|u - u_L\|_{L^2(\Omega)} \leq c_p 2^{-sL} \|u\|_{H^{s+1,0}(\Omega)}.
\]

The second part is bounded with the aid of Lemma 6.1 and stability of the physical Galerkin projector (Lemma 5.12):

\[
\left\| \int_S u_L \, ds - G_{L,M_S} \right\|_{L^2(S;L^2(D))} \leq M_S^{-1/2} \|u_L\|_{L^2(S;L^2(D))} \leq c_p M_S^{-1/2} \|u\|_1.
\]

Together these estimates yield the assertion of this Lemma. \( \square \)

**Remark 12.** As the Monte Carlo method samples the solution pointwise in the angular domain, we shall tacitly assume that the solutions are \( L^2(S) \) functions which can be evaluated almost everywhere in \( S \), i.e. the sets where pointwise values are not defined have zero measure. In finite precision arithmetics, there might be a positive probability for a sample \( s_j \) to be drawn from such a set of undefined point values. However, simply rejecting that sample and drawing anew resolves the situation. Then, the Monte Carlo method achieves the proved convergence results also for square summable solutions with e.g. finite discontinuities, point singularities, or line singularities (if \( d_S = 2 \)) in angle.
6.2.3 Error estimate for multilevel MC Galerkin method

As in the sparse methods presented in previous chapters, we will now take advantage of the multiscale structure of \( V^D_l \), \( l = 0, \ldots, L \), to define a randomized sparse discrete ordinates method. Instead of computing approximate solutions with full physical resolution for every sampled direction, we combine approximate solutions of all resolutions up to \( L \). In this combination, approximate solutions of coarser physical resolution will be sampled in angle more often as they can be computed comparatively cheaply, while costly high resolution transport subproblems will only be solved for relatively few directions. By choosing the number of samples for each resolution level appropriately we can reduce the overall complexity of the method such that it is essentially equal to that of purely physical solution method, while we maintain the convergence rate of the single-level method.

The total number of samples \( M_S \) of this multilevel Monte Carlo method is

\[
M_S = \sum_{l=0}^{L} M_l,
\]

with \( M_l \) the number of samples on level \( l \).

By expanding the solution \( u_L \) into a telescopic sum with the convention \( u_{-1} = 0 \) as

\[
u_{l,j} = \sum_{l=0}^{L} (u_{l,j} - u_{l-1,j}),
\]

we arrive at the definition of a sparse sampled approximate incident radiation:

\[
\hat{G}_{L,M_S} (x; S_{M_S}) := |S| \sum_{l=0}^{L} \frac{1}{M_l} \sum_{j=1}^{M_l} (u_{l,j}(x) - u_{l-1,j}(x)) \in L^2(S; L^2(D)), \quad (6.11)
\]

a random variable due to its dependence on the set \( S_{M_S} \) of sampled directions. Convergence of \( \hat{G}_{L,M_S} \) to \( G \) in mean square sense is ensured by the following theorem.

**Theorem 6.4 (Mean square error of multilevel incident radiation).** Assume \( u \in H^{s+1,0}(\Omega), s \in \{0, 1\} \), is the solution to the exact variational problem \((5.11)\), then the error of the sparse sampled approximate incident radiation from \((6.11)\) for any choice of the number of samples \( M_l \in \mathbb{N}, l = 0, \ldots, L \) satisfies the estimate

\[
\|G - \hat{G}_{L,M_S}\|_{L^2(S; L^2(D))} \lesssim (2^{-sL} + \sum_{l=0}^{L} M_l^{-1/2} 2^{-sl}) \|u\|_{H^{s+1,0}(\Omega)}. \quad (6.12)
\]

**Proof.** We follow the proof of Lemma 4.4 by Barth et al. (2011) and split the error as before:

\[
\|G - \hat{G}_{L,M_S}\|_{L^2(S; L^2(D))} \leq \left\| \int_S u_L ds \right\|_{L^2(S; L^2(D))} + \left\| \int_S u_L ds - \hat{G}_{L,M_S} \right\|_{L^2(S; L^2(D))}.
\]

where \( u_l, l = 0, \ldots, L \), is again the solution to the spatially discrete variational problem \((5.26)\) so that \( u_l \) is \( \|\cdot\|_1 \)-stable as asserted by Lemma 5.12 and converges to \( u \) as asserted by Lemma 5.13.
6.2 Approximation properties

The first term is estimated as in (6.10). We analyze the second term further by telescoping \( u_L = \sum_{i=0}^{L} (u_i - u_{i-1}) \) with \( u_{-1} = 0 \):

\[
II \leq \sum_{i=0}^{L} \left\| \int_{\Omega} (u_i - u_{i-1}) \, ds \right\|_{L^2(S;L^2(D))} \leq |S|^{1/2} \sum_{i=0}^{L} M_i^{-1/2} \| u_i - u_{i-1} \|_{L^2(S;L^2(D))}
\]

\[
\leq |S|^{1/2} \sum_{i=0}^{L} M_i^{-1/2} \left( \| u - u_i \|_{L^2(S;L^2(D))} + \| u - u_{i-1} \|_{L^2(S;L^2(D))} \right)
\]

\[
\leq \sum_{i=0}^{L} M_i^{-1/2} c_p \left( 2^{-sL} + 2^{-(L-1)s} \right) \| u \|_{H^{e+1,0}(\Omega)}
\]

\[
\leq c_p (1 + 2^{s}) \sum_{i=0}^{L} M_i^{-1/2} 2^{-sL} \| u \|_{H^{e+1,0}(\Omega)},
\]

where we have applied Lemma 6.1 from the first line to the second, and Lemma 5.13 from the third to the fourth.

As a next step we equilibrate the error contributions to achieve an overall rate of \( 2^{-sL} = O(h_L^e) \), if \( h_L \) is the largest mesh width of the triangulation over the physical domain mesh \( T^L \). A suitable choice is already given by Barth et al. (2011) for \( l \geq 1 \):

\[
M_i = l^{2+2e} \left( \frac{h_i}{h_L} \right)^{2s} = O(l^{2+2e} 2^{2s(l-1)}), \quad e > 0, \ l \geq 1.
\]  \hspace{1cm} (6.13)

We quickly convince ourselves that this choice yields the desired rate:

\[
\sum_{i=1}^{L} 2^{-sL} \left( l^{2+2e} 2^{2s(l-1)} \right)^{-1/2} = 2^{-sL} \sum_{i=1}^{L} \frac{1}{l^{1+e}} \leq C e 2^{-sL}.
\]

On level \( l = 0 \), we have to set \( M_0 = 2^{sL} \) in order to achieve the rate of \( 2^{-sL} \). Despite the high number of sampling directions, the work on level 0 remains feasible because very few spatial degrees of freedom exist on this level.

As a measure for the complexity of the problem to solve, we regard the total number of degrees of freedom \( \hat{M}_{L,M_S} \), which equals the sum of all spatial degrees of freedom used in the solution of each physical problem. The number of spatial degrees of freedom scales with the resolution level \( L \) as \( M_D = O(2^{dL}) \) (cf. relation (3.31)). Therefore the following lemma holds (pursuant to Thm. 4.5 by Barth et al., 2011):

\textbf{Lemma 6.5}. The total number of degrees of freedom employed in the randomized sparse DOM is

\[
\hat{M}_{L,M_S} \lesssim \begin{cases} M_D (\log M_D)^{3+2\epsilon} & \text{if } d = 2 \text{ and } s = 1, \\ M_D (\log M_D)^{2+2\epsilon} & \text{if } d = 3 \text{ or } s < 1, \end{cases} \quad \epsilon > 0 \text{ arbitrarily small.}
\]

The constant hidden by the relation “\( \lesssim \)” will depend on \( \epsilon \), but is independent of \( L, M_D, \) or \( M_S \).
Randomized sparse discrete ordinates method

Proof. Adapted from Barth et al. (2011):

\[ M_{L,M_S} \lesssim 2^{2sL} + \sum_{l=1}^{L} M_l 2^{dl} = 2^{2sL} + \sum_{l=1}^{L} l^{2+2e} 2^{2s(l-L)} 2^{dl} \]

\[ = 2^{2sL} + 2^{dl} \sum_{l=1}^{L} l^{2+2e} 2^{2s(l-L)} 2^{d(l-L)} = 2^{2sL} + 2^{dl} \sum_{l=1}^{L} l^{2+2e} 2^{d-2s}(l-L) \]

\[ \lesssim 2^{2sL} + 2^{dl} \left\{ \begin{array}{ll}
LL^{2+2e} & \text{if } d = 2 \text{ and } s = 1, \\
L^{2+2e}(2-2^{-L}) & \text{if } d = 3 \text{ or } s < 1
\end{array} \right. \]

\[ \lesssim M_D \left\{ \begin{array}{ll}
(\log M_D)^{3+2e} & \text{if } d = 2 \text{ and } s = 1, \\
(\log M_D)^{2+2e} & \text{if } d = 3 \text{ or } s < 1.
\end{array} \right. \]

The complexity remains essentially the same as for a purely spatial problem with \( M_D \) degrees of freedom.

6.2.4 Error of incident radiation in deterministic DOM

For purposes of comparison, we also investigate the convergence of the incident radiation in the standard deterministic DOM. The analysis below will cover the 2+1D setting because quadrature error estimates for functions of lower regularity are more easily attainable there, and leave the 3+2D setting to the numerical experiments.

Error estimate for angular quadrature. In the deterministic DOM in 2+1D, we use equispaced directions \( s_j, j = 1, \ldots, M_S \), to discretize the angular domain \( S \). Quadrature in \( S \) over the Galerkin solutions \( u(\cdot, s_j) \) of physical subproblems then yields the approximate incident radiation \( G_{L,N} \), in which the angular refinement level \( N \) relates to the number of DoFs in angle as \( M_S = 2N + 1 \).

Angular integrals are evaluated numerically by the trapezoidal rule. With this quadrature rule we may profit from high convergence rates over the periodic domain \( S \) if the integrand is sufficiently smooth. The following estimate for the convergence of the trapezoidal rule in terms of Sobolev norms of the integrand has been derived by Kurganov and Rauch (2009).

Lemma 6.6. For any periodic function \( w \in W^t_1(S) \) with \( S = [0,2\pi) \) and \( 1 < t \in \mathbb{N}, 1 \leq p \leq \infty \), the error of the trapezoidal quadrature rule \( Q_{M_S}^{M_S} \) with \( M_S \) nodes is bounded by

\[ | \int_S w \, d\varphi - Q_{M_S}^{M_S} w | \lesssim M_S^{-t} \| D^t w \|_{L^1(S)} \lesssim M_S^{-t} \| w \|_{W^t_1(S)} \]. \quad (6.14) \]

The relation “\( \lesssim \)” is given by Def. 2.1.

Due to the condition \( t > 1 \), the lemma does not apply at first to the nonsmooth \( L^2(S) \) functions of interest here. The following remarks attempt extensions to some functions of lower regularity.

Remark 13. Lemma 6.6 also holds for singular functions \( w(\varphi) = \varphi^\lambda \) with \( 0 > \lambda > -1 \), which are in \( W^t_1([0,2\pi)) \) with \( 0 < t < 1 \) if we assume the quadrature nodes to be
distributed as $\varphi_j = 2\pi j/N$ with $j = 0, \ldots, N$ and redefine and extend $w$ to

$$w(\varphi) = \begin{cases} \varphi^\lambda & \text{if } 0 < \varphi \leq 2\pi, \\ 0 & \text{if } \varphi = 0, \end{cases}$$ \hspace{1cm} (6.15)$$

which does not affect the value of the integral of $w$ over $S$. From the theory of fractional Sobolev spaces we know by the inequality

$$\lambda - t > -\frac{d_S}{p},$$ \hspace{1cm} (6.16)

which for $d_S = 1$ and $p = 1$ yields $\lambda - t > -1$, that $w \in W^{1+\lambda-\epsilon}_1$, $\epsilon > 0$, so that we expect a convergence rate just slightly worse than $M_S^{-1-\lambda}$.

Remark 14. For functions with a jump in the angular part, i.e. a finite discontinuity, a simple argument makes it plausible that the trapezoidal rule yields a $M_S^{-1}$ convergence rate. Assume the function $w \in L^1(S)$ is the step function with a jump height of $l$. Then the composite trapezoidal rule integrates the function on all subintervals of $[0,2\pi]$ exactly, except for the interval with the discontinuity. There, the error will be less than $1/2hl$, if $h$ is the distance between two quadrature points. As $h = O(M_S^{-1})$, the error decreases with a rate of $M_S^{-1}$.

Error estimate for incident radiation in full DOM. We derive an estimate for the error of the approximate incident radiation

$$G_{L,N} := Q^N_{S}p^D_{L}u$$ \hspace{1cm} (6.17)

in the full deterministic DOM, where $p^D_L$ is the Galerkin projector on the physical domain from Def. 5.11 and $Q^N_{S}u = \sum^{2N+1}_{i=1}w_iu(x,s_i)$ the quadrature operator of the trapezoidal rule on the angular domain.

Theorem 6.7. For $u \in H^{s+1,0}(\Omega) \cap H^1(D) \otimes W^1_1(S)$, $s \in \{0,1\}$, $1 < t \in \mathbb{N}$, the error of the approximate incident radiation $G_{L,N}$, $L, N \in \mathbb{N}$, satisfies

$$\|G - G_{L,N}\|_{L^2(D)} \lesssim 2^{-sL}\|u\|_{H^{s+1,0}(\Omega)} + N^{-t}\|u\|_{H^1(D) \otimes W^1_1(S)}. \hspace{1cm} (6.18)$$

Proof. We will use the notation $G = I_Su$, where $I_S$ stands for the integral operator over $S$. Then it holds

$$\|I_Su - Q^N_{S}p^D_{L}u\|_{L^2(D)} = \|I_Su - I_SP^D_{L}u + I_SP^D_{L}u - Q^N_{S}p^D_{L}u\|_{L^2(D)} \hspace{1cm} (6.19)$$

$$\lesssim \|u - P^D_{L}u\|_1 + N^{-t}\|P^D_{L}u\|_{W^1_1(S) \otimes L^2(D)}$$

$$\lesssim 2^{-sL}\|u\|_{H^{s+1,0}(\Omega)} + N^{-t}\|u\|_{W^1_1(S) \otimes H^1(D)}.$$ For the second term in the step from the third to the fourth line, we used stability of the physical Galerkin projector (Lemma 5.12). \qed
In terms of number of degrees of freedom, this estimate reads as
\[
\| G - G_{L,N} \|_{L^2(D)} \lesssim M_D^{-s/d} \| u \|_{H^{s+1,0}(\Omega)} + M_S^{-t/d_s} \| u \|_{H^1(D) \otimes W^1_t(S)}. \tag{6.20}
\]

In line (6.19) of the proof above, an estimate for the quadrature error on the angular domain is inserted. According to Remark 13, we would expect a convergence rate of the quadrature error of at least \( M_S^{-1/2} \) for singular functions of the type (6.15) which are additionally in \( L^2(S) \). Functions with finite discontinuity should even yield a rate of \( M_S^{-1} \) in the 2+1D setting.

Note again the difference in the notions of convergence: the estimate by Thm. 6.7 holds independently of the actual placement of the quadrature nodes of the trapezoidal rule, whereas the Monte Carlo estimates hold in expectation, nothing is asserted for any particular choice of directions, but the required angular regularity of the solution is only \( L^2(S) \) (subject to Remark 12).

If preferred transport directions are known in advance, the probability density can be adapted to a nonuniform density \( \rho(s) \) accounting for the expected importance of directions. With the adapted measure \( dP_S = \rho(s)ds \) and the replacement of \( L^2(S) \) by \( L^2(S;\rho) \), the results of Corollary 6.2 and Theorems 6.3 and 6.4 hold verbatim. In such a way, problems as the searchlight beam of an angularly collimated ray entering the domain from the outside could be handled as well.

We will investigate the Monte Carlo performance also numerically in the experiments of Chapter 8.
7 Implementation

An investigation of a numerical method would not be complete without at least a proof-of-concept implementation of the method. In this chapter, we therefore describe our implementation of the sparse discrete ordinates method from the targeted goals over design decisions and selected algorithms to the results of performance tests on parallel hardware.

7.1 Requirements specification

7.1.1 Problem statement

We have seen in the previous chapters that the computational effort required to solve the RTP (1.1) by standard methods as introduced in Sec. 1.3 increases strongly like $2^{L(d+d_S)}$ with physical resolution level $L$, physical dimension $d$, and angular dimension $d_S$, if we assume that physical resolution level and angular resolution level $N$ have to be coupled as $L \sim \log N$ to reduce the total error continuously.

With our sparse methods for radiative transfer, we could lower the increase in computational complexity to $\log LL_{\max}^{\max(d,d_S)}$, amounting to a significant difference for larger $L$.

In order to analyze our sparse methods, we would like to test them not only on 2+1D model problems, but also in more realistic 3+2D applications, and most of all, compare their performance to that of the standard full tensor methods. While 2+1D model problems and possibly sparse solutions of 3+2D problems of very limited resolution could be computed by a prototypical MATLAB implementation on one or a handful of processors, full tensor solutions of 3+2D problems of interesting resolution are out of reach with this kind of configuration.

Thus there was a need for a new implementation which would alleviate the existing limitations.

7.1.2 Goals

Leading from the problem statement, the software has the following goals:

1. Solve radiative transfer problems in 3+2D and 2+1D by standard full tensor DOM and sparse tensor DOM efficiently in a parallel way.

2. As a software for research, it should allow for simple exchange and modification of algorithmic components (quadrature rules, solution method for linear systems, generation of ordinates etc.) and extension to new applications (e.g. Boltzmann transport problems) without much restructuring effort.
The discrete ordinates method was selected as method for implementation because it promises a better modularity than direct sparse approach pursued for the spherical harmonics method. In principle only physical transport problems for fixed direction have to be solved in the DOM and the solutions have to be combined. The more modular structure also simplifies the parallelization as well as the overall implementation while still providing the benefits from sparse tensorization.

### 7.1.3 Use cases

Based on our research interests, two main use cases were identified:

1. Single run, possibly timed, with optional visualization of the solution.
2. Convergence run: several solutions varying in one or multiple parameters are compared to a reference solution, errors are computed and their dependence on the parameter(s) is shown.

### 7.1.4 Key functional features and nonfunctional requirements

To achieve the goals and support the use cases formulated above, a number of requirements for specific functions and requirements on other properties of the software, so-called nonfunctional requirements, were fixed. In the following, we list the requirements which the software fulfills.

**Key functional features**

These functions are provided by the software:

- Solve large, highly resolved 3+2D and 2+1D radiative transfer problems with sparse tensor DOM.
- For comparison purposes, also solve RT problems with full tensor DOM.
- Compute deviation between sparse and full tensor solution in terms of errors.
- Provide support for convergence rate calculations by running several solvers with varying resolution parameter in a row.
- Output solutions to file for archival and visualization by secondary software.

The software supports the solution of application problems with the properties:

- Polygonal domain
- Zero inflow boundary conditions $g(x) = 0 \ \forall x \in \partial D$, emitting wall boundary conditions $\exists x \in \partial D : g(x) \neq 0$
- Smooth scattering kernels, e.g. isotropic, Rayleigh, Henyey-Greenstein
• Separable black-body radiation functions on right hand side: \( f(x,s) = I_b(x)Y(s) \).

The implemented methods are:

• Sparse tensor combination technique DOM with flexible sparsity profile, full DOM

• Equidistributed ordinates in 2+1D, arbitrary ordinates loaded from files for 3+2D

• Multilevel Monte Carlo random ordinates

• Method for physical transport problems: Galerkin FEM with (non-)hierarchical bi-/trilinear elements

• Stabilization for physical problem: transport/SUPG stabilization

• Quadrilateral/hexahedral mesh cells in physical domain

• Solution of linear systems in parallel by fast Krylov subspace solvers.

Computed output quantities are:

• Incident radiation

• \( L^2 \)- and \( H^1 \)-error in incident radiation.

**Nonfunctional requirements**

Nonfunctional requirements satisfied by the software are:

• It is modular, easily extensible with new application problems and new algorithmic methods. Key algorithmic components such as solution procedure of linear systems, quadrature rules, FEM basis, are exchangeable.

• The code is dimension independent so that problems on \( (d,d_S) = (3,2) \) and \( (2,1) \) can be run without major changes to the problem definition.

• To tackle large highly resolved problems, it makes use of parallelism. Target machine is the Brutus cluster (2013) at ETH Zürich, with medium-scale parallelism by MPI up to about 1000 cores.

• For the solution of purely physical subproblems, an existing FEM library is used so that work can be focused on the implementation of the new parts.

• Efficiency: optimal runtime \( O(M \log M) \) desirable, with \( M \) being the total number of dofs.

• Output results are reproducible, that is, chosen input options are recorded along with generated output.
7.2 Software design

In this section we describe how the software has been designed to meet the requirements specified before.

7.2.1 Design decisions

The most relevant design decisions along with their rationale are:

• **Object oriented (OO) programming paradigm.** Object orientation supports the desired feature of a modular and extensible architecture by concepts such as encapsulation and polymorphism.

• **Programming language C++ with generic programming.** C++ ([Stroustrup, 1997](#)) is known to realize many concepts of the OO paradigm while at the same time to produce fast machine code.

  Generic programming adds a layer of abstraction to software design with the purpose of flexibility in the use of the code by leaving details of an implementation unspecified. Since the instantiation of required code is deferred to compile time, the compiler can produce executables containing only this required code so that the additional flexibility does not translate into bloating and therefore slowing of the program.

  Just as sparse tensor methods break the curse of dimensionality for the solution of PDEs, generic programming breaks the curse of dimensionality in software design: For instance, a method with a certain number of parameters which are allowed to vary in type would require a number of implementations exponential in the number of parameters if all possible combinations of parameter types were to be covered. Furthermore, not all parameter types might be known in advance. With generic programming, the decision for which parameter types the method should be implemented can be left open. All that is required from the possible parameter types is that they provide the interface the method expects. If one type does not satisfy this “contract”, it will be known at compile time.

  An insightful account of generic programming is given by [Alexandrescu (2001)](#). From his book, concepts such as traits, typelists, and policies have entered our software.

  Another reason for C++ is the fact that the chosen FEM library is programmed in it so that with the same choice of programming language no additional conversion layer for calls into the library is needed.

  We also decided to take advantage of some features of the latest C++11 standard such as smart pointers, `auto`, and the `nullptr` keyword which greatly simplify the development of heavily templated code and which are supported by recent versions of all common C++ compilers.

  Finally, the abundance of free libraries which can be interfaced with directly from C++ programs also speaks in favor of this programming language. Our software
is built upon the standard template library (STL) and some of the boost libraries (Dawes and Abrahams, 1998).

- **Header-only implementation.** A problem arising with the generic programming paradigm in C++ is that, as the actual required instantiations of the code are only generated during compile time, precompiled object files have to include instantiations for all combinations of template parameters that the user is going to employ in his code, otherwise linker errors will occur.

Since foreseeing the exact use of template classes is impossible and since this practice incurs the curse of dimension, thereby contradicting the main reason for generic programming, we moved all code to header files.

This leads to the disadvantage that every time a program is compiled, the complete code has to be processed by the compiler again. However, with modern compilers and computers, we found the resulting delays in compile times still quite bearable as the compilation durations for typical programs based on our software were usually between 10 and 30 seconds.

- **FEM library deal.II.** The library deal.II is a very mature object oriented FEM library (Bangerth et al., 2007). Its advantages over similar open source FEM libraries are:
  - Its interface is constructed in object oriented generic programming way with C++ as language, just as decided upon for the transport solver so that smooth and tight inclusion into the transport solver can be expected.
  - The problem dimension enters as template parameter of classes so that it only needs to be specified once in the beginning and is passed on afterwards. Internally, efficient code for each dimension can be generated, yet from a user perspective, the interface remains identical.
  - Already includes support for distributed parallel solution of PDEs via MPI, with and without domain decomposition. Interfaces to PETSc and Trilinos for efficient parallel solution of linear systems.
  - Has existed since at least 1999, currently in its seventh major version, extensively tested, appears mature and stable.
  - Well documented with tutorials and reference manual (Bangerth et al.).
  - An active community of developers continues to improve the code and supports users with their issues.

Disadvantages are:
  - Only supports quadrilateral and hexahedral mesh elements.
  - Compared with the FEM libraries of the FEniCS project (Logg et al., 2012) which allow the user to specify the variational formulation of the PDE to solve directly in the code, deal.II has a rather low-level interface, usually requiring the user to take care of the matrix assembly himself by looping over mesh cells and quadrature points explicitly. On the other hand, this
Implementation

also means that implementation details can be controlled and tuned more finely.

• **Parallel linear algebra with PETSc library.** PETSc (Balay et al., 1997, 2012) has been chosen as library for the solution of linear systems over the alternative Trilinos because it focuses only on this task of solving linear equations. Its lack of an object-oriented interface is in part made up by the wrapper classes provided in deal.II. Trilinos in contrast is divided in many packages which add functionality beyond linear equation solving to make it a more full-fledged collection of algorithms (Heroux et al., 2005), more than what is required from the library for our software. PETSc offers a sufficient number of parallel preconditioner-solver-pairings in connection with the preconditioner library Hypre (Chow et al., 1998) so that finding a performant pairing for the present problem was deemed possible.

• **Focus on combination technique and DOM as solution method.** To limit the overhead during development and execution of the software, we decided to focus on the discrete ordinates method in full and sparse version as solution method. However, adding other methods such as the spherical harmonics method should be feasible without much effort due to the modularity of the software design.

### 7.2.2 Architecture

The components of the software are ordered into several modules and packages. Modules combine components of related, particular purpose and are represented at the file system level by folders. Packages and subpackages group components of common origin and act as a layer of separation between components of identical interface but differing functionality. They are represented in C++ code by namespaces. The following examples should clarify the subdivision.

**Modules**

The Apps module contains definitions of application problems. They are generally defined by a specialization of the RadiativeTransportApplication class in the Radtrans module. The specialization consists in supplying the types of source function, absorption function, scattering kernel and coefficient function, physical domain generator and boundary conditions as template parameters to RadiativeTransportApplication. Each specialized application exists in its own package to facilitate the naming of recurring function classes. In this way, every application can have a function class named SourceFunction without name collisions, since the source function class is always referenced by its full nameAppName::SourceFunction.

In the DOM module, we find the main components of the software, the combination technique framework in CombTechSolver and the FullTransportSolver capable of solving radiative transfer problems by the full tensor DOM in a sequential manner. Also provided are classes for the solutions of either solver: a CombTechSolution is
the collection of one or many \texttt{FullTransportSolution} objects. For parallel operation, classes of identical name and interface are provided in the subpackage \texttt{parallel}. Switching from sequential to parallel execution is therefore (almost) as simple as substituting \texttt{parallel::CombTechSolver} for \texttt{CombTechSolver} (deal.II uses the same mechanism). In Sec. 7.2.3, a sequential and parallel program are juxtaposed.

The \texttt{Domain} module contains domain generator classes which are passed to the application. The solver uses the supplied domain generator to initialize the triangulation object.

The purpose of components in the \texttt{Eval} module is to evaluate a certain aspect of the computed solution. These evaluators act on the solution, extract information, and record it by writing to the console or a file. Examples are \texttt{EvalRunTime} to measure the run time of the solver, \texttt{EvalErrIncRad} to compute the error in incident radiation to a reference solution, or \texttt{EvalSolverList} to count the number of iterations of the linear solver.

Components in the \texttt{Radtrans} module specify the radiative transfer problem. A particular RTP is specified by a variational formulation and a specialization of the general application problem class \texttt{RadiativeTransportApplication}. Variational formulations with different stabilization operators such as \texttt{RTVarFormTStab} with T-stabilization and \texttt{RTVarFormSUPG} with SUPG stabilization (cf. Sec. 3.3.1) are available. They are passed to the solver via the \texttt{Method} parameter.

Typical scattering kernel functions are defined in the \texttt{Scattering} module.

The components in the \texttt{Scheduler} module deal with distribution and splitting of the subproblems of the combination technique to several processors. The actual scheduling strategy can be provided to the \texttt{parallel::Scheduler} class as policy. Existing strategies are explained in Sec. 7.3.

The \texttt{Transport} module combines abstract interface definitions for transport solvers and solutions in order to provide a common handling across possibly different solution methods in the future.

Some further modules not mentioned here offer frequently used function classes (e.g. constant function or zero function), support for data input and output, operating system access, and other auxiliary components.

\section*{Interfaces}

Here we present a short overview of the interfaces of the most important classes. The detailed interface description can be found in the code documentation.\par

\section*{TransportSolver}

The abstract base class \texttt{TransportSolver} only offers a \texttt{run()} method to start the solution process of a solver. The main purpose of this class is to provide a common base class for solvers of different type, which simplifies their handling in lists or as arguments to functions.\par

\section*{CombTechSolver}

This is the main solver class to solve RTPs by the combination technique. It is implemented as a class template with several template parameters which allow the
specialization of the type of application, the sparsity profile, the solver for subproblems, the scheduler, and the type of problem to compute. In the sequential version, some of these template parameters default to the appropriate sequential types so that the user need not specify them.

```cpp
namespace transo {
    template <class TransApp, class SparsityProfile, class SubprobSolver,
              class Scheduler = SerialScheduler<ComputeProblem<TransApp::spacedimV> >,
              class CompProb = ComputeProblem<TransApp::spacedimV> >
    class CombTechSolver;
}
```

Due to these template parameters, one implementation of this class suffices for sequential and parallel operation. The parallel CombTechSolver is simply a partial specialization of the general class template with a modified default for the scheduler. We achieve this by deriving from the original CombTechSolver while passing on some template parameters and setting others:

```cpp
namespace parallel {
    template <class TransApp, class SparsityProfile, class SubprobSolver,
              class Scheduler = parallel::Scheduler<SchedulingStrategySplitAll,
                                           TransApp::spacedimV> >
    class CombTechSolver : public transo::CombTechSolver<TransApp, SparsityProfile, SubprobSolver, Scheduler,
                                           parallel::ComputeProblem<TransApp::spacedimV> > { /* ... */ };}
```

The interface of the CombTechSolver adheres to the one of the TransportSolver base class. Additionally, the CombTechSolver provides a run(Solution& solution) method which accepts a CombTechSolution object, in which the solution to the application problem is provided.

The CombTechSolver is sufficiently flexible to be used as a general combination technique framework for any other kind of PDE problem stated on the Cartesian product of two component domains. To our knowledge, it is the first implementation of the combination technique based on deal.II.

FullTransportSolver
The FullTransportSolver class template performs the actual work of setting up a triangulation, degrees of freedom, the system matrix, and right hand side, and of solving the linear system of a full tensor transport problem. It can be specialized by the template parameters Method, which is a collection of types defining various methods used in the solver such as the spatial quadrature rule, the FEM type and order, the variational form, and the linear solver, and RTApp, which combines types defining all aspects of the application problem such as its dimension, the physical domain, boundary conditions, and all coefficient functions from the RTE.
The functionality of the FullTransportSolver is actually split over two classes. A class template FullTransportSolverBase implements the general order of execution of mesh refinement, distribution of degrees of freedom, matrix and vector assembly, and solution of the linear system, and those parts of these items which are identical to the sequential and parallel version. The FullTransportSolver derives from a sequential specialization of this base class template and implements the sequential versions of the execution steps, while parallel::FullTransportSolver does so for the parallel versions.

```cpp
namespace transo {
  template <class Method, class RTApp, class VeloGen>
  class FullTransportSolver : public FullTransportSolverBase<
      Method, RTApp, VeloGen,
      ComputeProblem<RTApp::spacedimV>,
      dealii::SparseMatrix<double>, dealii::Vector<double>,
      FullTransportSolution<RTApp::spacedimX, RTApp::spacedimV> > {
    /* ... */
  };
}
```

By passing the appropriate types for the compute problem, sparse matrix, and vector to the base class, we can re-use its existing code which would be identical in sequential and parallel version without having to rewrite it.

```cpp
namespace parallel {
  template <class Method, class RTApp, class VeloGen>
  class FullTransportSolver : public FullTransportSolverBase<
      Method, RTApp, VeloGen,
      parallel::ComputeProblem<RTApp::spacedimV>,
      dealii::PETScWrappers::MPI::SparseMatrix,
      dealii::PETScWrappers::MPI::Vector,
      parallel::FullTransportSolution<RTApp::spacedimX,
        RTApp::spacedimV> > {
    /* ... */
  };
}
```

Only parts of the code which differ between sequential and parallel version are re-implemented in the parallel FullTransportSolver. These are some initialization calls as the interfaces of the different data types are not completely identical and the matrix assembly method which requires additional checks and branches in the parallel version to ensure we only operate on local data.

The interface of the FullTransportSolver also adheres to the TransportSolver interface, with the addition of a run(Solution& solution) method returning the problem solution in the passed FullTransportSolution.
7.2.3 Interaction

In this section we illustrate the interfaces and interdependence of components in our software and show how they interact to solve radiative transfer problems.

To begin with, we juxtapose two main methods in Fig. 7.1 as the user would have to implement them to solve an application problem in a single run with fixed resolution, once in the sequential version, once in parallel.

The listings in Fig. 7.1 represent actual source code except for statements of includes, namespace directives, and parameter handling which have been omitted to highlight the crucial parts.

First, the requested problem dimension is set as a constant. Then, types for the application, method, solver, and evaluators are defined. Evaluators can also be chained as in the example, where the first evaluator takes the run time of the solver and the second computes the incident radiation from the solution produced during the timed run before. Only in the end of the listing, actual calculations are started by invoking eval.run(...).

The differences between a sequential and a parallel implementation are limited to parallel:: namespace qualifiers and class names which have been replaced by their parallel equivalent. Switching from a sequential code version to a parallel one therefore requires only minimal modifications by the user.

The evaluator which actually instantiates and executes the constructed solver does so by invoking the corresponding run() method of the solver, which will normally be a specialization of the CombTechSolver. In the CombTechSolver, subproblem solvers of the type passed to CombTechSolver are created for each subproblem, executed on the subproblem and immediately destroyed again in order to release all their resources as soon as possible.

Subproblems are solved by solvers of the FullTransportSolver type. This solver proceeds like a typical FEM solver by creating and refining a triangulation, distributing degrees of freedom on the mesh, setting up system matrix and right hand side vector and solving the linear system. Initially, it also invokes the passed velocity generator to produce a list of discrete ordinates with quadrature weights.

An example for the benefits of generic programming is the setup of the scattering factor matrix in the matrix assembly routine of the FullTransportSolver. The scattering factor matrix (cf. Sec. 5.2.5) contains the entries

\[ \omega_{ij} = w_j \Phi(s_i, s_j) \]

products of the quadrature weights \( w_j \) and evaluations of the scattering kernel \( \Phi(s_i, s_j) \) for different directions. Its assembly happens through a call to the constructor of the ScatterFactorsAssembler struct, whose sole purpose is to compute the scattering factor matrix:

```cpp
ScatterFactorsAssembler<RTApp::ScatteringKernelFunctionVV,
    RTApp::spacedimV> scfa(solution.directions(),
    solution.quad_weights(), scatter_factors);
```
7.2 Software design

(a) Sequential program.

```c
// Includes and namespace directives ...
int main(int argc, char *argv[]){

    const int dim = 3; // Phys. dimension
    // Define application
typedef GaussdegScatApp<dim> MyApp;

    // Define method
typedef Method<dealii::FE_Q<dim>, 1,
        dealii::QGauss<dim>, 2,
        RTVarFormSUPG,
        dealii::SolverBicgstab
    > MyMethod;

    // Define solver
typedef CombTechSolver<MyApp,
        SparseSparsityProfilePolicy,
        FullTransportSolver<MyMethod,
            MyApp,
            StandardDOMVelocityGenerator<dim> >
    > MySolver;

    // Define evaluators
typedef EvalChain<MySolver,
        EvalRunTime<MySolver>,
        EvalIncRad<MySolver>
    > Evalu;

    Evalu eval; // Create objects
    MySolver::Solution sol;
    Evaluator::ResultCollection rescol;
    // Run evaluators (and the solver)
    eval.run(sol, rescol);

    return EXIT_SUCCESS;
}
```

(b) Parallel program.

```c
// Includes and namespace directives ...
int main(int argc, char *argv[]){
    PetscInitialize(&argc, &argv, 0, 0);
    {
        const int dim = 3; // Phys. dimension
        // Define application
typedef GaussdegScatApp<dim> MyApp;

        // Define method
typedef Method<dealii::FE_Q<dim>, 1,
            dealii::QGauss<dim>, 2,
            RTVarFormSUPG,
            parallel::SolverPETSc,
            dealii::SolverBicgstab
        > MyMethod;

        // Define solver
typedef CombTechSolver<MyApp,
            SparseSparsityProfilePolicy,
            parallel::FullTransportSolver<MyMethod,
                MyApp,
                StandardDOMVelocityGenerator<dim> >
        > MyParallelSolver;

        // Define evaluators
typedef EvalChain<MyParallelSolver,
            EvalRunTime<MyParallelSolver>,
            EvalIncRad<MyParallelSolver>
        > Evalu;

        Evalu eval; // Create objects
        MyParallelSolver::Solution sol;
        Evaluator::ResultCollection rescol;
        // Run evaluators (and the solver)
        eval.run(sol, rescol);
    }
    PetscFinalize();
    return EXIT_SUCCESS;
}
```

Figure 7.1: Comparison of sequential and parallel main method. Differences are minor, they amount to prepended `parallel::` namespace qualifiers and exchanged class names. Some parts about input parameter handling identical in both versions have been omitted.
For a general scattering kernel function, all entries of this full matrix have to be computed separately. The general case is covered by the non-specialized template:

```cpp
template <class ScatteringKernelFunctionVV, int spacedimV>
struct ScatterFactorsAssembler {
    ScatterFactorsAssembler(const std::list<dealii::Tensor<1,spacedimV>>& dirs, const std::list<double>& quad_weights, dealii::FullMatrix<double>& scatter_factors) {
        // Compute entries of scattering factor matrix of this process ...
    }
};
```

For applications without scattering, nothing at all has to be computed. The specialization of the ScatterFactorsAssembler struct for scattering kernels of type NullFunction therefore computes exactly nothing:

```cpp
template <int spacedimV>
struct ScatterFactorsAssembler<NullFunction, spacedimV> {
    ScatterFactorsAssembler(const std::list<dealii::Tensor<1,spacedimV>>& dirs, const std::list<double>& quad_weights, dealii::FullMatrix<double>& scatter_factors) {};
};
```

Optimized routines for other scattering kernels could be provided by the user. Since the decision which routine to call happens already at compile time, additional decisions at run time by e.g. if-clauses or function overloading can be avoided and the compiler can even optimize the complete function call away if it notices that the function does nothing.

The type of call to the assembly of the scattering factor matrix has a low performance impact as this code is executed only once during the run of a subproblem solver. However, the situation might be different for the innermost lines of code in the loops of the system matrix assembly. To be able to exchange the type of variational formulation flexibly, we also supply this type via the Method template parameter to the FullTransportSolver. In the system matrix assembly routine we create the variational form of the desired type and call its methods to compute single matrix entries:

```cpp
// Declare quad_rule, fe_values, direction and cell iterators,
// dofs_per_cell, transport_cell_matrix ...
// Create variational form
typename Method::template RTVarForm<RTApp>::type rt_var_form(quad_rule, fe_values);
// Loop over all directions, mesh cells, and DoFs per cell
for( ; direction != solution.directions().end(); ++direction){
    rt_var_form.initDir(*direction); // prepare for current direction
    for( ; cell != endc; ++cell){
```
7.3 Algorithms

7.3.1 Quadrature

As quadrature rule for integration in the physical domain we select a Gauss-Legendre rule with two quadrature points along each axis by default. This rule integrates polynomials up to degree 3 exactly, which should be sufficient for linear finite elements if the coefficient and right hand side functions do not vary too rapidly over the mesh width \( h \). The type of quadrature rule and its number of points can also be set by the user as a parameter to the Method class if the default should prove inadequate.

Angular quadrature is determined with the choice of ordinates in angle. In 2+1D applications, quadrature on the circle is performed using the trapezoidal rule on equidistributed points. For 3+2D applications, the user can provide arbitrary sets of

```c++
for(unsigned int ii = 0; ii < dofs_per_cell; ++ii){
    for(unsigned int jj = 0; jj < dofs_per_cell; ++jj){
        transport_cell_matrix(ii,jj) =
            rt_var_form.calcTransportCellMatrixEntry(ii, jj);
    }
}
```

Any variational form then implements the actual computation of the matrix entry:

```c++
template <class RTApp>
class RTVarFormSUPG<RTApp> {
    // Constructors, member variables, coeff. functions etc. ...
    inline double calcTransportCellMatrixEntry(unsigned int kk,
                                               unsigned int ii){
        // Compute SUPG var. form for local DoFs on current cell ...
    }
};
```

With function inlining, the call to `calcTransportCellMatrixEntry()` can be replaced directly by the code of the method so that the additional layer of abstraction does not lead to any performance penalty. As the code in question is executed several thousand or million times during the computation of a subproblem solution, the performance difference between our generic programming solution and e.g. overloaded methods with virtual function calls could in fact become noticeable here, though we did not test this explicitly.

In summary, we use generic programming in various places to maintain flexibility while obtaining fast, tightly coupled executable code since branching decisions are brought forward to compile-time and irrelevant branches are completely omitted from the executable.
7 Implementation

points in tabular files of 4-tuples (coordinates and weight). Provided in the software are centers of a centroidal Voronoi tessellation generated by a modified version of a program by Burkardt (2002). These points correspond to a midpoint rule on a partition of the sphere (cf. Secs. 5.2.3 and 5.2.4) Alternatively, any other point set, e. g. the maximum determinant points by Womersley and Sloan (2001), can be imported into the software.

7.3.2 Linear solver and preconditioner

When first order finite difference schemes are applied to linear advection problems without scattering, the values of degrees of freedom further downstream only depend on the values of degrees of freedom upstream according to the flow of information in such a problem. This means that all degrees of freedom can be computed by sweeping over them in the direction of advection (so-called upwinding technique, see Kanschat (1996, Sec. 3.3.1), for example). Together with a downstream ordering of the degrees of freedom, the system matrix becomes lower triangular, and the computation sweep corresponds to a Gauss-Seidel iteration which converges in one step.

In Discontinuous Galerkin methods, the degrees of freedom of one mesh cell only depend on degrees of freedom in other upstream cells via the flux along the transport direction. With downstream ordering of cells, a block lower triangular structure appears in the system matrix so that the linear system can be solved efficiently using a block Gauss-Seidel preconditioner (Hartmann, 2002, Sec. 2.8.1).

Here, in our SUPG method with bi- or trilinear finite elements and nodal degrees of freedom, the value of each degree of freedom in general depends on all neighboring degrees of freedom, that is, on all degrees of freedom whose nodes in the mesh are connected by an edge or face. The $\delta$ parameter of the SUPG method (cf. Eq. (3.20)) is chosen as to reduce the influence of downstream degrees of freedom, but in general this influence is still nonzero, therefore there is no ordering of degrees of freedom which would result in a strictly lower-triangular system matrix.

Furthermore, in full transport problems as they arise in applications with scattering the physical transport subproblems for all directions are coupled and treated in one linear system. Thus there exists no single direction to order all degrees of freedom by. Rather, one would have to sort degrees of freedom in each direction block differently.

Hence, even though Gauss-Seidel preconditioning is a common choice for transport dominated problems, we investigated other preconditioners as well. In particular, we were interested in a preconditioner-solver-pair that converges quickly (with quickly referring to overall solution time rather than the sole number of iterative steps) and robustly for most applications, has a low memory footprint and most of all is already available as a parallel distributed implementation in the PETSc library.

After a series of tests with results reproduced partially in Table 7.1, we decided upon a combination of block ILU preconditioning (e. g. Saad, 2003) with zero level of fill-in and one of the Krylov subspace solvers BiCGStab (van der Vorst, 1992) or GMRES (Saad and Schultz, 1986). While BiCGStab showed smaller numbers of iteration, memory usage and run times of both solvers were almost identical. However, as BiCGStab
### 7.3 Algorithms

<table>
<thead>
<tr>
<th></th>
<th>$L = 6, N = 4$</th>
<th>6.8 MDoFs</th>
<th>$L = 3, N = 32$</th>
<th>0.8 MDoFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block ILU(0)/GMRES</td>
<td>165</td>
<td>89.9</td>
<td>111</td>
<td>413</td>
</tr>
<tr>
<td>Block ILU(0)/BiCGStab</td>
<td>155</td>
<td>92.1</td>
<td>54</td>
<td>436</td>
</tr>
<tr>
<td>Block SOR/GMRES</td>
<td>1011</td>
<td>92.1</td>
<td>1000*</td>
<td>450</td>
</tr>
<tr>
<td>Hypre Euclid(0)/BiCGStab</td>
<td>&gt;3600</td>
<td>138.4</td>
<td>n. a.</td>
<td>&gt;3600</td>
</tr>
</tbody>
</table>

Table 7.1: Performance of preconditioner-solver-pairs for two full tensor problems in 3D with Henyey-Greenstein scattering on 96 cores. The asterisk * indicates that the maximum number of iterations had been reached.

is known to suffer from breakdowns (Graves-Morris, 2002), we utilized GMRES for the performance tests later on. Note that the solver and preconditioner type can be changed comfortably by PETSc command line switches passed to our software if the need should arise.

In most applications, with an ILU level of fill-in of zero we achieved already good results. However, when applying this preconditioner-solver-pair to purely physical transport problems on a 2D unit square domain with a square mesh, we observed that the number of iterative steps would increase for directions diagonal to the coordinate axes and mesh axes (see Härdi, 2012, Fig. 1). This effect was intensified for higher physical resolution levels.

One solution to this problem was to set the ILU level of fill-in to values of 1 or 2, with the consequence of a higher memory usage. As the larger memory footprint could have proved detrimental to the solution of very large problems, we decided instead to combine the purely physical subproblems into one full subproblem even in applications without scattering and solve this larger linear system in parallel, since similar convergence issues did not occur in the full combined system.

In parallel, the chosen ILU preconditioner simply works on the block of rows owned by the local process without communicating with other processes. Communication takes place during matrix-vector-multiplication steps in the BiCGStab solver. Then, each process multiplies the iteration vector with the matrix rows it owns and shares the result with all other processes. Compared to sequential mode where the preconditioner can work on the whole matrix, the number of iterations increases slightly.

#### 7.3.3 Scheduling

When solving a radiative transfer problem with the sparse DOM, we have to compute solutions to a number of full tensor subproblems. These subproblems will in general all have different resolution parameters $L, N$ and can therefore vary considerably in
Implementation

their numbers of degrees of freedom.

Furthermore, in applications without scattering, transport problems for different
directions decouple and could be solved separately. However, we have already seen
in the previous section about preconditioning that it is beneficial for the convergence
of the linear solver to combine problems for different directions also in applications
without scattering. Also, splitting a full tensor subproblem of low physical resolution
$L$ and high angular resolution $N$ will result in a large number of very small physical
transport problems. Solving all these small problems individually multiplies the
overhead associated with the setup of a transport problem, too, and is therefore most
likely less efficient than tackling these problems together. For these reasons we refrain
from splitting up full tensor subproblems into single physical transport problems here.

Remark 15. A possible refinement of this decision in the future would be to split only
those full tensor subproblems into physical transport problems for which $N$ is below a
prescribed limit.

Still, the question remains how the subproblems can be distributed to available cores
in a parallel run most efficiently. In the literature on combinatorial problems, this task
called the “minimum makespan problem for parallel identical machines” is known to
be NP-complete (e.g. Hochbaum and Shmoys, 1987), which means that there exist no
polynomial time algorithms producing an optimal schedule (unless P=NP). Suboptimal
schedules can be computed polynomial time, often with worst-case bounds on the
objective function. A simple approximation scheme is the Longest Processing Time
(LPT) strategy, which sorts all problems descendingly according to their sizes and then
assigns them one after the other to the core with the least workload so far. Schedules
of this scheme achieve run times within $4/3 - 1/(3n_{\text{cores}})$ of the optimal times (also cf.
article by Hochbaum and Shmoys (1987) and references therein). The LPT strategy has
been implemented with some modifications in the SchedulingStrategyLPT class.

However, in typical subproblem sets such as those in Table 7.2, which arise from the
use of the combination technique with a minimum resolution level $N_0$ (cf. Sec. 4.3.4),
the largest problems can be several times larger than the smallest problems. Assigning

<table>
<thead>
<tr>
<th>Problem no.</th>
<th>$L$</th>
<th>$N$</th>
<th>#DoFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>4</td>
<td>6,865,625</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>972</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>4</td>
<td>675</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>8</td>
<td>648</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>5</td>
<td>288</td>
</tr>
</tbody>
</table>

Table 7.2: Example of full tensor subproblems to solve in the combination technique
for a 3D application with resolution levels $L = 6$, $N = 8$, $N_0 = 4$.

single problems to single cores would therefore produce very disparate workloads
on the cores, and even worse, restrict the maximum subproblem size to the available
resources of one core only. Instead, it must be possible to solve every subproblem in a
distributed manner on several cores.
7.4 Parallel performance results

With this modification, the previously NP-complete scheduling problem becomes much simpler to solve. It reduces essentially to the scheduling problem with preemption described by Brucker (2007, Sect. 5.1.5, referred to as \((P | pmtn | C_{\text{max}})\) problem). Ignoring the overhead of parallel solution of a subproblem, the straightforward approach of splitting all subproblems and distributing them evenly to all cores for parallel execution produces an optimal schedule in linear time with respect to the number of subproblems.

Hence, this strategy, implemented in the SchedulingStrategySplitAll class, is our preferred scheduling algorithm. To take the overhead of parallel execution into account, we order the subproblems descendingly by size and do not distribute problems below a certain size limit, but assign them in a round robin fashion to single cores. The size limit can be adapted by the user, we chose it relatively low at around one thousand degrees of freedom per core so that these sequentially solved problems affect the total run time only negligibly.

7.4 Parallel performance results

After describing the internals of our software in the previous sections we now analyze its performance in various settings and compare with the initially stated goals.

7.4.1 Computing environment

All run time tests were conducted on the ETH Zürich cluster Brutus (Brutus cluster, 2013), a heterogeneous computing environment comprising about 20,000 cores in differing configurations. As our software assembles the large FEM stiffness matrices in memory, it is memory-bound even if these matrices are sparse. Hence we ran our tests on some of the 80 large-memory (“fat”) nodes, each equipped with four 12-core AMD Opteron 6174 CPUs and 256 GB of RAM.

The AMD Opteron 6174 CPU “Magny-Cours” is clocked at 2.2 GHz and consists of two 6-core processors in one module (Stiller, 2010). Each core has 512 KB L2-cache and a 12 MB L3-cache is shared among all twelve cores of one module. Memory is connected via one dual channel DDR3 memory controller per processor, i.e. six cores share one memory controller. Access to memory not directly attached to the controller of a processor has to be routed over a bus system through the processor attached to the requested memory. Therefore access times to different parts of RAM may vary for one processor, and if all cores are trying to access memory at the same time, congestion on the memory bus might result.

All fat nodes of Brutus are connected to an InfiniBand QDR network which transmits the MPI communication during a program run.

The C++ compiler in use was gcc version 4.7.0. MPI library open mpi version 1.6.2 was tasked with interprocess communication. We submitted our program to the batch system of Brutus with the following command:
In this command, the desired number of processes to be run in parallel is split up into \(\langle \#\text{nodes} \rangle\), the number of computing nodes, and \(\langle \#\text{processes/node} \rangle\), the number of processes to launch on each node. The total number of parallel processes is then obviously determined by the product of \(\langle \#\text{nodes} \rangle\) and \(\langle \#\text{processes/node} \rangle\). This way, we ensure that we always request complete nodes so that no other user application can run during our timings on the same nodes, even if \(\langle \#\text{processes/node} \rangle\) is less than 48, the number of cores in a fat node. However, interference from operating system processes cannot be prevented.

The statement \'select[model==... mem=5000]\' lets the batch system allocate virtually all the available memory of a fat node to our program \((48 \cdot 5000 \text{MB} \approx 240 \text{GB}, \text{some memory is always occupied by the operating system})\) and allows this memory to be used by all processes which are launched on one node.

The mpirun option -bysocket distributes the processes on one node in a round-robin way over the eight processors of each node. Note that each Opteron 6174 counts as two processors to the operating system. This option should level the load on the memory controllers on one node.

The -bind-to-core option instructs the operating system not to move a process between different cores and thus helps to avoid time-consuming cache synchronization events during a program run.

All measurements have been conducted on the example application of a degenerate Gaussian with \(\kappa = 1\) and \(\sigma = 0.5\), see Sec. 8.2 for details.

Run times reported here comprise initial MPI setup, problem solution, and finalization of all processes.

### 7.4.2 Strong scaling

Two important performance measures for parallel software are the so-called strong and weak scaling. We will discuss strong scaling in this section, and weak scaling in Section 7.4.3.

Strong scaling measurements answer the question how well a program can take advantage of additional cores if the total load, the problem size, remains constant. Ideally, if the number of cores the program is executed on is \(n_{\text{cores}}\), one would hope for the run time to scale with \(n_{\text{cores}}^{-1}\). In practice, however, it is mostly the case that not the complete program can run independently in parallel. Frequently the duration of setup and communication phases increases with a larger number of cores. For this reason one often observes for a fixed problem size that there is an upper limit of cores beyond which the total run time does not decrease further, on the contrary, it rather increases again. Occasionally, run times profit from memory locality effects: With a larger number of cores, the amount of memory allocated by each process reduces so
7.4 Parallel performance results

Figure 7.2: Strong scaling for examples with Henyey-Greenstein scattering in 3D, full tensor DOM.

that it might fit into faster levels in the system memory hierarchy. Effects such as these sometimes lead to superlinear speedup, i.e. run times decrease faster than \( n^{-1} \) cores.

How well a program actually profits from additional cores is expressed in the parallel efficiency \( E_{\text{strong}} \). This variable relates the run time \( t_1 \) of a program on one core to the run time \( t_n \) on \( n_{\text{cores}} \) cores:

\[
E_{\text{strong}}(n_{\text{cores}}) := \frac{t_1}{n_{\text{cores}} t_n}. \tag{7.1}
\]

A value of 1 means linear scaling, practical values are usually below 1, values larger than 1 indicate superlinear speedup. If the run time \( t_1 \) on a single core is not available because the problem is too large to be solved on one core, for instance, then the definition of the parallel efficiency can be extended to

\[
E_{\text{strong}}(n_{\text{cores}}, n'_{\text{cores}}) := \frac{n'_{\text{cores}} t_{n'}}{n_{\text{cores}} t_n}, \tag{7.2}
\]

with \( n'_{\text{cores}} < n_{\text{cores}} \) the smaller number of cores to compare the run times to.

Next we present strong scaling results for a number of configurations.

**Full tensor DOM with scattering**

Figure 7.2 shows two strong convergence plots for problems of different resolution which have been solved with the full tensor DOM. We obtain strong scaling with efficiencies over 50% or 40%, resp., over approximately two orders of magnitudes of numbers of cores. In the smaller problem in Fig. 7.2a, we included the single core run of a purely sequential implementation.
Figure 7.3: Strong scaling for examples with Henyey-Greenstein scattering in 3D, sparse tensor DOM.

In this problem, we also see that the parallel efficiency is greater than 1 for an intermediate range of the number of cores. This superlinear speedup is most likely due to cache effects as the sequential code used about 27 GB of memory, the parallel code for eight cores on the other hand only 2.6 GB per core, which easily fits into the local memory of one CPU.

**Sparse tensor DOM with scattering**

In Figure 7.3 we present two runs with the sparse tensor DOM. The parallel efficiency remains above 40% over two orders of magnitude of number of cores in the smaller problem in Fig. 7.3a, and slightly over one order of magnitude in the larger problem in Fig. 7.3b.

It is noticeable that the run times decrease in a more irregular way than for the full tensor DOM, possibly because the ratio of problem setup operations to pure calculations is higher (more subproblems have to be set up and solved) and because the overall run times at the same number of cores are smaller (but observe that the chosen resolution levels may vary between full and sparse).

**Sparse tensor DOM without scattering**

Some scaling results for 2D applications without scattering have been presented by Härdi (2012) already, working on an earlier version of our software. Here we extend these results to greater numbers of cores.

Without scattering, we achieve efficiencies of 50% and more for core numbers between 1 and around 100, as can be seen from Fig. 7.4. Superlinear scaling does not appear
7.4 Parallel performance results

Figure 7.4: Strong scaling without scattering in 3D, sparse tensor DOM. $L = 6$, $N = 8$, $N_0 = 4$, 6.9 MDoFs. Parallel efficiency is $E_{\text{strong}}(n_{\text{cores}})$, cf. Eq. (7.1).

in this setting, possibly because the memory requirements of the version without scattering are far lower. At a memory footprint of 9.2 GB, the sequential single core version still fits comfortably into the 32 GB local RAM of one CPU.

**Comparison to expectations**

As the parallel solution of linear systems of equations could not be avoided with the subproblem size distribution at hand, strong scaling efficiencies of more than 40% over two orders of magnitude in the number of cores are satisfactory results. The code achieves very good scaling at low core numbers compared to the sequential single core version, and if the problem size is sufficiently large, four digit core numbers as available in Brutus can still be put to use.

A limitation certainly is the memory requirement of the version with scattering of around 20 – 40 KB per degree of freedom already at lower angular resolution levels, compared to resource needs of the order of 1 KB per degree of freedom for the non-scattering version. In future versions of our software, either some form of operator compression (e.g. ACA, hierarchical matrices) or a matrix-free implementation should reduce the memory requirements to bring even higher resolutions or larger problems into reach on the same hardware.

7.4.3 Weak scaling

Weak scaling tests try to answer the question how the communication overhead in a parallel program scales when the number of cores is increased along with the problem size so that the workload per core remains constant. To find out by how much the core count has to be increased when the problem size is raised we first require some kind of work model that predicts how the total workload grows with the problem size.
The number of degrees of freedom might serve as a straightforward measure for the problem size. However, problems of similar numbers of degrees of freedom might possess very different resolution levels in physical and angular space, and the corresponding system matrices also show different structure. While a system matrix stemming from a full transport problem of high physical resolution and low angular resolution will consist of few large, but rather sparse blocks, a matrix from a problem of low physical, but high angular resolution will be made up of many smaller blocks with a larger percentage of nonzeros so that the total matrix will be less sparse, too. Clearly, the resulting workload could differ among these two cases, not to mention the situation when applications with scattering are compared to those without.

Therefore we first conducted tests to analyze how the program run times scale with the input parameters full/sparse method, physical resolution $L$, angular resolution $N$, and scattering kernel in practice.

Figures 7.5–7.6 show that run times scale mostly linearly with the number of degrees of freedom when the resolution parameters are increased. Only when the angular resolution is increased in applications with scattering, run times depend on the number of degrees of freedom to the fourth order, even though the number of iterations of the linear solver remains constant. The slowdown rather has to do with the fact that the system matrix becomes very populated for larger $N$. For the subproblems of highest angular and lowest physical resolution in the combination technique, the matrix might even fill up completely. A representation of the system matrix in a sparse matrix data type is then certainly suboptimal, matrix operations become costly. Choosing an appropriate data type dynamically could possibly alleviate this problem.

Except for the case of angular refinement with scattering, our scaling goal for the total run times of $O(M \log M)$ from Sec. 7.1.4 with $M$ being the total number of degrees of freedom appears to have been achieved.

With these run time scalings as model for the dependence of the workload on the number of degrees of freedom, we proceed to the weak scaling measurements.

From Figure 7.7 we see that the parallel overhead grows considerably when the physical resolution $L$ is increased along with the number of cores, which is coupled linearly to the number of degrees of freedom. This also leads to approximately the same memory load per core in each run.

For increasing angular resolution (Fig. 7.8) we also notice a rise in run times, more irregular in the sparse method than the full method. Here, the number of cores was scaled up in order to keep the memory requirements per core approximately constant.

Splitting run times into the parts of setup of the degrees of freedom, system matrix assembly, and solution of the linear system in the full tensor examples in Figs. 7.7–7.9 reveals that the linear solver is responsible for the largest share of the parallel overhead. As problem sizes increase, the number of linear solve iterations grows accordingly necessitating more communication. This is the price to pay for the memory efficiency of our preconditioner-solver-pair. However, since memory was the scarcer resource than run time and commonly fewer than 1000 cores were available on Brutus for computations, only this memory conservative solver allowed to solve larger problems on a moderate number of cores in reasonable times.
7.4 Parallel performance results

Figure 7.5: Scaling of run times for fixed number of cores with varying resolution. 3D examples with Henyey-Greenstein scattering. With physical resolution $L$, run times scale approximately linear in (a) and (b), whereas with angular resolution $N$ in (c) and (d), scaling tends to fourth order for larger problems.
# DoFs

## Implementation

(a) Full: $L = 5$, $N = \{8, 16, 32, 64\}$.

(b) Sparse: $L = 6$, $N = \{8, 16, 32, 64, 128, 165\}$, $N_0 = 1$.

Figure 7.6: Scaling of run times in 3D examples without scattering when angular resolution $N$ is varied for $n_{\text{cores}} = 96$. Scaling is linear in the number of DoFs for the full method, or less than linear for the sparse method.

(a) Full, $L = \{4, 5, 6\}$, $N = 4$.

(b) Sparse, $L = \{4, 5, 6, 7\}$, $N = 8$, $N_0 = 3$.

Figure 7.7: Weak scaling with physical resolution for examples with Henyey-Greenstein scattering in 3D.
7.4 Parallel performance results

Figure 7.8: Weak scaling with angular resolution for examples with Henyey-Greenstein scattering in 3D.

Figure 7.9: Weak scaling for examples in 3D. Full, no scattering, $L = 5$, $N = \{8, 16, 32, 64\}$. 
8 Numerical experiments

This chapter presents the results of a number of numerical experiments.

In the first section, we compare the spherical harmonics method and the discrete ordinates method in 2D applications with source functions of different regularity. One aim of this section is to check that approximation rates of the spherical harmonics method derived in Thms. 4.7 and 4.8 without boundary conditions also hold for the solution of the RTE satisfying zero inflow boundary conditions in a strong sense.

In Section 8.2, we analyze the performance of the sparse DOM against that of the full tensor version in 3D applications.

To quantify the approximation error, we analyze the error of the incident radiation because of its importance in applications and for visualization (cf. Sec. 1.1). The relative error in the incident radiation is given as

\[ \text{err}(G_{L,N})_{H^k(D)} = \frac{\| G - G_{L,N} \|_{H^k(D)}}{\| G \|_{H^k(D)}}, \quad k = 0,1, \]  

(8.1)

where \( G \) is the exact solution for the incident radiation. This error relates to the error of the intensity solution by

\[ \text{err}(G_{L,N})_{H^k(D)} \lesssim \| u - u_{L,N} \|_{H^{k,0} (\Omega)}. \]

In experiments where no exact solution is available we approximate the errors by using a reference solution for \( u \) or \( G \), respectively, computed at high resolution.

8.1 Spherical harmonics and discrete ordinates in 2D

In the results of this section, we will use the following abbreviations to denote the different methods: ST stands for the sparse tensor spherical harmonics method and FT for the full tensor spherical harmonics method. By SC, we refer to the sparse combination technique discrete ordinates method, and by FC to its standard full version.

Except for the experiments in which the combination technique has been applied to the spherical harmonics method, the results of this section have been presented before (Grella and Schwab, 2011b).

8.1.1 Algorithms

As the spherical harmonics method has been implemented in a MATLAB code for 2D applications, we compare the four methods with the aid of this software.

Quadrature for integrals over the physical domain arising in the assembly of the stiffness matrix is performed by a Gauss-Legendre rule which is exact up to rounding
errors for products of piecewise affine nodal basis functions and constant coefficients. On the angular domain, the trapezoidal rule integrates periodic functions up to rounding errors. When exact integration is not achieved with these quadrature rules such as in integrals of the load vector or of products of full spherical harmonics and spectral functions over angular regions, iterative refinement ensures a relative integration error tolerance of $10^{-13}$.

The linear systems are solved by the BiCGStab algorithm (van der Vorst, 1992) with Jacobi preconditioning. We terminate the solver if the $\ell^2$-norm of the residual vector is less than $10^{-16}$.

In experiments without known solution, we compute a reference solution with a discrete ordinates method with method of lines/ray-tracing in the physical domain. Without scattering, ray tracing corresponds to integration along straight characteristics. In this reference method, the angular domain is discretized by choosing 256 directions of propagation, along which we calculate the solution by the method of lines with a standard non-stiff integrator in MATLAB. The line integrals are then interpolated to the FEM mesh in the physical domain corresponding to a resolution of $L = 7$.

### 8.1.2 Experiments

All experiments have been conducted on the physical domain $D = [0, 1]^2$, the unit square, with zero inflow boundary conditions. The absorption coefficient function is constant $\kappa(x) = 1$. Here, scattering is not included in the experiments ($\sigma \equiv 0$), it will be covered in the next section.

The spherical harmonics method utilizes basis functions as described in Sec. 4.2.

In the discrete ordinates method, we use hierarchical hat functions on a square mesh with mesh size $h = 2^{-L}$ as basis in the physical domain. Also, the relation $M_S = 2N + 1$ from (3.35) is adapted to $M_S = 2N + 3$ so that in the sparse DOM, each physical subproblem is solved in at least three different directions. $N_0$ is set to zero.

To determine the approximation rates of physical and angular approximation separately, we refine the physical resolution while fixing the angular resolution or vice versa. However, in normal operation one would rather use an equilibration relation to increase the resolution in $D$ and $S$ in a combined way. Experiments 2 and 3 are examples for combined refinement.

In the sparse DOM, the reported numbers of degrees of freedom represent the sum of all the degrees of freedom of the solved full physical subproblems.

**Experiment 1**

This example is an application with a degenerate Gaussian on the right hand side:

$$I_b(x) = \exp \left( -8(x - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix})^\top \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} (x - \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}) \right).$$
8.1 Spherical harmonics and discrete ordinates in 2D

Figure 8.1: Experiment 1: Incident radiation for resolution $L = 4/N = 16$ computed by (left to right, top to bottom) full spherical harmonics method (FT), sparse spherical harmonics method (ST), full DOM (FC), and sparse DOM (SC).
Figure 8.2: Experiment 1: Magnitude of active degrees of freedom (left full spherical harmonics method, right sparse spherical harmonics method). DoFs are ordered from lowest to finest resolution. Block columns correspond to different angular regions (cf. Fig. 4.4).

Figure 8.3: Experiment 1: Convergence in incident radiation (Eq. 1.2) of numerical solution $G_{L,N}$ to reference solution $G$ for $L = 0, 1, \ldots, 5$ and $N = 32$. 
8.1 Spherical harmonics and discrete ordinates in 2D

Figure 8.4: Experiment 1: Convergence in incident radiation (Eq. 1.2) of numerical solution \( G_{L,N} \) to reference solution \( G \) for \( N = 1, 2, 4, 8, 16, 32 \) and \( L = 5 \).

**Full and sparse spherical harmonics and discrete ordinates.** Fig. 8.1 compares the incident radiation from each of the four methods. Both full methods agree nicely. The incident radiation from the sparse spherical harmonics method appears broadened in comparison to the full spherical harmonics solution. As degrees of freedom representing higher combined variation have been omitted in the sparse solution, it necessarily becomes smoother or more diffused.

The full and sparse DOM incident radiations also agree qualitatively, while the sparse solution is noticeably rougher. However, the sparse solution has been computed with less than a quarter of the degrees of freedom of the full DOM solution.

Fig. 8.2 displays the magnitude of the degrees of freedom in the spherical harmonics method. We observe a decay for angular degrees of freedom of finer resolution, especially in the angular regions where Legendre polynomials are used as basis functions. The decrease of the physical degrees of freedom is also visible, although not as pronounced. The decay justifies the approach of the direct sparse approximation for hierarchic bases in the spherical harmonics method, where degrees of freedom of higher combined variation are omitted from the sparse approximation space.

Fig. 8.3 shows that the estimated convergence rates of 1/2 for the \( H^1 \)-error in refinement of the physical resolution are actually achieved for all four methods. Best in terms of efficiency, i.e. error per employed degrees of freedom, is the ST method, followed by the SC method. Both full tensor methods perform very similar with a larger ratio of error vs. number of degrees of freedom.

When only the angular resolution is refined (see Fig. 8.4), convergence rates of \( M_S^{-1} \)
are attained between lower resolutions. For higher resolutions the combined error saturates with the contribution from the physical side, as can be seen by the fact that the error reaches the value for \( L = 5 \) from the physical refinement study in Fig. 8.3.

**Combination technique for spherical harmonics.** As a comparison, we also solve the full subproblems in the combination technique by the full spherical harmonics method.

Discontinuities of the spectral functions on the angular regions appear to cause errors on the boundary, which are not present in the solution of the direct sparse approach (compare Fig. 8.5 with Fig. 8.1b).

Together with the inevitable overhead in the number of degrees of freedom of the combination technique, this method performs slightly worse than the direct sparse spherical harmonics method in terms of error per degrees of freedom, but still better than the full spherical harmonics method. Furthermore it is noteworthy that for the finest resolution level \( L = 5 \), convergence in Fig. 8.6 is still maintained, as opposed to the case for the combination technique with discrete ordinates in Fig. 8.3. This indicates that the combination technique as such works for radiative transfer in practice and that specific properties of the discrete ordinates method (namely the ray effect) are the reason for the slowing of the convergence in Fig. 8.3.

**Experiment 2**

The blackbody intensity for this experiment is chosen as the \( C_0^\infty (D) \) function

\[
I_b(x) = \begin{cases} 
10^4 \exp \left( \frac{-1}{0.25 - (x_1 - 0.5)^2} + \frac{-1}{0.25 - (x_2 - 0.5)^2} \right) & \text{if } 0.25 < x_1, x_2 < 0.75, \\
0 & \text{else.}
\end{cases}
\]

In this experiment, we refine in a combined manner: \( L \) is increased from 0 to 5 and \( N = \min\{2^{L+2}, 32\} \).
8.1 Spherical harmonics and discrete ordinates in 2D

Figure 8.6: Experiment 1: Convergence in incident radiation (Eq. 1.2) of numerical solution $G_{L,N}$ to reference solution $G$ for $L = 1, \ldots, 5$ and $N = 32$ with combination technique for spherical harmonics (HY), sparse spherical harmonics (ST), and full spherical harmonics (FT) method.

Figure 8.7: Experiment 2: Error in incident radiation $G$ for combined refinement $L = 0, 1, \ldots, 5$ and $N = \min\{2^{L+2}, 32\}$. 
Figure 8.8: Experiment 3: Error in incident radiation $G$ for combined refinement $L = 0, 1, \ldots, 5$ and $N = \min\{2^{L+2}, 32\}$.

With a smooth right hand side, we observe in Fig. 8.7 that the expected convergence rate of $1/2$ for the $H^1$-error is again achieved for the full tensor methods. The full DOM is slightly more efficient with a lower error per employed DoFs ratio. The sparse methods almost attain a rate of $1/2$ in the $H^1$ error, the difference to $1/2$ can be attributed to the log-factor in the error estimate. In terms of efficiency, both sparse methods are approximately equal, but outperform the full tensor methods. The $H^1$-error of the sparse DOM is up to 20% lower than the one of the full DOM, for the spherical harmonics methods, the reduction is even greater.

From this experiment we may conclude that the spherical harmonics method can actually realize the expected approximation rates also when boundary conditions are respected if the solution is sufficiently regular, at least up to variations which are indiscernible in our convergence plots.

**Experiment 3**

Here, the blackbody intensity represents a characteristic function of a circle in the physical domain:

$$I_b(x) = \begin{cases} 1 & \text{if } (x_1 - 0.5)^2 + (x_2 - 0.5)^2 < \frac{1}{4}, \\ 0 & \text{else.} \end{cases}$$

We again perform combined refinement with $L = 0, \ldots, 5$ and $N = \min\{2^{L+2}, 32\}$. 
Due to the lower regularity of the right hand side, the approximation rates of all methods and in all errors are lower than in the previous experiment. We achieve a rate of about $1/3$ in the $H^1$-error for both full and sparse methods (cf. Fig. 8.8). Again, for the same number of degrees of freedom, the sparse spherical harmonics method yields the smallest error, followed closely by the sparse DOM. While the sparse DOM converges more irregularly, for the finest resolution, the $H^1$-error even increases slightly because $N$ has been capped at 32, it is still noteworthy that also in this setting of lower smoothness, the sparse methods maintain an advantage in efficiency over the full tensor methods.

8.2 Discrete ordinates in 3D

The experiments in three dimensions have been conducted with our transport solver software described in Chap. 7. All experiments are set on the unit cube $[0, 1]^3$ as physical domain. If no exact solutions are known, reference solutions have been computed by the full DOM with resolution levels $L_{\text{ref}}$ and $N_{\text{ref}}$ as indicated at the respective experiment.

Experiment 1

We begin by testing with an exactly known prescribed solution

$$u(x, s) = \frac{3}{16\pi} (1 + (s \cdot s')^2) \prod_{i=1}^{d} (-4x_i(x_i - 1)),$$

with fixed $s' = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^\top$. In the angular part, we reuse the dipole scattering kernel from Rayleigh scattering to model a smooth function, the physical part represents a second order parabola. Absorption is constant at $\kappa = 1$, isotropic scattering (3.8) is applied with $\sigma = 0.5$. We compute the right hand side $f$ by evaluating
Figure 8.10: Experiment 2: Incident radiation for $L = 4$, $N = 10$ of full DOM (left) and sparse DOM with $N_0 = 4$ (right). The contour manifolds are spaced logarithmically. The facing octant of the cube has been clipped to expose the interior. Observe the different scales of the colorbar.

the RTE for the given solution, then solve the problem with the computed right hand side, which in general depends on $x$ and $s$ now.

The convergence behavior for combined refinement as shown in Fig. 8.9 is as expected: The full DOM exhibits the curse of dimension, the convergence rate in the $H^1$-error is about $1/(d + d_S) = 1/5$. The sparse DOM on the other hand achieves a rate of $1/3$ or even better with respect to the total number of degrees of freedom, which corresponds to the convergence rate on a purely physical three-dimensional problem.

Experiment 2

This experiment extends the two-dimensional experiment 1 from the previous section to three dimensions. The smooth blackbody radiation function now reads

$$I_b(x) = \exp(-8|x - \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}^T \begin{pmatrix} 8 & -2 & -2 \\ -2 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} [x - \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}]).$$

Absorption is $\kappa = 1$, scattering is of Henyey-Greenstein type (cf. Eq. (3.10)) with $\sigma = 0.5$. No radiation is entering the domain from the walls. Reference solutions in this and the following experiments are obtained by the full discrete ordinates method of high resolution.

Full and sparse DOM incident radiation for the same resolution levels as shown in Fig. 8.10 agree qualitatively. In the sparse solution, values in the vicinity of the boundary appear to be disturbed most. This is where ray effects have the largest impact. Note, however, that the magnitude of these values is around one order lower than in the center of the domain so that overall accuracy variations distort the shape of the contour manifolds more strongly.

In the plot of physical convergence (Fig. 8.11 left) we can discern the rate of $1/3$ for the $H^1$-error of the incident radiation, in line with expectations from the theoretical
8.2 Discrete ordinates in 3D

Figure 8.11: Experiment 2: Convergence in incident radiation when \( L \) is refined (left, \( N_0 = \{1, 3, 3, 4, 5\} \)) or \( N \) is refined (right, \( N_0 = \{1, 1, 2, 3, 4\} \)). Reference resolution was \( L_{\text{ref}} = 5, N_{\text{ref}} = 14 \).

analysis. As the reference solution was computed with the full DOM, an acceleration of the convergence of the full method can be observed for higher resolutions. In angle, errors converge fast before they are saturated from the physical contributions (Fig. 8.11 right).

**Experiment 3**

This experiment uses a \( C^0(D) \) bump function with center \( c = (0.5, 0.5, 0.5)\top \) in the physical domain as blackbody radiation and absorption coefficient:

\[
I_b(x) = \kappa(x) = \begin{cases} 
0.5 - \|x - c\|_2 & \text{if } \|x - c\|_2 \leq 0.5, \\
0 & \text{else}.
\end{cases}
\]

Scattering is isotropic with \( \sigma = 1 \) throughout the domain, zero inflow boundary conditions are imposed.

Due to the isotropic nature of the problem the sparse discrete ordinates method performs quite well here. Fig. 8.12 shows good visual agreement of the full and sparse incident radiation for the same resolution levels, even though the sparse version uses only a third of the degrees of freedom.

In the error vs. degrees of freedom plot in Fig. 8.13, the sparse DOM yields a smaller error per employed degrees of freedom than the full DOM from \( L = 3 \). Strong scattering helps to suppress ray effects which do not pollute the solution if we set \( N_0 \approx L \).

**Experiment 4**

This experiment models a cloud of scattering particles illuminated from one side. The cloud is represented by a Gaussian centered at \( c = (0.5, 0.5, 0.5)\top \) in the scattering coefficient:

\[
\sigma(x) = 5 \exp(-8^2 \|x - c\|_2^2).
\]
Figure 8.12: Experiment 3: Incident radiation for $L = 4$, $N = 10$ of full DOM with 594,473 DoFs (left) and sparse DOM with $N_0 = 5$ and 178,124 DoFs (right). The contour manifolds are spaced logarithmically. The facing octant of the cube has been clipped to expose the interior. Observe the different scales of the colorbar.

Figure 8.13: Experiment 3: Convergence in incident radiation when $L$ and $N$ are refined. $N_0 = \{1, 3, 3, 4, 5\}$, reference resolution $L_{\text{ref}} = 5$, $N_{\text{ref}} = 14$. 
8.2 Discrete ordinates in 3D

Figure 8.14: Experiment 4: Incident radiation for $L = 4, N = 10$ of full DOM with 594,473 DoFs (left) and sparse DOM with $N_0 = 3$ and 81,268 DoFs (right). The facing half of the cube has been clipped. Observe the different scales of the colorbar.

Figure 8.15: Experiment 4: Convergence in incident radiation when $L$ and $N$ are refined. Reference resolution was $L_{\text{ref}} = 5, N_{\text{ref}} = 14$. Points labeled “Sp.” obtained with $N_0 = \{2, 3, 4, 5, 6\}$, “Sp.2” with $N_0 = \{1, 3, 3, 3, 3\}$.
8 Numerical experiments

Scattering is isotropic. Absorption is set to zero. No source exists inside the domain, radiation enters the domain from the boundaries with at least one vanishing coordinate (faces with $x_1 = 0$, $x_2 = 0$, $x_3 = 0$). On these boundaries, we prescribe the data

$$g(x, s) = \begin{cases} 1 & \text{if } s = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^\top, \\ 0 & \text{else}. \end{cases}$$

The incident radiation computed by the sparse DOM with minimum angular resolution $N_0 = 3$ (Fig. 8.14 right) shows ray effects for the scattered radiation which should be isotropic as in the incident radiation of the full DOM (Fig. 8.14 left). Also, in the shaded region intensities become negative in some parts. A larger value of $N_0$ is therefore required.

Fig. 8.15 illustrates the effect of different values for $N_0$ on the convergence rates. In this example, the $L^2$-error in the incident radiation remains largely unaffected by the value of $N_0$, only for the highest resolution we observe a difference in the error. The efficiency in terms of error per employed degrees of freedom profits from the lower number of degrees of freedom for lower $N_0$. However, for continued convergence in the $H^1$-error, the minimum angular resolution must be increased with the physical resolution as $N_0 \approx L$. The reduction of the convergence rate for the finest resolution level is likely due to a saturation of the error from one of the component contributions since all methods suffer from it likewise.

8.3 Randomized discrete ordinates method

To analyze the randomized discrete ordinates method, we use manufactured exact solutions because the full deterministic discrete ordinates method might not work reliably on solutions of low regularity.

In order to simplify the computation of the right hand side we employ separable functions $u(x, s) = u_D(x)u_S(s)$ as exact solutions. The physical part is taken to be the parabola

$$u_D(x) = \prod_{i=1}^d (-4x_i(x_i - 1))$$

and only the angular part varies between experiments. The physical domain is the unit cube $[0, 1]^3$.

As the number of random ordinates $M_l$ to use for each full subproblem of physical resolution $l$ is given by Eq. (6.13) only up to a constant, we insert $N$ as a multiplicative factor into the formula for $M_l$:

$$M_l = \frac{N}{32} l^{2+2e} 2^{2(l-1)}.$$

The constant $e$ is set to 0.1. According to Thm. 6.4 we then expect the $L^2$-error in the incident radiation to scale with $N^{-1/2}$ if the physical resolution is fixed.
8.3 Randomized discrete ordinates method

Figure 8.16: Experiments 1 and 2: Convergence in incident radiation when \( N \) is refined (left Experiment 1, right Experiment 2). Full method: DOM, with errors defined in (8.1). Sparse method: MLMC, errors defined in (8.2). Exact solution computed with \( L_{\text{ref}} = 6 \).

In the following, we compare the relative error in the incident radiation of the full deterministic DOM as in (8.1) to the error

\[
\| G - \hat{G}_{LM,S} \|_{L^2(S,H^k(D))} / \| G \|_{H^k(D)}, \quad k = 0, 1 \tag{8.2}
\]

of the randomized sparse DOM, of which the \( L^2 \)-variant \((k = 0)\) has been estimated in Thm. 6.4. We approximate (8.2) by

\[
\left( \frac{1}{R} \sum_{i=1}^{R} \| G - \hat{G}_{LM,S}(i) \|_{H^k(D)}^2 \right)^{1/2} / \| G \|_{H^k(D)}
\]

where the dependence on \( i \) of the sparse sampled incident radiation indicates that each term in the sum is computed for a different set of realizations of the random variable \( s \). The number of sample runs is chosen as \( R = 40 \).

**Experiment 1**

We first test the randomized and the deterministic DOM on the solution with angular part

\[ u_S(s) = \Phi_{\text{Henyey},\gamma=0.8}(s,s'), \]

that is, we “abuse” the scattering phase function (3.10) of Henyey-Greenstein scattering with fixed \( s' = (1/\sqrt{3},1/\sqrt{3},1/\sqrt{3})^\top \) to represent a smooth, but highly peaked function in angle. No scattering is applied, absorption is set to \( \kappa = 1 \).

In Fig. 8.16 left, convergence of approximately \( 1/2 \) with respect to the angular degrees of freedom is attained as expected. However, the full deterministic DOM achieves a rate of about 1.5 for the \( L^2 \)-error of the incident radiation, the \( H^1 \)-error is saturated with the physical contribution for small \( N \) already.
8 Numerical experiments

**Experiment 2**

Here we choose a square integrable point singularity on the sphere as the angular part of the exact solution:

\[ u_S(s) = \arccos(s \cdot s')^{-2/3}. \]

The direction of the peak is again \( s' = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^\top \), absorption and scattering also remain unmodified from experiment 1.

Also in this experiment we observe convergence of the MLMC method with rate \( 1/2 \) or nearly \( 1/2 \) in Fig. 8.16 right when \( N \) is increased. However, the \( L^2 \)-error in the full deterministic DOM converges approximately with rate 1 and the level of error is much lower at the same number of degrees of freedom.
9 Conclusion

We conclude this thesis with a summary of its main results and an outline for directions of further research.

9.1 Summary

In the introduction, we motivated the need for efficient discretizations of the radiative transfer equation. Due to its five-dimensional phase space, standard discretizations quickly become prohibitively expensive for increasing resolution.

Direct sparse spherical harmonics method. As a first efficient discretization we derived a direct sparse tensor spherical harmonics method built upon a generic phase space Galerkin finite element framework with least squares stabilization. As bases of the component function spaces, we chose hierarchical piecewise affine nodal functions over the physical domain and spectral functions over the angular domain since these basis systems allow for a decomposition of the function spaces into detail spaces which is necessary for the direct sparse approach.

In Thms. 4.7 and 4.8 we proved that the approximation rate of the sparse tensor product approximation space of nodal functions and spherical harmonics equals the rate of approximation by the full tensor product space, at least up to a factor logarithmic in the number of degrees of freedom of one component domain, i.e. the physical or angular domain, if the solution is sufficiently smooth. At the same time, the total number of degrees of freedom of the sparse tensor approximation space is essentially (up to a logarithmic factor) equal to the number of degrees of freedom of one component domain. This benefit corresponds to expectations from the theory of sparse grids.

To avoid non-tensor-product terms in the variational formulation, we decided to satisfy boundary conditions in a strong sense and constructed conforming full and sparse approximation spaces with additional spectral functions of spherical regions. In contrast to common weak formulations of boundary conditions for spherical harmonics, the mathematical complexity of our construction does not increase with the order $N$ of the spherical harmonics. However, for complex domain geometries and in higher dimensions, the implementation of our construction becomes more involved.

In numerical examples in 2D with sufficiently regular solutions, the approximation rates of the approximation spaces with boundary conditions were found to coincide with our rate estimates for the spaces without boundary conditions. In those examples, the sparse spherical harmonics method delivered a smaller error per degrees of freedom than the standard full method. Especially if solutions are nearly isotropic in angle sparse spherical harmonics should be considered as the method of choice.
Sparse DOM. The second method we presented was a sparse discrete ordinates method based on a direct sparse approach for nested approximation spaces and based on the combination technique for nonnested approximation spaces. In the combination technique approach, solutions to smaller full tensor subproblems of varying resolution are combined, with their resolution levels chosen such that the total number of degrees of freedom essentially reduces to the number of degrees of freedom on one component domain. The full tensor subproblems are formulated as variational problems in the phase space Galerkin framework, in this case with SUPG stabilization and additional terms for the weak satisfaction of the boundary conditions.

We showed by Thms. 5.18 and 5.22 that the error of the sparse DOM to the exact solution decays essentially with the same rate as that of the standard full approximation. This estimate assumed exact scattering. In numerical computations, however, the scattering integral has to be replaced by quadrature over the angular domain. For the composite midpoint rule over a partition of the angular domain we showed in Thm. 5.21 that the additional consistency error decays faster than the approximation error and can therefore be neglected asymptotically. However, since solutions of low angular resolution also enter the combination technique approximation, a certain minimum angular resolution has to be maintained for subproblems in the combination technique.

Numerical experiments in 2D and 3D indicated that in applications in which the discrete ordinates method is plagued by ray effects the minimum angular resolution for subproblems helps to sustain convergence of the combination approximation with discrete ordinates for higher physical resolutions. Similar to the sparse spherical harmonics method, the sparse discrete ordinates method also worked best for solutions which are nearly isotropic in angle or at least sufficiently smooth. In such situations the achieved error per degrees of freedom was on par with or better than in the full discrete ordinates method. As an advantage over the spherical harmonics method, applications with strongly directed radiation could be handled as well by sparse discrete ordinates.

The main advantage of the combination technique approach, however, is that large radiative transfer problems are split up into smaller subproblems with far lower demands on computing time and memory without the need for a new direct sparse radiative transfer solver. At the cost of minor compromises in accuracy solutions to previously prohibitively expensive problems become accessible now.

Multilevel Monte Carlo DOM. As a third method for solutions of low angular regularity we proposed a novel multilevel Monte Carlo method for the radiative transfer equation without scattering. We proved in Thm. 6.4 that the solution mean over the angular domain converges in this method essentially as in a single-level Monte Carlo method, while the multilevel method only requires as many degrees of freedom as a purely physical problem.

In numerical experiments, the deterministic full tensor discrete ordinates method performed at least as efficient in terms of achieved error per employed number of degrees of freedom as the multilevel Monte Carlo method, even for solutions which are barely square integrable in angle for which no convergence proof exists. However, as a sparse method, the Monte Carlo multilevel technique also realizes great savings
in computational resources compared to the full discrete ordinates method and may therefore be of value in certain applications.

**Implementation.** A modular and flexible parallelized C++ implementation of the various variants of the discrete ordinates method analyzed before constitutes a major part of this thesis. The code uses the finite element library deal.II for the solution of full tensor subproblems. Generic programming allows to design interfaces almost independently of dimensions, case distinctions about scattering or parallel execution, while internally efficient code can be generated which is suited to the application at hand. The software permits the solution of radiative transfer problems in 2D and 3D on distributed memory computing systems and could be tested on more than 1000 cores. The enormous memory requirements for the standard full tensor DOM, which we used to compute reference solutions, in 3D problems required a careful selection of preconditioner and solver. The most viable choice was BiCGStab with an incomplete LU-factorization working in-place on the assembled matrix, even though other alternatives might have allowed for better parallel scaling of the code. However, other preconditioners and solvers can always be chosen by the user comfortably via command line parameters.

**9.2 Future work**

Satisfying boundary conditions in a strong sense in the direct sparse spherical harmonics method was accomplished by addition of specialized angular basis functions and further conditioning of the tensorization of bases. This construction could be removed if boundary conditions are satisfied in a weak sense instead. To comply with complexity goals of the sparse methods, the combination technique could also be applied to the spherical harmonics method. First tests into this direction proved promising, and quick progress appears possible thanks to the groundwork in this thesis.

The assumptions of the convergence proof for the combination technique by (Griebel and Harbrecht, 2013b) remain to be verified rigorously. On the physical domain, the extended stability assumptions of the Galerkin projectors could be derived for the periodic case by transfer to the Fourier representation. On the angular domain, the fact that the angular variable $s$ occurs in the directional derivative complicates the analysis.

To alleviate the memory requirements of the full discrete ordinates solver, matrix-free implementations could be explored, as already realized by Kanschat (1996). The advantages of the sparse method of lower resource utilization should still hold: Smaller linear systems require fewer instructions and therefore less time to solution.

For further improvements in efficiency, the methods presented here could be combined with a-posteriori adaptivity, as Widmer et al. (2008) investigated already for a Galerkin wavelet scheme. Adaptivity in the ordinates has been shown to reduce the impact of ray effects (Stone, 2007) so that this direction of research could also help to mitigate that issue of the discrete ordinates method.
## List of symbols

This table lists the most important symbols and abbreviations used throughout this work.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Introduced in</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Radiative transfer operator</td>
<td>Eq. (3.1)</td>
</tr>
<tr>
<td>D</td>
<td>Physical domain</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>d</td>
<td>Dimension of D</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>DOM</td>
<td>Discrete ordinates method</td>
<td>Sec. 1.3</td>
</tr>
<tr>
<td>DoF</td>
<td>Degree of freedom</td>
<td>Sec. 1.3</td>
</tr>
<tr>
<td>d_S</td>
<td>Dimension of S</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>f</td>
<td>Sparsity profile or source function</td>
<td>Eq. (2.19), Eq. (3.7)</td>
</tr>
<tr>
<td>G</td>
<td>Incident radiation</td>
<td>Eq. (1.2)</td>
</tr>
<tr>
<td>g</td>
<td>Boundary condition function</td>
<td>Eq. (1.1b)</td>
</tr>
<tr>
<td>(I_b)</td>
<td>Blackbody intensity</td>
<td>Eq. (1.1a)</td>
</tr>
<tr>
<td>L</td>
<td>Level of (physical) resolution</td>
<td>Sec. 3.4.1</td>
</tr>
<tr>
<td>M</td>
<td>No. of degrees of freedom</td>
<td>Sec. 2.2</td>
</tr>
<tr>
<td>N</td>
<td>Level of angular resolution</td>
<td>Sec. 3.4.1</td>
</tr>
<tr>
<td>n</td>
<td>Outer unit normal vector on boundary</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>N_0</td>
<td>Minimum angular resolution level</td>
<td>Sec. 4.3.4</td>
</tr>
<tr>
<td>P</td>
<td>Projection operator</td>
<td>Sec. 2.3.1</td>
</tr>
<tr>
<td>Q</td>
<td>Scattering operator</td>
<td>Eq. (3.6)</td>
</tr>
<tr>
<td>Q_1</td>
<td>Unity scattering operator</td>
<td>Eq. (3.6)</td>
</tr>
<tr>
<td>q</td>
<td>Heat flux</td>
<td>Eq. (1.3)</td>
</tr>
<tr>
<td>R</td>
<td>Stabilization operator</td>
<td>Eq. (3.18)</td>
</tr>
<tr>
<td>RTE</td>
<td>Radiative transfer equation</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>RTP</td>
<td>Radiative transfer problem</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>S</td>
<td>Angular domain</td>
<td>Eq. (1.1a)</td>
</tr>
<tr>
<td>SUPG</td>
<td>Streamline upwind Petrov Galerkin</td>
<td>Sec. 1.4</td>
</tr>
<tr>
<td>s</td>
<td>Angular variable</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>T</td>
<td>Transport operator</td>
<td>Eq. (3.5)</td>
</tr>
<tr>
<td>u</td>
<td>Radiative intensity or exact solution</td>
<td>Sec. 1.1</td>
</tr>
<tr>
<td>u_{L_1,L_2}</td>
<td>Full tensor approximation</td>
<td>Eq. (2.12)</td>
</tr>
<tr>
<td>(\hat{u}_{L_1,L_2})</td>
<td>Sparse tensor approximation</td>
<td>Eq. (2.20)</td>
</tr>
<tr>
<td>(\tilde{u}_{L_1,L_2})</td>
<td>Combination approximation</td>
<td>Eq. (2.24)</td>
</tr>
<tr>
<td>(\mathcal{V})</td>
<td>Infinite dimensional function space, trial space</td>
<td>Sec. 2.2, Sec. 3.3</td>
</tr>
<tr>
<td>(\mathcal{V}^L)</td>
<td>Finite dimensional function space of resolution L</td>
<td>Sec. 2.2</td>
</tr>
<tr>
<td>W_l</td>
<td>Detail space of resolution l</td>
<td>Eq. (2.14)</td>
</tr>
<tr>
<td>x</td>
<td>Spatial variable</td>
<td>Sec. 1.1</td>
</tr>
</tbody>
</table>
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Equation/Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma_-$</td>
<td>Inflow part of physical boundary</td>
<td>Eq. (3.3)</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Difference projector</td>
<td>Eq. (2.25)</td>
</tr>
<tr>
<td>$\partial\Omega_-$</td>
<td>Inflow boundary of phase space</td>
<td>Eq. (3.2)</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Absorption coefficient</td>
<td>Eq. (1.1a)</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>Scattering integral operator</td>
<td>Eq. (3.6)</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Scattering coefficient</td>
<td>Eq. (1.1a)</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>Scattering kernel</td>
<td>Eq. (1.1a)</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Cartesian product domain or phase space</td>
<td>Eq. (2.10), Sec. 3.1</td>
</tr>
</tbody>
</table>
References


References


References


References


References


References


