Technische Universität Ilmenau
Fakultät für Mathematik und Naturwissenschaften
Arbeitsgruppe Numerische Mathematik und Informationsverarbeitung

Classification of Lattice Group Models, High Order Discretizations of Boltzmann's Collision Operator and Parallelization

Dissertation zur Erlangung des akademischen Grades Dr. rer. nat.

Stefan Brechtken

betreut von
Prof. Dr. Hans Babovsky

Ilmenau, 21.11.2014

urn:nbn:de:gbv:ilm1-2015000223

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich während der Anfertigung dieser Arbeit unterstützt haben. Insbesondere möchte ich Prof. Dr. Hans Babovsky für die Ermöglichung dieser Arbeit sowie viele hilfreiche Diskussionen und Anmerkungen danken.

Weiterhin danke ich allen, die sich in den letzten Jahren an einer wissenschaftlichen Diskussion mit mir beteiligten, sei es auf einer Tagung oder einer entspannenden Tasse Kaffee gewesen. Besonders erwähnen möchte ich dabei meine Freunde und Kollegen Thomas Berger, Leslie Leben, Thomas Schröder und Lars Winterfeld.

Außerdem bedanke ich mich bei der Deutschen Forschungsgemeinschaft für die Ermöglichung dieser Arbeit durch die Finanzierung meiner Stelle.

Schließlich möchte ich meiner Mutter danken, die mich, so lange ich denken kann, bei all meinen Zielsetzungen unterstützt hat.

Abstract

In this thesis we are interested in so called *lattice group models* (LGpMs). This is a class of deterministic schemes which seem to be somehow linked to *discrete velocity models* (DVMs). Unfortunately there exists no convergence proof for the discretization of the collision operator through the LGpM. Additionally it is not clear if the convergence proofs for *discrete velocity models* (DVMs) apply to LGpMs, because there exists no exact classification of the LGpMs within the DVM framework. This work addresses these issues and gives a scheme for constructing discretizations that reach arbitrary high convergence orders (asymptotically) as well as a classification of the LGpMs within the DVM framework. The logically following step is a look at a practical implementation and numerical test of the resulting discretizations in order to numerically verify the theoretic convergence results as well as a closer look at the computational complexity. Finally we investigate the parallelization of a general LGpM solver. Here we pay special attention to the question whether it is possible to obtain a significantly better price to performance ratio when using *graphics processing units* (GPUs).

Contents

1 Introduction

The Boltzmann equation is a basic integro-differential equation in the kinetic gas theory. It mainly describes the statistic distribution of particles in a medium. This equation is primarily used if the mean free path of the particles becomes large compared to the characteristic length of the system under consideration. Or in other words when we look at the dynamic of rarefied gases. One typically uses the simpler equations of continuum mechanics if this condition is not met, for example the Navier-Stokes equations. On the other hand there needs to be a sufficient amount of particles in a given volume in order to do the statistics which give the Boltzmann equation its validity. If this condition is not met one comes into the field of molecular dynamics where one begins to describe every single particle and their interactions. Due to the position of the Boltzmann equation in between the macroscopic (Navier Stokes) and the microscopic (molecular dynamics) regime this equation is called a mesoscopic equation. In the case of a mono atomic gas this partial integro-differential equation is given by

$$(\partial_t+\mathfrak{v}\cdot\nabla_{\mathfrak{r}})f(\mathfrak{r},\mathfrak{v},t)=\alpha\int_{\mathbb{R}^n}\int_{S^{n-1}}\left[f\left(\mathfrak{v}'\right)f\left(\mathfrak{w}'\right)-f\left(\mathfrak{v}\right)f\left(\mathfrak{w}\right)\right]k\left(\mathfrak{v},\mathfrak{w},\eta\right)\mathrm{d}\eta\,\mathrm{d}\mathfrak{w}, n=2,3.$$

The right hand side is generally called Boltzmann collision operator, collision integral, collision operator or simply integral operator and we are mainly interested in the approximation of this operator.

The two main categories for numerical approaches aiming at solving the Boltzmann equation are probabilistic and deterministic schemes. The most successful approach (measured by the number of scientists and engineers using it) seems to be based on probabilistic Monte Carlo techniques for the approximation of the integral operator leading to *direct simulation Monte Carlo methods* (DSMC). We on the other hand are only interested in deterministic schemes. On the deterministic side of things are at least three approaches that can be considered as deterministic approximations of the Boltzmann equation. The first one is based on special collision models such as *Bhatnagar-Gross-Krook* (BGK) together with the discretization of the velocity space, which lead to *lattice Boltzmann methods* (LBM). These ones can hardly be considered as competitors to the other methods, because here it is typically assumed that the liquid is sufficiently dense and in equilibrium so that these discretizations aim at an approximation of the Navier-Stokes equations putting them more in the region of continuum mechanics. Due to the low computational costs and the wide field of applications these approaches have proven their potential in computational physics, for example see [SBH91, PDCD93, TR04, Raa04]. The second one relies on a Fourier transform (also called spectral methods) of the collision integral (or parts of it) in order to approximate the collision operator in a deterministic way with high accuracy and speed, see [BR97,BR00,IR02,FR03,FMP06]. The third one seems to be the most popular deterministic scheme and is called *discrete velocity scheme* (DVM). Schemes that are classified as DVMs do not have a common approach to the discretization problem, but they have the property that the discretization result can be represented as a standard DVM as given in a corresponding survey article [PI88]. And due to a substantial theory developed around DVMs one has a number of tools to verify if a discretization in DVM form possesses specific properties. A non exhaustive list of publications in this field (often coupled with specific discretization approaches and convergence proofs) is [CGL03, BG03, PSB97, FKW06, BV12, RS94, Wag95, Bue96, PS98, PH99, MS00]. There also exist hybrid schemes between the probabilistic and the deterministic schemes where the randomness can be freely chosen [IW93] which also led to developments in stochastic particle schemes [RW98, RSW98, RW07].

The main advantage of probabilistic schemes is the low computational complexity (probably the reason for its domination in the field of rarefied gases), the main disadvantage are solution fluctuations and accurate error estimations which originate in the use of random sequences. The deterministic schemes on the other hand have typically a high computational complexity and mixed results concerning the convergence order of specific discretizations. Until now there seem to be proofs for convergence orders between $\frac{1}{14}$ in [BPS95] (proofs in [PSB97]) and 3 in [FMP06] for approximations of the collision integral. To the best of our knowledge no one tried to construct a class of discretizations of the collision integral that can theoretically reach arbitrary high convergence orders. This is possibly linked to a number of additional properties a discretization must possess in order to reflect the basic properties of the continuous collision integral. The main drawback of the spectral methods are typically problems related to the non conservation of quantities which should be conserved according to the continuous problem. Even the newest publications in this field [FMP06] state that a trade-off between computational complexity and exactness in the conservation laws seems to be non-avoidable, at least when looking at very fast spectral algorithms for the calculation of the collision integral. When looking at spectral algorithms with a moderate computational complexity Bobylev and Rjasanov were able to construct a correction in order to achieve the correct conservation of macroscopic quantities [BR00].

In this work we are interested in so called *lattice group models* (LGpM) which were introduced by Babovsky in [Bab08, Bab09]. This is a class of deterministic schemes which seem to be somehow linked to DVMs. Further development of these LGpMs was done by Babovsky in [Bab11a, Bab11b, Bab12, Bab14]. Unfortunately there exists no convergence proof for the discretization of the collision operator through the LGpM (via kinetic models on integer lattices derived from the automorphism group of the lattice). Additionally it is not clear if the convergence proofs for *discrete velocity models* (DVMs) apply to LGpMs, because there exists no exact classification of the LGpMs within the DVM framework. This work addresses these issues and gives a scheme for constructing discretizations that reach arbitrary high convergence orders (asymptotically). In chapter 2 we aim at giving a short introduction or repetition of the basics around DVMs and LGpMs as well as a classification of the LGpMs within the DVM framework. In chapter 3 we begin with the construction of basic discretizations in two and three dimensions together with convergence proofs and their representation in the DVM and the LGpM framework. The last part of chapter 3 is devoted to a generalization of this discretization approach in order to reach arbitrary high convergence orders (at least asymptotically). The following chapter 4 aims at a practical implementation and numerical test of the resulting discretizations in order to numerically verify the theoretic convergence results. Here we apply and justify some final tweaks of the discretization in order to simplify the implementation and increase the numerical stability of the discretizations. Moreover we take a closer look at the computational complexity and the minimal size of the velocity space in order to be able to apply our discretization schemes. Finally we devote chapter 5 to our approach of a parallelization of a general LGpM solver, general in the sense that simple modifications lead to a solver that can apply any discretization fitting into the scheme of DVMs, as LGpMs generally do - see chapter 2.2. Here we pay special attention to the question whether it is possible to obtain a significantly better price to performance ratio when using *graphics processing units* (GPUs) that justifies the additional time expenditure needed for a GPU implementation.

2 The Lattice Group Model

The first part of this chapter is focused on giving a short repetition of the basic concepts around the Boltzmann equation, DVMs and LGpMs. The second part aims at creating a bridge between the LGpM developed by Babovsky, first published in [Bab08, Bab09], and the general DVM in which the models and discretizations of most scientists in the field of deterministic discretizations fit (for example [PI88, RS94, Bue96, BPS95, PS98, PH99, MS00]). At this point we strongly point at the necessity to suppress the majority of the dependencies in the formulas for the sake of readability. So at every point (especially in section 3) the reader is encouraged to take a break and think about the dependencies or to create a record about the dependency structure.

2.1 Basics

This section restates some well known facts about the Boltzmann equation, DVMs and LGpMs with corresponding references. Whenever the dimension n of the spaces and objects occurs it can be treated as 2 or 3, if not specified otherwise.

2.1.1 The Boltzmann equation

The definitions and results given here are well known and can be found in [CIP94, chapter 2,3]. The Boltzmann equation reads as:

$$(\partial_t + \mathfrak{v} \cdot \nabla_{\mathfrak{x}}) f(\mathfrak{x}, \mathfrak{v}, t) = I[f](\mathfrak{x}, \mathfrak{v}, t), \quad (\mathfrak{x}, \mathfrak{v}, t) \in \mathfrak{D} \subset \mathbb{R}^n_{\mathfrak{x}} \times \mathbb{R}^n_{\mathfrak{v}} \times \mathbb{R}_t \,, \tag{2.1.1}$$

where $f = f(\mathfrak{x}, \mathfrak{v}, t)$ is the distribution function of a gas and depends on the velocity $\mathfrak{v}$, space $\mathfrak{x}$ and time t variables. $I[f]$ is the collision operator and in the case of a rarefied mono-atomic gas it can be written as

$$I[f](\mathfrak{x}, \mathfrak{v}, t) := \alpha \int_{\mathbb{R}^n} \int_{S^{n-1}} \left[f\big(\mathfrak{x}, \mathfrak{v}', t\big) f\big(\mathfrak{x}, \mathfrak{w}', t\big) - f\big(\mathfrak{x}, \mathfrak{v}, t\big) f\big(\mathfrak{x}, \mathfrak{w}, t\big) \right] k\left(\mathfrak{v}, \mathfrak{w}, \eta\right) \, \mathrm{d}\eta \, \mathrm{d}\mathfrak{w} \,.$$

Here $\mathfrak{v}, \mathfrak{w}$ denote the pre-collision, $\mathfrak{v}', \mathfrak{w}'$ the post-collision velocities. The conservation of momentum ($\mathfrak{v}' + \mathfrak{w}' = \mathfrak{v} + \mathfrak{w}$) and energy ($\mathfrak{v}'^2 + \mathfrak{w}'^2 = \mathfrak{v}^2 + \mathfrak{w}^2$) in an elastic two particle collision give the following formulas for $\mathfrak{v}', \mathfrak{w}'$:

$$\mathfrak{v}'(\mathfrak{v}, \mathfrak{w}, \eta) = \mathfrak{v} + \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta, \qquad \mathfrak{w}'(\mathfrak{v}, \mathfrak{w}, \eta) = \mathfrak{w} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta \,.$$

Definition 2.1.1.1 (Collision invariant)
We call a local integrable function $\Phi : \mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}$ *collision invariant* of the operator I iff

$$\forall f \in \mathcal{L}^1(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}) \wedge \Phi \cdot f \text{ is integrable } : \int_{\mathbb{R}^n} \Phi(\mathfrak{v}) I[f](\mathfrak{v}) \, \mathrm{d}\mathfrak{v} = 0 \,.$$

Lemma 2.1.1.2 (Collision invariant)
An integrable function $\Phi \in \mathcal{L}^1_{\text{loc}}$ is a collision invariant, iff

$$\forall \mathfrak{v}, \mathfrak{w} \in \mathbb{R}^n_{\mathfrak{v}} \wedge \forall \eta \in S^{n-1} \; : \; \Phi(\mathfrak{v}) + \Phi(\mathfrak{w}) = \Phi\big(\mathfrak{v}'\big) + \Phi\big(\mathfrak{w}'\big) \; .$$

Remark 2.1.1.3 (Conserved quantities)
If we multiply the Boltzmann equation with a collision invariant and integrate over the velocity space we get

$$\partial_t \int_{\mathbb{R}^n} \Phi(\mathfrak{v}) f(\mathfrak{x}, \mathfrak{v}, t) \, \mathrm{d}\mathfrak{v} + \nabla_{\mathfrak{x}} \int_{\mathbb{R}^n} \mathfrak{v} \Phi(\mathfrak{v}) f(\mathfrak{x}, \mathfrak{v}, t) \, \mathrm{d}\mathfrak{v} = 0 \; .$$

Looking at a solution of the Boltzmann equation that does not depend on the space variable, we get a time independent operator

$$g : \mathcal{L}^1_{\text{loc}}(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}) \times \mathbb{R}^n_{\mathfrak{x}} \times \mathbb{R}^+_t \cup \{0\} \to \mathbb{R}, (\Phi, \mathfrak{x}, t) \mapsto \int_{\mathbb{R}^n} \Phi(\mathfrak{v}) f(\mathfrak{x}, \mathfrak{v}, t) \, \mathrm{d}\mathfrak{v} = \text{const} \; .$$

This operator maps every collision invariant to a corresponding *conserved physical quantity.*

Theorem 2.1.1.4 (Collision invariant)
A function $\Phi \in \mathcal{C}^0(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R})$ is a collision invariant iff it can be written as

$$\Phi(\mathfrak{v}) = a + \langle \mathfrak{b}, \mathfrak{v} \rangle + c \|\mathfrak{v}\|^2 \; , \qquad a, c \in \mathbb{R}, \mathfrak{b} \in \mathbb{R}^n.$$

Remark 2.1.1.5 (Space of collision invariants)
It follows from the last theorem that the functions

$$\Phi_0 : \mathfrak{v} \mapsto 1, \Phi_i : \mathfrak{v} \mapsto v_i, \Phi_{n+1} : \mathfrak{v} \mapsto \frac{1}{2}\|\mathfrak{v}\|^2, \quad \mathfrak{v} \in \mathbb{R}^n_{\mathfrak{v}}, i \in \{1, \ldots, n\} \; ,$$

are the base vectors of the space of all collision invariants. They correspond to the conserved quantities mass, momentum components and kinetic energy.

Now we take a brief look at the equilibrium solutions of the Boltzmann equation and their connection to collision invariants.

Definition 2.1.1.6 (Equilibrium solution, H-operator)

(i) Let f be a function with

$$f \in \mathcal{C} := \{ f | f \in \mathcal{L}^1(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}) \cap C^0(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}) \} \; .$$

Such a function is an *equilibrium solution* iff

$$f > 0 \text{ and } I[f] \equiv 0 \; .$$

(ii) The H-functional is defined as

$$\mathcal{H} : \mathcal{C} \to \mathbb{R}, f \mapsto \int_{\mathbb{R}^n} \ln\big(f(\mathfrak{v})\big) f(\mathfrak{v})\, \mathrm{d}\mathfrak{v}\ .$$

Theorem 2.1.1.7 (Equilibrium solution)
$f : \mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}^+ \cup \{0\}$ is an equilibrium solution iff there exist $a, c \in \mathbb{R}, \mathfrak{b} \in \mathbb{R}^n$ with

$$\forall\, \mathfrak{v} \in \mathbb{R}^n_{\mathfrak{v}} : f(\mathfrak{v}) = \exp(a + \langle \mathfrak{b}, \mathfrak{v} \rangle + c\|\mathfrak{v}\|^2)\ .$$

Remark 2.1.1.8

(i) Looking at 2.1.1.4 we see, that every equilibrium solution can be written as $f(\mathfrak{v}) = \exp(\Phi(\mathfrak{v}))$ with the corresponding collision invariant Φ.

(ii) The above theorem implies that every equilibrium solution can be written as

$$f(\mathfrak{v}) = M[\rho, \overline{\mathfrak{v}}, T](\mathfrak{v}) := \frac{\rho}{(2\pi T)^{\frac{n}{2}}} \exp\left(\frac{(\mathfrak{v} - \overline{\mathfrak{v}})^2}{2T}\right)\ .$$

These functions are called *Maxwell functions* and the values $\rho, \overline{\mathfrak{v}}, T$ can be calculated through

$$\begin{aligned} \rho &= \int_{\mathbb{R}^n} M[\rho, \overline{\mathfrak{v}}, T](\mathfrak{v}) \mathrm{d}^n\mathfrak{v} \\ \rho \cdot \overline{\mathfrak{v}} &= \int_{\mathbb{R}^n} \mathfrak{v} \cdot M[\rho, \overline{\mathfrak{v}}, T](\mathfrak{v}) \mathrm{d}^n\mathfrak{v} \\ \rho \cdot T &= \frac{1}{n} \int_{\mathbb{R}^n} \|\mathfrak{v} - \overline{\mathfrak{v}}\|_2^2 \cdot M[\rho, \overline{\mathfrak{v}}, T](\mathfrak{v}) \mathrm{d}^n\mathfrak{v}\ , \end{aligned}$$

Here ρ can be understood as the (mass) density, $\overline{\mathfrak{v}}$ as the flow velocity, and T as the temperature of the maxwell distribution. The following theorem shows that solutions of the Boltzmann equation can't increase their distance to the equilibrium (Maxwell) solution.

Theorem 2.1.1.9 (H - theorem)
Looking at the space homogeneous case and assuming that the collision kernel satisfies $k > 0$ almost everywhere we get

$$\frac{\partial \mathcal{H}[f]}{\partial t} \leq 0, \qquad \frac{\partial \mathcal{H}[f]}{\partial t} \equiv 0 \Longleftrightarrow f \text{ is a Maxwell function}\ .$$

The last thing we want to state is the calculation of the macroscopic values through the use of the density function f.

Definition 2.1.1.10 (Macroscopic Values)
Let $f \in \mathcal{L}^1(\mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}^+)$ be a nonnegative function. We can calculate the following *macroscopic values*:

- the *mass* ρ, the *momentum* $\mathfrak{m} = (m_i)_{i=1}^n$, the *kinetic energy* E

$$\rho := \int_{\mathbb{R}^n} f(\mathfrak{v})\,\mathrm{d}\mathfrak{v}, \qquad m_i := \int_{\mathbb{R}^n} v_i f(\mathfrak{v})\,\mathrm{d}\mathfrak{v}, \qquad E := \frac{1}{2}\int_{\mathbb{R}^n} \|\mathfrak{v}\|_2^2 f(\mathfrak{v})\,\mathrm{d}\mathfrak{v}\ ,$$

- the *stress tensor* $S = (s_{ij})_{i,j=1}^n$

$$s_{ij} := \int_{R^n} v_i v_j f(\mathfrak{v})\,\mathrm{d}\mathfrak{v} - \frac{1}{\rho} m_i m_j\ ,$$

- the *hydrostatic pressure* p

$$p := \frac{1}{n} \cdot \mathrm{tr}(S) = \frac{1}{n} \cdot \sum_{i=1}^n s_{ii}\ ,$$

- and the *temperature* T

$$T := \frac{p}{\rho}\ .$$

The first three macroscopic values correspond to the basic conserved quantities / collision invariants, compare 2.1.1.3, 2.1.1.5 .

2.1.2 Discrete Velocity Models

Discrete velocity models typically aim at a classical deterministic discretization of the collision operator through the discretization of the velocity space by using transformations and Newton-Cotes formulas or similar integral approximations. So in any case a discretization of the velocity space is needed, in the following thesis we will only look at a uniform discretization of the form

$$\overline{\mathfrak{V}} := \{\mathfrak{v} \in \mathbb{R}^n_{\mathfrak{v}} | v_j \in \Delta v \cdot \mathbb{Z}, j = 1, ..., n, \}, \quad \Delta v \in \mathbb{R}^+\ .$$

To avoid the problems that are associated with an infinite velocity space (and not topic of this work), we will restrict ourselves to the finite subset

$$\mathfrak{V} := \overline{\mathfrak{V}} \cap B_r(\mathbf{0}), \quad r \in \mathbb{R}^+\ .$$

We also need the corresponding index set $M_{\mathfrak{V}}$ to address single velocities. Now almost all discrete velocity discretizations of the Boltzmann equation 2.1.1 can be rewritten in the form of general DVMs (cp. [PI88]).

$$(\partial_t + \mathfrak{v}_i \cdot \nabla_{\mathfrak{x}}) f_i(\mathfrak{x},t) = \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(f_k(\mathfrak{x},t) f_l(\mathfrak{x},t) - f_i(\mathfrak{x},t) f_j(\mathfrak{x},t) \big) \tag{2.1.2}$$
$$=: J_i[f](\mathfrak{x},t), \qquad i \in \{1, \dots, |\mathfrak{V}|\} \,.$$

The standard way in referring to a specific class of DVMs is to give the dimensionality n and the number of velocities m in the form $n\mathrm{D}m$. To name the minimal requirements a DVM should fulfill we need some more definitions, in fact the discrete versions of 2.1.1.1, 2.1.1.6, 2.1.1.9, 2.1.1.10.

Definition 2.1.2.1 (Collision invariant, equilibrium, macroscopic values)

(i) We call a function $\Phi : \mathfrak{V} \mapsto \mathbb{R}$ a collision invariant iff $\sum\limits_{\mathfrak{v} \in \mathfrak{V}} \Phi(\mathfrak{v}) J[f](\mathfrak{v}) = 0$. We denote the space of collision invariants with $\mathbb{I}$.

(ii) An equilibrium solution can be characterized through $\mathbf{J}[f] = \begin{pmatrix} J_1[f] \\ \vdots \\ J_{|\mathfrak{V}|}[f] \end{pmatrix} = \mathbf{0}$

(iii) The macroscopic values can be obtained through

- *mass* ρ, *momentum* $\mathfrak{m} = (m_i)_{i=1}^2$, *kinetic energy* E

$$\rho := \sum_{\mathfrak{v} \in \mathfrak{V}} f(\mathfrak{v}), \qquad m_i := \sum_{\mathfrak{v} \in \mathfrak{V}} v_i f(\mathfrak{v}), \qquad E := \frac{1}{2} \sum_{\mathfrak{v} \in \mathfrak{V}} \|\mathfrak{v}\|_2^2 f(\mathfrak{v}),$$

- *stress tensor* $S = (s_{ij})_{i,j=1}^2$, with

$$s_{ij} := \sum_{\mathfrak{v} \in \mathfrak{V}} v_i v_j f(\mathfrak{v}) - \frac{1}{\rho} m_i m_j \ ,$$

- *hydrostatic pressure*

$$p := \frac{1}{2} \cdot \mathrm{tr}(S) = \frac{1}{2} \cdot \sum_{i=1}^{2} s_{ii} \ ,$$

- *temperature*

$$T := \frac{p}{\rho} \ .$$

Definition 2.1.2.2 (H - operator)
The discrete H - functional is defined as

$$H : \{f : \mathfrak{V} \to \mathbb{R}^+_{\{0\}}\} \to \mathbb{R}, f \mapsto \sum_{\mathfrak{v} \in \mathfrak{V}} f(\mathfrak{v}) \ln(f(\mathfrak{v})) \,.$$

Remark 2.1.2.3 (H - theorem)
Let $f : \mathfrak{V} \to \mathbb{R}^+_{\{0\}}$, then the discrete H - theorem is

$$\frac{\partial H[f]}{\partial t} \leq 0\,,$$

and

$$\frac{\partial H[f]}{\partial t} \equiv 0 \Longleftrightarrow f \text{ is an equilibrium solution.}$$

Remark 2.1.2.4 (Minimal requirements for DVMs)
It is well known that DVMs should obey some minimal requirements before they can be considered for serious applications. The most common requirements are :

(i) Properties of the Operator A :

a) All allowed particle interactions obey the momentum and energy conservation: $\mathfrak{v}_i + \mathfrak{v}_j \neq \mathfrak{v}_k + \mathfrak{v}_l \vee \mathfrak{v}_i^2 + \mathfrak{v}_j^2 \neq \mathfrak{v}_k^2 + \mathfrak{v}_l^2 \Longrightarrow A_{i,j}^{k,l} = 0$. And $A_{i,j}^{k,l} \geq 0$.

b) $A_{i,j}^{k,l} = A_{j,i}^{l,k} = A_{k,l}^{i,j}$, from these symmetries follows the alternative characterization of collisional invariants:

$$\begin{aligned} &\Phi \text{ is collisional invariant} \\ \Longleftrightarrow &\forall (i,j,k,l) : A_{i,j}^{k,l} \neq 0 \Rightarrow \Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) = \Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l)\,. \end{aligned}$$

From this follows that the space of collisional invariants $\mathbb{I}$ contains at least the discrete versions of $\Phi_0, \ldots, \Phi_{n+1}$ from 2.1.1.5, and finally we obtain the alternative characterization of the equilibrium solutions:

$$\begin{aligned} &\varphi \text{ is equilibrium solution} \\ \Longleftrightarrow &\forall (i,j,k,l) : A_{i,j}^{k,l} \neq 0 \Rightarrow \varphi(\mathfrak{v}_i)\varphi(\mathfrak{v}_j) = \varphi(\mathfrak{v}_k)\varphi(\mathfrak{v}_l) \\ \Longleftrightarrow &\exists \Phi \in \mathbb{I} : \Phi = \ln(\varphi)\,, \end{aligned}$$

as well as the discrete H - theorem.

(ii) $\Phi_0, \ldots, \Phi_{n+1}$ (as in 2.1.1.5) should be a basis of the space of collisional invariants, leading to equilibrium solutions of Maxwell type.

Proof of i) b):
In the literature one can often find the additional symmetries $A_{i,j}^{k,l} = A_{j,i}^{k,l} = A_{i,j}^{l,k}$. These are not necessary to obtain the result i) b), we recall the corresponding proofs to reassure the reader of this fact. We start with the alternative characterization of collisional invariants:

$$\sum_{i \in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_i) J_i[f] = \sum_{i \in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_i) \cdot \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} (f_k f_l - f_i f_j) \right)$$

$$
\begin{aligned}
&= \sum_{i,j,k,l\in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_i) A_{i,j}^{k,l}(f_k f_l - f_i f_j) = \sum_{j,i,l,k\in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_j) \overbrace{A_{i,j}^{k,l}}^{=A_{j,i}^{l,k}} (f_l f_k - f_j f_i) \\
&= \sum_{k,l,i,j\in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_k) \overbrace{A_{i,j}^{k,l}}^{=A_{k,l}^{i,j}} (f_i f_j - f_k f_l) = \sum_{l,k,j,i\in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_l) \overbrace{A_{i,j}^{k,l}}^{=A_{l,k}^{j,i}} (f_j f_i - f_l f_k)\,.
\end{aligned}
$$

Summation over the last 4 equivalent sums gives

$$
4 \sum_{i\in M_{\mathfrak{V}}} \Phi(\mathfrak{v}_i) J_i[f] = \sum_{i,j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) - \Phi(\mathfrak{v}_k) - \Phi(\mathfrak{v}_l)\big)(f_k f_l - f_i f_j)\,. \tag{2.1.3}
$$

Now we assume that Φ is a collisional invariant and

$$
\exists\,(i,j,k,l) \in \left\{(i,j,k,l) | A_{i,j}^{k,l} \neq 0\right\} : \Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) \neq \Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l)\,.
$$

Let (i,j,k,l) be the multi index with $\Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) \neq \Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l)$ and $r \in \mathbb{R}$. Then by using

$$
f_n := \begin{cases} 1, & \text{if } n = i \vee n = j \\ \varepsilon \in \mathbb{R}^+, & \text{otherwise} \end{cases}
$$

we obtain

$$
\begin{aligned}
\sum_{a,b,c,d\in M_{\mathfrak{V}}} & A_{a,b}^{c,d} \big(\Phi(\mathfrak{v}_a) + \Phi(\mathfrak{v}_b) - \Phi(\mathfrak{v}_c) - \Phi(\mathfrak{v}_d)\big)(f_c f_d - f_a f_b) \\
&= 2\big(\Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) - \Phi(\mathfrak{v}_k) - \Phi(\mathfrak{v}_l)\big)(-f_i f_j) \\
&+ 2\big(\Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l) - \Phi(\mathfrak{v}_i) - \Phi(\mathfrak{v}_j)\big)(f_i f_j) + \varepsilon \cdot r \\
&= 4\big(\Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l) - \Phi(\mathfrak{v}_i) - \Phi(\mathfrak{v}_j)\big) + \varepsilon \cdot r \\
&\neq 0, \text{ if } \varepsilon \text{ sufficiently small,}
\end{aligned}
$$

which is a contradiction to the assumption that Φ is a collisional invariant. The minimal set of collisional invariants the DVM possesses can now be calculated by:

$$
\begin{aligned}
\Phi_0(\mathfrak{v}_i) + \Phi_0(\mathfrak{v}_j) &\overset{2.1.1.5}{=} 1 + 1 = \Phi_0(\mathfrak{v}_k) + \Phi_0(\mathfrak{v}_l) \\
\Phi_a(\mathfrak{v}_i) + \Phi_a(\mathfrak{v}_j) &\overset{2.1.1.5}{=} \mathfrak{v}_{i,a} + \mathfrak{v}_{j,a} \overset{(i)a)}{=} \mathfrak{v}_{k,a} + \mathfrak{v}_{l,a} = \Phi_a(\mathfrak{v}_k) + \Phi_a(\mathfrak{v}_l), \quad a = 1,\dots,n \\
\Phi_{n+1}(\mathfrak{v}_i) + \Phi_{n+1}(\mathfrak{v}_j) &\overset{2.1.1.5}{=} \frac{1}{2}\|\mathfrak{v}_i\|^2 + \frac{1}{2}\|\mathfrak{v}_j\|^2 \overset{(i)a)}{=} \frac{1}{2}\|\mathfrak{v}_l\|^2 + \frac{1}{2}\|\mathfrak{v}_k\|^2 \\
&= \quad \Phi_{n+1}(\mathfrak{v}_k) + \Phi_{n+1}(\mathfrak{v}_l)\,.
\end{aligned}
$$

We need some preliminary considerations to prove the alternative characterization of the equilibrium solutions and the correspondence of collisional invariants:

$$\sum_{\mathfrak{v}\in\mathfrak{V}} \ln\big(f(\mathfrak{v})\big) J[f](\mathfrak{v}) \overset{(2.1.3)}{=} \frac{1}{4} \sum_{i,j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l} \left[\ln(f_i) + \ln(f_j) - \ln(f_k) - \ln(f_l)\right] (f_k f_l - f_i f_j) \tag{2.1.4}$$

$$= \frac{1}{4} \sum_{i,j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l} \overbrace{\ln\left(\frac{f_i f_j}{f_k f_l}\right)\left(1 - \frac{f_i f_j}{f_k f_l}\right)}^{\ln(\lambda)(1-\lambda)} f_k f_l, \text{ and we know} \tag{2.1.5}$$

$$\forall \lambda \in \mathbb{R}^+ \setminus \{1\} : \ln(\lambda)(1-\lambda) < 0 \ \wedge \ \ln(\lambda)(1-\lambda) = 0 \Longleftrightarrow \lambda = 1\,. \tag{2.1.6}$$

At this point we need to explain why $\lambda > 0$. This comes from the positivity of f. The positivity can be obtained by looking at the space homogeneous initial value problem

$$\partial_t f_i(t) = J_i[f](t), \quad f_i(0) = f_0(\mathfrak{v}_i) > 0\,, \tag{2.1.7}$$

and the helper function $g_i : t \mapsto f_i(t)e^{\rho t}, \quad \rho := c \sum_{j,k,l\in M_{\mathfrak{V}}} f_j$:

$$\partial_t g_i(t) = \partial_t f_i e^{\rho t} + f_i \rho e^{\rho t} = \sum_{j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l} (f_k f_l - f_i f_j) e^{\rho t} + c f_i \sum_{j,k,l\in M_{\mathfrak{V}}} f_j e^{\rho t}$$
$$= e^{\rho t} \left(\sum_{j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l} f_k f_l + \sum_{j,k,l\in M_{\mathfrak{V}}} \left(c - A_{i,j}^{k,l}\right) f_i f_j \right)\,.$$

By knowing (i)a) $A_{i,j}^{k,l} \geq 0$, $f(0) > 0$ and by choosing c sufficiently large ($c \geq \max_{i,j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l}$) we obtain the positivity of the derivative of g. This leads to the positivity of f. Now we can go back to topic: the equilibrium solutions. Using their definition we get:

$$J[f] = 0 \Longrightarrow \ln(f)J[f] = 0 \Longrightarrow \sum_{i\in M_{\mathfrak{V}}} \ln(f_i) J_i[f] = 0$$
$$\Longleftrightarrow \left(\forall (i,j,k,l) : A_{i,j}^{k,l} \neq 0 \Rightarrow f_i f_j = f_k f_l\right), \text{ by } (2.1.4) - (2.1.6)$$
$$\Longleftrightarrow \left(\forall (i,j,k,l) : A_{i,j}^{k,l} \neq 0 \Rightarrow \ln(f_i) + \ln(f_j) = \ln(f_k) + \ln(f_l)\right)\,.$$

If we follow the argumentation in the opposite direction we obtain $\ln(f)J[f] = 0 \Longleftarrow \sum_{i\in M_{\mathfrak{V}}} \ln(f_i)J_i[f] = 0$ by (2.1.4) - (2.1.6) and the positivity of f, A. Assuming $f \not\equiv 1$ we also get $J[f] = 0 \Longleftarrow \ln(f)J[f] = 0$, where the case $f \equiv 1$ is already covered by the knowledge that $J_i[f] = \sum_{j,k,l\in M_{\mathfrak{V}}} A_{i,j}^{k,l}(1-1) = 0$. The last thing remaining is the proof of the H-theorem:

$$\partial_t H[f] = \sum_{\mathfrak{v}\in\mathfrak{V}} \dot f(\mathfrak{v}) \ln(f(\mathfrak{v})) + f(\mathfrak{v}) \frac{1}{f(\mathfrak{v})} \dot f(\mathfrak{v})$$

$$
\begin{aligned}
&= \sum_{\mathfrak{v}\in\mathfrak{V}} \dot{f}(\mathfrak{v})\big(\ln(f(\mathfrak{v}))+1\big) \overset{(2.1.7)}{=} \sum_{\mathfrak{v}\in\mathfrak{V}} J[f](\mathfrak{v})(\ln(f(\mathfrak{v}))+1) \\
&= \sum_{\mathfrak{v}\in\mathfrak{V}} J[f](\mathfrak{v})(\ln(f(\mathfrak{v}))+\Phi_0(\mathfrak{v})) = \sum_{\mathfrak{v}\in\mathfrak{V}} J[f](\mathfrak{v})\ln(f(\mathfrak{v})) \\
&= \sum_{i\in M_{\mathfrak{V}}} \ln(f_i) \sum_{j,k,l\in M_{\mathfrak{V}}} A^{k,l}_{i,j}(f_kf_l - f_if_j) = \sum_{i,j,k,l\in M_{\mathfrak{V}}} A^{k,l}_{i,j}\ln(f_i)(f_kf_l - f_if_j) \\
&= \frac{1}{2}\begin{bmatrix} \sum\limits_{i,j,k,l\in M_{\mathfrak{V}}} \ln(f_i)A^{k,l}_{i,j}(f_kf_l - f_if_j) \\ + \sum\limits_{j,i,l,k\in M_{\mathfrak{V}}} \ln(f_j)A^{l,k}_{j,i}(f_lf_k - f_jf_i) \end{bmatrix} = \frac{1}{2}\sum_{i,j,k,l\in M_{\mathfrak{V}}} \ln(f_if_j)A^{kl}_{ij}(f_kf_l - f_if_j) \\
&= \frac{1}{4}\begin{bmatrix} \sum\limits_{i,j,k,l\in M_{\mathfrak{V}}} \ln(f_if_j)A^{k,l}_{i,j}(f_kf_l - f_if_j) \\ - \sum\limits_{k,l,i,j\in M_{\mathfrak{V}}} \ln(f_kf_l)A^{i,j}_{k,l}(f_kf_l - f_if_j) \end{bmatrix} \\
&= \frac{1}{4}\sum_{i,j,k,l\in M_{\mathfrak{V}}} (\ln(f_if_j) - \ln(f_kf_l))A^{kl}_{ij}(f_kf_l - f_if_j) \leq 0, \text{ because} \\
&\forall x,y\in\mathbb{R}^+ : (\ln(x)-\ln(y))(y-x) \leq 0 \\
&\wedge\, (\ln(x)-\ln(y))(y-x) = 0 \Longleftrightarrow x = y \wedge A^{k,l}_{i,j} \geq 0\,.
\end{aligned}
$$

The second part of the H-theorem can be obtained by

$$
\begin{aligned}
\partial_t H[f] &= \sum_{\mathfrak{v}\in\mathfrak{V}} \ln(f(\mathfrak{v}))J[f](\mathfrak{v}), (\text{ see above}) \\
&\overset{(2.1.3)}{=} \frac{1}{4}\sum_{i,j,k,l\in M_{\mathfrak{V}}} (\ln(f_i)+\ln(f_j)-\ln(f_k)-\ln(f_l))A^{kl}_{ij}(f_kf_l - f_jf_i) = 0 \\
&\Longleftrightarrow \ln(f_i)+\ln(f_j)-\ln(f_k)-\ln(f_l) = 0 \\
&\overset{(\mathrm{i})b}{\Longleftrightarrow} \ln(f) \text{ is collisional invariant} \\
&\overset{(\mathrm{i})b}{\Longleftrightarrow} f \text{ is equilibrium solution.}
\end{aligned}
$$

□

Remark 2.1.2.5

Many authors also consider the additional symmetries $A^{k,l}_{i,j} = A^{k,l}_{j,i} = A^{l,k}_{i,j}$. These symmetries seem to be a consequence of the interpretation of the $A^{\bullet,\bullet}_{\bullet,\bullet}$ as transition probabilities from pre- to post collisional velocity pairs. But it seems that these symmetries have no impact on any discretization property considered throughout this work. So we neglect these symmetries when it comes to the derivation of different discretizations and their transformation into DVMs, but we give remarks whether these symmetries exist or can be artificially created.

Lemma 2.1.2.6 (Collision spheres)
The Operator A of a DVM satisfying the minimal requirements possesses the properties

$$A_{i,j}^{k,l} \neq 0 \Longrightarrow \{\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l\} \in S_{ij}^{\mathfrak{V}} := \left\{ \mathfrak{v} \in \mathfrak{V} \,\middle|\, \left\| \mathfrak{v} - \frac{\mathfrak{v}_i + \mathfrak{v}_j}{2} \right\| = \left\| \frac{\mathfrak{v}_i - \mathfrak{v}_j}{2} \right\| \right\} ,$$
$$A_{i,j}^{k,l} \neq 0 \Longrightarrow \mathfrak{v}_l = \mathfrak{v}_i + \mathfrak{v}_j - \mathfrak{v}_k \Longrightarrow A_{i,j}^{k,l} = A_{i,j}^{k,l(i,j,k)} =: A_{i,j}^{k} .$$

Proof: By using 2.1.2.4, (i)a a commonly known proof is

$$\begin{aligned} A_{i,j}^{k,l} \neq 0 &\Longrightarrow \mathfrak{v}_i + \mathfrak{v}_j = \mathfrak{v}_k + \mathfrak{v}_l \wedge \mathfrak{v}_i^2 + \mathfrak{v}_j^2 = \mathfrak{v}_k^2 + \mathfrak{v}_l^2 \\ &\Longrightarrow (\mathfrak{v}_i + \mathfrak{v}_j)^2 = (\mathfrak{v}_k + \mathfrak{v}_l)^2 \wedge \mathfrak{v}_i^2 + \mathfrak{v}_j^2 = \mathfrak{v}_k^2 + \mathfrak{v}_l^2 \\ &\Longrightarrow \langle \mathfrak{v}_i, \mathfrak{v}_j \rangle = \langle \mathfrak{v}_k, \mathfrak{v}_l \rangle \\ &\Longrightarrow (\mathfrak{v}_i - \mathfrak{v}_j)^2 = (\mathfrak{v}_k - \mathfrak{v}_l)^2 \\ &\Longrightarrow \|\mathfrak{v}_i - \mathfrak{v}_j\| = \|\mathfrak{v}_k - \mathfrak{v}_l\| . \end{aligned}$$

This together with $\frac{\mathfrak{v}_i+\mathfrak{v}_j}{2} = \frac{\mathfrak{v}_k+\mathfrak{v}_l}{2}$ gives the result. The second claim follows directly from $\mathfrak{v}_i + \mathfrak{v}_j = \mathfrak{v}_k + \mathfrak{v}_l$. □

The next theorem restates the common ways of checking 2.1.2.4 (ii) .

Theorem 2.1.2.7 (Calculating artificial collision invariants)
Let $\mathfrak{V}$ be a uniform (equidistant) velocity space and M be the set of all collision pairs used by the DVM. For classical DVMs this means

$$M := \left\{ (i,j,k,l) \in M_{\mathfrak{V}}^4 \,\middle|\, A_{i,j}^{k,l} \neq 0 \right\} .$$

Assuming that the DVM satisfies the minimal requirements 2.1.2.4 (i) the space of collisional invariants $\mathbb{I}$ is a equal to

$$\ker \begin{pmatrix} (e_{k_1} + e_{l_1} - e_{i_1} - e_{j_1})^T \\ \vdots \\ (e_{k_{|M|}} + e_{l_{|M|}} - e_{i_{|M|}} - e_{j_{|M|}})^T \end{pmatrix} = \ker \left(D_\varphi^{-1} \nabla_\varphi J[\varphi] D_\varphi \right) ,$$

where $(i_\bullet, j_\bullet, k_\bullet, l_\bullet) \in M$, $D_\varphi := \operatorname{diag}(\varphi(\mathfrak{v}_1), \ldots, \varphi(\mathfrak{v}_{|\mathfrak{V}|}))$ and φ represents an arbitrary equilibrium solution.

Proof:
We divide the proof into two sections, that correspond to the two kernels.

(i) Due to the compliance with the minimal requirements 2.1.2.4 (i) we obtain the alternative characterization of the collision invariants

$$\Phi \in \mathbb{I} \Longleftrightarrow \forall (i,j,k,l) \in M : \Phi(\mathfrak{v}_i) + \Phi(\mathfrak{v}_j) = \Phi(\mathfrak{v}_k) + \Phi(\mathfrak{v}_l) .$$

From this follows (by using the shortcut $\Phi_i := \Phi(\mathfrak{v}_i)$) that the space of collision invariants is equal to the solution space of the system of equations

$$\begin{aligned}\Phi_{k_1} + \Phi_{l_1} - \Phi_{i_1} - \Phi_{j_1} &= 0\\ \Phi_{k_2} + \Phi_{l_2} - \Phi_{i_2} - \Phi_{j_2} &= 0\\ \vdots \qquad &= \vdots\\ \Phi_{k_{|M|}} + \Phi_{l_{|M|}} - \Phi_{i_{|M|}} - \Phi_{j_{|M|}} &= 0\,,\end{aligned}$$

which can be rewritten by using the corresponding matrix

$$\overbrace{\begin{pmatrix} (e_{k_1} + e_{l_1} - e_{i_1} - e_{j_1})^T \\ \vdots \\ (e_{k_{|M|}} + e_{l_{|M|}} - e_{i_{|M|}} - e_{j_{|M|}})^T \end{pmatrix}}^{\hat{M}:=} \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_{|M|} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}\,.$$

The solution of this is (by definition) the kernel of the matrix. Due to this we get $\mathbb{I} = \ker(\hat{M})$.

(ii) Starting at

$$J[\varphi] = \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\varphi_k \varphi_l - \varphi_i \varphi_j\big) \right)_{i=1}^{|\mathfrak{V}|}\,,$$

we can simply derive

$$\begin{aligned}\nabla_\varphi J[\varphi] &= \left[\frac{\partial}{\partial \varphi_m} \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\varphi_k \varphi_l - \varphi_i \varphi_j\big) \right)_{i=1}^{|\mathfrak{V}|} \right]_{m=1}^{|\mathfrak{V}|}\\ &= \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\varphi_k e_l + \varphi_l e_k + \varphi_i e_j + \varphi_j e_i\big)^T \right)_{i=1}^{|\mathfrak{V}|}\,.\end{aligned}$$

Now we choose an arbitrary $f \in ker(\nabla_\varphi J[\varphi])$. This f can always be represented as $f = (\varphi, \tilde{f})$, by using the element-wise multiplication $(\bullet, \bullet)$. Now we get

$$\begin{aligned}\nabla_\varphi J[\varphi] f &= \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\varphi_k f_l + \varphi_l f_k + \varphi_i f_j + \varphi_j f_i\big)^T \right)_{i=1}^{|\mathfrak{V}|}\\ &= \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(\varphi_k \tilde{f}_l \varphi_l + \varphi_l \tilde{f}_k \varphi_k + \varphi_i \tilde{f}_j \varphi_j + \varphi_j \tilde{f}_i \varphi_i\big)^T \right)_{i=1}^{|\mathfrak{V}|}\end{aligned}$$

$$= \left(\varphi_i \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \varphi_j \big(\tilde{f}_l + \tilde{f}_k + \tilde{f}_j + \tilde{f}_i \big)^T \right)_{i=1}^{|\mathfrak{V}|} \text{, by using 2.1.2.4(i)}b$$

$$\overset{!}{=} 0 \Longleftrightarrow \forall (i,j,k,l) \in M_{\mathfrak{V}}^4 : A_{i,j}^{k,l} \neq 0 \Rightarrow \tilde{f}_i + \tilde{f}_j = \tilde{f}_k + \tilde{f}_l \Longleftrightarrow \tilde{f} \in \mathbb{I}$$

We get the last conclusion by the fact that the above calculation holds true for linear independent φ. At this point we have proved

$$\ker \left(\nabla_\varphi J[\varphi] \right) = \operatorname{span}((\varphi, \tilde{\mathbf{f}}_1), \ldots, (\varphi, \tilde{\mathbf{f}}_m)) \,,$$

where $\tilde{\mathbf{f}}_i$ are the basis vectors of $\mathbb{I}$. Due to

$$\nabla_\varphi J[\varphi] f = \nabla_\varphi J[\varphi](\varphi, \tilde{f}) = \nabla_\varphi J[\varphi] D_\varphi \tilde{f}$$

it becomes clear that

$$\mathbb{I} = \ker(\nabla_\varphi J[\varphi] D_\varphi) = \ker(D_\varphi^{-1} \nabla_\varphi J[\varphi] D_\varphi) \,. \qquad \square$$

Due to the fact that the number of collision pairs $|M|$ typically grows at least quadratically (in two and three dimensions) with the number of velocities $|\mathfrak{V}|$ it is desirable to get less (computational) complex tools to rule artificial collision invariants out.

Corollary 2.1.2.8 (Ruling artificial collision invariants out)
Let $\mathfrak{V}$ be a uniform (equidistant) velocity space and M be the set of all collision pairs used by the DVM. Let the set $\tilde{M} \subset M$ contain exactly all collision pairs representing squares with a diagonal of $2\Delta v$ or $\sqrt{2}\Delta v$ and let the DVM satisfy the minimal requirements 2.1.2.4 (i).

(i) The space of collisional invariants $\mathbb{I}$ is a linear subspace of

$$\ker \begin{pmatrix} (e_{k_1} + e_{l_1} - e_{i_1} - e_{j_1})^T \\ \vdots \\ (e_{k_{|\tilde{M}|}} + e_{l_{|\tilde{M}|}} - e_{i_{|\tilde{M}|}} - e_{j_{|\tilde{M}|}})^T \end{pmatrix} = \ker \left(D_\varphi^{-1} \nabla_\varphi \tilde{J}[\varphi] D_\varphi \right) ,$$

where $(i_\bullet, j_\bullet, k_\bullet, l_\bullet) \in \tilde{M}$, $D_\varphi := \operatorname{diag}(\varphi(\mathfrak{v}_1), \ldots, \varphi(\mathfrak{v}_{|\mathfrak{V}|}))$, φ represents an arbitrary equilibrium solution and

$$\tilde{J}[\varphi] := \left(\sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \begin{Bmatrix} \big(\varphi_k \varphi_l - \varphi_i \varphi_j\big) & \text{, if } (i,j,k,l) \in \tilde{M} \\ 0 & \text{, else} \end{Bmatrix} \right)_{i=1}^{|\mathfrak{V}|}$$

is the collision operator induced by $\tilde{M}$. If the dimension of these kernels is equal to $n+2$ the model possesses no artificial collision invariants.

(ii) Let $\mathfrak{V}$ contain at least the origin and all neighboring points (9 in 2D and 27 in 3D). Assuming $A \subset \Delta v\mathbb{Z}^n, \mathfrak{a} \in \Delta v\mathbb{Z}^n$ and using the definitions

$$\begin{aligned}
N_B(A) &:= \{\mathfrak{a} \in \Delta v\mathbb{Z}^n \,|\, \mathfrak{a} \notin A \wedge \exists \mathfrak{b} \in A : \|\mathfrak{a} - \mathfrak{b}\| \in \Delta v B\} \,, \\
N_{i,j}(A, \mathfrak{a}) &:= A \cap N_{\{1,\sqrt{2}\}}(\{\mathfrak{a}\}) \cap \{\mathfrak{b} \in \Delta v\mathbb{Z}^n | \mathfrak{b} = \mathfrak{a} + \lambda_1 e_i + \lambda_2 e_j; \lambda_1, \lambda_2 \in \Delta v\mathbb{Z}\} \,, \\
\hat{N}(A) &:= \left\{ \mathfrak{a} \in \Delta v\mathbb{Z}^n \,\middle|\, \begin{array}{l} \exists \mathfrak{b} \in N_{1,2}(A, \mathfrak{a}) : |N_{1,2}(A, \mathfrak{a}) \cap N_{\{1\}}(\{\mathfrak{b}\})| = 2 \quad \vee \\ \exists \mathfrak{b} \in N_{1,3}(A, \mathfrak{a}) : |N_{1,3}(A, \mathfrak{a}) \cap N_{\{1\}}(\{\mathfrak{b}\})| = 2 \quad \vee \\ \exists \mathfrak{b} \in N_{2,3}(A, \mathfrak{a}) : |N_{2,3}(A, \mathfrak{a}) \cap N_{\{1\}}(\{\mathfrak{b}\})| = 2 \end{array} \right\} \,, \\
\mathfrak{o} &:= (0, \ldots, 0)^T \,,
\end{aligned}$$

the following algorithm can be used to falsify the existence of artificial collision invariants:

```
S1 $\hat{\mathfrak{V}} := \{\mathfrak{v} \in \Delta v\mathbb{Z}^n | \mathfrak{v} \in N_{\{1,\sqrt{2}\}}(\mathfrak{o}) \cup \mathfrak{o}\}$
S2 $A := \hat{N}(\hat{\mathfrak{V}}) \cap \mathfrak{V}$
S3 • if $A = \emptyset$
       ⇒ if $\mathfrak{V} = \hat{\mathfrak{V}}$
          output: no artificial collision invariants [END]
       ⇒ else
          output: artificial collision invariants possible [END]
   • else
     $\hat{\mathfrak{V}} := \hat{\mathfrak{V}} \cup A$, goto S2
```

This algorithm holds true in two and three dimensions, but in two dimensions we can simply disregard $N_{1,3}, N_{2,3}$.

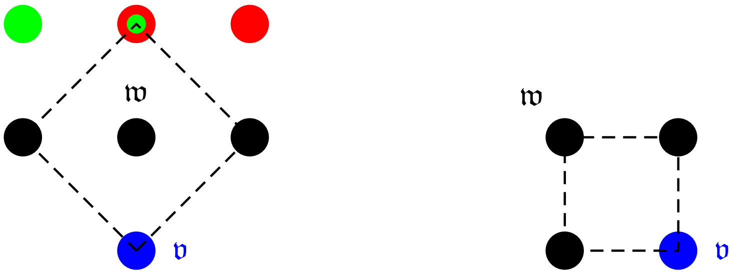
Proof:

(i) The proof of the first part is equal to the proof of the last theorem adding that the introduction of additional collision pairs can only shrink the space of collision invariants and that $\mathbb{I}$ must contain at least the linear independent $\Phi_0, \ldots, \Phi_{n+1}$ as described in 2.1.2.4 (i)b.

(ii) We start with the finite velocity space $\hat{\mathfrak{V}}$ that contains only zero and the neighboring points (in total 9 points in two dimensions and 27 in three dimensions) and we assume that the set of collision pairs $\tilde{M}$ contains all squares (in the velocity space) with diameter $\sqrt{2}\Delta v$ and $2\Delta v$. This leads to 5 resp. 45 collision pairs resulting in $\hat{M}_{\text{2D9}} \in \{0, 1, -1\}^{5\times 9}, \hat{M}_{\text{3D27}} \in \{0, 1, -1\}^{45\times 27}$. Here $\hat{M}$ is the representing matrix of the collision pair equations ($f_k + f_l = f_i + f_j$). Calculating the rank of these matrices gives $\text{rank}(\hat{M}_{\text{2D9}}) = 5$, $\text{rank}(\hat{M}_{\text{3D27}}) = 22$ resulting in a kernel dimension of 4 resp. 5. Due to the size of these matrices an Octave (MATLAB) function to calculate these matrices and the dimension of the kernels can be found in appendix A.1.1. This can be seen as the begin of our mathematical induction. Now we conduct the inductive step by introducing a new point $\hat{\mathfrak{v}}$ to the velocity space set $\mathfrak{V}$ that has the property of possessing three direct neighbors in $\hat{\mathfrak{V}}$ that are neighbors to each other and are lying in the same axis parallel plane:

$$\begin{pmatrix} \exists \mathfrak{w} \in N_{1,2}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) : |N_{1,2}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) \cap N_{\{1\}}(\{\mathfrak{w}\})| = 2 & \vee \\ \exists \mathfrak{w} \in N_{1,3}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) : |N_{1,3}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) \cap N_{\{1\}}(\{\mathfrak{w}\})| = 2 & \vee \\ \exists \mathfrak{w} \in N_{2,3}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) : |N_{2,3}(\hat{\mathfrak{V}}, \hat{\mathfrak{v}}) \cap N_{\{1\}}(\{\mathfrak{w}\})| = 2 & \end{pmatrix} .$$

Due to the position in axis parallel planes we can reduce our consideration to the two dimensional case and because of the symmetries we have only too look at exactly two different cases.



(a) *Case i)* (b) *Case ii)*

Figure 2.1: *Possible cases of grid expansion*

i) Figure 2.1a illustrates the first case where all 3 neighbors of $\mathfrak{v}$ that lie within $\hat{\mathfrak{V}}$ are on a straight line. Every of these three points must have another three neighbors that are neighbors to each other, because this holds true for the begin of the induction and for every point that was introduced since then. From this comes the fact that either the red or the green dots are the neighbors of $\mathfrak{w}$ in $\hat{\mathfrak{V}}$ resulting in the existence of the dashed collision pair ruling out the introduction of a new collision invariant (because $\Phi(\mathfrak{v})$ is determined by the three points in $\hat{\mathfrak{V}}$ belonging to the collision pair).

ii) Figure 2.1b illustrates the second case where $\mathfrak{v}$ is "surrounded" by three points from $\hat{\mathfrak{V}}$ in such a way that we can directly see the existence of a collision pair ruling out the introduction of a new collision invariant. □

Definition 2.1.2.9 (Normal grid)
Let $\mathfrak{V}$ be a uniform (equidistant) velocity space and M the set of all collision pairs representing squares with a diagonal of $2\Delta v$ or $\sqrt{2}\Delta v$. We call such a grid in n dimensions normal iff

$$\ker \begin{pmatrix} (e_{k_1} + e_{l_1} - e_{i_1} - e_{j_1})^T \\ \vdots \\ (e_{k_{|M|}} + e_{l_{|M|}} - e_{i_{|M|}} - e_{j_{|M|}})^T \end{pmatrix} = n + 2, \qquad (i_\bullet, j_\bullet, k_\bullet, l_\bullet) \in M\,.$$

Remark 2.1.2.10 (Computational complexity and conclusions)

(i) It is interesting to notice that 2.1.2.8 (i) leads to the calculation of the kernel of a matrix that can not grow larger than $\{-1,0,1\}^{8|\mathfrak{V}|\times|\mathfrak{V}|}$ where only 4 elements per row are non zero. Using Givens rotations to determine the kernel the computational complexity becomes $\mathcal{O}(|\mathfrak{V}|)$.

(ii) An interesting implication from algorithm 2.1.2.8 (ii) is that every velocity grid $\mathfrak{V}$ that fulfills the requirements of 2.1.2.8 and has the property that every point of the grid can be reached from the origin through a "three point wide way" is normal. $(\mathfrak{v}_0,\dots,\mathfrak{v}_m)$ is such a way iff

$$\forall i \in \{0,\dots,m-1\} : \begin{cases} \left|N_{1,2}(\mathfrak{V},\mathfrak{v}_{i+1}) \cap N_{\{1\}}(\{\mathfrak{v}_i\})\right| = 2 & \vee \\ \left|N_{1,3}(\mathfrak{V},\mathfrak{v}_{i+1}) \cap N_{\{1\}}(\{\mathfrak{v}_i\})\right| = 2 & \vee \\ \left|N_{2,3}(\mathfrak{V},\mathfrak{v}_{i+1}) \cap N_{\{1\}}(\{\mathfrak{v}_i\})\right| = 2 & \end{cases} .$$

(iii) From the above follows that every velocity grid that is used throughout this work (only uniform, equidistant discretizations of origin centered balls and cubes) are normal resp. artificial collision invariant free as long as our numerical schemes include all collision pairs representing squares with a diameter of $2\Delta v$ or $\sqrt{2}\Delta v$.

2.1.3 Lattice Group Models

The LGpM framework was established by Babovsky in [Bab09]. Throughout this work we will restrict this (very general) approach to binary collisions and the definitions that we give in this section.

Definition 2.1.3.1 (Orthonormal group, relation)
The orthonormal group G (a finite subset of the orthogonal group $O(n)$) of $\mathfrak{V}$ is the group which is generated by all reflections and rotations around zero that map the grid $\mathfrak{V}$ onto itself. The automorphism groups of the most common integer lattices may be found in [CSB87]. In this work we denote the automorphism group of the lattice under consideration by simply saying "automorphism group". In two dimensions and for Cartesian grids this group contains the identity, negative identity, 90° rotation as well as the reflection around the x-axis and around the identity. A corresponding relation is given by

$$\mathfrak{v} \sim_{\mathfrak{a}} \mathfrak{w} \iff \exists \varphi \in G : \mathfrak{v} - \mathfrak{a} = \varphi(\mathfrak{w} - \mathfrak{a}) .$$

And let H be the subgroup of G that contains only $id, -id$.

Remark 2.1.3.2 (Right coset class)
Using the relation

$$\varphi, \varphi' \in G, \quad \varphi \sim_H \varphi' \iff \varphi\varphi'^{-1} \in H ,$$

the group G decomposes in the right coset classes $G\diagup\sim_H$ and the order of this is

$$|G\diagup\sim_H| = \mathrm{idx}_G H = \frac{|G|}{|H|} = \frac{|G|}{2} \, .$$

Definition 2.1.3.3 (Some necessary mappings)
These definitions were originally introduced in [Bab09, section 2.1, 3.2].

(i) Let $\hat{\alpha} : G\diagup\sim_H \times G\diagup\sim_H \rightarrow \mathbb{R}^+$ be some coefficients satisfying

a) group invariance: There is a mapping $\tilde{\alpha} : G\diagup\sim_H \rightarrow \mathbb{R}^+$ such that

$$\hat{\alpha}_{[\varphi],[\varphi']} = \tilde{\alpha}_{[\varphi\varphi'^{-1}]}$$

b) micro-reversibility: For all $\varphi \in G$

$$\tilde{\alpha}_{[\varphi]} = \tilde{\alpha}_{[\varphi^{-1}]} \, .$$

(ii) Let $\mathfrak{c} \in \mathfrak{C}, \mathfrak{v} \in \mathfrak{V} \subset \mathfrak{C}, \psi_{\mathfrak{c},\mathfrak{v}} : G \rightarrow \mathfrak{V}$ be the mapping that is defined by $\varphi \mapsto \mathfrak{c} + \varphi(\mathfrak{v} - \mathfrak{c})$. This mapping can be used to determine the points that are used to approximate the spherical integral in $J[f](\mathfrak{v})$ above a sphere with the center in $\mathfrak{c}$. The inverse of this mapping gives all operators that map from a point $\mathfrak{v}$ on a sphere with center in $\mathfrak{c}$ to another given point on the sphere, $\psi^{-1}(\mathfrak{w}) = \{\varphi \in G | \mathfrak{w} = \mathfrak{c} + \varphi(\mathfrak{v} - \mathfrak{c})\}$.

(iii) Let $\overline{\alpha} : \mathfrak{V} \times \mathfrak{V} \rightarrow \mathbb{R}^+$ be a mapping defined by

$$\overline{\alpha}_{\mathfrak{w},\mathfrak{v}} := \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v})} \sum_{\varphi' \in \psi^{-1}(\mathfrak{w})} \hat{\alpha}_{[\varphi'],[\varphi]}$$

(iv) Let $\gamma : \mathfrak{V} \times \mathbb{R}^+ \rightarrow \mathbb{R}_0^+$ be some nonnegative coefficients and let

$$\alpha_{\mathfrak{c},\mathfrak{v}}^{\varphi} := \gamma(\mathfrak{c}, |\mathfrak{v} - \mathfrak{c}|) \cdot \overline{\alpha}_{\mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}),\mathfrak{v}}$$

be an abbreviation to handle these constants in a more convenient way.

Definition 2.1.3.4 (Lattice group model, LGpM)
By introducing a set of potential center points $\mathfrak{C}$, we get a LGpM above $\mathfrak{C}, \mathfrak{V}$ as

$$J[f](\mathfrak{v}) = \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G\diagup\sim_H} \alpha_{\mathfrak{c},\mathfrak{v}}^{\varphi} \left(\prod_{\varphi'\in[\varphi]} f\left(\mathfrak{c} + \varphi'(\mathfrak{v} - \mathfrak{c})\right) - \prod_{\varphi''\in H} f\left(\mathfrak{c} + \varphi''(\mathfrak{v} - \mathfrak{c})\right) \right) \, .$$

In the following work we will call this LGpM the “original” LGpM.

Remark 2.1.3.5
The above definitions summarize the result and the ingredients of the LGpM, due to the aim to develop some theory on top of this we simply use it as a definition of the model - concealing the basics of this concept. These basics can be found in [Bab09], and the above definition can be obtained by using [Bab09, section 3.2, formula 3.20] and recursively replacing the objects in this formula with the precedent definitions.
A simple question that instantly arises is: "are the points $\mathfrak{c} + \varphi(\mathfrak{v} - \mathfrak{c})$ points of the used grid $\mathfrak{V}$?" To answer this question we have to specify $\mathfrak{V}$,$\mathfrak{C}$ and we need to take a closer look at the associated discrete spheres.

Lemma 2.1.3.6 (Sphere decomposition)
Let $\mathfrak{v}, \mathfrak{w} \in \overline{\mathfrak{V}}$, $\hat{\mathfrak{V}} := \overline{\mathfrak{V}} + \frac{\Delta v}{2}\mathbf{1}$, $\mathbf{1} := (1, \dots, 1)^T$, $\mathfrak{a} := \frac{\mathfrak{v}+\mathfrak{w}}{2}$ and $r := \left\| \frac{\mathfrak{v}-\mathfrak{w}}{2} \right\|$ and let us call a point $\mathfrak{b}$

- "on the grid" iff $\mathfrak{b} \in \mathfrak{V}$
- "partially off grid" iff $\exists b_i \in \Delta v \mathbb{Z}^n \wedge \exists b_j \notin \Delta v \mathbb{Z}^n$, $i, j \in \{1, \dots, n\}$ and
- "completely off grid" iff $\nexists b_i \in \Delta v \mathbb{Z}^n$.

With these ingredients we get

(i) a sphere with the endpoints of a diagonal on the grid ($\mathfrak{v}, \mathfrak{w}, \in \overline{\mathfrak{V}}$) and the center $\mathfrak{a}$ on the grid or completely off grid $\left(\mathfrak{a} \in \overline{\mathfrak{V}} \cup \hat{\mathfrak{V}}\right)$ decomposes in equivalence classes over $\overline{\mathfrak{V}}$:
$$S_r^{\overline{\mathfrak{V}}}(\mathfrak{a}) = \bigcup_{[\mathfrak{v}] \in S_r^{\overline{\mathfrak{V}}}(\mathfrak{a}) / \sim_{\mathfrak{a}}} [\mathfrak{v}] \, .$$

(ii) a sphere with a diagonal on the grid ($\mathfrak{v}, \mathfrak{w} \in \overline{\mathfrak{V}}$) and a center partially off the grid ($\mathfrak{a} \in \frac{1}{2}\overline{\mathfrak{V}} \setminus (\overline{\mathfrak{V}} \cup \hat{\mathfrak{V}})$) generally does not decompose in equivalence classes above $\overline{\mathfrak{V}}$.

Proof:

At first we simplify the problem by transforming the prerequisites into a simpler setting. Without loss of generality we look at the integer lattice $2\mathbb{Z}^n$ and the finer grid $\mathbb{Z}^n$ instead of $\overline{\mathfrak{V}}$ and $\frac{1}{2}\overline{\mathfrak{V}}$. We assume $\mathfrak{v}, \mathfrak{w} \in 2\mathbb{Z}^n$.

(i) We split this part of the proof in two additional parts. In the first one the center of the circle is on the grid and in the second one the center is completely off grid.

  a) The operators in G are isometric mappings, and they map the grid onto itself. So it is clear, that
$$\forall \mathfrak{b} \in 2\mathbb{Z}^n : [\mathfrak{b}] \subset S_{\|\mathfrak{b}\|}^{2\mathbb{Z}^n}(\mathbf{0}) \, , \text{ this together with } \mathfrak{a}, \mathfrak{a} - \mathfrak{v} \in 2\mathbb{Z}^n$$

gives the desired result:

$$\forall \mathfrak{a}, \mathfrak{v} \in 2\mathbb{Z}^n : \mathfrak{a} + [\mathfrak{a} - \mathfrak{v}] \subset S^{2\mathbb{Z}^n}_{\|\mathfrak{a}-\mathfrak{v}\|}(\mathfrak{a}) \ .$$

Now it is clear that the discrete spheres $S^{\mathfrak{V}}_r(\mathfrak{a})$ can be decomposed in (disjoint) sets corresponding to equivalence classes of the above relation.

b) Now we consider completely off lattice center points ($\mathfrak{a} \in 2\mathbb{Z}^n + \mathfrak{1}$). From this follows, that $\mathfrak{a} - \mathfrak{v} \in 2\mathbb{Z}^n + \mathfrak{1}$. All operators $\varphi \in G$ can be interpreted as permutations with sign changes. This implies $\forall \varphi \in G : \varphi(\mathfrak{a} - \mathfrak{v}) \in 2\mathbb{Z}^n + \mathfrak{1}$. These considerations give the result

$$\begin{aligned} &\forall \mathfrak{a} \in 2\mathbb{Z}^n + \mathfrak{1} \forall \mathfrak{v} \in 2\mathbb{Z}^n \forall \varphi \in G : \mathfrak{a} + \varphi(\mathfrak{a} - \mathfrak{v}) \in 2\mathbb{Z}^n + \mathfrak{2} = 2\mathbb{Z}^n \\ \Longrightarrow &\forall \mathfrak{a} \in 2\mathbb{Z}^n + \mathfrak{1} \forall \mathfrak{v} \in 2\mathbb{Z}^n : \mathfrak{a} + [\mathfrak{a} - \mathfrak{v}] \subset S^{2\mathbb{Z}^n}_{\|\mathfrak{a}-\mathfrak{v}\|}(\mathfrak{a}) \end{aligned}$$

(ii) We only look at the three dimensional case, because it includes the two dimensional one. The center of the considered sphere $\mathfrak{a}$ lies on $\mathbb{Z}^3 \setminus 2\mathbb{Z}^3 \cup 2\mathbb{Z}^3 + \mathfrak{1}$ and the ends of the diagonal $(\mathfrak{v}, \mathfrak{w})$ lie on $2\mathbb{Z}^3$. We know from (i) that the corresponding sphere on $\mathbb{Z}^3$ decomposes into disjoint equivalence classes. Now we prove a little bit more than the necessary counter example. We do this to understand the underlying problem better. We prove, that in all except one case off grid equivalence classes possess at least one representative on the grid:

$$[\mathfrak{b}] \in \left(S^{\mathbb{Z}^3}_r(\mathfrak{a}) / \sim_{\mathfrak{a}} \right) \exists \mathfrak{c} \in [\mathfrak{b}] : \mathfrak{c} \in S^{2\mathbb{Z}^3}_r(\mathfrak{a}) \ .$$

From this fact follows instantly (ii).
Let $\mathfrak{b}$ be an arbitrary representative of $[\mathfrak{b}]$ that lies on $\mathbb{Z}^3 \setminus 2\mathbb{Z}^3$, so at least one component of $\mathfrak{b}$ must be odd. In this proof we denote odd scalars with a $\sim$ and even without a $\sim$ above.

a) $\mathfrak{c} = \mathfrak{b} - \mathfrak{a}$ contains exactly one odd element
Wlog we assume $\mathfrak{c} = (c_1, \tilde{c}_2, c_3)^T$.

i. $\mathfrak{a}$ contains exactly one odd element
Wlog $\mathfrak{a} = (\tilde{a}_1, a_2, a_3)^T$, because we can generate every permutation of the elements in $\mathfrak{c}$ through operators in G, it is obvious that : $\exists \varphi \in G$: $\varphi(\mathfrak{c}) = (\tilde{d_1}, d_2, d_3)^T$ and therefore $\mathfrak{a} + \varphi(\mathfrak{c}) \in 2\mathbb{Z}^3 \wedge \mathfrak{a} + \varphi(\mathfrak{c}) \in [\mathfrak{b}]$

ii. $\mathfrak{a}$ contains exactly two odd elements
Wlog we assume $\mathfrak{a} = (\tilde{a}_1, \tilde{a}_2, a_3)^T, \mathfrak{c} = (c_1, \tilde{c}_2, c_3)^T$. There exists no such sphere on $2\mathbb{Z}^3$ in other words: $\left| S^{2\mathbb{Z}^3}_r(\mathfrak{a}) \right| = \emptyset$, because the equation

$$\overbrace{\overbrace{(x - \tilde{a}_1)^2}^{\in 2\mathbb{Z}+1} + \overbrace{(y - \tilde{a}_2)^2}^{\in 2\mathbb{Z}+1} + \overbrace{(z - a_3)^2}^{\in 2\mathbb{Z}}}^{\in 2\mathbb{Z}} = \overbrace{\overbrace{c_1^2 + \tilde{c}_2^2 + c_3^2}^{\in 2\mathbb{Z}+1}}^{\notin 2\mathbb{Z}}$$

contains no solution $(x, y, z)^T \in 2\mathbb{Z}^2$.

b) $\mathfrak{c} = \mathfrak{b} - \mathfrak{a}$ contains exactly two odd elements

  i. $\mathfrak{a}$ contains exactly one odd element
     analogical to (ii)(a)ii - there exists no such sphere on $2\mathbb{Z}^3$.

  ii. $\mathfrak{a}$ contains exactly two odd elements
     analogical to (ii)(a)i - permutations.

c) $\mathfrak{c} = \mathfrak{b} - \mathfrak{a}$ contains exactly three odd elements

  i. $\mathfrak{a}$ contains exactly one odd element
     Such spheres exist and they possess completely off lattice equivalence classes.

  ii. $\mathfrak{a}$ contains exactly two odd elements
     analogical to (ii)(a)ii - there exists no such sphere on $2\mathbb{Z}^3$.

□

Remark 2.1.3.7
In two dimensions spheres with partially off grid centers decompose into equivalence classes giving the sphere on the finer grid:

$$S_r^{\frac{1}{2}\overline{\mathfrak{V}}}(\mathfrak{a}) = \bigcup_{[\mathfrak{v}] \in S_r^{\overline{\mathfrak{V}}}(\mathfrak{a}) / \sim_{\mathfrak{a}}} [\mathfrak{v}] \; .$$

The proof can be taken from the last proof (ii)a,(ii)b. But that does not hold true in three dimensions (due to the existence of completely off lattice equivalence classes, see (ii)(c)i).

2.2 Classification

In this section we try to give an idea about the relations between DVMs and LGpMs. The aim is to classify the LGpM in the context of the well established DVM framework. As far as the author knows this is the first attempt to create a rigorous bridge between these approaches.

Theorem 2.2.1 (LGpM as a DVM)
The standard LGpM 2.1.3.4 can be directly transformed into a DVM. With the definitions from 2.1.3.3 and the additional definitions

$$M_i := \left\{ (i,j,k,l) \;\middle|\; \begin{array}{l} (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_k) \in \mathfrak{V}^4 \\ \mathfrak{v}_j = \mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c}), \\ \mathfrak{v}_k = \mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c}), \\ \mathfrak{v}_l = \mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c}), \\ \mathfrak{c} \in \mathfrak{C}, \varphi \in G \end{array} \right\}, \; \alpha^k_{i,j} := \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left|\mathfrak{v}_i - \frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}\right|\right) \cdot \overline{\alpha}_{\mathfrak{v}_k, \mathfrak{v}_i},$$

$$A : M^4_{\mathfrak{V}} \to \mathbb{R}, \qquad A^{k,l}_{i,j} = \frac{1}{2}\mathbb{1}_{M_i}(i,j,k,l)\alpha^k_{i,j}\ ,$$

we get

$$\begin{aligned} J[f](\mathfrak{v}_i) &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \alpha^{\varphi}_{\mathfrak{c},\mathfrak{v}_i} \left(\prod_{\varphi'\in[\varphi]} f\left(\mathfrak{c} + \varphi'(\mathfrak{v}_i - \mathfrak{c})\right) - \prod_{\varphi''\in H} f\left(\mathfrak{c} + \varphi''(\mathfrak{v}_i - \mathfrak{c})\right) \right) \\ &= \sum_{j,k,l\in M_{\mathfrak{V}}} A^{k,l}_{i,j}(f(\mathfrak{v}_k)f(\mathfrak{v}_l) - f(\mathfrak{v}_i)f(\mathfrak{v}_j))\ . \end{aligned}$$

Proof:

$$\begin{aligned} J[f](\mathfrak{v}_i) &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \alpha^{\varphi}_{\mathfrak{c},\mathfrak{v}_i} \left(\prod_{\varphi'\in[\varphi]} f\left(\mathfrak{c} + \varphi'(\mathfrak{v}_i - \mathfrak{c})\right) - \prod_{\varphi''\in H} f\left(\mathfrak{c} + \varphi''(\mathfrak{v}_i - \mathfrak{c})\right) \right) \\ &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \alpha^{\varphi}_{\mathfrak{c},\mathfrak{v}_i} \left[\begin{array}{l} f\left(\mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c})\right) f\left(\mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c})\right) - \\ f\left(\mathfrak{c} + \mathfrak{v}_i - \mathfrak{c}\right) f\left(\mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c})\right) \end{array} \right] \text{ by } \begin{array}{c} 2.1.3.1 \\ 2.1.3.2 \end{array} \\ &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \alpha^{\varphi}_{\mathfrak{c},\mathfrak{v}_i} (f(\overbrace{\mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c})}^{\mathfrak{v}_k:=}) f(\overbrace{\mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c})}^{v_l:=}) - f(\mathfrak{v}_i) f(\overbrace{2\mathfrak{c} - \mathfrak{v}_i}^{\mathfrak{v}_j:=})) \\ &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \gamma(\mathfrak{c}, |\mathfrak{v}_i - \mathfrak{c}|) \cdot \overline{\alpha}_{\mathfrak{c}+\varphi(\mathfrak{v}_i - \mathfrak{c}),\mathfrak{v}_i} (f(\mathfrak{v}_k) f(\mathfrak{v}_l) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)) \text{ by } 2.1.3.3 \\ &= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\varphi]\in G/\sim_H} \overbrace{\gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left|\mathfrak{v}_i - \frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}\right|\right) \cdot \overline{\alpha}_{\mathfrak{v}_k,\mathfrak{v}_i}}^{\alpha^k_{i,j}:=} (f(\mathfrak{v}_k) f(\mathfrak{v}_l) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)) \\ &= \sum_{(i,j,k,l)\in M_i} A^{k,l}_{i,j} (f(\mathfrak{v}_k) f(\mathfrak{v}_l) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)) \\ &= \sum_{j,k,l\in M_{\mathfrak{V}}} A^{k,l}_{i,j} (f(\mathfrak{v}_k) f(\mathfrak{v}_l) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)) \end{aligned}$$

□

Theorem 2.2.2 (DVM as a LGpM)
The standard DVM with the minimal requirements 2.1.2.4 (i) and

$$\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2} \notin \mathfrak{V} \cup \mathfrak{V}_{\frac{1}{2}} \Longrightarrow A_{i,j}^{k,l} = 0 \,, \tag{2.2.1}$$

can be transformed into the LGpM framework. With the definitions

$$\begin{aligned}
&\mathfrak{C} := \mathfrak{V} \cup \mathfrak{V}_{\frac{1}{2}},\ \mathfrak{V}_{\frac{1}{2}} := \left\{ \mathfrak{a} \,\middle|\, \exists \mathfrak{v}, \mathfrak{w} \in \mathfrak{V} : \|\overrightarrow{\mathfrak{v}\mathfrak{w}}\| = \sqrt{n}\Delta v \wedge \mathfrak{a} = \frac{\mathfrak{v} + \mathfrak{w}}{2} \right\}, \\
&\tilde{G} := G \diagup \sim_H,\ \alpha_{\mathfrak{c},\mathfrak{v}_i}^{\varphi,\mathfrak{v}} := \frac{2A_{i,j}^{k}}{|\{\varphi' \in G | \mathfrak{c} + \varphi(\mathfrak{v} - \mathfrak{c}) = c + \varphi'(\mathfrak{v} - \mathfrak{c})\}|}\,, \text{ with} \\
&k = \tilde{k} \Longleftrightarrow \exists \tilde{k} \in M_{\mathfrak{V}} : \mathfrak{v}_{\tilde{k}} = \mathfrak{c} + \varphi(\mathfrak{v} - \mathfrak{c})\,, \\
&j = \tilde{j} \Longleftrightarrow \exists \tilde{j} \in M_{\mathfrak{V}} : \mathfrak{v}_{\tilde{j}} = 2\mathfrak{c} - \mathfrak{v}_i, \qquad\qquad \tilde{S}_{i,\mathfrak{c}}^{\mathfrak{V}} := S_{ij}^{\mathfrak{V}} \diagup \sim_{\mathfrak{c}}\,,
\end{aligned}$$

and the knowledge that the existence of $\tilde{k}, \tilde{j}$ is guaranteed by (2.2.1) and 2.1.3.6, we get

$$\begin{aligned}
J[f](\mathfrak{v}_i) &= \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l}(f_k f_l - f_i f_j) \\
&= \sum_{\mathfrak{c} \in \mathfrak{C}} \sum_{[\mathfrak{v}] \in \tilde{S}_{i,\mathfrak{c}}^{\mathfrak{V}}} \sum_{[\varphi] \in \tilde{G}} \alpha_{\mathfrak{c},\mathfrak{v}_i}^{\varphi,\mathfrak{v}} \left(\prod_{\varphi' \in [\varphi]} f(\mathfrak{c} + \varphi'(\mathfrak{v} - \mathfrak{c})) - \prod_{\varphi' \in H} f(\mathfrak{c} + \varphi'(\mathfrak{v}_i - \mathfrak{c})) \right) .
\end{aligned}$$

Proof:
Using the abbreviation $\mathfrak{c}_{ij} := \frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}$ and the identities

$$\begin{aligned}
A_{i,j}^{k} &= A_{\mathfrak{v}_i,\mathfrak{v}_j}^{\mathfrak{v}_k} = A_{\mathfrak{v}_i,\mathfrak{c}_{ij}-(\mathfrak{v}_i-\mathfrak{c}_{ij})}^{\mathfrak{v}_k} =: A_{i,\mathfrak{c}_{ij}}^{k} \\
S_{ij}^{\mathfrak{V}} &= S_{\mathfrak{v}_i,\mathfrak{v}_j}^{\mathfrak{V}} = S_{\mathfrak{v}_i,\mathfrak{c}_{ij}-(\mathfrak{v}_i-\mathfrak{c}_{ij})}^{\mathfrak{V}} =: S_{i\mathfrak{c}_{ij}}^{\mathfrak{V}}, \quad \tilde{S}_{i\mathfrak{c}}^{\mathfrak{V}} := S_{i\mathfrak{c}}^{\mathfrak{V}} \diagup \sim_{\mathfrak{c}}
\end{aligned}$$

we can calculate:

$$\begin{aligned}
J[f](\mathfrak{v}_i) &= \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l}(f_k f_l - f_i f_j) = \sum_{j \in M_{\mathfrak{V}}} \sum_{k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l}(f_k f_l - f_i f_j) \\
&= \sum_{j \in M_{\mathfrak{V}}} \sum_{\mathfrak{v}_k,\mathfrak{v}_l \in S_{ij}^{\mathfrak{V}}} A_{i,j}^{k,l}(f_k f_l - f_i f_j) \quad \text{by 2.1.2.6} \\
&= \sum_{\mathfrak{v}_j \in \mathfrak{V}} \sum_{\mathfrak{v}_k \in S_{ij}^{\mathfrak{V}}} A_{i,j}^{k}(f(\mathfrak{v}_k) f(\mathfrak{v}_i + \mathfrak{v}_j - \mathfrak{v}_k) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)) \quad \text{by 2.1.2.6} \\
&= \sum_{\mathfrak{v}_j \in \mathfrak{V}} \sum_{\mathfrak{v}_k \in S_{ij}^{\mathfrak{V}}} A_{i,j}^{k} \left(f(\mathfrak{v}_k) f(\mathfrak{c}_{ij} - (\mathfrak{v}_k - \mathfrak{c}_{ij})) - f(\mathfrak{v}_i) f(\mathfrak{c}_{ij} - (\mathfrak{v}_i - \mathfrak{c}_{ij})) \right) \\
&= \sum_{\mathfrak{c} \in \mathfrak{C}} \sum_{\mathfrak{v}_k \in S_{i\mathfrak{c}}^{\mathfrak{V}}} A_{i,\mathfrak{c}}^{k} \left(f(\mathfrak{v}_k) f(\mathfrak{c} - (\mathfrak{v}_k - \mathfrak{c})) - f(\mathfrak{v}_i) f(\mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c})) \right) \text{ by (2.2.1)}
\end{aligned}$$

$$= \sum_{\mathfrak{c}\in\mathfrak{C}} \left(\sum_{[\mathfrak{v}]\in S^{\mathfrak{V}}_{i\mathfrak{c}}/\sim_c} \left(\sum_{\mathfrak{v}_k\in[\mathfrak{v}]} A^k_{i,\mathfrak{c}} \begin{bmatrix} f(\mathfrak{v}_k)f(\mathfrak{c}-(\mathfrak{v}_k-\mathfrak{c}))- \\ f(\mathfrak{v}_i)f(\mathfrak{c}-(\mathfrak{v}_i-\mathfrak{c})) \end{bmatrix} \right)\right) \qquad \text{by 2.1.3.6}$$

$$= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\mathfrak{v}]\in \tilde{S}^{\mathfrak{V}}_{i\mathfrak{c}}} \sum_{\varphi\in G} \frac{1}{2}\alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} \begin{bmatrix} f(\mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}))f(\mathfrak{c}-\varphi(\mathfrak{v}-\mathfrak{c}))- \\ f(\mathfrak{c}+(\mathfrak{v}_i-\mathfrak{c}))f(\mathfrak{c}-(\mathfrak{v}_i-\mathfrak{c})) \end{bmatrix} \quad \text{by 2.1.3.1}$$

$$= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\mathfrak{v}]\in \tilde{S}^{\mathfrak{V}}_{i\mathfrak{c}}} \sum_{[\varphi]\in G/\sim_H} \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} \begin{bmatrix} \prod_{\varphi'\in[\varphi]} f(\mathfrak{c}+\varphi'(\mathfrak{v}-\mathfrak{c}))- \\ \prod_{\varphi'\in H} f(\mathfrak{c}+\varphi'(\mathfrak{v}_i-\mathfrak{c})) \end{bmatrix} \qquad \text{by 2.1.3.2}$$

$$= \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\mathfrak{v}]\in \tilde{S}^{\mathfrak{V}}_{i\mathfrak{c}}} \sum_{[\varphi]\in \tilde{G}} \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} \left(\prod_{\varphi'\in[\varphi]} f(\mathfrak{c}+\varphi'(\mathfrak{v}-\mathfrak{c})) - \prod_{\varphi'\in H} f(\mathfrak{c}+\varphi'(\mathfrak{v}_i-\mathfrak{c})) \right) \qquad \square$$

Corollary 2.2.3 (Relation between DVMs and LGpMs)

(i) From the above theorems comes the fact, that the LGpMs are a subclass of the DVMs (LGpM$\subsetneq$ DVM).

(ii) The LGpMs transformed into a DVM satisfy the minimal requirements 2.1.2.4 (i).

(iii) For LGpMs the common ways of checking property 2.1.2.4 (ii) (artificial collision invariants) can be used. That means that 2.1.2.7 and 2.1.2.8 also hold true for LGpMs, where the set of collision pairs is equal to

$$M = \bigcup_{i\in M_{\mathfrak{V}}} M_i\,.$$

Proof: The first claim follows directly from the two preceding theorems. For the second claim we have to check the three statements from 2.1.2.4 (i).

a) We want to show $\mathfrak{v}_i+\mathfrak{v}_j \neq \mathfrak{v}_k+\mathfrak{v}_l \vee \mathfrak{v}_i^2+\mathfrak{v}_j^2 \neq \mathfrak{v}_k^2+\mathfrak{v}_l^2 \Longrightarrow A^{kl}_{ij}=0$. To do this we prove $\forall(i,j,k,l) : \mathfrak{v}_i+\mathfrak{v}_j = \mathfrak{v}_k+\mathfrak{v}_l \wedge \mathfrak{v}_i^2+\mathfrak{v}_j^2 = \mathfrak{v}_k^2+\mathfrak{v}_l^2 \Longleftarrow (i,j,k,l)\in M_i$. This is sufficient, because $A^{k,l}_{i,j} = \frac{1}{2}\mathbb{1}_{M_i}(i,j,k,l)\alpha^k_{i,j}$.

$$(i,j,k,l)\in M_i \Longleftrightarrow \exists\mathfrak{c}\in\mathfrak{C}\exists\varphi\in G : \begin{cases} \mathfrak{v}_i = \mathfrak{c}+(\mathfrak{v}_i-\mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c}-(\mathfrak{v}_i-\mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c}+\varphi(\mathfrak{v}_i-\mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c}-\varphi(\mathfrak{v}_i-\mathfrak{c}) \end{cases}$$

$$\begin{aligned} &\Longrightarrow \mathfrak{v}_i+\mathfrak{v}_j = \mathfrak{c}+(\mathfrak{v}_i-\mathfrak{c})+\mathfrak{c}-(\mathfrak{v}_i-\mathfrak{c}) = 2\mathfrak{c} \\ &= \mathfrak{c}+\varphi(\mathfrak{v}_i-\mathfrak{c})+\mathfrak{c}-\varphi(\mathfrak{v}_i-\mathfrak{c}) = \mathfrak{v}_k+\mathfrak{v}_l \\ &\Longrightarrow \mathfrak{v}_i^2+\mathfrak{v}_j^2 = (\mathfrak{c}+(\mathfrak{v}_i-\mathfrak{c}))^2+(\mathfrak{c}-(\mathfrak{v}_i-\mathfrak{c}))^2 = 2\mathfrak{c}^2+2(\mathfrak{v}_i-\mathfrak{c})^2 \\ &\overset{\varphi\text{ is an isometry}}{=} 2\mathfrak{c}^2+2\varphi^2(\mathfrak{v}_i-\mathfrak{c}) = (\mathfrak{c}+\varphi(\mathfrak{v}_i-\mathfrak{c}))^2+(\mathfrak{c}-\varphi(\mathfrak{v}_i-\mathfrak{c}))^2 \\ &= \mathfrak{v}_k^2+\mathfrak{v}_l^2 \end{aligned}$$

b) Prerequisite:

$$
\begin{aligned}
(*) \quad \psi^{-1}(\mathfrak{v}_{i(k)}) &= \left\{ \varphi \in G \,\middle|\, \mathfrak{v}_{i(k)} = \frac{\mathfrak{v}_{i(k)} + \mathfrak{v}_{j(l)}}{2} + \varphi\left(\frac{\mathfrak{v}_{i(k)} - \mathfrak{v}_{j(l)}}{2} \right) \right\} \\
&= \left\{ \varphi \in G \,\middle|\, \mathfrak{v}_{j(l)} = \frac{\mathfrak{v}_{i(k)} + \mathfrak{v}_{j(l)}}{2} - \varphi\left(\frac{\mathfrak{v}_{i(k)} - \mathfrak{v}_{j(l)}}{2} \right) \right\} \\
&= \left\{ \varphi \in G \,\middle|\, \mathfrak{v}_{j(l)} = \frac{\mathfrak{v}_{j(l)} + \mathfrak{v}_{i(k)}}{2} + \varphi\left(\frac{\mathfrak{v}_{j(l)} - \mathfrak{v}_{i(k)}}{2} \right) \right\} \quad \text{because } \varphi \text{ is a linear map} \\
&= \psi^{-1}(\mathfrak{v}_{j(l)})
\end{aligned}
$$

I) We want to prove $A_{i,j}^{k,l} = A_{j,i}^{k,l}$. Due to $A_{i,j}^{k,l} = \mathbb{1}_{M_i}(i,j,k,l)\alpha_{ij}^k$ we have to prove the corresponding symmetry for M_i as well as α.

$$
\begin{aligned}
(i,j,k,l) \in M_i &\iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi \in G : \begin{cases} \mathfrak{v}_i = \mathfrak{c} + (\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} - (\mathfrak{v}_j - \mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} + (\mathfrak{v}_j - \mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} + \varphi(\mathfrak{c} - \mathfrak{v}_j) \\ \mathfrak{v}_l = \mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} - \varphi(\mathfrak{c} - \mathfrak{v}_j) \end{cases} \\
&\iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi' = -\varphi \in G : \begin{cases} \mathfrak{v}_j = \mathfrak{c} + (\mathfrak{v}_j - \mathfrak{c}) \\ \mathfrak{v}_i = \mathfrak{c} - (\mathfrak{v}_j - \mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c} + \varphi'(\mathfrak{v}_j - \mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c} - \varphi'(\mathfrak{v}_j - \mathfrak{c}) \end{cases} \quad \text{because } \varphi \text{ is a linear map} \\
&\iff (j,i,k,l) \in M_j
\end{aligned}
$$

$$
\begin{aligned}
\alpha_{ij}^k &= \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left| \frac{\mathfrak{v}_i - \mathfrak{v}_j}{2} \right| \right) \overline{\alpha}_{\mathfrak{v}_k \mathfrak{v}_i} \\
&= \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left| \frac{\mathfrak{v}_i - \mathfrak{v}_j}{2} \right| \right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_i)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_k)} \hat{\alpha}_{[\varphi'],[\varphi]} \\
&= \gamma\left(\frac{\mathfrak{v}_j + \mathfrak{v}_i}{2}, \left| \frac{\mathfrak{v}_j - \mathfrak{v}_i}{2} \right| \right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_j)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_k)} \hat{\alpha}_{[\varphi'],[\varphi]} \qquad \text{by } (*) \\
&= \alpha_{ji}^k
\end{aligned}
$$

II) We want to prove $A_{i,j}^{k,l} = A_{i,j}^{l,k}$. As above we have to prove the corresponding symmetry for M_i as well as α.

$$
(i,j,k,l) \in M_i \iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi \in G : \begin{cases} \mathfrak{v}_i = \mathfrak{c} + (\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c}) \end{cases}
$$

$$\iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi' = -\varphi \in G : \begin{cases} \mathfrak{v}_i = \mathfrak{c} + (\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c} + \varphi'(\mathfrak{v}_i - \mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c} - \varphi'(\mathfrak{v}_i - \mathfrak{c}) \end{cases}$$

$$\iff (i,j,l,k) \in M_i$$

$$\begin{aligned} \alpha_{ij}^k &= \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left|\frac{\mathfrak{v}_i - \mathfrak{v}_j}{2}\right|\right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_i)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_k)} \hat{\alpha}_{[\varphi'],[\varphi]} \\ &= \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left|\frac{\mathfrak{v}_i - \mathfrak{v}_j}{2}\right|\right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_i)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_l)} \hat{\alpha}_{[\varphi'],[\varphi]} \qquad \text{by } (*) \\ &= \alpha_{ij}^l \end{aligned}$$

c) Now we prove $A_{i,j}^{k,l} = A_{k,l}^{i,j}$. Same procedure as above.

$$(i,j,k,l) \in M_i \iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi \in G : \begin{cases} \mathfrak{v}_i = \mathfrak{c} + (\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} + \varphi^{-1}(\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c} - (\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} - \varphi^{-1}(\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_k = \mathfrak{c} + \varphi(\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} + (\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c} - \varphi(\mathfrak{v}_i - \mathfrak{c}) = \mathfrak{c} - (\mathfrak{v}_k - \mathfrak{c}) \end{cases}$$

$$\iff \exists \mathfrak{c} \in \mathfrak{C}\ \exists \varphi' = \varphi^{-1} \in G : \begin{cases} \mathfrak{v}_k = \mathfrak{c} + (\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_l = \mathfrak{c} - (\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_i = \mathfrak{c} + \varphi'(\mathfrak{v}_k - \mathfrak{c}) \\ \mathfrak{v}_j = \mathfrak{c} - \varphi'(\mathfrak{v}_k - \mathfrak{c}) \end{cases}$$

$$\iff (k,l,i,j) \in M_k$$

$$\begin{aligned} \alpha_{ij}^k &= \gamma\left(\frac{\mathfrak{v}_i + \mathfrak{v}_j}{2}, \left|\frac{\mathfrak{v}_i - \mathfrak{v}_j}{2}\right|\right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_i)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_k)} \hat{\alpha}_{[\varphi'],[\varphi]} \\ &= \gamma\left(\frac{\mathfrak{v}_k + \mathfrak{v}_l}{2}, \left|\frac{\mathfrak{v}_k - \mathfrak{v}_l}{2}\right|\right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_i)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_k)} \hat{\alpha}_{[\varphi'],[\varphi]} \text{ by } a) \\ &= \gamma\left(\frac{\mathfrak{v}_k + \mathfrak{v}_l}{2}, \left|\frac{\mathfrak{v}_k - \mathfrak{v}_l}{2}\right|\right) \frac{1}{2} \sum_{\varphi \in \psi^{-1}(\mathfrak{v}_k)} \sum_{\varphi' \in \psi^{-1}(\mathfrak{v}_i)} \hat{\alpha}_{[\varphi'],[\varphi]} \text{ by 2.1.3.3 (i)} \\ &= \alpha_{kl}^i \end{aligned}$$

The last claim follows from the fact, that LGpMs transformed into DVMs satisfy the minimal requirements 2.1.2.4 (i). □

Remark 2.2.4 (Difference to the original LGpM)

(i) The additional requirement for the DVM comes from the way in which discrete spheres decompose into equivalence classes 2.1.3.6 . This requirement is equivalent to the assumption, that the DVM discretizes the velocity space with $\mathfrak{V}$, but uses only spheres with centers in $\mathfrak{V} \cup \mathfrak{V}_{\frac{1}{2}}$. This can be interpreted as a standard DVM where the approximation of the spherical integral (compare (2.1.1)) gets improved without improving the approximation of the outer integral.

(ii) There are two differences in the original LGpM and the one obtained from a DVM transformation:

$$\sum_{\mathfrak{c}\in\mathfrak{C}} \overbrace{\sum_{[\mathfrak{v}]\in\tilde{S}^{\mathfrak{V}}_{i,\mathfrak{c}}}}^{(*)} \sum_{[\varphi]\in G/\sim_H} \overbrace{\alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i}}^{(**)} \left(\prod_{\varphi'\in[\varphi]} f(\mathfrak{c} + \varphi'(\mathfrak{v} - \mathfrak{c})) - \prod_{\varphi''\in H} f(\mathfrak{c} + \varphi''(\mathfrak{v}_i - \mathfrak{c})) \right) ,$$

the sum $(*)$ is non-existent in the original LGpM 2.1.3.4 (compare [Bab09, chapter 2,3], [Bab11a, introduction]). The LGpM was not designed by Babovsky to be consistent with the Boltzmann equation, but to give an easy to calculate model that possesses and reflects the main properties of the Boltzmann equation. This "missing" sum $(*)$ is necessary to realize the convergence of the spherical integral in the Boltzmann equation, because without it every sphere is approximated by a maximum of eight points (at least in two dimensions). The second difference $(**)$ can be found in $\alpha^{\varphi;\bullet}_{\mathfrak{c},\mathfrak{v}_i}$, which now depends on the equivalence classes $[\mathfrak{v}]$ in which the discrete spheres decompose. For every fixed $[\mathfrak{v}]$ the coefficient $\alpha^{\varphi;\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i}$ should possess the same general properties as the original $\alpha^{\varphi}_{\mathfrak{c},\mathfrak{v}_i}$. Due to the modifications (with partially unknown consequences) of the original LGpM we call this modified version eLGpM (extended lattice group model). We will investigate the theoretic implications of this modified LGpM in another work, because in this thesis we aim at consistency and convergence results.

3 Consistency

In this section we aim at large models for rarefied gases, where the convergence of the discretization becomes important. To that aim we give a specific discretization of the collision operator through Farey angles and transform the resulting scheme into a DVM and an eLGpM. This allows us to use the simple representation as a DVM for calculations as well as the eLGpM representation of this discretization to obtain theoretical results about this discretization through the LGpM approach in the future. The underlying approach is not new, but we refine the approaches of other authors, like [RS94, PH99, MS00] and use an interpretation through automorphism groups to reach the point at which we can generalize our discretization to obtain arbitrary convergence orders without the loss of the correct collisional invariants, exact conservation of the associated quantities, the correct equilibrium solutions and the H-Theorem. Throughout this section we restrict ourselves to look on “inner” points of the velocity space. This typically means that the point $\mathfrak{v}$ has at least a distance of $L \in \mathbb{R}^+$ to the next boundary point.

3.1 Preliminaries

Because we want to use Farey angles for a discretization of the Boltzmann equation we shortly state the idea and some facts around this approach. The first application of this approach in the case of the Boltzmann equation was introduced by Rogier and Schneider in [RS94]. We will use a similar approach, but with other transformations of the Boltzmann equation. Basics about the Farey sequence can be found in [HW60, chapter 3].

Definition 3.1.1 (Farey sequences)
Let the Farey sequence $\tilde{\mathfrak{F}}_n$ of order n be a vector of ascending elements

$$\tilde{\mathfrak{F}}_n := (F_1, \dots, F_N)^T \in ([0,1] \cap \mathbb{Q})^N, \quad F_i := \frac{p_i}{q_i}; p_i, q_i \in \mathbb{N}_0 \ ,$$

with

$$\{F_1, \dots, F_N\} = \left\{ \frac{p_i}{q_i} \,\middle|\, \begin{array}{l} i \in \{1, \dots, N\};\ p_i, q_i, \tilde{p}_i, \tilde{q}_i \in \mathbb{N}_0; 0 \le \tilde{p}_i \le \tilde{q}_i \le n \\ \wedge \frac{p_i}{q_i} = \frac{\tilde{p}_i}{\tilde{q}_i} \wedge \gcd(p_i, q_i) = 1 \wedge \frac{p_{j-1}}{q_{j-1}} < \frac{p_j}{q_j}, j > 1 \end{array} \right\} .$$

The elements of the Farey sequence $F_i = \frac{p_i}{q_i}$ are maximal reduced fractions and can be seen as equivalence classes for the non reduced fractions $\frac{\tilde{p}_i}{\tilde{q}_i}$. In the following work we will put letters in the second lower right index of numbers in the context of Farey sequences to specify the sequence to which these numbers belong. For example $F_{1,n}, F_{1,m}$ is the first element of the Farey sequence $\tilde{\mathfrak{F}}_n$ resp. $\tilde{\mathfrak{F}}_m$. We suppress this index if there is no risk of confusion.

Remark 3.1.2 (Geometric interpretation of Farey sequences)
This sequence is of special interest, because the elements of the Farey sequences correspond to all possible growth rates of lines that fit onto a uniform discretization of the velocity space and go through zero, at least to the lower half of the first quadrant, see figure 3.1. The first Farey sequences are given by

$$
\begin{aligned}
\tilde{\mathfrak{F}}_1 &= \{0, 1\}\,, & \tilde{\mathfrak{F}}_4 &= \left\{0, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1\right\},\\
\tilde{\mathfrak{F}}_2 &= \left\{0, \frac{1}{2}, 1\right\}, & \tilde{\mathfrak{F}}_5 &= \left\{0, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, 1\right\},\\
\tilde{\mathfrak{F}}_3 &= \left\{0, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1\right\}, & \tilde{\mathfrak{F}}_6 &= \left\{0, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, 1\right\}
\end{aligned}
$$

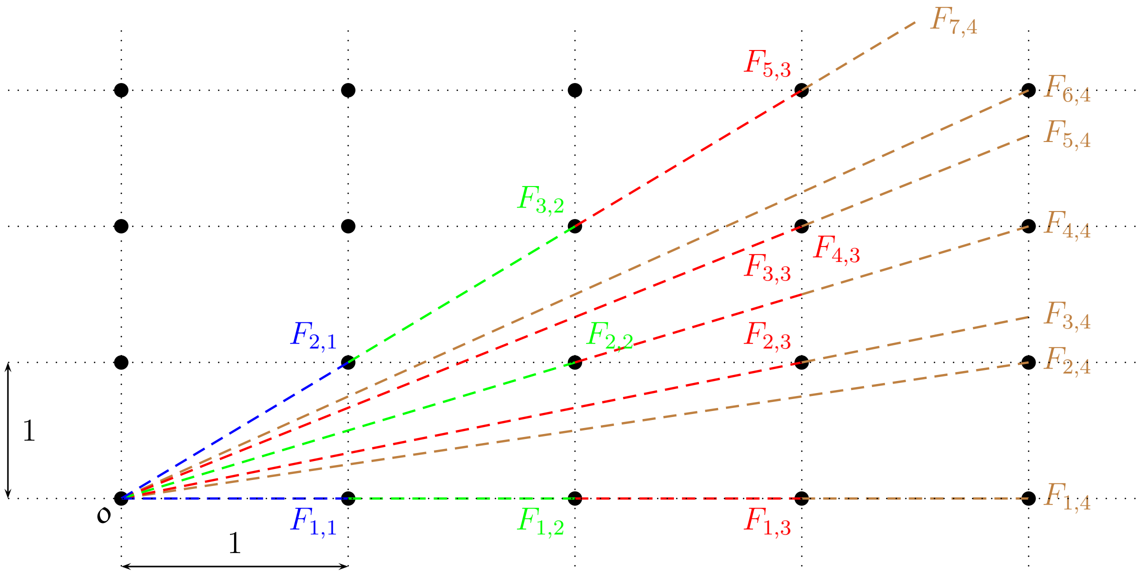


Figure 3.1: *Correspondence between the first four Farey sequences and lines on the grid*

This correspondence can be used to approximate integrals over spheres.

Definition 3.1.3 (Farey angles)
We get the ascending sequence of Farey angles by

$$\mathfrak{F}_n := \arctan\left(\tilde{\mathfrak{F}}_n\right) = \left(\arctan(F_1), \ldots, \arctan(F_N)\right)^T = (\theta_1, \ldots, \theta_N)^T\,.$$

And the last ingredient are the circle arcs corresponding to the so called Farey arcs:

$$\alpha_i := \mu_{i+1} - \mu_i, \quad \mu_1 := \arctan(0), \mu_{N+1} := \arctan(1), \mu_i := \arctan\left(\frac{p_i + p_{i-1}}{q_i + q_{i-1}}\right)\,.$$

The fractions $\frac{p_i+p_{i-1}}{q_i+q_{i-1}}$ are so called mediants, as we can see in the next proof they have the property

$$\frac{p_{i-1}}{q_{i-1}} < \frac{p_i + p_{i-1}}{q_i + q_{i-1}} < \frac{p_i}{q_i} < \frac{p_{i+1} + p_i}{q_{i+1} + q_i}\,.$$

The μ_i are simply the associated angles and we use the α_i as the step sizes for the approximation of the integral in spherical coordinates. We use this detour to calculate the step sizes, because there is a result for the α_i ((iii) of the next lemma) that is necessary to get a convergence result for the following approximation.

Lemma 3.1.4 (Basic properties of Farey arcs)
A similar version of this lemma can be found in [RS94, section 1.2].

(i) $0 \leq \mu_i < \theta_i < \mu_{i+1} \leq \frac{\pi}{4}$

(ii) $\alpha_i^2 > (\theta_i - \mu_i)^2 \wedge \alpha_i^2 > (\mu_{i+1} - \theta_i)^2$

(iii) $0 \leq \alpha_i < \frac{2}{nq_i}$

(iv) $\sum\limits_{i=1}^{N} \alpha_{i,n} = \frac{\pi}{4}$

Proof:

(i) Due to the monotonicity of arctan it is sufficient to take a look at the fractions associated to μ, θ:

$$\begin{aligned}\frac{p_{i-1}}{q_{i-1}} < \frac{p_i}{q_i} &\iff p_{i-1}q_i < p_i q_{i-1} \iff q_i p_i + q_i p_{i-1} < p_i q_i + p_i q_{i-1} \\ &\iff q_i(p_i + p_{i-1}) < p_i(q_i + q_{i-1}) \iff \frac{p_i + p_{i-1}}{q_i + q_{i-1}} < \frac{p_i}{q_i} \\ &\iff \mu_i < \theta_i\end{aligned}$$

and analogical

$$\begin{aligned}\frac{p_{i+1}}{q_{i+1}} > \frac{p_i}{q_i} &\iff p_{i+1}q_i > p_i q_{i+1} \iff q_i p_i + q_i p_{i+1} > p_i q_i + p_i q_{i+1} \\ &\iff q_i(p_i + p_{i+1}) > p_i(q_i + q_{i+1}) \iff \frac{p_i + p_{i+1}}{q_i + q_{i+1}} > \frac{p_i}{q_i} \\ &\iff \mu_{i+1} > \theta_i\,.\end{aligned}$$

(ii) From (i) follows:

$$\alpha_i = \mu_{i+1} - \mu_i > \mu_{i+1} - \theta_i \implies \alpha_i^2 = (\mu_{i+1} - \mu_i)^2 > (\mu_{i+1} - \theta_i)^2\,,$$
$$\alpha_i = \mu_{i+1} - \mu_i > \theta_i - \mu_i \implies \alpha_i^2 = (\mu_{i+1} - \mu_i)^2 > (\theta_i - \mu_i)^2\,,$$

(iii) [HW60, Chapter 3, Theorem 35]

(iv) $\sum\limits_{i=1}^{N} \alpha_i = \sum\limits_{i=1}^{N} \mu_{i+1} - \mu_i = \mu_{N+1} - \mu_1 = \arctan(1) - \arctan(0) = \frac{\pi}{4}$

□

The following proposition as well as the main steps of the proof can be extracted from [RS94, Theorem 3]. This proposition corresponds to a part of this theorem, is adapted to fit into this work and we give a more detailed proof.

Proposition 3.1.5 (Farey approximation)
Let $H \in C^1(\mathbb{R} \to \mathbb{R})$. Using the Farey angles $\mathfrak{F}_n$, the approximation

$$\sum_{i=1}^{N} \alpha_{i,n} H(\theta_i) \approx \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta$$

has the following order of convergence:

$$\mathcal{O}\left(\frac{\ln(n)}{n^2}\right) .$$

An upper bound of the error is

$$e := \left| \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta - \sum_{i=1}^{N} \alpha_{i,n} H(\theta_i) \right| < \sup_{\theta \in \left[0,\frac{\pi}{4}\right]} |H'(\theta)| \, \frac{4\ln(n)+4}{n^2} .$$

Proof:

$$\begin{aligned}
e &= \left| \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta - \sum_{i=1}^{N} \alpha_{i,n} H(\theta_i) \right| \\
&= \left| \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta - \sum_{i=1}^{N} (\mu_{i+1}-\mu_i) H(\theta_i) \right| && \text{by 3.1.3} \\
&= \left| \sum_{i=1}^{N} \int_{\mu_i}^{\mu_{i+1}} H(\theta) - H(\theta_i)\mathrm{d}\theta \right| && \text{by 3.1.4 (i)} \\
&= \left| \sum_{i=1}^{N} \int_{\mu_i}^{\theta_i} H(\theta) - H(\theta_i)\mathrm{d}\theta - \int_{\theta_i}^{\mu_{i+1}} H(\theta_i) - H(\theta)\mathrm{d}\theta \right| \\
&= \left| \sum_{i=1}^{N} \int_{\mu_i}^{\theta_i} \frac{H(\theta) - H(\theta_i)}{\theta-\theta_i}(\theta-\theta_i)\mathrm{d}\theta - \int_{\theta_i}^{\mu_{i+1}} \frac{H(\theta_i) - H(\theta)}{\theta_i-\theta}(\theta_i-\theta)\mathrm{d}\theta \right| \\
&= \left| \sum_{i=1}^{N} \int_{\mu_i}^{\theta_i} H'(\xi(\theta))(\theta-\theta_i)\mathrm{d}\theta - \int_{\theta_i}^{\mu_{i+1}} H'(\tilde{\xi}(\theta)(\theta_i-\theta)\mathrm{d}\theta \right| && \text{by MVT} \\
&\leq \sum_{i=1}^{N} \left| \int_{\mu_i}^{\theta_i} H'(\xi(\theta))(\theta-\theta_i)\mathrm{d}\theta \right| + \left| \int_{\theta_i}^{\mu_{i+1}} H'(\tilde{\xi}(\theta)(\theta-\theta_i)\mathrm{d}\theta \right|
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{N} \left| \sup_{\varphi\in[\mu_i,\theta_i]} |H'(\varphi)| \int_{\mu_i}^{\theta_i} (\theta-\theta_i)\mathrm{d}\theta \right| + \left| \sup_{\hat{\varphi}\in[\theta_i,\mu_{i+1}]} |H'(\hat{\varphi})| \int_{\theta_i}^{\mu_{i+1}} (\theta-\theta_i)\mathrm{d}\theta \right| \\
&\leq \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \sum_{i=1}^{N} \left| \frac{\theta_i^2}{2} - \frac{\mu_i^2}{2} - \theta_i^2 + \theta_i\mu_i \right| + \left| \frac{\mu_{i+1}^2}{2} - \frac{\theta_i^2}{2} - \theta_i\mu_{i+1} + \theta_i^2 \right| \\
&= \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \sum_{i=1}^{N} \frac{1}{2}(\theta_i-\mu_i)^2 + \frac{1}{2}(\mu_{i+1}-\theta_i)^2 \\
&< \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \sum_{i=1}^{N} \alpha_{i,n}^2 \qquad \text{by 3.1.4 (ii)} \\
&< \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \sum_{i=1}^{N} \left(\frac{2}{nq_i}\right)^2 \qquad \text{by 3.1.4 (iii)} \\
&= \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \frac{2^2}{n^2} \sum_{i=1}^{N} \left(\frac{1}{q_i}\right)^2 \leq \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \frac{2^2}{n^2} \sum_{q=1}^{n} \sum_{p=1}^{q} \frac{1}{q^2} \\
&= \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \frac{2^2}{n^2} \sum_{q=1}^{n} \frac{1}{q} \leq \sup_{\varphi\in\left[0,\frac{\pi}{4}\right]} |H'(\varphi)| \frac{4\ln(n)+4}{n^2} \qquad \square
\end{aligned}
$$

Remark 3.1.6

(i) The above discretization can be interpreted as

$$
\sum_{i=1}^{N} \alpha_{i,n} \tilde{H}(\omega(\theta_i)) \approx \int_{S^1_{\frac{1}{8}}} \tilde{H}(\mathfrak{w})\mathrm{d}\mathfrak{w}, \qquad \omega(\theta) = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}
$$

with $H(\theta) = \tilde{H}(\omega(\theta))$ and $S^1_{\frac{1}{8}}$ being $\frac{1}{8}$ of the unit sphere in two dimensions. We use the representation in the above proposition, because in the next subsection we first simplify the transformation of the collision operator before we use the Farey approximation.

(ii) At a first glance this discretization looks uninteresting. The nice thing about this discretization is that it only uses angles that have the important property to correspond to the gradient of lines that fit onto our uniform grid. This makes this discretization especially suited for integration in polar or spherical coordinates.

3.2 Two dimensions

Now we start to derive a specific discretization of the collision operator by using the Farey approximation. For this we need to transform the collision operator in a convenient way for the following discretization.

Lemma 3.2.1 (Transformation of the collision operator)
Let us assume that the density f has the property $\mathrm{supp}(f) \subset B_{\frac{L}{2}}(\mathfrak{o}), L \in \mathbb{R}^+$ and let $\mathfrak{v} \in B_{\frac{L}{2}}(\mathfrak{o})$, $\omega(\theta) := \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$, $\omega^{\perp}(\theta) := \omega(\theta + \frac{\pi}{2})$, then the following holds true:

$$\begin{aligned} I[f](\mathfrak{v}) &= \int_{\mathbb{R}^2} \int_{S^1} \left[f\left(\mathfrak{v}_1'\right) f\left(\mathfrak{w}_1'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}\right)\right] k\left(\mathfrak{v}, \mathfrak{w}, \eta\right) \, \mathrm{d}\eta \, \mathrm{d}\mathfrak{w} \\ &= \int_0^{2\pi} \int_0^{2\pi} \int_0^L \left[f\left(\mathfrak{v}_2'\right) f\left(\mathfrak{w}_2'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}_2\right)\right)\right] lk\left(\mathfrak{v}, \mathfrak{w}_2, \omega(\theta)\right) \, \mathrm{d}l \, \mathrm{d}\lambda \, \mathrm{d}\theta \; , \end{aligned}$$

with

$$\begin{aligned} \mathfrak{v}_1' &:= \mathfrak{v} + \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta, & \mathfrak{w}_1' &:= \mathfrak{w} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta \; , \\ \mathfrak{v}_2' &:= \mathfrak{v} + l \langle \omega(\theta + \lambda), \omega(\theta) \rangle \omega(\theta), & \mathfrak{w}_2' &:= \mathfrak{v} + l \langle \omega(\theta + \lambda), \omega^{\perp}(\theta) \rangle \omega^{\perp}(\theta) \; , \\ \mathfrak{w}_2 &:= \mathfrak{v} + l\omega(\theta + \lambda) \; . \end{aligned}$$

Proof:

(i) start

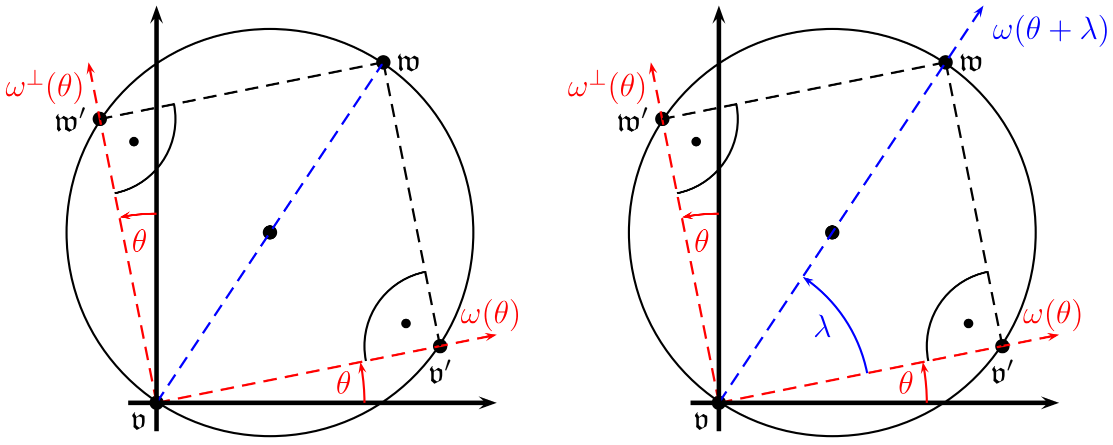
$$\begin{aligned} &I[f](\mathfrak{v}) = \int_{\mathbb{R}^2} \int_{S^1} \left[f\left(\mathfrak{v}'\right) f\left(\mathfrak{w}'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}\right)\right] k\left(\mathfrak{v}, \mathfrak{w}, \eta\right) \, \mathrm{d}\eta \, \mathrm{d}\mathfrak{w}, \\ &\mathfrak{v}' = \mathfrak{v} + \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta, \qquad \mathfrak{w}' = \mathfrak{w} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \eta \rangle \eta \end{aligned}$$

(ii) transformation 1 (compare figure 3.2a):

$$\begin{aligned} &\Phi(\{1\} \times [0, 2\pi]) = S^1, \quad \Phi \begin{pmatrix} 1 \\ \theta \end{pmatrix} = \omega(\theta) := 1 \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} (= \eta), \quad \det(D\Phi) = 1 \\ &\Longrightarrow I[f](\mathfrak{v}) = \int_{\mathbb{R}^2} \int_0^{2\pi} \left[f\left(\mathfrak{v}'\right) f\left(\mathfrak{w}'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}\right)\right] k\left(\mathfrak{v}, \mathfrak{w}, \omega(\theta)\right) \, \mathrm{d}\theta \, \mathrm{d}\mathfrak{w} \; , \\ &\mathfrak{v}' = \mathfrak{v} + \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \omega(\theta) \rangle \omega(\theta), \quad \mathfrak{w}' = \mathfrak{w} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \omega(\theta) \rangle \omega(\theta) = \mathfrak{v} + \overrightarrow{\mathfrak{v}\mathfrak{w}} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \omega(\theta) \rangle \omega(\theta) \end{aligned}$$

(iii) transformation 2 (compare figure 3.2b):

$$\begin{aligned} &\Phi(\mathbb{R}^+_{\{0\}} \times [0, 2\pi]) = \mathbb{R}^2, \quad \Phi \begin{pmatrix} l \\ \lambda \end{pmatrix} = l\omega(\theta + \lambda) (= \overrightarrow{\mathfrak{v}\mathfrak{w}}), \quad \det(D\Phi) = l \\ &\Longrightarrow I[f](\mathfrak{v}) = \int_0^{2\pi} \int_{\mathbb{R}^2} \left[f\left(\mathfrak{v}'\right) f\left(\mathfrak{w}'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{v} + \overrightarrow{\mathfrak{v}\mathfrak{w}}\right)\right] lk\left(\mathfrak{v}, \mathfrak{w}, \omega(\theta)\right) \, \mathrm{d}\mathfrak{w} \, \mathrm{d}\theta \\ &= \int_0^{2\pi} \int_0^{2\pi} \int_0^L \Big[f\left(\mathfrak{v}'\right) f\left(\mathfrak{w}'\right) - f\left(\mathfrak{v}\right) f\big(\overbrace{\mathfrak{v} + l\omega(\theta + \lambda)}^{\mathfrak{w}_2 :=}\big)\Big] lk\left(\mathfrak{v}, \mathfrak{w}_2, \omega(\theta)\right) \, \mathrm{d}l \, \mathrm{d}\lambda \, \mathrm{d}\theta \, , \end{aligned}$$



(a) *Transformation 1* (b) *Transformation 2*

Figure 3.2: *Visualization of the transformations*

because $\operatorname{supp}(f) \subset B_{\frac{L}{2}}(\mathfrak{o})$ and with

$$\begin{aligned}
\mathfrak{v}' &= \mathfrak{v} + \langle l\omega(\theta+\lambda), \omega(\theta)\rangle \omega(\theta) =: \mathfrak{v}_2' \\
\mathfrak{w}' &= \mathfrak{v} + \overrightarrow{\mathfrak{v}\mathfrak{w}} - \langle \overrightarrow{\mathfrak{v}\mathfrak{w}}, \omega(\theta)\rangle\omega(\theta) = \mathfrak{v} + l\omega(\theta+\lambda) - \langle l\omega(\theta+\lambda), \omega(\theta)\rangle\omega(\theta) \\
&\overset{(*)}{=} \mathfrak{v} + l\cos(\lambda)\omega(\theta) + l\sin(\lambda)\omega\left(\theta + \frac{\pi}{2}\right) \\
&- l\Big(\cos(\lambda)\underbrace{\langle\omega(\theta),\omega(\theta)\rangle}_{=1} + \sin(\lambda)\underbrace{\left\langle \omega\left(\theta+\frac{\pi}{2}\right), \omega(\theta)\right\rangle}_{=0}\Big)\omega(\theta) \\
&= \mathfrak{v} + l\sin(\lambda)\omega\left(\theta+\frac{\pi}{2}\right) = v + l\cos\left(\lambda - \frac{\pi}{2}\right)\omega\left(\theta+\frac{\pi}{2}\right) \\
&= \mathfrak{v} + l\left\langle \omega\left(\lambda - \frac{\pi}{2} + \theta\right), \omega(\theta)\right\rangle \omega\left(\theta+\frac{\pi}{2}\right) \\
&= \mathfrak{v} + l\left\langle \omega(\lambda+\theta), \overbrace{\omega\left(\theta+\frac{\pi}{2}\right)}^{\omega^\perp(\theta):=}\right\rangle \omega\left(\theta+\frac{\pi}{2}\right) \\
&= \mathfrak{v} + \langle l\omega(\theta+\lambda), \omega^\perp(\theta)\rangle\omega^\perp(\theta) =: \mathfrak{w}_2', \qquad \text{here we used} \\
(*): \omega(a+b) &= \begin{pmatrix}\cos(a+b)\\ \sin(a+b)\end{pmatrix} = \begin{pmatrix}\cos(a)\cos(b) - \sin(a)\sin(b)\\ \sin(a)\cos(b) + \cos(a)\sin(b)\end{pmatrix} \\
&= \cos(b)\omega(a) + \sin(b)\omega\left(a + \frac{\pi}{2}\right)
\end{aligned}$$

(iv) simplification: There are 4 symmetries that can be used to reduce the domain of the second integral to a quarter of the original one. We do not use these symmetries at this point, but we want to point out, that this fact causes no harm. We simply get the integrals multiplied by a factor of 4. We justify this

with the following calculation:
Assuming that

$$\theta - \frac{3}{2}\pi = \theta_1 - \pi = \theta_2 - \frac{\pi}{2} = \theta_3$$

and knowing that

$$-\omega(\theta) = -\begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix} = \omega(\theta+\pi) = \omega^\perp\left(\theta + \frac{\pi}{2}\right)$$

we get

$$\begin{aligned}
\mathfrak{w}_2(\lambda,\theta) &= \mathfrak{w}_2\left(\left[\lambda + \frac{\pi}{2}\right] + \theta_1\right) = \mathfrak{w}_2\left([\lambda+\pi] + \theta_2\right) = \mathfrak{w}_2\left(\left[\lambda + \frac{3\pi}{2}\right] + \theta_3\right)\ ,\\
\mathfrak{w}_2'(\lambda,\theta) &= \mathfrak{v} + l\langle\omega(\lambda+\theta), \omega^\perp(\theta)\rangle\omega^\perp(\theta)\\
&= \mathfrak{v} + l\left\langle\omega\left(\left[\lambda+\frac{\pi}{2}\right]+\theta_1\right), \omega^\perp\left(\theta_1 + \frac{\pi}{2}\right)\right\rangle\omega^\perp\left(\theta_1+\frac{\pi}{2}\right)\\
&= \mathfrak{v} + l\left\langle\omega\left(\left[\lambda+\frac{\pi}{2}\right]+\theta_1\right), \omega\left(\theta_1\right)\right\rangle\omega\left(\theta_1\right)\\
&= \mathfrak{v}_2'\left(\lambda + \frac{\pi}{2}, \theta_1\right)\\
&= \mathfrak{v} + l\langle\omega\left([\lambda+\pi]+\theta_2\right), \omega^\perp\left(\theta_2+\pi\right)\rangle\omega^\perp\left(\theta_2+\pi\right)\\
&= \mathfrak{v} + l\langle\omega\left([\lambda+\pi]+\theta_2\right), \omega^\perp\left(\theta_2\right)\rangle\omega^\perp\left(\theta_2\right)\\
&= \mathfrak{w}_2'(\lambda+\pi, \theta_2)\\
&= \mathfrak{v} + l\left\langle\omega\left(\left[\lambda+\frac{3\pi}{2}\right]+\theta_3\right), \omega^\perp\left(\theta_3+\frac{3\pi}{2}\right)\right\rangle\omega^\perp\left(\theta_3+\frac{3\pi}{2}\right)\\
&= \mathfrak{v} + l\left\langle\omega\left(\left[\lambda+\frac{3\pi}{2}\right]+\theta_3\right), \omega\left(\theta_3+\pi\right)\right\rangle\omega\left(\theta_3+\pi\right)\\
&= \mathfrak{v} + l\left\langle\omega\left(\left[\lambda+\frac{3\pi}{2}\right]+\theta_3\right), \omega\left(\theta_3\right)\right\rangle\omega\left(\theta_3\right)\\
&= \mathfrak{v}_2'\left(\lambda+\frac{3\pi}{2}, \theta_3\right)\ ,\ \text{and analog:}\\
\mathfrak{v}_2'(\lambda,\theta) &= \mathfrak{w}_2'\left(\lambda+\frac{\pi}{2},\theta_1\right) = \mathfrak{v}_2'\left(\lambda+\pi,\theta_2\right) = \mathfrak{w}_2'\left(\lambda+\frac{3\pi}{2},\theta_3\right)\ .
\end{aligned}$$

□

Remark 3.2.2
The condition $\operatorname{supp}(f) \subset B_{\frac{L}{2}}(\mathfrak{o}), L \in \mathbb{R}^+$ is obviously not met, because f is typically a distorted normal distribution. Nevertheless the exponential decline of $f(\mathfrak{v})$ for growing $\|\mathfrak{v}\|$ justifies this assumption, especially because nearly every numerical scheme using a finite grid is forced to this truncation. Due to the exponential decline it is possible to hold the corresponding error into check by calculating a compact region where a

specific (large) percentage of the integral is situated and set f to zero outside of this domain. This approach seems to be the standard in this field of research and the corresponding assumption on f or the asymptotic exponential decline and truncation is generally used in convergence proofs, for example in [RS94, PSB97]. From now on we will always assume

$$\operatorname{supp}(f) \subset B_{\frac{L}{2}}(\mathfrak{o}), L \in \mathbb{R}^+ .$$

The following theorem can be seen as an extension of the discretization obtained by Rogier and Schneider in [RS94]. In contrast to them we use another transformation of the collision integral (they only transform the inner integral into polar coordinates and thus have only one angle) and we look at general Newton-Cotes formula for the approximation of the innermost integral.

Theorem 3.2.3 (Farey discretization of the collision operator)
Defining

$$h(l,\lambda,\theta) := \left[f\big(\mathfrak{v}_2'\big)f\big(\mathfrak{w}_2'\big) - f\big(\mathfrak{v}\big)f\big(\mathfrak{w}_2\big)\right] lk\left(\mathfrak{v},\mathfrak{w}_2,\omega(\theta)\right) ,$$

and assuming $f \in C^r(\mathbb{R}^2 \to \mathbb{R}),\ k \in C^r(\mathbb{R}^2 \times \mathbb{R}^2 \times S^1 \to \mathbb{R})$, the transformed collision operator

$$I[f](\mathfrak{v}) = \int_0^{\frac{\pi}{4}} \int_0^{\frac{\pi}{4}} \int_0^L h(l,\lambda,\theta)\,\mathrm{d}l\,\mathrm{d}\lambda\,\mathrm{d}\theta$$

can be approximated by

$$\tilde{I}[f](\mathfrak{v}) := \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} g(k) h(l_k,\lambda_j,\theta_i) ,$$

where g represents the weight function corresponding to the Newton-Cotes formula used to approximate the innermost integral and

$$\Delta v_{ij} := \Delta v \sqrt{p_{i,n}^2 + q_{i,n}^2}\sqrt{p_{j,m}^2 + q_{j,m}^2}, \quad l_k = k \cdot \Delta v_{ij} .$$

This approach yields an upper error bound of

$$\left| I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v}) \right| < 4K_\theta \frac{\ln(n)+1}{n^2} + \pi K_\lambda \frac{\ln(m)+1}{m^2} + 3 \cdot 2^{r+1} c L K_l (\Delta v)^r n^r m^r ,$$

where K_l, K_λ, K_θ are some constants depending only on f, k and r, c correspond to the used Newton-Cotes formula, r being the error order and c corresponding to some error constants. A simplification in the calculation of

$$h(l_k,\lambda_j,\theta_i) = \left[f(\mathfrak{v}_2'(i,j,k)) f(\mathfrak{w}_2'(i,j,k)) - f(\mathfrak{v}) f(\mathfrak{w}_2(i,j,k))\right] lk(\mathfrak{v},\mathfrak{w}_2(i,j,k),\omega(\theta_i))$$

is given through:

$$\mathfrak{w}_2(l_k,\lambda_j,\theta_i) = \mathfrak{v} + l_k \omega(\theta_i + \lambda_j) \qquad\qquad = \mathfrak{v} + k\Delta v \begin{pmatrix} q_i q_j - p_i p_j \\ p_i q_j + p_j q_i \end{pmatrix} ,$$

$$\begin{aligned}
\mathfrak{v}'_2(l_k,\lambda_j,\theta_i) &= \mathfrak{v} + l_k\langle\omega(\theta_i+\lambda_j),\omega(\theta_i)\rangle\omega(\theta_i) &&= \mathfrak{v} + k\Delta v q_j \begin{pmatrix} q_i \\ p_i \end{pmatrix} , \\
\mathfrak{w}'_2(l_k,\lambda_j,\theta_i) &= \mathfrak{v} + l_k\langle\omega(\theta_i+\lambda_j),\omega^{\perp}(\theta_i)\rangle\omega^{\perp}(\theta_i) &&= \mathfrak{v} + k\Delta v p_j \begin{pmatrix} -p_i \\ q_i \end{pmatrix} .
\end{aligned}$$

Proof:
In the above theorem and throughout this proof we use the Farey sequences $\mathfrak{F}_n, \mathfrak{F}_m$ with the corresponding sequence lengths N, M. To prove the error bound we define

$$H_1(\lambda,\theta) := \int_0^L h(l,\lambda,\theta)\mathrm{d}l, \qquad H_2(\theta) := \int_0^{\frac{\pi}{4}} H_1(\lambda,\theta)\mathrm{d}\lambda ,$$

and with this we will take a closer look at the three occurring approximation errors:

$$\begin{aligned}
e_3 &:= \int_0^{\frac{\pi}{4}} H_2(\theta)\mathrm{d}\theta - \sum_{i=1}^{N} \alpha_{i,n} H_2(\theta_i) , \\
e_2(i) &:= H_2(\theta_i) - \sum_{j=1}^{M} \alpha_{j,m} H_1(\lambda_j,\theta_i), \qquad e_2 := \sum_{i=1}^{N} \alpha_{i,n} e_2(i) ,
\end{aligned}$$

we want to use a closed Newton-Cotes formula for the innermost integration, so we bloat the innermost integral by adding a zero:

$$\begin{aligned}
e_1(i,j) &:= H_1(\lambda_j,\theta_i) - \lceil L/\Delta v_{ij}\rceil \Delta v_{ij} \sum_{k=1}^{\lfloor L/\Delta v_{ij}\rfloor} g(k) h(l_k,\lambda_j,\theta_i) \\
&= \int_0^L h(l,\lambda,\theta)\mathrm{d}l - \lceil L/\Delta v_{ij}\rceil \Delta v_{ij} \sum_{k=1}^{\lfloor L/\Delta v_{ij}\rfloor} g(k) h(l_k,\lambda_j,\theta_i) \\
&- \lceil L/\Delta v_{ij}\rceil \Delta v_{ij} \big[g(\lceil L/\Delta v_{ij}\rceil) \overbrace{h(l_{\lceil L/\Delta v_{ij}\rceil},\lambda_j,\theta_i)}^{0=} - g(0) \overbrace{h(0,\lambda_j,\theta_i)}^{0=} \big] \\
&= \int_0^{\lceil L/\Delta v_{ij}\rceil \Delta v_{ij}} h(l,\lambda,\theta)\mathrm{d}l - \lceil L/\Delta v_{ij}\rceil \Delta v_{ij} \sum_{k=0}^{\lceil L/\Delta v_{ij}\rceil} g(k) h(l_k,\lambda_j,\theta_i), \\
e_1 &:= \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} e_1(i,j) .
\end{aligned}$$

Here $g(k)$ is the weight function corresponding to the used Newton-Cotes formula. To understand the step size of the Newton-Cotes formula we calculate the grid point corresponding to $\omega(\theta_i+\lambda_j)$ by using the underlying Farey sequences:

$$\tan(\theta_i+\lambda_j) = \frac{\tan(\theta_i)+\tan(\lambda_j)}{1-\tan(\theta_i)\tan(\lambda_j)} = \frac{\left(\frac{p_i}{q_i}+\frac{p_j}{q_j}\right)}{1-\frac{p_i p_j}{q_i q_j}} = \frac{p_i q_j + p_j q_i}{q_i q_j - p_i p_j}$$

$$\begin{aligned}\Longrightarrow \begin{pmatrix} q_iq_j - p_ip_j \\ p_iq_j + p_jq_i \end{pmatrix} &= \left|\begin{pmatrix} q_iq_j - p_ip_j \\ p_iq_j + p_jq_i \end{pmatrix}\right| \cdot \omega(\theta_i + \lambda_j) \\ &= \sqrt{(q_iq_j - p_ip_j)^2 + (p_iq_j + p_jq_i)^2} \cdot \omega(\theta_i + \lambda_j) \\ &= \sqrt{(p_i^2 + q_i^2)(p_j^2 + q_j^2)} \cdot \omega(\theta_i + \lambda_j)\,.\end{aligned}$$

For the next step we need the abbreviations

$$\begin{aligned} P_i &:= \begin{pmatrix} q_i \\ p_i \end{pmatrix} = |P_i|\omega(\theta_i), & r_i &:= |P_i| = \sqrt{q_i^2 + p_i^2}\,, \\ P_{ij} &:= \begin{pmatrix} q_iq_j - p_ip_j \\ p_iq_j + p_jq_i \end{pmatrix} = |P_{ij}|\omega(\theta_i + \lambda_j), & r_{ij} &:= |P_{ij}| = \sqrt{(p_i^2 + q_i^2)(p_j^2 + q_j^2)}\,. \end{aligned}$$

Using this we can take a closer look at $\mathfrak{w}_2, \mathfrak{v}'_2, \mathfrak{w}'_2$. These velocities are given by

$$\begin{aligned} \mathfrak{w}_2 &:= \mathfrak{v} + l_k\omega(\theta_i + \lambda_j)\,, \\ \mathfrak{v}'_2 &:= \mathfrak{v} + \langle l_k\omega(\theta_i + \lambda_j), \omega(\theta_i)\rangle\omega(\theta_i), \qquad \mathfrak{w}'_2 := \mathfrak{v} + \langle l_k\omega(\theta_i + \lambda_j), \omega^\perp(\theta_i)\rangle\omega^\perp(\theta_i)\ , \end{aligned}$$

and now we want to calculate the occurring scalar products as well as l_k. At first we want that these velocities lie on our grid. So a first guess for l_k is (according to $\mathfrak{w}_2$)

$$l_k := k \cdot \overbrace{\Delta v \cdot r_{ij}}^{\Delta v_{ij} :=} = k \cdot \Delta v_{ij}\,.$$

Now let us see if this results in $\mathfrak{v}'_2, \mathfrak{w}'_2$ lying on the grid:

$$\begin{aligned} &\langle \omega(\theta_i + \lambda_j), \omega(\theta_i)\rangle = x \Longleftrightarrow x = \frac{\langle r_{ij}\omega(\theta_i + \lambda_j), r_i\omega(\theta_i)\rangle}{r_{ij}r_i} \\ \Longleftrightarrow & x = \frac{\langle P_{i,j}, P_i\rangle}{r_{ij}r_i} \Longleftrightarrow x = \frac{q_i^2q_j - q_ip_ip_j + p_i^2q_j + p_iq_ip_j}{r_{ij}r_i} = \frac{q_jr_i^2}{r_{ij}r_i} \\ \Longleftrightarrow & x = \frac{q_jr_i}{r_{ij}} \\ \Longrightarrow & \mathfrak{v}'_2 = \mathfrak{v} + k\Delta v_{ij}\frac{q_jr_i}{r_{ij}}\omega(\theta_i) = \mathfrak{v} + k\Delta v q_j r_i\omega(\theta_i) = \mathfrak{v} + k\Delta v q_j P_i \in \mathfrak{V} \end{aligned}$$

and

$$\begin{aligned} x &= \left\langle \omega(\theta_i + \lambda_j), \omega^\perp(\theta_i)\right\rangle = \left\langle \omega(\theta_i + \lambda_j), \omega\left(\theta_i + \frac{\pi}{2}\right)\right\rangle \\ &= \left\langle \omega(\theta_i + \lambda_j), \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\omega(\theta_i)\right\rangle = \frac{\left\langle P_{ij}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} P_i\right\rangle}{r_{ij}r_i} \\ &= \frac{p_jr_i^2}{r_{ij}r_i} = \frac{p_jr_i}{r_{ij}} \end{aligned}$$

$$\Longrightarrow \mathfrak{w}_2' = \mathfrak{v} + k\Delta v_{ij}\frac{p_j r_i}{r_{ij}}\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}\omega(\theta_i) = \mathfrak{v} + k\Delta v p_j\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}P_i \in \mathfrak{V}\,.$$

As we see the resulting $\mathfrak{v}_2', \mathfrak{w}_2'$ lie on the grid, so $\mathfrak{w}_2$ is given by

$$\mathfrak{w}_2 = \mathfrak{v} + k\Delta v_{ij}\omega(\theta_i + \lambda_j) = \mathfrak{v} + k\Delta v P_{ij}\,.$$

Figure 3.3 illustrates the preceding argumentation. Here we can see that the Farey approximation can be interpreted as a rotation of the coordinate system around zero by the Farey angle θ_i. Then we conduct the integration on a line that is determined by an additional rotation around zero by the Farey angle λ_j. This results in the aforementioned step size of

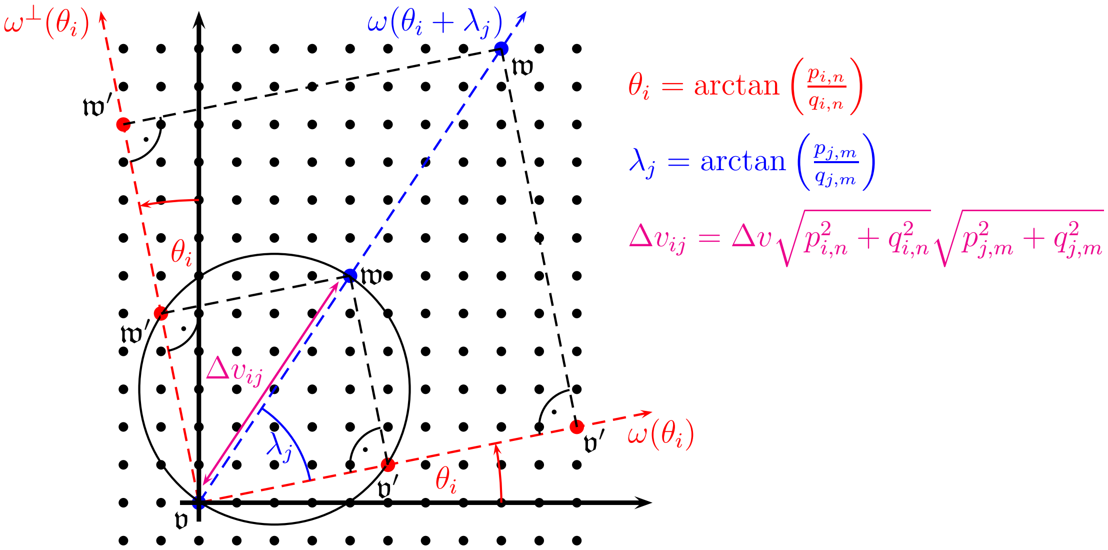

$\omega^{\perp}(\theta_i)$
$\omega(\theta_i + \lambda_j)$
$\mathfrak{w}$
$\mathfrak{w}'$
$\theta_i = \arctan\left(\frac{p_{i,n}}{q_{i,n}}\right)$
$\lambda_j = \arctan\left(\frac{p_{j,m}}{q_{j,m}}\right)$
$\Delta v_{ij} = \Delta v\sqrt{p_{i,n}^2 + q_{i,n}^2}\sqrt{p_{j,m}^2 + q_{j,m}^2}$
θ_i
$\mathfrak{w}$
$\mathfrak{w}'$
Δv_{ij}
λ_j
$\mathfrak{v}'$
$\omega(\theta_i)$
$\mathfrak{v}'$
θ_i
$\mathfrak{v}$

Figure 3.3: *Scheme for determining* $\mathfrak{w}, \mathfrak{v}', \mathfrak{w}'$ *and line of integration, in this example we have chosen* $p_i = 1, q_i = 5, p_j = 1, q_j = 1$

$$\Delta v_{ij} = \Delta v\sqrt{p_{i,n}^2 + q_{i,n}^2}\sqrt{p_{j,m}^2 + q_{j,m}^2}$$

for the innermost integration. Now we use a composite Newton-Cotes formula with an error order of r to approximate the innermost integral resulting in

$$\begin{aligned}|e_1(i,j)| &\le \lceil L / \Delta v_{ij}\rceil \Delta v_{ij} c \Delta v_{ij}^r \max_{l\in[0,L]}\left|\frac{\partial^r h(l,\lambda_j,\theta_i)}{\partial l^r}\right|\\ &< \frac{3}{2} Lc\Delta v_{ij}^r \max_{l\in[0,L]}\left|\frac{\partial^r h(l,\lambda_j,\theta_i)}{\partial l^r}\right|\\ &= \frac{3}{2} Lc(\Delta v)^r ({p_{i,n}}^2 + {q_{i,n}}^2)^{\frac{r}{2}}({p_{j,m}}^2 + {q_{j,m}}^2)^{\frac{r}{2}} \sup_{l\in[0,L]}\left|\frac{\partial^r h(l,\lambda_j,\theta_i)}{\partial l^r}\right|\\ \Longrightarrow |e_1| &\le \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_{i,n}\alpha_{j,m}e_1(i,j)\end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_{i,n}\alpha_{j,m} c\frac{3}{2}L(\Delta v)^r (p_{i,n}^2+q_{i,n}^2)^{\frac{r}{2}}(p_{j,m}^2+q_{j,m}^2)^{\frac{r}{2}} \sup_{l\in[0,L]}\left|\frac{\partial^r h(l,\lambda_j,\theta_i)}{\partial l^r}\right| \\
&\le \frac{3}{2}cL(\Delta v)^r \overbrace{\sup_{\substack{l\in[0,L]\\ \theta,\lambda\in\left[0,\frac{\pi}{4}\right]}}\left|\frac{\partial^r h(l,\lambda,\theta)}{\partial l^r}\right|}^{=:K_l} \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_{i,n}\alpha_{j,m}(p_{i,n}^2+q_{i,n}^2)^{\frac{r}{2}}(p_{j,m}^2+q_{j,m}^2)^{\frac{r}{2}} \\
&< \frac{3}{2}cL(\Delta v)^r K_l \sum_{i=1}^{N}\sum_{j=1}^{M} \frac{2}{nq_{i,n}}\frac{2}{mq_{j,m}}(p_{i,n}^2+q_{i,n}^2)^{\frac{r}{2}}(p_{j,m}^2+q_{j,m}^2)^{\frac{r}{2}} \qquad \text{by 3.1.4 (iii)} \\
&< \frac{3}{2}cL(\Delta v)^r K_l \frac{4}{mn}\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{(2q_{i,n}^2)^{\frac{r}{2}}}{q_{i,n}}\frac{(2q_{j,m}^2)^{\frac{r}{2}}}{q_{j,m}} \\
&= \frac{6cL(\Delta v)^r}{nm}K_l 2^r \sum_{i=1}^{N} q_{i,n}^{r-1}\sum_{j=1}^{M} q_{j,m}^{r-1} \\
&\le \frac{3\cdot 2^{r+1}cL(\Delta v)^r}{nm}K_l \sum_{q=1}^{n}\sum_{p=1}^{q} q^{r-1}\sum_{Q=1}^{m}\sum_{P=1}^{Q} Q^{r-1} \\
&= \frac{3\cdot 2^{r+1}cL(\Delta v)^r}{nm}K_l \sum_{q=1}^{n} q^{r}\sum_{Q=1}^{m} Q^{r} \\
&< \frac{3\cdot 2^{r+1}cL(\Delta v)^r}{nm}K_l n^{r+1}m^{r+1} = 3\cdot 2^{r+1}cLK_l(\Delta v)^r n^r m^r\ .
\end{aligned}$$

For the second error we get

$$\begin{aligned}
|e_2(i)| &< \sup_{\lambda\in\left[0,\frac{\pi}{4}\right]}\left|\frac{\partial H_1(\lambda,\theta_i)}{\partial\lambda}\right|\frac{4\ln(m)+4}{m^2} \qquad \text{by 3.1.5} \\
\Longrightarrow |e_2| &< \sum_{j=0}^{N}\alpha_{i,n}\sup_{\lambda\in\left[0,\frac{\pi}{4}\right]}\left|\frac{\partial H_1(\lambda,\theta_i)}{\partial\lambda}\right|\frac{4\ln(m)+4}{m^2} \\
&\le \overbrace{\sup_{\theta,\lambda\in\left[0,\frac{\pi}{4}\right]}\left|\frac{\partial H_1(\lambda,\theta_i)}{\partial\lambda}\right|}^{=:K_\lambda}\sum_{i=0}^{N}\alpha_{i,n}\frac{4\ln(m)+4}{m^2} \\
&= \frac{\pi}{4}K_\lambda\frac{4\ln(m)+4}{m^2}\ . \qquad \text{by 3.1.4 (iv)}
\end{aligned}$$

And analogously the last error results in

$$|e_3| \overset{3.1.5}{<} \overbrace{\sup_{\theta\in\left[0,\frac{\pi}{4}\right]} \left|\frac{\partial H_2(\theta)}{\partial\theta}\right|}^{=:K_\theta} \frac{4\ln(n)+4}{n^2} = K_\theta \frac{4\ln(n)+4}{n^2} \ .$$

The sum of these three errors gives the claim of this theorem. The last thing we have to look at is the differentiability of h:

$$h(l,\lambda,\theta) = \left[f\left(\mathfrak{v}_2'\right)f\left(\mathfrak{w}_2'\right) - f\left(\mathfrak{v}\right)f\left(\mathfrak{w}_2\right)\right] lk\left(\mathfrak{v},\mathfrak{w}_2,\omega(\theta)\right) \ ,$$

we know that products and sums of total differentiable functions (f, k by assumption) are total differentiable. We also know that the composition of total differentiable functions is total differentiable. Knowing that

$$\mathfrak{v}_2'(l,\lambda,\theta) := \mathfrak{v} + \langle l\omega(\theta+\lambda),\omega(\theta)\rangle\omega(\theta), \quad \mathfrak{w}_2'(l,\lambda,\theta) := \mathfrak{v} + \langle l\omega(\theta+\lambda),\omega^{\perp}(\theta)\rangle\omega^{\perp}(\theta) \ ,$$
$$\mathfrak{w}_2(l,\lambda,\theta) := \mathfrak{v} + l\omega(\theta+\lambda) \ ,$$

are elements of $C^\infty([0,\infty)\times[0,2\pi]^2 \to \mathbb{R}^3)$ and using the assumptions on f we arrive at

$$\left[f\left(\mathfrak{v}_2'\right)f\left(\mathfrak{w}_2'\right) - f\left(\mathfrak{v}\right)f\left(\mathfrak{w}_2\right)\right] \in \mathbb{C}^r\left((0,\infty)\times(0,2\pi)^2 \to \mathbb{R}\right) \ .$$

We also know that the first r total derivatives of k exist. Plugging in the r times total differentiable function

$$\Phi(l,\lambda,\theta) := \begin{pmatrix}\mathfrak{v}\\ \mathfrak{w}_2\\ \omega\end{pmatrix}(l,\lambda,\theta) = \begin{pmatrix}\mathfrak{v}\\ \mathfrak{w}_2(l,\lambda,\theta)\\ \omega(\theta)\end{pmatrix} = \begin{pmatrix}\mathfrak{v}\\ \mathfrak{v}+l\omega(\theta+\lambda)\\ \omega(\theta)\end{pmatrix}$$
$$\in \mathbb{C}^r\left((0,\infty)\times(0,2\pi)^2 \to \mathbb{R}^3\times\mathbb{R}^3\times S^2\right)$$

results into $k(\Phi(l,\lambda,\theta)) = k(\mathfrak{v},\mathfrak{w}_2,\omega(\theta)) \in \mathbb{C}^r\left((0,\infty)\times(0,2\pi)^2 \to \mathbb{R}\right)$. This leads to the conclusion that the product $h(l,\lambda,\theta)$ also lies in $\mathbb{C}^r\left((0,\infty)\times(0,2\pi)^2 \to \mathbb{R}\right)$. The last thing we want to remark is that k normally only depends on $|\mathfrak{v}-\mathfrak{w}|$ and θ, so it only depends on two scalars. In the above argumentation we treat k as a black box and do not use this structure. Nevertheless the results hold true for k being r times differentiable in these two scalar variables. □

Now we do a detailed derivation of the associated convergence order in terms of Δv for the above discretization. Similar considerations were done in [MS00, PH99]. Our result gives a convergence order up to one, which is in agreement with [MS00, PH99].

Theorem 3.2.4 (Convergence order in terms of Δv)
Assuming that $\Delta v \in \mathbb{R}^+, r \in \mathbb{N}$ are given constants satisfying

$$\frac{L}{2\Delta v \tilde{n}(\Delta v, r)\tilde{m}(\Delta v, r)} > r\,,$$

with

$$\begin{aligned}
\tilde{n}(\Delta v, r) &= \left(\frac{W\left(-e^{-2r-2}(2r+2)(\Delta v)^r 3 \cdot 2^{r-1} cL \left(\frac{\pi}{4}\right)^{\frac{r}{2}}\right)}{-(2r+2)(\Delta v)^r 3 \cdot 2^{r-1} cL \left(\frac{\pi}{4}\right)^{\frac{r}{2}}} \right)^{\frac{1}{2r+2}}, \\
\tilde{m}(\Delta v, r) &= \sqrt{\frac{\pi}{4}} n(\Delta v, r)\,,
\end{aligned}$$

and choosing n, m according to $n = \lceil \tilde{n}(\Delta v, r) \rceil,\ m = \lceil \tilde{m}(\Delta v, r) \rceil$, the convergence order of the Farey discretization 3.2.3 is

$$\forall \delta > 0 : |I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})| < e(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{r}{r+1}-\delta}\right).$$

Here $W(\bullet)$ is the minus first branch W_{-1} or W_{m} of the Lambert W function, see [OLBC10, 4.13]. We have to remark that $W(x)$ for $x \to -0$ asymptotically grows like the logarithm, implying that

$$\forall \varepsilon > 0 : n, m \in \mathcal{O}\left((\Delta v)^{-\frac{r}{2r+2}+\varepsilon}\right) .$$

So the main conclusion is that if n, m grow sufficiently slow with $\frac{1}{\Delta v}$ and in comparison to $\frac{1}{\Delta v}$ than the whole approximation converges with the order given above. The first requirement ensures that at least r points lie on every line that is used for the innermost integration, a condition that is necessary for the application of a Newton-Cotes formula of order r. The minimal number of points $\tilde{r}$ on any given line associated with the approximation of the innermost integral grows asymptotically as is given by

$$\forall \varepsilon > 0 : \tilde{r}(\Delta v) \in \mathcal{O}\left((\Delta v)^{-\frac{1}{r+1}+2\varepsilon}\right) .$$

Proof:
Let $K \geq \max\{K_l, K_\lambda, K_\theta\}$, $n > 2$ and $c_l := 3 \cdot 2^{r+1} cL, c_\lambda := \pi, c_\theta := 4$. Then the error formula

$$\left|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})\right| < 4K_\theta \frac{\ln(n)+1}{n^2} + \pi K_\lambda \frac{\ln(m)+1}{m^2} + 3 \cdot 2^{r+2} cLK_l (\Delta v)^r n^r m^r\ ,$$

holds true even when K_l, K_λ, K_θ get substituted by K. This leads to the ansatz

$$Kc_\theta \frac{\ln(n)+1}{n^2} = Kc_\lambda \frac{\ln(m)+1}{m^2} = Kc_l (\Delta v)^r n^r m^r\ , \tag{3.2.1}$$

this ansatz forces the three integrations to deliver approximately the same error, only disturbed by the differences between K_λ, K_θ, K_l. The most simple ansatz would be to do the following upward estimation

$$\begin{aligned}
&\frac{\ln(n)+1}{n^2} < \frac{1}{n} \wedge \frac{\ln(m)+1}{m^2} < \frac{1}{m} \\
\Longrightarrow &\left|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})\right| < Kc_\theta \frac{1}{n} + Kc_\lambda \frac{1}{m} + Kc_l (\Delta v)^r n^r m^r \,, \\
&Kc_\theta \frac{1}{n} = Kc_\lambda \frac{1}{m} = Kc_l (\Delta v)^r n^r m^r \Longrightarrow n = \text{const} \cdot m, n \in \mathcal{O}((\Delta v)^{-\frac{r}{2r+1}}) \\
\Longrightarrow &\left|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})\right| \in \mathcal{O}((\Delta v)^{\frac{r}{2r+1}}) \approx \mathcal{O}((\Delta v)^{\frac{1}{2}}) \,,
\end{aligned}$$

which essentially leads to a convergence order around $\frac{1}{2}$. But we want to do it in a more precise way where the correct handling of the logarithm results into some complicated expressions involving the Lambert W function. So let us start, at first we derive m in terms of n.

$$\begin{aligned}
&c_\theta \frac{\ln(n)+1}{n^2} = c_\lambda \frac{\ln(m)+1}{m^2} && \Longleftrightarrow \ln(m) = \frac{c_\theta}{c_\lambda} \frac{\ln(n)+1}{n^2} m^2 - 1 \\
\Longleftrightarrow &\frac{1}{2} \ln(m^2) = \frac{c_\theta}{c_\lambda} \frac{\ln(n)+1}{n^2} m^2 - 1 && \Longleftrightarrow m^2 = e^{\frac{2c_\theta}{c_\lambda} \frac{\ln(n)+1}{n^2} m^2 - 2} \\
\Longleftrightarrow &1 = \frac{m^2}{e^{\frac{2c_\theta}{c_\lambda} \frac{\ln(n)+1}{n^2} m^2 - 2}} && \Longleftrightarrow (*)
\end{aligned}$$

At this point we want to use the Lambert W function, for this we introduce the abbreviation $a := \frac{2c_\theta}{c_\lambda} \frac{\ln(n)+1}{n^2}$ and the substitution $\tilde{m} = m^2$ leading to

$$\begin{aligned}
(*) \Longleftrightarrow &\ 1 = \tilde{m} \cdot e^{-a\tilde{m}+2} = \tilde{m} e^2 \cdot e^{-a\tilde{m}} \\
\Longleftrightarrow &\ -ae^{-2} = -ae^{-2} \tilde{m} e^2 e^{-a\tilde{m}} = -a\tilde{m} e^{-a\tilde{m}} \\
\Longleftrightarrow &\ W(-ae^{-2}) = -a\tilde{m} && \Longleftrightarrow m = \sqrt{-\frac{W(-ae^{-2})}{a}} \,.
\end{aligned}$$

From (3.2.1) follows that m has to grow proportionally to n. This together with the demand that m should be real and the fact that $\forall n \in \mathbb{N} : ae^{-2} \in \left(0, \frac{1}{e}\right)$ leads to the conclusion that we have to choose the branch W_{-1} of the Lambert W function. We will plainly call it W, because we do not need other branches. Now we search for an upper bound and approximation of $m(n)$. For this we investigate the monotonicity and limit of $-\frac{W(-a(n)e^{-2})}{\frac{2c_\theta}{c_\lambda}(\ln(n)+1)}$.

(i) Monotonicity: The derivative is given by

$$-\frac{\frac{1}{-a(n)e^{-2}} \frac{W(-a(n)e^{-2})}{1+W(-a(n)e^{-2})} \cdot (-a'(n)e^{-2})(\ln(n)+1) + \frac{1}{n(\ln(n)+1)^2} W(-a(n)e^{-2})}{(\ln(n)+1)^2}$$

$$= -\frac{\frac{a'(n)}{a(n)}\frac{W(-a(n)e^{-2})}{1+W(-a(n)e^{-2})}(\ln(n)+1)+\frac{W(-a(n)e^{-2})}{n(\ln(n)+1)^2}}{(\ln(n)+1)^2}\,.$$

We know

- $c_\theta > 0, \quad c_\lambda > 0, \quad 2\frac{c_\theta}{c_\lambda} < e\,,$
- $a : \mathbb{N} \to \left(0, 2\frac{c_\theta}{c_\lambda}\right), n \mapsto \frac{2c_\theta}{c_\lambda}\frac{\ln(n)+1}{n^2}\,,$
- $\forall b \in \left(-\frac{1}{e}, 0\right) : W(b) < -1, \quad \lim\limits_{b\to -0} W(b) = -\infty\,,$

and W is strict monotonically decreasing in $\left(-\frac{1}{e}, 0\right)$. This can be used to get

- $-a(n)e^{-2} \in \left(-\frac{2c_\theta}{c_\lambda e^2}, 0\right) \subset \left(-\frac{1}{e}, 0\right) \implies W(-a(n)e^{-2}) < -1\,,$
- $a(n) > 0, \quad \frac{\partial a(n)}{\partial n} = -2c_\theta \frac{2\ln(n)+1}{c_\lambda n^3} < 0\,.$

This implies the positivity of the derivative:

$$-\frac{\overbrace{\underbrace{\frac{a'(n)}{a(n)}}_{}\!\!\!\overset{<0}{}\frac{W(-a(n)e^{-2})}{1+W(-a(n)e^{-2})}}^{<0}(\ln(n)+1)+\overbrace{\frac{W(-a(n)e^{-2})}{n(\ln(n)+1)^2}}^{<0}}{(\ln(n)+1)^2} > 0\,.$$

(ii) Limit: The limit can be calculated by using L'Hôpital:

$$\begin{aligned}
&\lim_{n\to\infty}\frac{c_\lambda}{2c_\theta}\frac{-W\left(\overbrace{-2\frac{c_\theta}{c_\lambda}\frac{\ln(n)+1}{n^2}e^{-2}}^{-b(n):=}\right)}{\ln(n)+1}\\
&= \lim_{n\to\infty}\frac{c_\lambda}{2c_\theta}\frac{\frac{-W(-b(n))}{1+W(-b(n))}\frac{1}{-b(n)}\left(-2\frac{c_\theta}{c_\lambda}e^{-2}\left(-\frac{2\ln(n)+1}{n^3}\right)\right)}{\frac{1}{n}}\\
&= \lim_{n\to\infty}\frac{W(-b(n))}{1+W(-b(n))}\frac{e^{-2}}{b(n)}\left(\frac{2\ln(n)+1}{n^2}\right)\\
&= \lim_{n\to\infty}\frac{W(-b(n))}{1+W(-b(n))}\frac{e^{-2}}{2\frac{c_\theta}{c_\lambda}\frac{\ln(n)+1}{n^2}e^{-2}}\left(\frac{2\ln(n)+1}{n^2}\right)\\
&= \lim_{n\to\infty}\frac{W(-b(n))}{1+W(-b(n))}\frac{n^2c_\lambda}{2c_\theta(\ln(n)+1)}\left(\frac{2\ln(n)+1}{n^2}\right)\\
&= \frac{c_\lambda}{c_\theta}\lim_{n\to\infty}\frac{W(-b(n))}{1+W(-b(n))} = \frac{c_\lambda}{c_\theta}\lim_{n\to\infty}\frac{W(-a(n)e^{-2})}{1+W(-a(n)e^{-2})} = \frac{c_\lambda}{c_\theta}\lim_{z\to -0}\frac{W(z)}{1+W(z)}\\
&= \frac{c_\lambda}{c_\theta}\,.
\end{aligned}$$

The upper bound (as well as an approximation) of $m(n)$ is now given by

$$\begin{aligned} m(n) &= \sqrt{-\frac{W(-a(n)e^{-2})}{a(n)}} = \sqrt{-\frac{W(-a(n)e^{-2})}{\frac{2c_\theta}{c_\lambda}\frac{\ln(n)+1}{n^2}}} = \sqrt{-\frac{W(-a(n)e^{-2})}{\frac{2c_\theta}{c_\lambda}(\ln(n)+1)}}n \\ &< \sqrt{\frac{c_\lambda}{c_\theta}}n, \text{ with } \forall\varepsilon > 0 \exists n_0 \in \mathbb{N} \forall n > n_0 : \sqrt{\frac{c_\lambda}{c_\theta}}n - \varepsilon \leq m(n) \end{aligned} \tag{3.2.2}$$

and we can use it to derive $n(\Delta v)$. By using $m(n)$ in (3.2.1) we get

$$c_\theta \frac{\ln(n)+1}{n^2} = c_l(\Delta v)^r n^r m^r < c_l(\Delta v)^r n^r \left(\sqrt{\frac{c_\lambda}{c_\theta}}n\right)^r ,$$

and now we can calculate a $n(\Delta v)$ that condemns the above inequality to be true:

$$\begin{aligned} & c_\theta \frac{\ln(n)+1}{n^2} = c_l(\Delta v)^r n^r \left(\sqrt{\frac{c_\lambda}{c_\theta}}n\right)^r = c_l(\Delta v)^r n^{2r} \left(\frac{c_\lambda}{c_\theta}\right)^{r/2} \\ \iff & \ln(n) = \frac{c_l}{c_\theta}(\Delta v)^r n^{2r+2} \left(\frac{c_\lambda}{c_\theta}\right)^{r/2} - 1 \\ \iff & \frac{1}{2r+2} \ln\left(n^{2r+2}\right) = \frac{c_l}{c_\theta}(\Delta v)^r n^{2r+2} \left(\frac{c_\lambda}{c_\theta}\right)^{r/2} - 1 \\ \iff & \ln\left(n^{2r+2}\right) = \overbrace{(2r+2)\frac{c_l}{c_\theta}(\Delta v)^r n^{2r+2} \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}}^{d(n,\Delta v):=} - 2r - 2 \\ \iff & n^{2r+2} = e^{d(n,\Delta v)-2r-2} \iff n^{2r+2}e^{-d(n,\Delta v)+2r+2} = 1 \\ \iff & -(2r+2)\frac{c_l}{c_\theta}(\Delta v)^r n^{2r+2} \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}} e^{-d(n,\Delta v)} = -e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}} \\ \iff & -d(n,\Delta v)e^{-d(n,\Delta v)} = -e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}} \\ \iff & W\left(-e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}\right) = -d(n,\Delta v) \\ \iff & n^{2r+2} = \frac{W\left(-e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}\right)}{-(2r+2)\frac{c_l}{c_\theta}(\Delta v)^r \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}} \\ \iff & n(\Delta v) = \left(\frac{W\left(-e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}\right)}{-(2r+2)\frac{c_l}{c_\theta}(\Delta v)^r \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}}\right)^{\frac{1}{2r+2}} . \end{aligned}$$

Now that we know sufficient criteria for $m(n)$ and $n(\Delta v)$ we start to derive the error in terms of Δv. For that we have to get upper and lower boundaries of $n(\Delta v)$ in terms of Δv. To get a lower bound let's start again at (3.2.1):

$$
\begin{aligned}
c_\theta \frac{\ln(n)+1}{n^2} &= c_l(\Delta v)^r n^r m^r < c_l(\Delta v)^r n^r \left(\sqrt{\frac{c_\lambda}{c_\theta}}n\right)^r = c_l(\Delta v)^r n^{2r} \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}} \\
\Longleftrightarrow & \frac{c_\theta}{n^2} < c_l(\Delta v)^r n^{2r}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}} \Longleftrightarrow n^{2r+2} > \frac{c_\theta}{c_l}\left(\frac{c_\theta}{c_\lambda}\right)^{\frac{r}{2}} (\Delta v)^{-r} \\
\Longleftrightarrow & n > \left(\frac{c_\theta}{c_l}\left(\frac{c_\theta}{c_\lambda}\right)^{\frac{r}{2}}\right)^{\frac{1}{2r+2}} \cdot (\Delta v)^{-\frac{r}{2r+2}} \quad \Longrightarrow \quad n \in \Omega\left((\Delta v)^{-\frac{r}{2r+2}}\right) \\
\Longrightarrow & \frac{1}{n^2} \in \mathcal{O}\left((\Delta v)^{\frac{2r}{2r+2}}\right) . \qquad (3.2.3)
\end{aligned}
$$

For the upper bound we use (3.2.1), (3.2.2) and we choose an arbitrary $\varepsilon \in \mathbb{R}^+$ with $\varepsilon < \sqrt{\frac{c_\lambda}{c_\theta}}$. This leads to

$$
\begin{aligned}
\exists n_0 \in \mathbb{N} \forall n > n_0 : c_\theta \frac{\ln(n)+1}{n^2} &> c_l(\Delta v)^r n^r \left(\sqrt{\frac{c_\lambda}{c_\theta}}n - \varepsilon\right)^r \\
&= c_l(\Delta v)^r n^{2r} \left(\sqrt{\frac{c_\lambda}{c_\theta}} - \frac{\varepsilon}{n}\right)^r > c_l(\Delta v)^r n^{2r} \left(\sqrt{\frac{c_\lambda}{c_\theta}} - \varepsilon\right)^r \\
\Longleftrightarrow \exists n_0 \in \mathbb{N} \forall n > n_0 : \frac{n^{2r+2}}{\ln(n)+1} &< \frac{c_\theta}{c_l}\left(\sqrt{\frac{c_\lambda}{c_\theta}} - \varepsilon\right)^{-r} (\Delta v)^{-r} \\
\Longleftrightarrow \forall \delta > 0 \exists n_0 \in \mathbb{N} \forall n > n_0 : n^{2r+2-\delta} &< \frac{c_\theta}{c_l}\left(\sqrt{\frac{c_\lambda}{c_\theta}} - \varepsilon\right)^{-r} (\Delta v)^{-r} \\
\Longleftrightarrow \forall \delta > 0 \exists n_0 \in \mathbb{N} \forall n > n_0 : n &< \left(\frac{c_\theta}{c_l}\right)^{\frac{1}{2r+2-\delta}} \left(\sqrt{\frac{c_\lambda}{c_\theta}} - \varepsilon\right)^{-\frac{r}{2r+2-\delta}} (\Delta v)^{-\frac{r}{2r+2-\delta}} \\
\Longrightarrow n \in \mathcal{O}\left((\Delta v)^{-\frac{r}{2r+2-\delta}}\right) \quad &\Longrightarrow \quad \ln(n) \in \mathcal{O}\left(\ln(\Delta v)\right) \qquad (3.2.4)
\end{aligned}
$$

Now we come to the convergence order. By choosing n, m according to

$$
n = \left\lceil \left(\frac{W\left(-e^{-2r-2}(2r+2)(\Delta v)^r \frac{c_l}{c_\theta}\left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}\right)}{-(2r+2)\frac{c_l}{c_\theta}(\Delta v)^r \left(\frac{c_\lambda}{c_\theta}\right)^{\frac{r}{2}}} \right)^{\frac{1}{2r+2}} \right\rceil , \qquad m = \left\lceil \sqrt{\frac{c_\lambda}{c_\theta}}n \right\rceil \qquad (3.2.5)
$$

we can "simplify" the error bound:

$$
|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})| < K\left(c_\theta \frac{\ln(n)+1}{n^2} + c_\lambda \frac{\ln(m)+1}{m^2} + c_l(\Delta v)^r n^r m^r\right)
$$

$$= K\left(c_\theta\frac{\ln(n)+1}{n^2} + c_\lambda\frac{\ln\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)+1}{\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)^2} + c_l(\Delta v)^r n^r\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)^r\right) .$$

The usage of (3.2.3) and (3.2.4) implies:

$$\underbrace{c_\theta(\ln(n)+1)}_{\in\mathcal{O}(\ln(\Delta v))}\cdot \underbrace{\frac{1}{n^2}}_{\in\mathcal{O}\left((\Delta v)^{\frac{2r}{2r+2}}\right)} + \underbrace{c_\lambda\frac{\ln\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)+1}{\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)^2}}_{\in\mathcal{O}\left(\ln(\Delta v)(\Delta v)^{\frac{2r}{2r+2}}\right)} + \underbrace{c_l(\Delta v)^r n^r\left(\left\lceil\sqrt{\frac{c_\lambda}{c_\theta}}n\right\rceil\right)^r}_{\in\mathcal{O}\left((\Delta v)^r(\Delta v)^{-\frac{2r^2}{2r+2-\delta}}\right)},$$

where the last term can be simplified

$$\begin{aligned}(\Delta v)^r(\Delta v)^{-\frac{2r^2}{2r+2-\delta}} &= (\Delta v)^{-\frac{2r^2}{2r+2-\delta}+r} = (\Delta v)^{\frac{2r^2+2r-\delta r-2r^2}{2r+2-\delta}} = (\Delta v)^{\frac{2r-\delta r}{2r+2-\delta}}\\ &= (\Delta v)^{\frac{2r}{2r+2-\delta}-\tilde{\delta}} \in \mathcal{O}\left((\Delta v)^{\frac{2r}{2r+2-\delta}-\tilde{\delta}}\right) \subset \mathcal{O}\left((\Delta v)^{\frac{2r}{2r+2}-\tilde{\delta}}\right) ,\end{aligned}$$

and reveals a final result:

$$|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})| < e(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{2r}{2r+2}-\delta}\right) .$$

The last question that remains is: how many points are at least on any line for the innermost integration, because this determines what the largest r (order of the Newton-Cotes formula) can be. We know that the number of points corresponds to

$$\forall i,j : \frac{L}{\Delta v_{ij}} = \frac{L}{\Delta v\sqrt{(p_i^2+q_i^2)(p_j^2+q_j^2)}} > \frac{L}{2\Delta v n m} .$$

We also know that the number of points necessary to achieve an convergence order of r in the Newton-Cotes formulas is given by $\hat{r} \leq r$ (2 points for the trapezoid method, 3 points for the Simpson's rule etc.). So

$$\frac{L}{2\Delta v n m} \geq r \geq \hat{r}$$

has to be true for the chosen $\Delta v, r$. To verify this we can put $n(\Delta v, r)$ and $m(\Delta v, r)$ from (3.2.5) into this inequality. Unfortunately the equations become so complex that we surrender the task to calculate this to numerical algorithms on computers. Nevertheless we can derive the asymptotic behavior of the minimal number of points on any given line associated with the innermost integral by taking a closer look at

$$\tilde{r} = \frac{L}{2\Delta v n m} .$$

We know that $W(x)$ grows as the logarithm for $x \to -0$:

$$\lim_{x\to -0} \frac{W(x)}{\ln(x)}, \text{ using L'Hôpital gives}$$
$$= \lim_{x\to -0} \frac{\frac{W(x)}{x(1+W(x))}}{\frac{1}{x}} = \lim_{x\to -0} \frac{W(x)}{1+W(x)} = 1\,,$$

resulting into

$$\forall \varepsilon > 0 : n \in \mathcal{O}\left((\Delta v)^{-\frac{r}{2r+2}+\varepsilon}\right), \quad m \in \mathcal{O}\left((\Delta v)^{-\frac{r}{2r+2}+\varepsilon}\right)\,.$$

Using this knowledge for the calculation of the asymptotic behavior of the minimal number of points on lines results into

$$\forall \varepsilon > 0 : \tilde{r} = \frac{L}{2\Delta v n m} \in \mathcal{O}\left((\Delta v)^{\frac{-2r-2+2r}{2r+2}+2\varepsilon}\right) = \mathcal{O}\left((\Delta v)^{-\frac{1}{r+1}+2\varepsilon}\right)\,. \qquad \square$$

Corollary 3.2.5 (Completion of the approximation)
Using the same assumptions as in 3.2.3 the full Boltzmann collision operator

$$I[f](\mathfrak{v}) = \int_0^{2\pi}\int_0^{2\pi}\int_0^L h(l,\lambda,\theta)\,\mathrm{d}l\,\mathrm{d}\lambda\,\mathrm{d}\theta$$

can be approximated by the discretization

$$\hat{I}[f](\mathfrak{v}) = \sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{M}\alpha_{j,m}\Delta v_{ij}\sum_{k=0}^{\lfloor L/\Delta v_{ij}\rfloor}\sum_{\alpha,\beta\in A} g(k)\left(h(l_k,\beta\pm\lambda_j,\alpha\pm\theta_i)\right)$$

with $A := \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$. This approximation yields an upper error bound of

$$\left|I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})\right| < 64\left(4\hat{K}_\theta\frac{\ln(n)+1}{n^2} + \pi\hat{K}_\lambda\frac{\ln(m)+1}{m^2} + 2^{r+2}3cL\hat{K}_l(\Delta v)^r n^r m^r\right)\,,$$

where $\hat{K}_l, \hat{K}_\lambda, \hat{K}_\theta$ are some constants depending only on f. By choosing n, m according to theorem 3.2.4 we get the same convergence order for the above Discretization:

$$\forall \delta > 0 : |I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v})| < e(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{r}{r+1}-\delta}\right)\,.$$

Proof:
At first we have to take a look at an approximation of a spherical integral using the Farey approximation, that needs to take more than $\left(0, \frac{\pi}{4}\right)$ into consideration. One could think that the following approach does the trick:

$$\int_0^{\frac{\pi}{2}} H(\theta)\mathrm{d}\theta = \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta + \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} H(\theta)\mathrm{d}\theta = \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta + \int_0^{\frac{\pi}{4}} H\left(\theta+\frac{\pi}{4}\right)\mathrm{d}\theta$$

$$\approx \sum_{i=1}^{N} \alpha_{i,n} H(\theta_i) + \sum_{i=1}^{N} \alpha_{i,n} H\left(\theta_i + \frac{\pi}{4}\right) = \sum_{i=1}^{N} \alpha_{i,n} \left(H(\theta_i) + H\left(\theta_i + \frac{\pi}{4}\right)\right) .$$

But this approach has the nasty feature of increasing the necessary number of points in the velocity space (in one direction) by a factor up to 2. Or in other words: this approach creates step sizes for the innermost integration that are twice as long as necessary.

To give an example: the approximation using a Farey sequence of order 3 results in $F_{4,3} = \frac{p_4}{q_4} = \frac{2}{3}$ being the largest gradient of a line of integration for the innermost integration, hitting the first grid point in $(3,2)^T$. The given approach results in the usage of the Farey angle $\arctan\left(\frac{2}{3}\right) + \frac{\pi}{4}$. Now we can simply calculate the gradient of the corresponding line of integration by $\tan\left(\arctan\left(\frac{2}{3}\right) + \frac{\pi}{4}\right) = 5$ and we realize that we have to have at least 5 points in the direction of the ordinate to get one point for the innermost integration. This would lead to an unnecessary degradation of the efficiency and error rate. A more sophisticated approach is to transform the second integral in such a way that we can use the “inverse” Farey sequence

$$\left\{ \frac{1}{F_{N,n}} = \tan\left(\frac{\pi}{2} - \arctan(F_{N,n})\right), \ldots, \frac{1}{F_{1,n}} = \tan\left(\frac{\pi}{2} - \arctan(F_{N,1})\right) \right\} ,$$

for the approximation. This can be achieved by backward integration:

$$\begin{aligned} \int_0^{\frac{\pi}{2}} H(\theta)\mathrm{d}\theta &= \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta + \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} H(\theta)\mathrm{d}\theta = \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta + \int_{-\frac{\pi}{4}}^{0} H\left(\theta + \frac{\pi}{2}\right)\mathrm{d}\theta \\ &= \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta - \int_0^{-\frac{\pi}{4}} H\left(\theta + \frac{\pi}{2}\right)\mathrm{d}\theta = \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta + \int_0^{\frac{\pi}{4}} H\left(\frac{\pi}{2} - \theta\right)\mathrm{d}\theta \\ &\approx \sum_{i=1}^{N} \alpha_{i,n} H(\theta_i) + \sum_{i=1}^{N} \alpha_{i,n} H\left(\frac{\pi}{2} - \theta_i\right) = \sum_{i=1}^{N} \alpha_{i,n} \left(H(\theta_i) + H\left(\frac{\pi}{2} - \theta_i\right)\right) . \end{aligned}$$

The successive application of this approach to all four quadrants for the outer and middle integral results in the approximation of 64 integrals and gives the approximation formula

$$\hat{I}[f](\mathfrak{v}) = \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\alpha,\beta \in A} g(k) \left(h(l_k, \beta \pm \lambda_j, \alpha \pm \theta_i)\right)$$

with $A := \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$. Due to the fact that this corresponds to 64 Farey approximations we get a 64 times larger bound for the error:

$$\left| I[f](\mathfrak{v}) - \hat{I}[f](\mathfrak{v}) \right| < 64 \left(4\hat{K}_\theta \frac{\ln(n) + 1}{n^2} + \pi \hat{K}_\lambda \frac{\ln(m) + 1}{m^2} + 2^{r+2} 3cL\hat{K}_l (\Delta v)^r n^r m^r \right) ,$$

with

$$\hat{K}_l := \sup_{\substack{l \in [0,L] \\ \theta,\lambda \in [0,2\pi]}} \left| \frac{\partial^r h(l,\lambda,\theta)}{\partial l^r} \right| , \qquad \hat{K}_\lambda := \sup_{\theta,\lambda \in [0,2\pi]} \left| \frac{\partial H_1(\lambda,\theta)}{\partial \lambda} \right| ,$$

$$\hat{K}_\theta := \sup_{\theta \in [0,2\pi]} \left| \frac{\partial H_2(\theta)}{\partial \theta} \right| .$$

The approximation of every single integral has an order of convergence corresponding to 3.2.4, and consequently the sum of these 64 approximations has the same order of convergence. □

Remark 3.2.6
It would be more precise to define

$$A := \left\{ \alpha : x \mapsto a + x \middle| a \in \left\{ 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2} \right\} \right\} \cup \left\{ \alpha : x \mapsto a - x \middle| a \in \left\{ 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2} \right\} \right\}$$

and to use this to write

$$\hat{I}[f](\mathfrak{v}) = \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\alpha,\beta \in A} g(k) h(l_k, \beta(\lambda_j), \alpha(\theta_i)) ,$$

but in the following proofs and argumentations we will use different descriptions of these operators as functions (like above), as angles and as elements of the automorphism group (next corollary) interchangeably. So we give the hint to remember the more precise definition above if some argumentation results into confusion.

Corollary 3.2.7 (Completion of the approximation using the automorphism group)
Due to the grid symmetries being represented through the operators in the automorphism group G the above approximation can be rewritten using this group:

$$\begin{aligned} \hat{I}[f](\mathfrak{v}) &= \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\alpha,\beta \in A} g(k) \left(h(l_k, \beta \pm \lambda_j, \alpha \pm \theta_i) \right) \\ &= \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{M} \alpha_{j,m} \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\varphi_\alpha,\varphi_\beta \in G} g(k) \left(\tilde{h} \left(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta \right) \right) . \end{aligned}$$

Here we use a reinterpretation of h through rotations:

$$\begin{aligned} \tilde{h}(l,\lambda,\theta,\varphi_\alpha,\varphi_\beta) &:= \left[f\left(\mathfrak{v}_3'\right) f\left(\mathfrak{w}_3'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}_3\right) \right] lk\left(\mathfrak{v}, \mathfrak{w}_3, \varphi_\alpha \omega(\theta)\right) , \\ \mathfrak{w}_3(l,\lambda,\theta,\varphi_\alpha,\varphi_\beta) &:= \mathfrak{v} + l \varphi_\alpha R_\theta \varphi_\beta R_\lambda \mathfrak{x} , \\ \mathfrak{v}_3'(l,\lambda,\theta,\varphi_\alpha,\varphi_\beta) &:= \mathfrak{v} + l \left\langle \varphi_\alpha R_\theta \varphi_\beta R_\lambda \mathfrak{x}, \varphi_\alpha \omega(\theta) \right\rangle \varphi_\alpha \omega(\theta) , \\ \mathfrak{w}_3'(l,\lambda,\theta,\varphi_\alpha,\varphi_\beta) &:= \mathfrak{v} + l \langle \varphi_\alpha R_\theta \varphi_\beta R_\lambda \mathfrak{x}, R_{\frac{\pi}{2}} \varphi_\alpha \omega(\theta) \rangle R_{\frac{\pi}{2}} \varphi_\alpha \omega(\theta) , \\ R_\bullet &:= \begin{pmatrix} \cos(\bullet) & -\sin(\bullet) \\ \sin(\bullet) & \cos(\bullet) \end{pmatrix} , \ \mathfrak{x} := \begin{pmatrix} 1 \\ 0 \end{pmatrix} . \end{aligned}$$

Proof:
We start with the reinterpretation of the velocities $\mathfrak{w}, \mathfrak{v}', \mathfrak{w}'$ through rotations. For this we only look at the vectors $\overrightarrow{\mathfrak{v}\mathfrak{w}}, \overrightarrow{\mathfrak{v}\mathfrak{v}'}$ and $\overrightarrow{\mathfrak{v}\mathfrak{w}'}$. As explained in 3.2.1, 3.2.3 and figure 3.2, we can understand $\overrightarrow{\mathfrak{v}\mathfrak{v}'}$ as a rotation of the x-axis $\mathfrak{x} = (1,0)^T$ by an angle of θ. Now the operators φ_α map $\overrightarrow{\mathfrak{v}\mathfrak{v}'}$ into one of the 8 quadrant halves. Then an additional rotation by λ results into $\overrightarrow{\mathfrak{v}\mathfrak{w}}$ which gets also mapped into one of the 8 quadrant halves (of the system rotated by θ) by φ_β. This leads to the diagram

$$\mathfrak{x} \xrightarrow{R_\theta} \mathfrak{x}_1 \xrightarrow{\varphi_\alpha} \mathfrak{x}_2 = \varphi_\alpha\omega(\theta) = \frac{\overrightarrow{\mathfrak{v}\mathfrak{v}_3'}}{\|\overrightarrow{\mathfrak{v}\mathfrak{v}_3'}\|} \xrightarrow{\varphi_\alpha R_\theta R_\lambda(\varphi_\alpha R_\theta)^{-1}} \mathfrak{x}_3 \xrightarrow{\varphi_\alpha R_\theta \varphi_\beta(\varphi_\alpha R_\theta)^{-1}} \mathfrak{x}_4$$

$$\begin{aligned}\Longrightarrow \mathfrak{x}_4 &= \varphi_\alpha R_\theta \varphi_\beta (\varphi_\alpha R_\theta)^{-1} \varphi_\alpha R_\theta R_\lambda (\varphi_\alpha R_\theta)^{-1} \varphi_\alpha R_\theta \mathfrak{x} \\ &= \varphi_\alpha R_\theta \varphi_\beta R_\lambda \mathfrak{x} = \frac{\overrightarrow{\mathfrak{v}\mathfrak{w}_3}}{\|\overrightarrow{\mathfrak{v}\mathfrak{w}_3}\|} \,.\end{aligned}$$

Now $\overrightarrow{\mathfrak{v}\mathfrak{w}'}$ is simply given by a 90° rotation of $\overrightarrow{\mathfrak{v}\mathfrak{v}'}$.
Using this knowledge and the resulting reinterpretation through the usage of a rotation matrix R as well as the geometric interpretation of the automorphism group the proof of the above remark becomes easy, so let us take a look at the main idea. The automorphism group for a two dimensional uniform grid consists of 8 elements

$$\begin{aligned}
&\varphi_1 := \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \quad &&\varphi_2 := \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \quad &&\varphi_3 := \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}, \quad &&\varphi_4 := \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\\
&\varphi_5 := \begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}, \quad &&\varphi_6 := \begin{pmatrix}-1 & 0\\ 0 & 1\end{pmatrix}, \quad &&\varphi_7 := \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}, \quad &&\varphi_8 := \begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix}.
\end{aligned}$$

Let $A := \left(0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right) = (\alpha_1, \ldots, \alpha_4)$, $\alpha \in A$, $\mathfrak{x} := (1,0)^T$, then the following holds true:

$$\begin{aligned}
\omega(\alpha_k + \theta_i) &= R_{\alpha_k}\omega(\theta_i) &&= R_{\alpha_k} R_{\theta_i}\mathfrak{x} &&= \varphi_{2k-1} \cdot R_{\theta_i}\mathfrak{x} \,,\\
\omega(\alpha_k - \theta_i) &= R_{\alpha_k}\omega(-\theta_i) &&= R_{\alpha_k} R_{-\theta_i}\mathfrak{x} &&= \varphi_{2k} \cdot R_{\theta_i}\mathfrak{x} \,.
\end{aligned} \tag{3.2.6}$$

Now we can write the sum $\sum_{\alpha\in A} h(\bullet, \bullet, \alpha \pm \theta_i)$ out and begin to identify the angles in A with the corresponding rotation or reflection in G through the reinterpretation of h:

$$\begin{aligned}
\hat{h}(l, \lambda, \omega(\theta)) &:= \left[f\left(\hat{\mathfrak{v}}_3'\right) f\left(\hat{\mathfrak{w}}_3'\right) - f\left(\mathfrak{v}\right) f\left(\hat{\mathfrak{w}}_3\right)\right] lk\left(\mathfrak{v}, \hat{\mathfrak{w}}_3, \omega(\theta)\right) = h(l, \lambda, \theta)\,,\\
\hat{\mathfrak{w}}_3(l, \lambda, \omega(\theta)) &:= \mathfrak{v} + l R_\lambda \omega(\theta),\\
\hat{\mathfrak{v}}_3'(l, \lambda, \omega(\theta)) &:= \mathfrak{v} + l \left\langle R_\lambda\omega(\theta), R_\theta \mathfrak{x}\right\rangle \omega(\theta),\\
\hat{\mathfrak{w}}_3'(l, \lambda, \omega(\theta)) &:= \mathfrak{v} + l \left\langle R_\lambda\omega(\theta), R_{\frac{\pi}{2}}\omega(\theta)\right\rangle R_{\frac{\pi}{2}}\omega(\theta),
\end{aligned}$$

$$\begin{aligned}
\sum_{\alpha\in A} h(\bullet, \bullet, \alpha \pm \theta) &= \sum_{\alpha\in A} \hat{h}(\bullet, \bullet, \omega(\alpha \pm \theta)) = \sum_{\alpha\in A} \hat{h}(\bullet, \bullet, R_\alpha\omega(\pm\theta))\\
&= \hat{h}(\overbrace{R_0}^{\varphi_1=}\omega(\theta)) + \hat{h}(\overbrace{R_0\omega(-\theta)}^{\varphi_2(\omega(\theta))=}) + \hat{h}(\overbrace{R_{\frac{\pi}{2}}\omega(\theta)}^{\varphi_3(\omega(\theta))=}) + \hat{h}(\overbrace{R_{\frac{\pi}{2}}\omega(-\theta)}^{\varphi_4(\omega(\theta))=})
\end{aligned}$$

$$
\begin{aligned}
&+ \hat{h}(\underbrace{R_\pi \omega(\theta)}_{=\varphi_5(\omega(\theta))}) + \hat{h}(\underbrace{R_\pi \omega(-\theta)}_{=\varphi_6(\omega(\theta))}) + \hat{h}(\underbrace{R_{\frac{3\pi}{2}} \omega(\theta)}_{=\varphi_7(\omega(\theta))}) + \hat{h}(\underbrace{R_{\frac{3\pi}{2}} \omega(-\theta)}_{=\varphi_8(\omega(\theta))}) \\
&= \sum_{\varphi_\alpha \in G} \tilde{h}(\bullet, \bullet, \theta, \varphi_\alpha, \text{identity}) .
\end{aligned}
$$

Analog considerations for λ give the final result. In the end we see that the automorphism group can be used to map a point in the first half of the first quadrant into every half of every quadrant in such a way that it seems natural to use this group for the completion of the Farey approximation. □

Lemma 3.2.8 (Simplification)
The last corollary can be used to obtain a simplification of the discretization:

$$
\begin{aligned}
\mathfrak{w}_3(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + k\Delta v \left(\varphi_\alpha \left[P_i, P_i^\perp\right]\right) \varphi_\beta P_j \,, \\
\mathfrak{v}_3'(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + k\Delta v (\varphi_\beta P_j)_1 \varphi_\alpha P_i \,, \\
\mathfrak{w}_3'(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + \mathfrak{w}_3 - \mathfrak{v}_3' \,, \\
P_i := \begin{pmatrix} q_i \\ p_i \end{pmatrix}, &\quad P_i^\perp := \begin{pmatrix} -p_i \\ q_i \end{pmatrix} .
\end{aligned}
$$

Proof:
We know from 3.2.7 and 3.2.3 that

$$
\begin{aligned}
\mathfrak{w}_3(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &:= \mathfrak{v} + l_k \varphi_\alpha R_{\theta_i} \varphi_\beta R_{\lambda_j} \mathfrak{x} \,, \\
\mathfrak{v}_3'(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &:= \mathfrak{v} + l_k \left\langle \varphi_\alpha R_{\theta_i} \varphi_\beta R_{\lambda_j} \mathfrak{x}, \varphi_\alpha \omega(\theta_i) \right\rangle \varphi_\alpha \omega(\theta_i) \,, \\
\mathfrak{w}_3'(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &:= \mathfrak{v} + l_k \left\langle \varphi_\alpha R_{\theta_i} \varphi_\beta R_{\lambda_j} \mathfrak{x}, R_{\frac{\pi}{2}} \varphi_\alpha \omega(\theta_i) \right\rangle R_{\frac{\pi}{2}} \varphi_\alpha \omega(\theta_i) \,, \\
P_i, P_i^\perp \text{ as above and } P_{ij} &:= \begin{pmatrix} q_i q_j - p_i p_j \\ p_i q_j + p_j q_i \end{pmatrix}, r_i := \|P_i\|, \; r_{ij} = \|P_{ij}\| \,,
\end{aligned}
$$

and with the knowledge $R_{\theta_i} = [\omega(\theta_i), \omega^\perp(\theta_i)]$ we obtain

$$
\begin{aligned}
\mathfrak{w}_3 &= \mathfrak{v} + l_k \varphi_\alpha R_{\theta_i} \varphi_\beta R_{\lambda_j} \mathfrak{x} = \mathfrak{v} + k\Delta v r_i r_j \varphi_\alpha [\omega(\theta_i), \omega^\perp(\theta_i)] \varphi_\beta \omega(\lambda_j) \\
&= \mathfrak{v} + k\Delta v \left(\varphi_\alpha \left[P_i, P_i^\perp\right]\right) \varphi_\beta P_j \,, \\
\mathfrak{v}_3' &= \mathfrak{v} + l_k \left\langle \varphi_\alpha R_{\theta_i} \varphi_\beta R_{\lambda_j} \mathfrak{x}, \varphi_\alpha \omega(\theta_i) \right\rangle \varphi_\alpha \omega(\theta_i) \\
&= \mathfrak{v} + k\Delta v r_i \underbrace{\left\langle \left(\varphi_\alpha \left[\omega(\theta_i), \omega^\perp(\theta_i)\right]\right) \varphi_\beta P_j, \varphi_\alpha \omega(\theta_i) \right\rangle} \varphi_\alpha \omega(\theta_i) \\
&= \mathfrak{v} + k\Delta v \qquad\qquad\qquad (\varphi_\beta P_j)_1 \qquad\qquad\qquad \varphi_\alpha P_i \,, \\
\mathfrak{w}_3' &= \mathfrak{v} + \mathfrak{w}_3 - \mathfrak{v}_3' \,.
\end{aligned}
$$

□

Lemma 3.2.9 (Discretization as a DVM and its properties)
Assuming that

$$k(\mathfrak{v}_i, \mathfrak{w}_2, \omega(\theta)) = \tilde{k}(\|\mathfrak{v}_i - \mathfrak{w}_2\|, \angle(\omega(\theta), R_\lambda \omega(\theta))) = \tilde{k}(l, \lambda)\,,$$

possesses the same symmetries as the grid (which means $\tilde{k}(l,\lambda) = \tilde{k}(l, \beta \pm \lambda)$) and by using 3.2.5 or 3.2.7 for the calculation of the Operator $A_{\bullet,\bullet}^{\bullet,\bullet}$ we obtain

$$\begin{aligned}
I_{\mathrm{DVM}}[f](\mathfrak{v}_i) &= \sum_{x=1}^{N} \alpha_{x,n} \sum_{j=1}^{M} \alpha_{j,m} L_{xj} \sum_{k=0}^{\lfloor L/\Delta v_{xj} \rfloor} \sum_{\alpha,\beta \in A} g(k) h(l_k, \beta \pm \lambda_j, \alpha \pm \theta_x) \\
&= \sum_{x=1}^{N} \alpha_{x,n} \sum_{j=1}^{M} \alpha_{j,m} L_{xj} \sum_{k=0}^{\lfloor L/\Delta v_{xj} \rfloor} \sum_{\varphi_\alpha,\varphi_\beta \in G} g(k) \tilde{h}\left(l_k, \lambda_j, \theta_x, \varphi_\alpha, \varphi_\beta\right) \\
&= \sum_{j,k,l} A_{i,j}^{k,l} \left(f(\mathfrak{v}_k) f(\mathfrak{v}_l) - f(\mathfrak{v}_i) f(\mathfrak{v}_j)\right)\,,
\end{aligned}$$

with

$$\begin{aligned}
A_{i,j}^{k,l} &= \begin{cases} \begin{matrix} l_{abc}\alpha_{a,n}s_{abc} \\ \cdot\alpha_{b,m}L_{ab}g(c) \\ \cdot k(a,b,c) \end{matrix}, & \text{if} \begin{matrix} \exists a,b \in \overline{N}, c \in N_{ab}, \alpha,\beta \in A : \\ (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')(a,b,c,\alpha,\beta) \end{matrix} \\ 0, & \text{else} \end{cases}, \\
&= \begin{cases} \begin{matrix} l_{abc}\alpha_{a,n}s_{abc} \\ \cdot\alpha_{b,m}L_{ab}g(c) \\ \cdot k(a,b,c) \end{matrix}, & \text{if} \begin{matrix} \exists a,b \in \overline{N}, c \in N_{ab}, \varphi_\alpha,\varphi_\beta \in G : \\ (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_3, \mathfrak{v}_3', \mathfrak{w}_3')(a,b,c,\varphi_\alpha,\varphi_\beta) \end{matrix} \\ 0, & \text{else} \end{cases},
\end{aligned}$$

$$\begin{aligned}
&\overline{N} := \{1,\dots,N\} \times \{1,\dots,M\}, \qquad N_{ab} := \{0,\dots,\lfloor L/\Delta v_{ab} \rfloor\}\,, \\
&s_{abc} := \left|\{(\alpha,\beta)\,|(\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')(a,b,c,\alpha,\beta)\}\right| \\
&\mathfrak{w}_2 = \mathfrak{w}_2(l_c, \beta \pm \lambda_b, \alpha \pm \theta_a),\ \mathfrak{v}_2' = \mathfrak{v}_2'(l_c, \beta \pm \lambda_b, \alpha \pm \theta_a),\ \mathfrak{w}_2' = \mathfrak{w}_2'(l_c, \beta \pm \lambda_b, \alpha \pm \theta_a)\,, \\
&\mathfrak{w}_3 = \mathfrak{w}_3(l_c, \lambda_b, \theta_a, \varphi_\alpha, \varphi_\beta),\ \mathfrak{v}_3' = \mathfrak{v}_3'(l_c, \lambda_b, \theta_a, \varphi_\alpha, \varphi_\beta),\ \mathfrak{w}_3' = \mathfrak{w}_3'(l_c, \lambda_b, \theta_a, \varphi_\alpha, \varphi_\beta)\,, \\
&\mathfrak{w}_3, \mathfrak{v}_3', \mathfrak{w}_3' \text{ as in 3.2.8,} \quad \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2' \text{ as in 3.2.1, assuming } \mathfrak{v} = \mathfrak{v}_i\,, \\
&\quad \Delta v_{ab},\ g(c),\ L_{ab} \text{ as in 3.2.3, } l_{abc} = c\Delta v_{ab} \qquad k(a,b,c) = k(\mathfrak{v}_i, \mathfrak{w}_2, \omega(\theta_a))\,.
\end{aligned}$$

This DVM fulfills the minimal requirements 2.1.2.4 and possesses no artificial collision invariants on normal grids.

Proof:
We prove the transformation by using the explicit form of $A_{i,j}^{k,l}$ and the knowledge, that the sum $\sum_{j,k,l}$ goes through all possible point combinations $\mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l$. This can

be used for the following calculation:

$$\begin{aligned}
&\sum_{j,k,l} A^{k,l}_{i,j}\left(f(\mathfrak{v}_i)f(\mathfrak{v}_j) - f(\mathfrak{v}_k)f(\mathfrak{v}_l)\right)\\
&= \sum_{(a,b,c)\in\overline{N}\times N_{ab}} \sum_{\varphi_\alpha,\varphi_\beta\in G} l_{abc}\alpha_{a,n}\alpha_{b,m}L_{ab}g(c)k(a,b,c)\left(f(\mathfrak{v}'_3)f(\mathfrak{w}'_3) - f(\mathfrak{v}_i)f(\mathfrak{w}_3)\right)\\
&= \sum_{(a,b,c)\in\overline{N}\times N_{ab}} \sum_{\varphi_\alpha,\varphi_\beta\in G} \alpha_{a,n}\alpha_{b,m}L_{ab}g(c)\tilde{h}(l_{abc},\lambda_b,\theta_a,\varphi_\alpha,\varphi_\beta)\,.
\end{aligned}$$

The transformation using the angles α instead of the operators φ is analog to the above. Now we show that $A^{k,l}_{i,j}$ possesses the symmetry properties

$$A^{k,l}_{i,j} = A^{i,j}_{k,l} = A^{l,k}_{j,i}\,,$$

and that there exists a one to one correspondence between the nonzero elements $A^{k,l}_{i,j}$ and the points used by the discretization, except for some special points for which s_{ijkl} takes care of (later we will take a closer look at s_{ijkl}). This finally justifies the above calculation. Now we need some additional assumptions about the structure of k:

$$k(\mathfrak{v}_i,\mathfrak{w}_2,\omega(\theta_a)) = \tilde{k}(\|\mathfrak{v}_i-\mathfrak{w}_2\|,\angle(\mathfrak{v}_i-\mathfrak{w}_2,\omega(\theta_a))) = \hat{k}(l_{abc},\lambda_b)\,,$$

and we assume that k possesses at least the grid symmetries - making it independent of $\beta\pm$ or φ_β. Due to the definition of $A^{k,l}_{i,j}$ and the additional knowledge about the structure of k we see, that A only depends on a,b,c. So we can restate the task of proving the above symmetry by proving that a,b,c is independent of the ordering of $\mathfrak{v}_i,\mathfrak{v}_j,\mathfrak{v}_k,\mathfrak{v}_l$ resp. $\mathfrak{v}_i,\mathfrak{w}_{2/3},\mathfrak{v}'_{2/3},\mathfrak{w}'_{2/3}$. This is the case iff order changes in these four velocities only lead to changes of $\varphi_\alpha,\varphi_\beta$ and not in a,b,c. So let us prove this by calculating through the necessary permutations. We start with some results given by the reinterpretation in 3.2.7 and 3.2.8. Here the main one is given by the fact that a reflection of $\overrightarrow{\mathfrak{v}\mathfrak{w}}$ on $\omega(\theta)$ gives $\overrightarrow{\mathfrak{w}'\mathfrak{v}'}$, see figure 3.2. This reflection is given by $\varphi_\alpha R_\theta\varphi_\gamma(\varphi_\alpha R_\theta)^{-1}$ with $\varphi_\gamma := \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ and results into:

$$\begin{aligned}
\overrightarrow{\mathfrak{w}'_3\mathfrak{v}'_3}(\varphi_\alpha,\varphi_\beta) &= \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(\varphi_\alpha,\varphi_\gamma\varphi_\beta) = k\Delta v(\varphi_\alpha[P_i,P_i^\perp])\varphi_\gamma\varphi_\beta P_j = \overrightarrow{\mathfrak{w}'_3\mathfrak{v}'_3}(\varphi_\alpha,\varphi_\gamma^2\varphi_\beta)\,,\\
\overrightarrow{\mathfrak{v}\mathfrak{w}_3}(\varphi_\alpha,\varphi_\beta) &= -\overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha,\varphi_\beta) = -\overrightarrow{\mathfrak{v}\mathfrak{w}_3}(\varphi_\alpha,-\varphi_\beta) = \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha,-\varphi_\beta)\,,\\
\overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha,\varphi_\beta) &= k\Delta v(\varphi_\beta P_j)_1\varphi_\alpha P_i = -\overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha,\varphi_\beta) = -\overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha,-\varphi_\beta)\\
&= \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha,-\varphi_\beta) = \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha,\varphi_\gamma\varphi_\beta)\,,\\
&= \overrightarrow{\mathfrak{w}'_3\mathfrak{w}_3} \Longrightarrow \overrightarrow{\mathfrak{w}'_3\mathfrak{v}'_3} = \overrightarrow{\mathfrak{w}'_3\mathfrak{w}_3} - \overrightarrow{\mathfrak{v}\mathfrak{w}_3} + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} = 2\overrightarrow{\mathfrak{v}\mathfrak{v}'_3} - \overrightarrow{\mathfrak{v}\mathfrak{w}_3}\,.
\end{aligned}$$

Now we prove that the different orderings of $\mathfrak{v}_i,\mathfrak{v}_j,\mathfrak{v}_k,\mathfrak{v}_l$ can be realized by using different elements of the automorphism group $\varphi_\alpha,\varphi_\beta$ resulting into the same coefficients

$A_{i,j}^{k,l}$ for the permutations of these velocities (because A does not depend on $\varphi_\alpha, \varphi_\beta$). So let us calculate through the necessary permutations:

$$\begin{pmatrix} \mathfrak{v}_i \\ \mathfrak{v}_j \\ \mathfrak{v}_k \\ \mathfrak{v}_l \end{pmatrix} = \begin{pmatrix} \mathfrak{v}_i \\ \mathfrak{v}_i + \overrightarrow{\mathfrak{v}\mathfrak{w}_3} \\ \mathfrak{v}_i + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \\ \mathfrak{v}_i + \overrightarrow{\mathfrak{v}\mathfrak{w}_3} - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \end{pmatrix}$$

$$\Longrightarrow \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_l \\ \mathfrak{v}_i \\ \mathfrak{v}_j \end{pmatrix} = \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} + \overrightarrow{\mathfrak{v}\mathfrak{w}_3} - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} + \overrightarrow{\mathfrak{v}\mathfrak{w}_3} \end{pmatrix}$$

$$= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - 2\overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_k - 2\overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(\varphi_\alpha, \varphi_\beta) + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) \end{pmatrix}$$

$$= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{w}'_3\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{w}'_3\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(\varphi_\alpha, \varphi_\beta) \end{pmatrix}$$

$$= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha, \varphi_\gamma\varphi_\beta) \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha, \varphi_\gamma\varphi_\beta) \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha, \varphi_\gamma\varphi_\beta) - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha, \varphi_\gamma\varphi_\beta) \end{pmatrix}$$

$$\Longrightarrow \begin{pmatrix} \mathfrak{v}_j \\ \mathfrak{v}_i \\ \mathfrak{v}_l \\ \mathfrak{v}_k \end{pmatrix} = \begin{pmatrix} \mathfrak{v}_j \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_3} \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_3} + \overrightarrow{\mathfrak{v}\mathfrak{w}_3} - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_3} + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3} \end{pmatrix}$$

$$= \begin{pmatrix} \mathfrak{v}_j \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha, \varphi_\beta) \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{w}_3}(-\varphi_\alpha, \varphi_\beta) - \overrightarrow{\mathfrak{v}\mathfrak{v}'_3}(-\varphi_\alpha, \varphi_\beta) \end{pmatrix} .$$

Now we have to think about the one to one correspondence between the nonzero elements of $A_{i,j}^{k,l}$ and the points used by the discretization. For this we take a look at $\mathfrak{w}_2, \mathfrak{v}'_2$:

$$\mathfrak{w}_2 = \mathfrak{v} + l_c\omega(\theta_a + \lambda_b), \quad \mathfrak{v}'_2 = \mathfrak{v} + l_c\langle\omega(\theta_a + \lambda_b), \omega(\theta_a)\rangle\omega(\theta_a) \,,$$

here we see that (by construction - 3.2.3) we have the property

$$\forall(a,b,c) \neq (\tilde{a},\tilde{b},\tilde{c}) : (\mathfrak{w}_2, \mathfrak{v}_2')(a,b,c) \neq (\mathfrak{w}_2, \mathfrak{v}_2')(\tilde{a},\tilde{b},\tilde{c})\,.$$

So we see that we have at most one (a,b,c) corresponding to a (i,j,k,l) (mapping from (a,b,c) to (i,j,k,l) is injective). We can use a similar argumentation for the reflections and rotations $\alpha\pm, \beta\pm$:

$$\mathfrak{w}_2 = \mathfrak{v} + l_c\omega(\alpha \pm \theta_a + \beta \pm \lambda_b), \quad \mathfrak{v}_2' = \mathfrak{v} + l_c\langle\omega(\alpha \pm \theta_a + \beta \pm \lambda_b), \omega(\alpha \pm \theta_a)\rangle\omega(\alpha \pm \theta_a)\,,$$

here one can realize that $\alpha\pm, \beta\pm$ correspond to 90° rotations and reflections around symmetry axes of the grid which have the property of being mappings between the 8 regions

$$S_i := \left\{ r\begin{pmatrix}\cos(a)\\ \sin(a)\end{pmatrix} \middle| \, r \in \mathbb{R}_0^+, a \in \left[\frac{i\pi}{4}, \frac{(i+1)\pi}{4}\right] \right\},\ i = 0, \ldots, 7\,.$$

These mappings correspond to the 8 elements of the automorphism group $\varphi \in G$ (see 3.2.7). Using the alternative representation $\mathfrak{w}_3, \mathfrak{v}_3', \mathfrak{w}_3'$ it becomes easy to see that for every φ (or $\alpha\pm$) $\mathfrak{v}_3' - \mathfrak{v} = \mathfrak{v}_2' - \mathfrak{v}$ gets mapped into another S_i whereas $\beta\pm$ only changes the length of $\mathfrak{v}_3' - \mathfrak{v}$ and that for every $\beta\pm$, $\mathfrak{w}_2 - \mathfrak{v}$ moves into another S_i. The last thing that can happen in this context is the case where $\theta \in \left\{0, \frac{\pi}{4}\right\} \vee \lambda = \frac{\pi}{4}$ (the case $\lambda = 0$ is irrelevant because then $\mathfrak{v} = \mathfrak{w}', \mathfrak{v}' = \mathfrak{w} \implies j = k, i = l \implies f_k f_l - f_i f_j = 0$). In such a case we have for example $\omega(\theta) = \omega(0) = (x,0)^T$ and a reflection on the x-axis would result into the same point $\omega(0) = \omega(-0)$. In such a case we need the additional argument that our automorphism group gets partitioned into 4 equivalence classes each having 2 elements. In the given example $\theta = 0$ these would correspond to the sets $0\pm := \left\{0+, 0-\right\}, \frac{\pi}{2}\pm, \pi\pm, \frac{3\pi}{2}\pm$, in the other possible situation $\theta = \frac{\pi}{4}$ it would be $\left\{0+, \frac{\pi}{2}-\right\}, \left\{\frac{\pi}{2}+, \pi-\right\}, \left\{\pi+, \frac{3\pi}{2}-\right\}, \left\{\frac{3\pi}{2}+, 0-\right\}$. So we realize that the case $\theta \in \left\{0, \frac{\pi}{4}\right\}, \lambda \neq \frac{\pi}{4}$ results into a two to one correspondence (two multi indexes (a,b,c,α,β) for one (i,j,k,l)) and that the case $\theta \in \left\{0, \frac{\pi}{4}\right\}, \lambda = \frac{\pi}{4}$ results into a four to one correspondence. Fortunately these two resp. four indexes only differ in α, β. So the same coefficient $l_{abc}\alpha_{a,n} \cdot \alpha_{b,m}L_{ab}g(c) \cdot k(a,b,c)$ corresponds to the multi indexes and we can simply merge the multiple occurrences of this coefficient into $A_{i,j}^{k,l}$ by multiplying $A_{i,j}^{k,l}$ with

$$s_{ijkl} := \left|\left\{(a,b,c,\alpha,\beta) \left| (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')(a,b,c,\alpha,\beta)\right.\right\}\right|\,,$$

giving the final form of $A_{i,j}^{k,l}$ as in the lemma. Due to the fact that the mapping from (a,b,c) to (i,j,k,l) is injective (when taking $\alpha\pm, \beta\pm$ into account) we can write

$$s_{ijkl} = s_{abc} := \left|\left\{(\alpha,\beta) \left| (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')(a,b,c,\alpha,\beta)\right.\right\}\right|\,.$$

This eliminates the necessity of looking at the symmetries of s_{ijkl}, because these symmetries are created by the α, β. One of the last things to look at is the non negativity

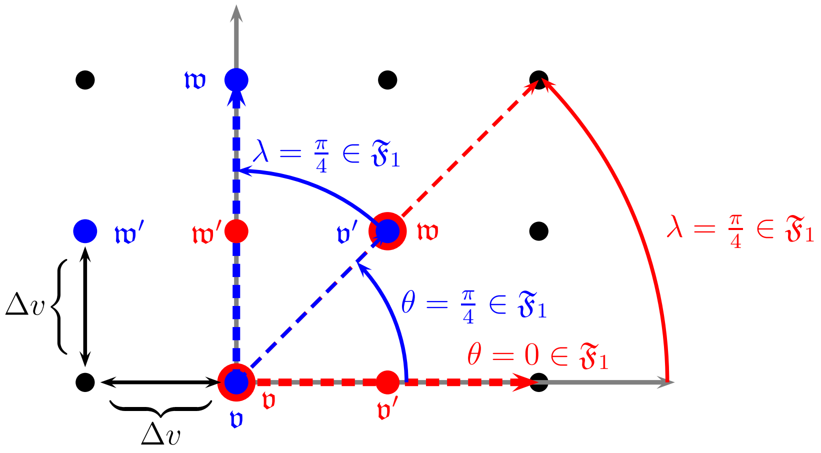

$\mathfrak{w}$
$\lambda = \frac{\pi}{4} \in \mathfrak{F}_1$
$\mathfrak{w}'$
$\mathfrak{w}'$
$\mathfrak{v}'$
$\mathfrak{w}$
$\lambda = \frac{\pi}{4} \in \mathfrak{F}_1$
Δv
$\theta = \frac{\pi}{4} \in \mathfrak{F}_1$
$\theta = 0 \in \mathfrak{F}_1$
$\mathfrak{v}$
$\mathfrak{v}$
$\mathfrak{v}'$
Δv

Figure 3.4: *Visualization of the fact that using $n \geq 1, m \geq 1$ results into having all squares with a diameter of $2\Delta v$, $\sqrt{2}\Delta v$*

of $A_{i,j}^{k,l}$. This is given by the fact that all factors $A_{i,j}^{k,l}$ consists of are non negative. Claiming that a Farey discretization with $n \geq 1, m \geq 1$ results into a set of collision pairs M that contains all squares in the velocity space with a diagonal of $2\Delta v$, $\sqrt{2}\Delta v$ together with remark 2.1.2.10(iii) implies that a uniform discretization of a origin centered sphere or a cube with at least 9 points possesses no artificial collision invariants. So the last thing to do is to justify this claim. For this we take a look at figure 3.4 and savor the self explaining picture. □

Remark 3.2.10

(i) The above DVM may not possess the symmetries $A_{i,j}^{k,l} = A_{j,i}^{k,l} = A_{i,j}^{l,k}$ as can be seen in the corresponding proof. Nonetheless these symmetries can be created by setting

$$\tilde{A}_{i,j}^{k,l} := \frac{A_{i,j}^{k,l} + A_{j,i}^{k,l} + A_{i,j}^{l,k}}{3},$$

and using $\tilde{A}_{i,j}^{k,l}$ instead of $A_{i,j}^{k,l}, A_{j,i}^{k,l}, A_{i,j}^{l,k}$. Taking a look at the actual calculation of this discretization one realizes that this adaptation does not change the discretization (the result remains exactly the same), except that it creates these symmetries.

(ii) Unfortunately it is not possible to transform this DVM into an eLGpM, because in general it does not possess the necessary property (2.2.1) that is needed for theorem 2.2.2 . This condition is violated as soon as the order of the Farey sequences n, m gets larger than one. Due to this we need an according adjustment of the discretization to make it compatible with the eLGpMs.

Lemma 3.2.11 (Farey approximation as an eLGpM)
An eLGpM based on the Farey approximation (using $\mathfrak{F}_n, \mathfrak{F}_{\mathfrak{m}}$) can be written as

$$I_{\mathrm{LGpM}}[f](\mathfrak{v}_i) = \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\mathfrak{v}]\in\tilde{S}^{\mathfrak{V}}_{i,\mathfrak{c}}} \sum_{[\varphi]\in\tilde{G}} \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} \left(\prod_{\varphi'\in[\varphi]} f(\mathfrak{c}+\varphi'(\mathfrak{v}-\mathfrak{c})) - \prod_{\varphi'\in H} f(\mathfrak{c}+\varphi'(\mathfrak{v}_i-\mathfrak{c})) \right)$$

with

$$\begin{aligned}
&\mathfrak{C} := \mathfrak{V}\cup\mathfrak{V}_{\frac{1}{2}},\ \mathfrak{v}_k(\mathfrak{c},\mathfrak{v},\varphi) := \mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}),\ \mathfrak{v}_j(\mathfrak{c},i) := 2\mathfrak{c}-\mathfrak{v}_i\,,\\
&\tilde{S}^{\mathfrak{V}}_{i,\mathfrak{c}} := S^{\mathfrak{V}}_{ij}/\sim_{\mathfrak{c}},\ \tilde{G} := G/\sim_H,\ \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} := \frac{2\overline{A^{k}_{i,j}}}{|\{\varphi'\in G|\mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}) = c+\varphi'(\mathfrak{v}-\mathfrak{c})\}|}\,,\\
&\overline{A^{k,l}_{i,j}} := A^{k,l}_{i,j}\cdot\overline{\Delta v_{ij}},\quad \overline{\Delta v_{ij}} := \begin{cases} 1, & \text{if } \frac{\overrightarrow{\mathfrak{v}_i\mathfrak{v}_j}}{2c}\in\mathfrak{C}\\ 2, & \text{else}\end{cases},\quad c = \max\left\{\tilde{c}\in\mathbb{N}\,\middle|\,\frac{\overrightarrow{\mathfrak{v}_i\mathfrak{v}_j}}{\tilde{c}}\in\mathfrak{V}\right\}
\end{aligned}$$

where $A^{k,l}_{i,j}$ comes from 3.2.9. The convergence order is the same as in corollary 3.2.5, the error boundary is increased by a factor of 2^r and this LGpM has no artificial collision invariants.

Proof:
It is not possible to transform the DVM 3.2.9 directly into an eLGpM. So we introduce a little adaption of the Farey approximation that results into a DVM that can be transformed into an eLGpM (see theorem 2.2.2). Beginning at the Farey approximation 3.2.3,

$$\begin{aligned}
\tilde{I}[f](\mathfrak{v}) &:= \sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{M}\alpha_{j,m}\Delta v_{ij}\sum_{k=0}^{\lfloor L/\Delta v_{ij}\rfloor} g(k)h(l_k,\lambda_j,\theta_i)\,,\\
h(l,\lambda,\theta) &:= \left[f\left(\mathfrak{v}'_2\right)f\left(\mathfrak{w}'_2\right) - f\left(\mathfrak{v}\right)f\left(\mathfrak{w}_2\right)\right] lk\left(\mathfrak{v},\mathfrak{w}_2,\omega(\theta)\right)\,,
\end{aligned}$$

we want to realize that we only use such $\mathfrak{w}$, which have the property that $\frac{\mathfrak{v}+\mathfrak{w}}{2}\in\mathfrak{V}\cup\mathfrak{V}_{\frac{1}{2}}$. This can be achieved by an adaption of the innermost integral approximation. By doubling the step size of the innermost integration for "problematic" integration lines we force the corresponding centers to lie on $\mathfrak{V}$. The only problem that remains is to identify these "problematic" lines. We know $\frac{\mathfrak{w}-\mathfrak{v}}{c}$ corresponds to the first point that is used by the innermost integration (c corresponds to the step number) in the direction $\overrightarrow{\mathfrak{v}\mathfrak{w}}$. Let us call a direction problematic iff the first step in this direction (of length Δv_{ij}) leads to a corresponding center that is not in $\mathfrak{C}$

$$\frac{1}{2}\frac{\overrightarrow{\mathfrak{v}\mathfrak{w}}}{c}\notin\mathfrak{C}\,.$$

Now we have to adjust the integration step size to avoid centers that are not in $\mathfrak{C}$ leading to a an adaption of the step size by

$$\overline{\Delta v_{ij}} := \begin{cases} 1, & \text{if } \frac{1}{2}\frac{\overrightarrow{\mathfrak{vw}}}{c} \in \mathfrak{C} \\ 2, & \text{else} \end{cases} .$$

Knowing this little tweak to tune the Farey approximation for use in the LGpM context we can simply use the DVM derived in 3.2.9 with $\overline{\Delta v_{ij}}$ (also resulting in $\overline{L}_{ij}, \overline{l}_k$) and transform it via 2.2.2 into an eLGpM. The doubled step size directly translates into a 2^r increase for the error in the innermost integral (see proof of 3.2.3) . But this is not as bad as it looks, because this directly results into a relaxation of the necessary order of the Farey sequences, as can be seen in equation (3.2.5) in the proof of theorem 3.2.4 (c_l becomes $2^{2r+2}cL$). Due to the fact that the error only changes by a constant the convergence order remains. The lack of artificial collision invariants comes from the intermediate step of transforming the Farey approximation in a DVM and 3.2.9. □

3.3 Three dimensions

At this point we repeat the procedure of the last subsection in three dimensions. To avoid the danger of being repetitive we reduce all necessary proofs to the point where they are understandable and mainly new ideas remain. For parts of the proof being analog to the 2 dimensional case we point to the corresponding region in the last subsection. In [MS00] the authors generalized the Farey sequence approach by [RS94] to three dimensions and obtained a convergence order of $\frac{6}{7}$. The approach in [MS00] is more sophisticated, more complex, involves more number theory and probably yields a superior convergence when compared against first order quadratures within our scheme. But it becomes significantly harder to apply any kind of regular quadrature and to partition the domain of integration into separate regions which eases the inspection and handling of grid and operator symmetries. Beside this we learned about the existence of this publication after we developed our own approach. The reason for this seems to be that [MS00] was published in French. We use another approach for three dimensions in order to obtain higher convergence orders by the application of quadrature formulas on specific symmetry regions.

Remark 3.3.1
In the last subsection we have seen that the velocity space gets divided into smallest "symmetry regions" $S_i, i = 1, \ldots, 8$ which can be mapped onto each other by the operators in the automorphism group. In 2 dimensions such a smallest region is given by one half of one of the quadrants, see figure 3.5. In 3 dimensions the space is divided into 48 of these regions, figure 3.5 shows all 6 of these regions lying in the positive

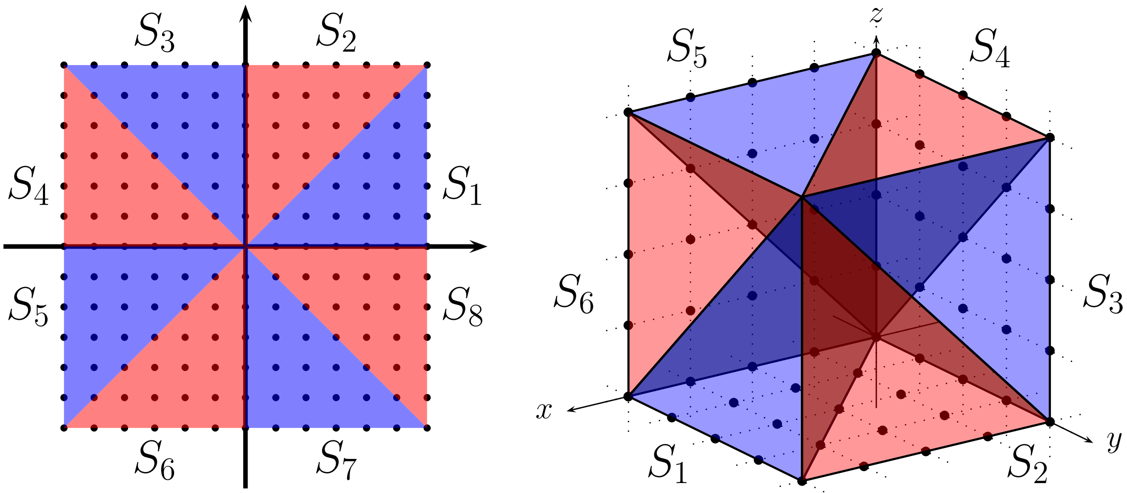


Figure 3.5: *Symmetry regions created by the automorphism group in two and three dimensions*

quadrant. In spherical coordinates these regions can be represented by

$$S_1 = \left\{ r \begin{pmatrix} \cos(\theta)\cos(\varphi) \\ \sin(\theta)\cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \middle| \, r \in \mathbb{R}_0^+, \theta \in \left[0, \frac{\pi}{4}\right], \varphi \in [0, \arctan(\sin(\theta))] \right\},$$

$$S_1 \cup S_6 = \left\{ r \begin{pmatrix} \cos(\theta)\cos(\varphi) \\ \sin(\theta)\cos(\varphi) \\ \sin(\varphi) \end{pmatrix} \middle| \, r \in \mathbb{R}_0^+, \theta \in \left[0, \frac{\pi}{4}\right], \varphi \in [0, \arctan(\cos(\theta))] \right\}.$$

The image of one such region S_i under the automorphism group results into a set of regions (the sets of inner points $\hat{S}_i$ of these regions are pairwise disjunct) equaling the complete space:

$$\forall i \in \{1, \dots, 48\} : \left(\forall \alpha \neq \beta \in G : \alpha(\hat{S}_i) \cap \beta(\hat{S}_i) = \emptyset\right) \wedge \left(\bigcup_{\alpha \in G} \alpha(S_i) = \mathbb{R}^n\right).$$

Now one of the main messages of the last subsection is that it is sufficient to discretize one of these smallest symmetry regions, because we can then apply the automorphism group (or simply the corresponding rotations and reflections) to obtain a discretization of the whole space. We use this insight to create a Farey-based discretization in 3 dimensions of one such symmetry region and then prove that this discretization can be extended to the whole space by applying the necessary coordinate transformations and taking a closer look at the occurring functional determinants. The desired side effect of this approach is that the resulting discretization possesses all grid related symmetries by design, because these symmetries are reflected by the automorphism group. This together with a goal oriented transformation of the collision integral finally leads to the situation that the minimal requirements 2.1.2.4 of a DVM are fulfilled, giving all the properties we want.

So let us start with the discretization of one symmetry region (S_1):

Proposition 3.3.2 (Farey approximation in spherical coordinates)
Let $H \in C^1\left(\left[0, \frac{\pi}{4}\right]^2 \to \mathbb{R}\right)$. Using the Farey sequence $\tilde{\mathfrak{F}}_n$, the approximation

$$\int_0^{\frac{\pi}{4}} \int_0^{\arctan(\sin(\theta))} H(\theta, \varphi) \mathrm{d}\varphi \mathrm{d}\theta \approx \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{N_i} \alpha_{i,j} H(\theta_i, \varphi_{i,j}) \,,$$

yields an upper error bound of

$$\begin{aligned} &\left| \int_0^{\frac{\pi}{4}} \int_0^{\arctan(\sin(\theta))} H(\theta, \varphi) \mathrm{d}\varphi \mathrm{d}\theta - \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{N_i} \alpha_{i,j} H(\theta_i, \varphi_{i,j}) \right| \\ &< 8K_{\theta,\varphi} \cdot \frac{1}{n} + 4K_{\theta} \cdot \frac{\ln(n) + 1}{n^2} \,, \end{aligned}$$

where $K_{\theta,\varphi}, K_\theta$ are some constants depending only on H and its derivative. The approximation uses the definitions

$$\begin{aligned} \tilde{\mathfrak{F}}_{i,n} &:= \left(\left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j} \right) \,\middle|\, \begin{array}{l} p_{i,j} \leq q_{i,j}, p \in \mathbb{N}_0, t \in \left\{1, \ldots, \left\lfloor \frac{n}{q_i} \right\rfloor \right\}, q = t \cdot q_i : \\ 0 \leq p \leq t \cdot p_i \ \wedge \ \gcd(p_{i,j}, q_{i,j}) = 1 \ \wedge \\ \left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j} \right) = \left(\frac{p}{q}, t \right) \ \wedge \ \forall j > 1 : \frac{p_{i,j-1}}{q_{i,j-1}} < \frac{p_{i,j}}{q_{i,j}} \\ \wedge \text{ when multiple } t \text{ s are possible choose the smallest one} \end{array} \right) \\ &= \left(\left(\frac{p_{i,1}}{q_{i,1}}, t_{i,1} \right), \ldots, \left(\frac{p_{i,N_i}}{q_{i,N_i}}, t_{i,N_i} \right) \right) = \left((F_{i,1}, t_{i,1}), \ldots, (F_{i,N_i}, t_{i,N_i}) \right), \quad N_i := |\tilde{\mathfrak{F}}_{i,n}| \,, \end{aligned}$$

$$\alpha_{i,j} := \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2}, \ \varphi_{i,j} = \arctan\left(F_{i,j} \frac{q_i}{\sqrt{q_i^2 + p_i^2}} \right), \ \varphi_{i,0} := \varphi_{i,1}, \ \varphi_{i,N_i+1} := \varphi_{i,N_i} \,.$$

At this point the reader should not try to understand the meaning of $\tilde{\mathfrak{F}}_{i,n}$. Geometrically this set corresponds to the angles $\varphi_{i,j}$ (which can be used for the approximation in the 3rd dimension) over the lines corresponding to the angles θ_i (which are responsible for the approximation in the x-y plane). The construction and explanation of this set can be found in the following proof.

Proof:
Our aim is to discretize an integral with the structure

$$\int_0^{\frac{\pi}{4}} \int_0^{\arctan(\sin(\theta))} H(\theta, \varphi) \mathrm{d}\varphi \mathrm{d}\theta$$

over the symmetry region S_1. We start with a look at the approximation for the outermost integral over θ using the Farey approximation in two dimensions, assuming that θ corresponds to the Cartesian coordinates in the x-y plane. For this we need to find out how we can extract the correct angles in spherical coordinates from a given point in $\mathbb{R}^3$. To avoid index confusion we define (only for this proof) that all objects containing only the index i correspond to $\tilde{\mathfrak{F}}_n$. To get the angles we use the following parametrization of a sphere

$$\omega(\theta,\varphi) := \begin{pmatrix} \cos(\varphi)\cos(\theta) \\ \cos(\varphi)\sin(\theta) \\ \sin(\varphi) \end{pmatrix} .$$

Now we calculate the Cartesian coordinates corresponding to the elements in the Farey sequence for the approximation of the integral over θ. That means the coordinates of the first point in which a line from the origin in the x-y plane with the gradient $\frac{p_i}{q_i}$ would cross a point on the uniform grid $\mathbb{Z}^3$. As described in 3.1.2 we can use $\frac{p_i}{q_i}$ to obtain the point in the $x-y$ plane by $\mathfrak{a} := \begin{pmatrix} q_{i,n} \\ p_{i,n} \\ 0 \end{pmatrix}$. Transforming this into spherical coordinates yields

$$r_{i,j} = \sqrt{q_i^2 + p_i^2}, \quad \theta_i = \arctan\left(\frac{p_i}{q_i}\right), \quad \varphi_{i,j} = \arcsin\left(0\right) .$$

Now we can define (assuming that the domain of integration for φ depends on θ)

$$H(\theta) := \int_0^{\arctan(\sin(\theta))} H(\varphi,\theta)\mathrm{d}\varphi \,,$$

and by 3.1.5 we instantly get

$$e_1 := \left| \sum_{i=1}^{N} \alpha_i H(\theta_i) - \int_0^{\frac{\pi}{4}} H(\theta)\mathrm{d}\theta \right| \leq 4 \sup_{\theta\in\left[0,\frac{\pi}{4}\right]} |H'(\theta)| \frac{\ln(n)+1}{n^2} .$$

Now we need to take a closer look at the expansion into 3 dimensions. For this we take a look at figure 3.6 (left), the solid lines correspond to the discretization in the x-y plane and the dashed lines to the usable gradients that correspond to a specific F_i. Here we see that we have the same situation (number of points and position) as in the x-y plane. Unfortunately this case is the best case scenario. In the worst case we can only use the points directly above the point $(q_i, p_i, 0)^T$, because there are no other grid points above the line associated with F_i, an example for this is given in figure 3.6 (right). With this knowledge we can collect all usable gradients corresponding to a specific F_i. At this point we have to consider the requirement that we want to discretize the region S_1 (see figure 3.5), resulting in the (possibly) arbitrarily looking borders and corresponding inequalities.

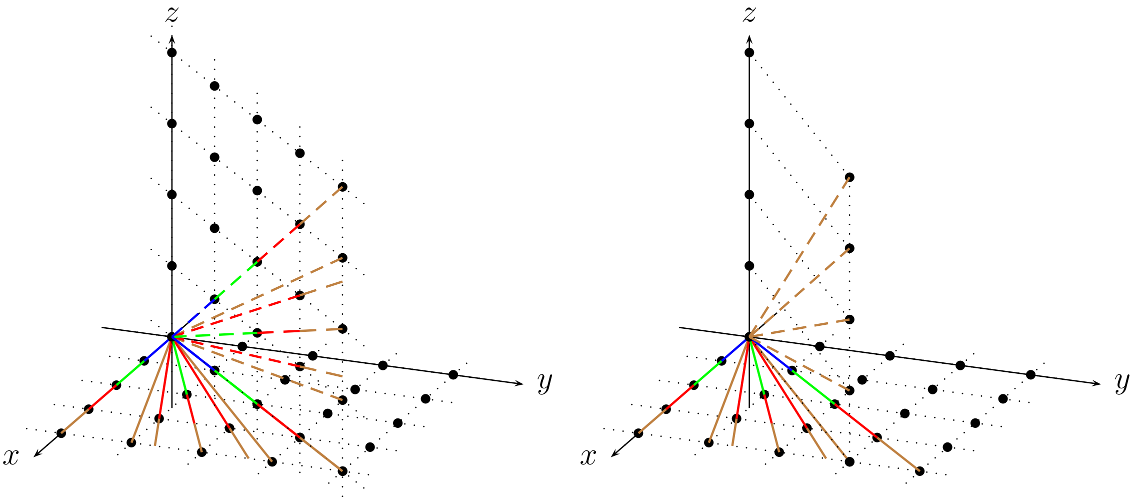


Figure 3.6: *Visualization of the Farey sequence in the x-y plane and a best / worst case expansion into 3 dimensions above F_N, F_{N-1}, compare figure 3.1*

Looking at one line in the x-y plane (corresponding to $F_i = \frac{p_i}{q_i}$) we can generally use all points above the last point on this line for our discretization (see 3.6, right). So let us assume that one such point is $(q_i, p_i, p_{i,j})^T$. Now $p_{i,j}$ equals the z-component of our discretization point. But we are generally interested in the angle φ corresponding to this point. To systematically obtain these angles we need to calculate the gradient of the line through

$$\begin{pmatrix} q_i \\ p_i \\ p_{i,j} \end{pmatrix}$$

in the z-direction. This gradient is given by $\frac{p_{i,j}}{\sqrt{q_i^2+p_i^2}}$. And all gradients above $(q_i, p_i, 0)^T$ in S_1 are given by

$$\overline{\tilde{A}}_{i,1} := \left\{ \frac{p}{q} \middle| p \in \mathbb{N}_0, 0 \le p \le 1 \cdot p_i,\ q = 1 \cdot \sqrt{q_i^2 + p_i^2} \right\} .$$

Before we can proceed we need to take a closer look at the obtained gradients $\frac{p}{q}$. The fact that $q \notin \mathbb{N}$ will become a constant pain due to fraction reductions necessary to obtain the final gradient set. Without proper fraction reductions the following argumentation would become way more complex. So we introduce a little trick to obtain more suitable fractions characterizing our gradients:

$$\frac{p}{\sqrt{p_i^2 + q_i^2}} = \frac{p}{q_i} \frac{q_i}{\sqrt{p_i^2 + q_i^2}} .$$

From now on we interpret $\frac{q_i}{\sqrt{p_i^2+q_i^2}}$ as a constant relating to our new gradient set

$$\tilde{A}_{i,1} := \left\{ \frac{p}{q} \middle| p \in \mathbb{N}_0, 0 \le p \le 1 \cdot p_i,\ q = 1 \cdot q_i \right\} .$$

The next step results from the case that there lie more than one point of the uniform grid on our discretization line through $(q_i, p_i, 0)^T$ (see 3.6, left), which is the case when $\left\lfloor \frac{n}{q_i} \right\rfloor > 1$. In such a case we want to use the additional points above $t \cdot (q_i, p_i, 0)^T$. This results into the discretization points

$$\begin{pmatrix} t \cdot \begin{pmatrix} q_i \\ p_i \end{pmatrix} \\ p_{i,j} \end{pmatrix} ,$$

with the corresponding gradient being $\frac{p_{i,j}}{tq_i}$ and the relating constant $\frac{q_i}{\sqrt{q_i^2+p_i^2}}$. All gradients above $t \cdot (q_i, p_i, 0)^T$ are now given by

$$\tilde{A}_{i,t} := \left\{ \left(\frac{p}{q}, t \right) \middle| p \in \mathbb{N}_0, 0 \le p \le t \cdot p_i,\ q = t \cdot q_i \right\} .$$

So in the end all usable gradients corresponding to one $F_i = \frac{p_i}{q_i}$ are given by

$$\tilde{A}_i := \bigcup_{t=1,\dots,\left\lfloor \frac{n}{q_i} \right\rfloor} \tilde{A}_{i,t} .$$

By neglecting multiples of maximal reduced fractions and using only the points with the smallest distances from zero for the gradients (minimization of $t_{i,j}$) we obtain the set $\tilde{\mathfrak{F}}_{i,n}$ that corresponds to all gradients that can be used to calculate all $\varphi_{i,j}$ (angle in z - $(q_i, p_i, 0)^T$ plane) corresponding to one θ_i (angle in x-y plane):

$$\begin{aligned} \tilde{\mathfrak{F}}_{i,n} &:= \left(\left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j} \right) \middle| \begin{array}{l} p_{i,j} \le q_{i,j}, t_{i,j} \in \mathbb{N}_0 \wedge \gcd(p_{i,j}, q_{i,j}) = 1 \\ \wedge \exists a \in \tilde{A}_i : \left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j} \right) = a \wedge \forall j > 1 : \frac{p_{i,j-1}}{q_{i,j-1}} < \frac{p_{i,j}}{q_{i,j}} \\ \wedge t_{i,j} = \min \left\{ t \in \mathbb{N} \middle| \exists a \in \tilde{A}_i : \left(\frac{p_{i,j}}{q_{i,j}}, t \right) = a \right\} \end{array} \right) \\ &= \left(\left(\frac{p_{i,1}}{q_{i,1}}, t_{i,1} \right), \dots, \left(\frac{p_{i,N_i}}{q_{i,N_i}}, t_{i,N_i} \right) \right) = \left((F_{i,1}, t_{i,1}), \dots, (F_{i,N_i}, t_{i,N_i}) \right) , \\ N_i &:= \left| \tilde{\mathfrak{F}}_{i,n} \right| \le N , \quad (\le N \text{ by construction}) . \end{aligned}$$

Here we used the fraction reduction $\gcd(p_{i,j}, q_{i,j}) = 1$ only to neglect multiple appearances of gradients. But by doing so we introduced a new problem where fractions $\frac{p_{i,j}}{q_{i,j}}$ get reduced (by $c \in \mathbb{N}$) in the set $\tilde{\mathfrak{F}}_{i,n}$ to the point where

$$\begin{pmatrix} \frac{1}{c} t_{i,j} \cdot \begin{pmatrix} q_i \\ p_i \end{pmatrix} \\ p_{i,j} \end{pmatrix} ,$$

does not lie on our grid. So we need to to compensate the reduction. Such a fraction reduction fulfills the equation $q_{i,j} \cdot c = q_i \cdot t_{i,j}$ (because $q_{i,j} \cdot c$ is a q from $\tilde{A}_{i,t_{i,j}}$). This equation directly leads to the conclusion that we have to use the discretization points

$$\begin{pmatrix} t_{i,j} \cdot \begin{pmatrix} q_i \\ p_i \end{pmatrix} \\ c p_{i,j} \end{pmatrix} = \begin{pmatrix} t_{i,j} \cdot \begin{pmatrix} q_i \\ p_i \end{pmatrix} \\ \frac{t_{i,j} q_i}{q_{i,j}} p_{i,j} \end{pmatrix} .$$

For the following convergence prove we need some preliminary considerations. At first we recall how we can extract the angles in spherical coordinates from a given point $(x, y, z)^T \in \mathbb{R}^3$. In this work we use the following parametrization of a sphere

$$\omega(\theta, \varphi) := \begin{pmatrix} \cos(\varphi)\cos(\theta) \\ \cos(\varphi)\sin(\theta) \\ \sin(\varphi) \end{pmatrix} .$$

and the associated spherical coordinates (so we do not use the "standard" spherical coordinates). Transforming $(x, y, z)^T$ into our spherical coordinates yields

$$r = \sqrt{x^2 + y^2 + z^2}, \quad \theta = \arctan\left(\frac{y}{x}\right), \quad \varphi = \arcsin\left(\frac{z}{r}\right) . \tag{3.3.1}$$

Thinking about the fact that r, $\sqrt{x^2 + y^2}$, z form a right triangle with r being the hypotenuse we get

$$\varphi = \arcsin\left(\frac{z}{r}\right) = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) . \tag{3.3.2}$$

More considerations:

(a) Let $A \subset B \subset \mathbb{R}_0^+, |A| < |B| < \infty$, where $A = (a_1, \ldots, a_{|A|})$ and $B = (b_1, \ldots, b_{|B|})$ are increasing sequences with $a_1 = b_1, a_{|A|} = b_{|B|}$. In this case we instantly get

$$(a_{i+1} - a_i) = \sum_{j: b_j \in [a_i, a_{i+1})} (b_{j+1} - b_j) \qquad \wedge$$
$$\sum_{i=1}^{|A|-1} (a_{i+1} - a_i) = \sum_{i=1}^{|A|-1} \left(\sum_{j: b_j \in [a_i, a_{i+1})} (b_{j+1} - b_j) \right) = \sum_{j=1}^{|B|-1} (b_{j+1} - b_j) .$$

We use this to prove an inequality of the form

$$\sum_{i=2}^{|A|-1} (a_{i+1} - a_{i-1})^2 > \frac{1}{4} \sum_{j=2}^{|B|-1} (b_{i+1} - b_{i-1})^2 \quad :$$

$$\begin{aligned}
a_{i+1} - a_{i-1} = a_{i+1} - a_i + a_i - a_{i-1} &= \sum_{b_j \in [a_i, a_{i+1})} (b_{j+1} - b_j) + \sum_{b_j \in [a_{i-1}, a_i)} (b_{j+1} - b_j) \\
&= \frac{1}{2} \left[\begin{array}{l} \sum_{b_j \in [a_i, a_{i+1})} (b_{j+1} - b_j) + \sum_{b_j \in [a_{i-1}, a_i)} (b_{j+1} - b_j) \\ + \sum_{b_j \in (a_i, a_{i+1}]} (b_j - b_{j-1}) + \sum_{b_j \in (a_{i-1}, a_i]} (b_j - b_{j-1}) \end{array} \right] \\
&> \frac{1}{2} \left(\sum_{b_j \in (a_{i-1}, a_{i+1})} (b_{j+1} - b_j) + \sum_{b_j \in (a_{i-1}, a_{i+1})} (b_j - b_{j-1}) \right) \\
&= \frac{1}{2} \sum_{b_j \in (a_{i-1}, a_{i+1})} (b_{j+1} - b_{j-1}) \\
\implies \sum_{i=2}^{|A|-1} (a_{i+1} - a_{i-1})^2 &> \sum_{i=2}^{|A|-1} \left(\frac{1}{2} \sum_{b_j \in (a_{i-1}, a_{i+1})} (b_{j+1} - b_{j-1}) \right)^2 \\
&> \frac{1}{4} \sum_{i=2}^{|A|-1} \sum_{b_j \in (a_{i-1}, a_{i+1})} (b_{j+1} - b_{j-1})^2 > \frac{1}{4} \sum_{b_j \in (a_1, a_{|A|})} (b_{j+1} - b_{j-1})^2 \\
&= \frac{1}{4} \sum_{j=2}^{|B|-1} (b_{i+1} - b_{i-1})^2 .
\end{aligned}$$

Here we have used some very rough estimates to simplify the above and the following argumentation. A more precise evaluation would only change the constant $\frac{1}{4}$ to $\frac{1}{2}$ in exchange for a more complicated formula and higher order error terms in the following argumentation due to the necessary special treatment of the sequence start and ending.

(b) Let $a < b$; $a, b \in \mathbb{R}$. We know that $\frac{\mathrm{d}}{\mathrm{d}x} \arctan(x) = \frac{1}{1+x^2} \leq 1$, from this follows

$$\arctan(b) - \arctan(a) < b - a \,. \tag{3.3.3}$$

Now we need the point on the grid corresponding to $F_i, F_{i,j}$. It is clear that $(q_i, p_i, 0)^T$ corresponds to F_i. And we have seen that the additional points that are associated with $F_{i,j}$ are given by

$$P_{ij} := \begin{pmatrix} t_{i,j} \begin{pmatrix} q_i \\ p_i \end{pmatrix} \\ \frac{t_{i,j} q_i}{q_{i,j}} p_{i,j} \end{pmatrix} .$$

Here $t_{i,j}$ corresponds to the case where multiple grid points lie on the line corresponding to F_i and $\frac{t_{i,j} q_i}{q_{i,j}} \in \mathbb{N}$ corresponds to the possible reduction of the fractions that can occur in the set $\widetilde{\mathfrak{F}}_{i,n}$. So if there was a fraction reduction by $c \in \mathbb{N}$ then $t_{i,j} q_i = c q_{i,j}$. From this and (3.3.1), (3.3.2) follows

$$r_{i,j} = t_{i,j}\sqrt{q_i^2 + p_i^2 + \left(\frac{p_{i,j}}{q_{i,j}}q_i\right)^2}, \theta_i = \arctan\left(\frac{p_i}{q_i}\right), \varphi_{i,j} = \arctan\left(\frac{p_{i,j}}{q_{i,j}}\frac{q_i}{\sqrt{q_i^2+p_i^2}}\right).$$

The set of usable angles for our approximation is now given by

$$\mathfrak{F}_{i,n} := \left(\arctan\left(\frac{F_{i,1}q_i}{\sqrt{q_i^2+p_i^2}}\right), \dots, \arctan\left(\frac{F_{i,N_i}q_i}{\sqrt{q_i^2+p_i^2}}\right)\right) = (\varphi_{i,1}, \dots, \varphi_{i,N_i}) ,$$

which corresponds to the following circle arcs that can be used for the approximation:

$$\alpha_{i,j} := \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2}, \quad \varphi_{i,0} := 0, \quad \varphi_{i,N_i+1} := \varphi_{i,N_i} .$$

Here we know that

$$\varphi_{i,1} = 0, \qquad \varphi_{i,N_i} = \arctan(\sin(\theta_i)) .$$

The last one φ_{i,N_i} needs some clarification. Assuming that we look at the line in the $x - y$ plane that is given through $\frac{p_i}{q_i}$, the largest φ is given by $p_{i,j} = p_i, q_{i,j} = q_i$. This results into $P_{i,j} = (q_i, p_i, p_{i,j})^T = (q_i, p_i, p_i)^T$ and gives (by (3.3.2))

$$\begin{aligned}\varphi_{\max} &= \arctan\left(\frac{p_i}{\sqrt{p_i^2+q_i^2}}\right) = \arctan\left(\frac{1}{\sqrt{\frac{q_i^2}{p_i^2}+1}}\right) \\ &= \arctan\left(\cos\left(\arctan\left(\frac{q_i}{p_i}\right)\right)\right) \\ &= \arctan\left(\cos\left(\arctan\left(\tan\left(\frac{\pi}{2} - \arctan\left(\frac{p_i}{q_i}\right)\right)\right)\right)\right) \\ &= \arctan(\sin(\theta_i)) .\end{aligned}$$

At this point we use a little trick to get to the point where we can prove convergence. We look at the set $\tilde{A}_{i,\left\lfloor\frac{n}{q_i}\right\rfloor}$ given by

$$\tilde{A}_{i,\left\lfloor\frac{n}{q_i}\right\rfloor} = \left\{\frac{0}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i}, \frac{1}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i}, \dots, \frac{p_i\left\lfloor\frac{n}{q_i}\right\rfloor}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i}\right\} \times \left\{\left\lfloor\frac{n}{q_i}\right\rfloor\right\} ,$$

and we know from the definition of $\tilde{\mathfrak{F}}_{i,n}$ that $\tilde{A}_{i,\left\lfloor\frac{n}{q_i}\right\rfloor} \subset \tilde{\mathfrak{F}}_{i,n}$ (ignoring possible fraction reductions). This leads to the conclusion that the increasing sequence

$$A_{i,\left\lfloor\frac{n}{q_i}\right\rfloor} := \left(\arctan\left(\frac{0}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i}\frac{q_i}{\sqrt{q_i^2+p_i^2}}\right), \dots, \arctan\left(\frac{p_i\left\lfloor\frac{n}{q_i}\right\rfloor}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i}\frac{q_i}{\sqrt{q_i^2+p_i^2}}\right)\right)$$

$$=: \left(\tilde{\varphi}_{i,1}, \dots, \tilde{\varphi}_{i,p_i\left\lfloor\frac{n}{q_i}\right\rfloor+1}\right),$$

is a subsequence of $\mathfrak{F}_{i,n}$ with $\overbrace{\varphi_{i,1} = \tilde{\varphi}_{i,1} \ \wedge \ \varphi_{\max} = \varphi_{i,N_i} = \tilde{\varphi}_{i,p_i\left\lfloor\frac{n}{q_i}\right\rfloor+1}}^{(*)}$. With the above arguments we can use consideration (b) to obtain an upper bound for the differences of successive $\varphi_{i,\bullet}$:

$$\forall j \in \left\{1, \dots, q_i \left\lfloor\frac{n}{q_i}\right\rfloor\right\}:$$

$$\tilde{\varphi}_{i,j+1} - \tilde{\varphi}_{i,j} \overset{(b)}{<} \frac{j+1}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2+p_i^2}} - \frac{j}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2+p_i^2}} = \frac{1}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2+p_i^2}} \tag{3.3.4}$$

$$\tilde{A}_{i,\left\lfloor\frac{n}{q_i}\right\rfloor} \subset \tilde{\mathfrak{F}}_{i,n} \overset{(*)}{\Longrightarrow} \forall j \in \{1, \dots, N_i\} : \varphi_{i,j+1} - \varphi_{i,j} < \frac{1}{\left\lfloor\frac{n}{q_i}\right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2+p_i^2}}. \tag{3.3.5}$$

This together with the following equivalent of the first part of lemma 3.1.4 can be used to calculate the error.

Lemma 3.3.3

(i) $0 \le \mu_{i,j} := \frac{\varphi_{i,j}+\varphi_{i,j-1}}{2} \le \varphi_{i,j} \le \mu_{i,j+1} \le \arctan(\sin(\theta_i))$

(ii) $\alpha_{i,j}^2 \ge (\varphi_{i,j} - \mu_{i,j})^2 \wedge \alpha_{i,j}^2 \ge (\mu_{i,j+1} - \varphi_{i,j})^2$

(iii) $0 \le \alpha_{i,j} = \mu_{i,j+1} - \mu_{i,j} \le \frac{1}{\left\lfloor\frac{n}{q_i}\right\rfloor\sqrt{q_i^2+p_i^2}}$

(iv) $\sum\limits_{i=1}^{N_i} \alpha_{i,j} = \arctan(\sin(\theta_i)) \le \frac{\pi}{4}$

Proof: The first one is obvious, because $\varphi_{i,j}$ is an increasing sequence in j and due to the definition of $\mu_{i,j}$. The second one is a consequence of the first one as can be seen in the proof of 3.1.4. The third follows from

$$\alpha_{i,j} = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2} = \frac{\varphi_{i,j+1} - \varphi_{i,j} + \varphi_{i,j} - \varphi_{i,j-1}}{2}$$

together with result (3.3.5) and the last one follows from

$$\sum_{i=1}^{N_i} \alpha_{i,j} = \frac{\varphi_{i,N_i+1} + \varphi_{i,N_i}}{2} - \frac{\varphi_{i,1} + \varphi_{i,0}}{2} = \frac{2 \cdot \arctan(\sin(\theta_i)) - 2 \cdot 0}{2}. \qquad \square$$

We apply this to reuse the main part of the proof of proposition 3.1.5:

$$\begin{aligned}
a(\theta) &:= \varphi_{\max}(\theta) = \arctan(\sin(\theta)) \\
e_{2,i} &:= \left| \int_0^{a(\theta_i)} H(\theta_i, \varphi) \mathrm{d}\varphi - \sum_{j=1}^{N_i} \alpha_{i,j} H(\theta_i, \varphi_j) \right| \\
&= \ldots \qquad \text{see proof of 3.1.5} \qquad \ldots \\
&< \overbrace{\sup_{\varphi \in [0, a(\theta_i)]} \left| \frac{\partial H(\theta_i, \varphi)}{\partial \varphi} \right|}^{c:=} \sum_{j=1}^{N_i} \alpha_{i,j}^2 = c \frac{1}{4} \sum_{j=1}^{N_i} (\varphi_{i,j+1} - \varphi_{i,j-1})^2 \\
&\overset{\text{(a)}}{<} c \sum_{j=1}^{p_i \left\lfloor \frac{n}{q_i} \right\rfloor} (\tilde{\varphi}_{i,j+1} - \tilde{\varphi}_{i,j-1})^2 \\
&\overset{(3.3.4)}{<} c \sum_{j=1}^{p_i \left\lfloor \frac{n}{q_i} \right\rfloor} \left(\frac{j+1}{\left\lfloor \frac{n}{q_i} \right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2 + p_i^2}} - \frac{j-1}{\left\lfloor \frac{n}{q_i} \right\rfloor q_i} \frac{q_i}{\sqrt{q_i^2 + p_i^2}} \right)^2 \\
&= 2c \sum_{j=1}^{p_i \left\lfloor \frac{n}{q_i} \right\rfloor} \left(\frac{1}{\left\lfloor \frac{n}{q_i} \right\rfloor} \frac{1}{\sqrt{q_i^2 + p_i^2}} \right)^2 = 2c p_i \left\lfloor \frac{n}{q_i} \right\rfloor \left(\frac{1}{\left\lfloor \frac{n}{q_i} \right\rfloor \cdot \sqrt{q_i^2 + p_i^2}} \right)^2 \\
&\leq 2c \frac{p_i}{\left\lfloor \frac{n}{q_i} \right\rfloor \cdot (q_i^2 + p_i^2)} .
\end{aligned}$$

We have to take another sum over this error to get the final result:

$$\begin{aligned}
e_2 := \sum_{i=1}^{N} \alpha_i e_{2,i} &< \sum_{i=1}^{N} \alpha_i 2 \sup_{\varphi \in [0, a(\theta_i)]} \left| \frac{\partial H(\theta_i, \varphi)}{\partial \varphi} \right| \frac{p_i}{\left\lfloor \frac{n}{q_i} \right\rfloor \cdot (q_i^2 + p_i^2)} \\
&< \sum_{i=1}^{N} \frac{2}{n q_i} 2 \sup_{\varphi \in [0, a(\theta_i)]} \left| \frac{\partial H(\theta_i, \varphi)}{\partial \varphi} \right| \frac{p_i}{\left\lfloor \frac{n}{q_i} \right\rfloor \cdot (q_i^2 + p_i^2)}, \qquad \text{by 3.1.4(iii)} \\
&\leq 4 \overbrace{\sup_{\substack{\varphi \in [0, a(\theta)] \\ \theta \in [0, \frac{\pi}{4}]}} \left| \frac{\partial H(\theta, \varphi)}{\partial \varphi} \right|}^{\tilde{c}} \sum_{i=1}^{N} \frac{1}{n \cdot \left\lfloor \frac{n}{q_i} \right\rfloor \cdot (q_i^2 + p_i^2)} \\
&< 4\tilde{c} \sum_{q=1}^{n} \sum_{p=1}^{q} \frac{1}{n \cdot \left\lfloor \frac{n}{q} \right\rfloor \cdot q^2} = 4\tilde{c} \sum_{q=1}^{n} \frac{1}{n \left(\left\lfloor \frac{n}{q} \right\rfloor q \right)} \\
&< 4\tilde{c} \sum_{q=1}^{n} \frac{2}{n^2}, \qquad \text{because } \left\lfloor \frac{n}{q} \right\rfloor q > \frac{n}{2}
\end{aligned}$$

$$= 8\tilde{c}\frac{1}{n}\,.$$

Finally we get the approximation and the upper bound for the total error through

$$\sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{N_i}\alpha_{i,j}H(\theta_i,\varphi_{i,j}) \approx \int_0^{\frac{\pi}{4}}\int_0^{a(\theta)} H(\theta,\varphi)\mathrm{d}\varphi\mathrm{d}\theta\,,$$

$$e_2 + e_1 < 8\cdot \overbrace{\sup_{\substack{\varphi\,\in\,[0,\,a(\theta)]\\ \theta\,\in\,\left[0,\,\frac{\pi}{4}\right]}}\left|\frac{\partial H(\theta,\varphi)}{\partial\varphi}\right|}^{K_{\theta,\varphi}:=}\cdot\frac{1}{n} + 4\cdot\overbrace{\sup_{\theta\in\left[0,\frac{\pi}{4}\right]}|H'(\theta)|}^{K_\theta}\cdot\frac{\ln(n)+1}{n^2}\,. \qquad \square$$

Lemma 3.3.4 (Completion of the approximation)
Let $H \in C^1\left([0,2\pi]\times\left[-\frac{\pi}{2},\frac{\pi}{2}\right]\to\mathbb{R}\right)$ and G be the automorphism group corresponding to the uniform grid. G consists of 48 elements in 3 dimensions. Now let Φ_k be the transformation from the spherical coordinates we used in the last proposition into the spherical coordinates that are given by the following discretization of the sphere: $\varphi_k\begin{pmatrix}\cos(\hat\theta)\cos(\hat\varphi)\\ \sin(\hat\theta)\cos(\hat\varphi)\\ \sin(\hat\varphi)\end{pmatrix}$, $\varphi_k \in G$ with pairwise distinct φ_k. Using the Farey sequence $\tilde{\mathfrak{F}}_n$, the approximation

$$\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta,\varphi)\cos(\varphi)\mathrm{d}\varphi\mathrm{d}\theta \approx \sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{N_i}\alpha_{i,j}\sum_{k=1,\ldots,48} H(\Phi_k(\theta_i,\varphi_{i,j}))\cos(\varphi_{i,j})\,,$$

yields an upper error bound of

$$\left|\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta,\varphi)\cos(\varphi)\mathrm{d}\varphi\mathrm{d}\theta - \sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{N_i}\alpha_{i,j}\sum_{k=1,\ldots,48} H(\Phi_k(\theta_i,\varphi_{i,j}))\cos(\varphi_{i,j})\right|$$
$$< 192\cdot\left(2K_{\theta,\varphi}\cdot\frac{1}{n} + K_\theta\cdot\frac{\ln(n)+1}{n^2}\right)\,,$$

where $K_{\theta,\varphi}, K_\theta$ are some constants depending only on H and its derivative.

Proof:
The idea of this proof is that we divide the domain of integration into 48 regions and then use coordinate transformations to reduce these 48 integrals to the point where we can use 3.3.2 to approximate them. Then we realize that the 48 necessary coordinate transformations correspond to the 48 elements of the automorphism group and that the functional determinants of these transformations in combination with $\cos(\varphi_{i,j})$ essentially vanish. So let us begin with the splitting of the domain, let $a_i := \frac{i-1}{4}\pi$, $a(x) := \arctan(\sin(x)), b(x) := \arctan(\cos(x))$, then

$$\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta =$$

$$\begin{aligned}\sum_{i=1}^{4}&\int_{a_{2i-1}}^{a_{2i}}\int_{0}^{a(\theta-a_{2i-1})}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i-1}}^{a_{2i}}\int_{a(\theta-a_{2i-1})}^{b(\theta-a_{2i-1})}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\\
&+\int_{a_{2i}}^{a_{2i+1}}\int_{0}^{a(a_{2i+1}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i}}^{a_{2i+1}}\int_{a(a_{2i+1}-\theta)}^{b(a_{2i+1}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\\
&+\int_{a_{2i-1}}^{a_{2i}}\int_{b(\theta-a_{2i-1})}^{\frac{\pi}{2}}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i}}^{a_{2i+1}}\int_{b(a_{2i+1}-\theta)}^{\frac{\pi}{2}}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\\
&+\int_{a_{2i-1}}^{a_{2i}}\int_{-a(\theta-a_{2i-1})}^{0}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i-1}}^{a_{2i}}\int_{-b(\theta-a_{2i-1})}^{-a(\theta-a_{2i-1})}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\\
&+\int_{a_{2i}}^{a_{2i+1}}\int_{-a(a_{2i+1}-\theta)}^{0}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i}}^{a_{2i+1}}\int_{-b(a_{2i+1}-\theta)}^{-a(a_{2i+1}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\\
&+\int_{a_{2i-1}}^{a_{2i}}\int_{-\frac{\pi}{2}}^{-b(\theta-a_{2i-1})}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta+\int_{a_{2i}}^{a_{2i+1}}\int_{-\frac{\pi}{2}}^{-b(a_{2i+1}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta\,.\end{aligned}$$

Here we see the 48 integrals corresponding to the 48 symmetry regions. To get a better grip of the problem we take a look at the 6 integrals corresponding to the positive quadrant (see figure 3.5, angle θ lying in the $x-y$ plane)

$$\begin{aligned}Q_1:=&\overbrace{\int_{0}^{\frac{\pi}{4}}\int_{0}^{a(\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_1\sim}+\overbrace{\int_{0}^{\frac{\pi}{4}}\int_{a(\theta)}^{b(\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_6\sim}\\
&+\overbrace{\int_{\frac{\pi}{4}}^{\frac{\pi}{2}}\int_{0}^{a(\frac{\pi}{2}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_2\sim}+\overbrace{\int_{\frac{\pi}{4}}^{\frac{\pi}{2}}\int_{a(\frac{\pi}{2}-\theta)}^{b(\frac{\pi}{2}-\theta)}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_3\sim}\\
&+\overbrace{\int_{0}^{\frac{\pi}{4}}\int_{b(\theta)}^{\frac{\pi}{2}}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_5\sim}+\overbrace{\int_{\frac{\pi}{4}}^{\frac{\pi}{2}}\int_{b(\frac{\pi}{2}-\theta)}^{\frac{\pi}{2}}H(\theta,\varphi)\cos(\varphi)\,\mathrm{d}\varphi\mathrm{d}\theta}^{S_4\sim}\,.\end{aligned}$$

Now we exemplarily prove that we can transform the integrals corresponding to S_6, S_5 into S_1, where the only difference occurs to be the transformation of the variables in H (no change in cos). The transformation of all other (46) integrals is then analog to this one. Starting with S_6 we see that S_6 is essentially the same as S_1 the only difference being a change of coordinates from x, y to x, z. So we want to apply the Farey approach in the $x-z$ plane and the extension of the approximation in the y direction. This corresponds to using another set of spherical coordinates based on another parametrization of the sphere:

$$\text{old}:\ \begin{pmatrix}\cos(\theta)\cos(\varphi)\\ \sin(\theta)\cos(\varphi)\\ \sin(\varphi)\end{pmatrix}=\begin{pmatrix}x\\ y\\ z\end{pmatrix}=\begin{pmatrix}\cos(\hat\theta)\cos(\hat\varphi)\\ \sin(\hat\varphi)\\ \sin(\hat\theta)\cos(\hat\varphi)\end{pmatrix}:\ \text{new}\,. \qquad (3.3.6)$$

The transformation from Cartesian $(x, y, z)^T$ to the new spherical coordinates is given by

$$\hat{r} = \sqrt{x^2 + y^2 + z^2}, \quad \hat{\theta} = \arctan\left(\frac{z}{x}\right), \quad \hat{\varphi} = \arctan\left(\frac{y}{\sqrt{x^2 + z^2}}\right).$$

The self inverse coordinate transformation Φ_6 between the new and the old spherical coordinates is given by

$$\begin{pmatrix} r \\ \theta \\ \varphi \end{pmatrix} = \Phi_6 \begin{pmatrix} \hat{r} \\ \hat{\theta} \\ \hat{\varphi} \end{pmatrix} = \begin{pmatrix} \hat{r} \\ \arctan\left(\frac{\tan(\hat{\varphi})}{\cos(\hat{\theta})}\right) \\ \arctan\left(\frac{\sin(\hat{\theta})}{\sqrt{\tan^2(\hat{\varphi}) + \cos^2(\hat{\theta})}}\right) \end{pmatrix},$$

$$|\det(D\Phi)| = \frac{\cos(\hat{\varphi})}{\sqrt{1 - \cos^2(\hat{\varphi}) + \cos^2(\hat{\varphi})\cos^2(\hat{\theta})}}.$$

The transformation of the domain of integration becomes a bit problematic, because the new coordinates depend on both φ, θ. Fortunately it is geometrically clear (see figure 3.5) that the domain remains the same. So we get

$$\begin{aligned} S_6 &\sim \int_0^{\frac{\pi}{4}} \int_{a(\theta)}^{b(\theta)} H(\theta, \varphi) \cos(\varphi)\, \mathrm{d}\varphi \mathrm{d}\theta \\ &= \int_0^{\frac{\pi}{4}} \int_0^{a(\hat{\theta})} H(\Phi_6(\hat{\theta}, \hat{\varphi})) \left[\begin{array}{c} \cos\left(\arctan\left(\frac{\sin(\hat{\theta})}{\sqrt{\tan^2(\hat{\varphi}) + \cos^2(\hat{\theta})}}\right)\right) \cdot \\ \frac{\cos(\hat{\varphi})}{\sqrt{1 - \cos^2(\hat{\varphi}) + \cos^2(\hat{\varphi})\cos^2(\hat{\theta})}} \end{array} \right] \mathrm{d}\hat{\varphi} \mathrm{d}\hat{\theta} \\ &= \int_0^{\frac{\pi}{4}} \int_0^{a(\hat{\theta})} H(\Phi_6(\hat{\theta}, \hat{\varphi})) \cos(\hat{\varphi}) \frac{\sqrt{1 - \cos^2(\hat{\varphi}) + \cos^2(\hat{\varphi})\cos^2(\hat{\theta})}}{\sqrt{1 - \cos^2(\hat{\varphi}) + \cos^2(\hat{\varphi})\cos^2(\hat{\theta})}} \mathrm{d}\hat{\varphi} \mathrm{d}\hat{\theta} \\ &= \int_0^{\frac{\pi}{4}} \int_0^{a(\hat{\theta})} H(\Phi_6(\hat{\theta}, \hat{\varphi})) \cos(\hat{\varphi})\, \mathrm{d}\hat{\varphi} \mathrm{d}\hat{\theta} \sim S_1(\Phi_6). \end{aligned}$$

And this last integral can be approximated in the same way we used for S_1, see 3.3.2. We can see that this transformation corresponds to a simple permutation of the Cartesian coordinates, see (3.3.6), and we see that the functional determinant of the transformation "magically" vanishes when we multiply it with the cosine of the transformed φ.

Remark:

This is obvious because this transformation can be interpreted as a two step transformation, from spherical coordinates back to Cartesian coordinates and then into the new spherical coordinates. Another interpretation is given by applying different spherical transformations to the Cartesian integrals and seeing afterwards that the proper choice of transformations leads to integrals with the same domain of integration, distinguishable only by the permutation of the Cartesian coordinates.

This (the vanishing of the functional determinant) happens every time we use a permutation of the Cartesian coordinates to obtain another set of spherical coordinates. Now it is easy to see that we can transform all 6 integrals in Q_1 by using all 6 possible permutations of the Cartesian coordinates in the way described above. This finally leads to

$$Q_1 = \sum_{i=1}^{6} \int_0^{\frac{\pi}{4}} \int_0^{a(\theta)} H(\Phi_i(\theta, \varphi)) \cos(\varphi) \, \mathrm{d}\varphi \mathrm{d}\theta \, .$$

Interestingly these transformations correspond to the following operators of the automorphism group which happen to be permutation matrices

$$\Phi_1 \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} =: M_1, \quad \Phi_2 \sim \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} =: M_2, \quad \Phi_3 \sim \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} =: M_3,$$
$$\Phi_4 \sim \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} =: M_4, \quad \Phi_5 \sim \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} =: M_5, \quad \Phi_6 \sim \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} =: M_6 \, .$$

Now it is easy to see that all possible sign changes inside of these matrices give in total 8 operators per transformation $\Phi_1, \ldots, \Phi_6$. These 48 operators correspond to the 48 coordinate transformations, and all these operators together form the automorphism group. The successive application of these 48 coordinate transformations lead to

$$\int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta, \varphi) \cos(\varphi) \, \mathrm{d}\varphi \mathrm{d}\theta = \sum_{k=1}^{48} \int_0^{\frac{\pi}{4}} \int_0^{a(\theta)} H(\Phi_k(\theta, \varphi)) \cos(\varphi) \, \mathrm{d}\varphi \mathrm{d}\theta \, .$$

Now we can apply 3.3.2 on these 48 integrals giving

$$\int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta, \varphi) \cos(\varphi) \mathrm{d}\varphi \mathrm{d}\theta \approx \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{N_i} \alpha_{i,j} \sum_{k=1,\ldots,48} H(\Phi_k(\theta_i, \varphi_{i,j})) \cos(\varphi_{i,j}) \, ,$$

with an upper error bound of

$$\left| \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\theta, \varphi) \cos(\varphi) \mathrm{d}\varphi \mathrm{d}\theta - \sum_{i=1}^{N} \alpha_{i,n} \sum_{j=1}^{N_i} \alpha_{i,j} \sum_{k=1,\ldots,48} H(\Phi_k(\theta_i, \varphi_{i,j})) \cos(\varphi_{i,j}) \right|$$
$$< 48 \cdot \left(8K_{\theta,\varphi} \cdot \frac{1}{n} + 4K_\theta \cdot \frac{\ln(n) + 1}{n^2} \right) \, ,$$

$$\text{with } K_{\theta,\varphi} := \sup_{\substack{\varphi \, \in \, [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \theta \, \in \, [0, \, 2\pi]}} \left| \frac{\partial H(\theta, \varphi) \cos(\varphi)}{\partial \varphi} \right| , K_\theta := \sup_{\theta \in [0, 2\pi]} \left| \frac{\partial \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H(\varphi, \theta) \cos(\varphi) \mathrm{d}\varphi}{\partial \theta} \right| . \square$$

Remark 3.3.5
Comparable to 3.1.6 we have to remark that

(i) the above discretization corresponds to

$$\sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{N_i}\alpha_{i,j}\sum_{k=1,\dots,48}\tilde{H}(\omega(\Phi_k(\theta_i,\varphi_{i,j})))\cos(\varphi_{i,j})\approx\int_{S^2}\tilde{H}(\mathfrak{w})\mathrm{d}\mathfrak{w}$$

with $H(\theta,\varphi)=\tilde{H}(\omega(\theta,\varphi))$. We use the representation in the above proposition, because simplifications of the transformed collision operator are possible. And they should be done before the application of the Farey approximation.

(ii) The main properties of this discretization are that the used angles correspond to the gradient of lines that fit onto our uniform grid and that we have the freedom to handle the different symmetry regions independently. The last one finally leads to the point where we can choose which grid symmetries get preserved during the discretization.

Definition 3.3.6 (Rotations)
Let $\mathfrak{x}=(1,0,0)^T,\mathfrak{y}=(0,1,0)^T,\mathfrak{z}=(0,0,1)^T$ and $R_{\mathfrak{w}}(\alpha)$ be the rotation around a vector $\mathfrak{w}$ with an angle of α. We call a rotation

(i) $R_{\mathfrak{w}}(\alpha)$ extrinsic if it is a rotation around one of the "global" axis, $\mathfrak{w}\in\{\mathfrak{x},\mathfrak{y},\mathfrak{z}\}$,

(ii) $R_{\mathfrak{w}}(\alpha)$ intrinsic if it is a rotation around one of the axis after an arbitrary rotation $\tilde{R}$ that rotated the coordinate system and anything inside it, $\mathfrak{w}\in\{\tilde{R}\mathfrak{x},\tilde{R}\mathfrak{y},\tilde{R}\mathfrak{z}\}$.

Lemma 3.3.7 (Simplification of intrinsic rotations)
An intrinsic rotation can be expressed by an extrinsic rotation :

$$R_{\tilde{R}\mathfrak{a}}(\alpha)\tilde{R}\mathfrak{b}=\tilde{R}R_{\mathfrak{a}}(\alpha)\mathfrak{b},\qquad \mathfrak{a}\in\{\mathfrak{x},\mathfrak{y},\mathfrak{z}\},\mathfrak{b}\in\mathbb{R}^3\alpha\in[0,2\pi)\,.$$

Proof:
Looking at the diagram

$$\begin{tikzcd}\mathfrak{b} \arrow[r, "\tilde{R}"] & \tilde{\mathfrak{b}} \arrow[r, "\tilde{R}R_{\mathfrak{a}}(\alpha)\tilde{R}^{-1}"] & \tilde{\tilde{\mathfrak{b}}}\,,\end{tikzcd}$$

and knowing that a rotation $R_{\tilde{R}\mathfrak{a}}(\alpha)$ corresponds to a rotation within another basis given by the basis transformation $\tilde{R}R_{\mathfrak{a}}(\alpha)\tilde{R}^{-1}$ we get

$$R_{\tilde{R}\mathfrak{a}}(\alpha)\tilde{R}\mathfrak{b}=\tilde{\tilde{\mathfrak{b}}}=\tilde{R}R_{\mathfrak{a}}(\alpha)\tilde{R}^{-1}\tilde{\mathfrak{b}}=\tilde{R}R_{\mathfrak{a}}(\alpha)\tilde{R}^{-1}\tilde{R}\mathfrak{b}=\tilde{R}R_{\mathfrak{a}}(\alpha)\mathfrak{b}\,.\qquad\square$$

Remark 3.3.8
The last definition and lemma 3.3.6,3.3.7 can be used to understand and substantially simplify the following rotation sequences, because these rotations can be expressed by a product of two dimensional rotations in three dimensions. Two dimensional in the sense that $R_{\mathfrak{x}},R_{\mathfrak{y}},R_{\mathfrak{z}}$ rotate only within the corresponding planes and correspond to the simplest 3D rotations.

Lemma 3.3.9 (Transformation of the collision operator)
Let us assume that the density f has the property $\operatorname{supp}(f) \subset B_{\frac{L}{2}}(\mathfrak{o}), L \in \mathbb{R}^+$ and let $\mathfrak{v} \in B_{\frac{L}{2}}(\mathfrak{o})$, then the following holds true:

$$I[f](\mathfrak{v}) = \int_{\mathbb{R}^3}\int_{S^2} [f(\mathfrak{v}_1')f(\mathfrak{w}_1') - f(\mathfrak{v})f(\mathfrak{w})]k(\mathfrak{v}-\mathfrak{w},\eta)\mathrm{d}\eta\mathrm{d}\mathfrak{w} =$$
$$\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_0^L [f(\mathfrak{v}_2')f(\mathfrak{w}_2') - f(\mathfrak{v})f(\mathfrak{w}_2)]k(r\omega_2,\omega)D\begin{bmatrix} r,\theta, \\ \varphi,\psi \end{bmatrix} \mathrm{d}r\mathrm{d}\psi\mathrm{d}\lambda\mathrm{d}\varphi\mathrm{d}\theta\,,$$

with

$$\begin{aligned} \mathfrak{v}_1' &:= \mathfrak{v} + \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\eta\rangle\eta, & \mathfrak{v}_2' &:= \mathfrak{v} + r\langle\omega_2(\theta,\varphi,\lambda,\psi),\omega(\theta,\varphi)\rangle\omega(\theta,\varphi)\,, \\ \mathfrak{w}_1' &:= \mathfrak{w} - \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\eta\rangle\eta, & \mathfrak{w}_2' &:= \mathfrak{w}_2 - r\langle\omega_2(\theta,\varphi,\lambda,\psi),\omega(\theta,\varphi)\rangle\omega(\theta,\varphi)\,, \\ \mathfrak{w}_2 &:= \mathfrak{v} + r\omega_2(\theta,\varphi,\lambda,\psi), & \omega(\theta,\varphi) &:= \begin{pmatrix}\cos(\varphi)\cos(\theta)\\ \cos(\varphi)\sin(\theta)\\ \sin(\varphi)\end{pmatrix} = \mathfrak{x}'\,, \end{aligned}$$

$$\mathfrak{x} := \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad \mathfrak{y} := \begin{pmatrix}0\\1\\0\end{pmatrix}, \quad \mathfrak{z} := \begin{pmatrix}0\\0\\1\end{pmatrix},$$

$$\mathfrak{y}' := R_{\mathfrak{z}}(\theta)\mathfrak{y}, \quad \mathfrak{x}' := R_{\mathfrak{y}'}(-\varphi)R_{\mathfrak{z}}(\theta)\mathfrak{x}, \quad \mathfrak{z}' = R_{\mathfrak{y}'}(-\varphi)\mathfrak{z}, \quad \mathfrak{y}'' = R_{\mathfrak{z}'}(\lambda)\mathfrak{y}'\,,$$

$$\omega_2(\theta,\varphi,\lambda,\psi) := R_{\mathfrak{y}''(\theta,\varphi,\lambda)}(-\psi)R_{\mathfrak{z}'(\theta,\varphi)}(\lambda)\operatorname{diag}(r_{\mathfrak{x}}(\theta,\varphi), r_{\mathfrak{y}}(\theta,\varphi), r_{\mathfrak{z}}(\theta,\varphi))\mathfrak{x}'(\theta,\varphi)$$
$$= \begin{pmatrix} r_{\mathfrak{x}}\cos(\theta)\cos(\varphi)\cos(\lambda)\cos(\psi) - r_{\mathfrak{y}}\sin(\lambda)\cos(\psi)\sin(\theta) - r_{\mathfrak{z}}\sin(\varphi)\sin(\psi)\cos(\theta) \\ r_{\mathfrak{x}}\sin(\theta)\cos(\varphi)\cos(\lambda)\cos(\psi) + r_{\mathfrak{y}}\sin(\lambda)\cos(\psi)\cos(\theta) - r_{\mathfrak{z}}\sin(\varphi)\sin(\psi)\sin(\theta) \\ r_{\mathfrak{x}}\sin(\varphi)\cos(\lambda)\cos(\psi) + r_{\mathfrak{z}}\cos(\varphi)\sin(\psi) \end{pmatrix},$$

$$D(r,\theta,\varphi,\psi) := r^2 r_{\mathfrak{x}}(\theta,\varphi)r_{\mathfrak{y}}(\theta,\varphi)r_{\mathfrak{z}}(\theta,\varphi)\cos(\varphi)\cos(\psi)\,.$$

Proof:
Our aim is to apply the same approach as in the 2 dimensional case, so we want to identify the inner integration with a rotation of the coordinate system. We then use this analogy to conduct the outer integration within these new coordinates. This directly corresponds to the integral transformation in 2 dimensions using $\omega(\theta+\lambda)$. This ansatz is essential to obtain the interchangeability of the pre- and post collisional velocities. To obtain a deeper understanding of this approach we look at the transformation into spherical coordinates using rotations. Defining

$$\mathfrak{x} := \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad \mathfrak{y} := \begin{pmatrix}0\\1\\0\end{pmatrix}, \quad \mathfrak{z} := \begin{pmatrix}0\\0\\1\end{pmatrix}$$

and using the knowledge

$$R_{\mathfrak{a}}(\alpha)\mathfrak{b} = \mathfrak{a}\langle\mathfrak{a},\mathfrak{b}\rangle + \cos(\alpha)(\mathfrak{a}\times\mathfrak{b})\times\mathfrak{a} + \sin(\alpha)(\mathfrak{a}\times\mathfrak{b}),$$

we get spherical coordinates by

$$\mathfrak{y}' := R_{\mathfrak{z}}(\theta)\mathfrak{y},\ \omega(\theta,\varphi) = \begin{pmatrix}\cos(\theta)\cos(\varphi)\\ \sin(\theta)\cos(\varphi)\\ \sin(\varphi)\end{pmatrix} = R_{\mathfrak{y}'}(-\varphi)R_{\mathfrak{z}}(\theta)\mathfrak{x}\,.$$

So we can understand the spherical coordinates $\omega(\theta,\varphi)$ as two successive rotations. The first one around $\mathfrak{z}$ and the second around the rotated y-axis $R_{\mathfrak{z}}(\theta)\mathfrak{y}$. Now we use the resulting rotated coordinate system

$$\mathfrak{x}' = \omega(\theta,\varphi), \quad \mathfrak{z}' = R_{\mathfrak{y}'}(-\varphi)\mathfrak{z}, \quad \mathfrak{y}'' = R_{\mathfrak{z}'}(\lambda)\mathfrak{y}'$$

to transform the outer integral into ellipsoidal coordinates within this rotated coordinate system:

$$\begin{aligned}
&\omega_2(\lambda,\psi) := R_{\mathfrak{y}''(\theta,\varphi,\lambda)}(-\psi)R_{\mathfrak{z}'(\theta,\varphi)}(\lambda)\mathrm{diag}(r_{\mathfrak{x}}(\theta,\varphi), r_{\mathfrak{y}}(\theta,\varphi), r_{\mathfrak{z}}(\theta,\varphi))\mathfrak{x}'(\theta,\varphi)\\
&= \begin{pmatrix} r_{\mathfrak{x}}\cos(\theta)\cos(\varphi)\cos(\lambda)\cos(\psi) - r_{\mathfrak{y}}\sin(\lambda)\cos(\psi)\sin(\theta) - r_{\mathfrak{z}}\sin(\varphi)\sin(\psi)\cos(\theta)\\ r_{\mathfrak{x}}\sin(\theta)\cos(\varphi)\cos(\lambda)\cos(\psi) + r_{\mathfrak{y}}\sin(\lambda)\cos(\psi)\cos(\theta) - r_{\mathfrak{z}}\sin(\varphi)\sin(\psi)\sin(\theta)\\ r_{\mathfrak{x}}\sin(\varphi)\cos(\lambda)\cos(\psi) + r_{\mathfrak{z}}\cos(\varphi)\sin(\psi)\end{pmatrix}\\
&=: \omega_2(\theta,\varphi,\lambda,\psi)\,.
\end{aligned}$$

Here $r_{\mathfrak{x}}(\theta,\varphi), r_{\mathfrak{y}}(\theta,\varphi), r_{\mathfrak{z}}(\theta,\varphi)$ are piecewise constant functions. An extensive explanation of this coordinate rescaling can be found in the next theorem. The short version is: these scalar functions correspond to the new length of the unit vectors $\mathfrak{x}', \mathfrak{y}', \mathfrak{z}'$ which are necessary to guarantee that the unit vectors end on a grid point. This results into the second transformation corresponding to spherical coordinates on a sub-grid of $\mathfrak{V}$. This is similar to the 2D case, see figure 3.5.

a) Now we start at

$$\begin{aligned}
&I[f](\mathfrak{v}) = \int_{\mathbb{R}^3}\int_{S^2}[f(\mathfrak{v}')f(\mathfrak{w}') - f(\mathfrak{v})f(\mathfrak{w})]k(\mathfrak{v}-\mathfrak{w},\eta)\mathrm{d}\eta\mathrm{d}\mathfrak{w}\,,\\
&\mathfrak{v}' := \mathfrak{v} + \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\eta\rangle\eta, \qquad \mathfrak{w}' := \mathfrak{w} - \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\eta\rangle\eta\,,
\end{aligned}$$

b) and the first Transformation remains simple:

$$\begin{aligned}
&\Phi\left(\{1\}\times[0,2\pi]\times\left[-\frac{\pi}{2},\frac{\pi}{2}\right]\right) = S^2, \quad \Phi\begin{pmatrix}1\\ \theta\\ \varphi\end{pmatrix} = \omega(\theta,\varphi)\\
&= 1\begin{pmatrix}\cos(\varphi)\cos(\theta)\\ \cos(\varphi)\sin(\theta)\\ \sin(\varphi)\end{pmatrix}(=\eta), \qquad \det(D\Phi) = \cos(\varphi)\,,\\
&\Longrightarrow \mathfrak{v}' = \mathfrak{v} + \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\omega(\theta,\varphi)\rangle\omega(\theta,\varphi), \qquad \mathfrak{w}' = \mathfrak{w} - \langle\overrightarrow{\mathfrak{v}\mathfrak{w}},\omega(\theta,\varphi)\rangle\omega(\theta,\varphi)\\
&\Longrightarrow I[f](\mathfrak{v}) = \int_{\mathbb{R}^3}\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}[f(\mathfrak{v}')f(\mathfrak{w}') - f(\mathfrak{v})f(\mathfrak{w})]\cos(\varphi)k(\mathfrak{v}-\mathfrak{w},\omega(\theta,\varphi))\mathrm{d}\varphi\mathrm{d}\theta\mathrm{d}\mathfrak{w}.
\end{aligned}$$

c) Whereas the second transformation becomes a bit painful due to the necessary extra rotations depending on θ, φ:

$$\Phi\left(\mathbb{R}^+_{\{0\}} \times [0, 2\pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right) = \mathbb{R}^3, \quad \Phi\begin{pmatrix} r \\ \lambda \\ \psi \end{pmatrix} = r\omega_2(\theta, \varphi, \lambda, \psi) = \overrightarrow{\mathfrak{v}\mathfrak{w}},$$

$$\det(D\Phi) = r^2 r_{\mathfrak{x}}(\theta, \varphi) r_{\mathfrak{y}}(\theta, \varphi) r_{\mathfrak{z}}(\theta, \varphi) \cos(\psi)\,,$$

$$D(r, \theta, \varphi, \psi) := r^2 r_{\mathfrak{x}}(\theta, \varphi) r_{\mathfrak{y}}(\theta, \varphi) r_{\mathfrak{z}}(\theta, \varphi) \cos(\psi) \cos(\varphi)\,,$$

$$\Longrightarrow \mathfrak{w} = \overbrace{\mathfrak{v} + r\omega_2(\theta, \varphi, \lambda, \psi)}^{\mathfrak{w}_2:=}, \qquad \mathfrak{v}' = \overbrace{\mathfrak{v} + r\langle\omega_2(\theta, \varphi, \lambda, \psi), \omega(\theta, \varphi)\rangle\omega(\theta, \varphi)}^{\mathfrak{v}_2':=}$$

$$\Longrightarrow \mathfrak{w}' = \overbrace{\mathfrak{w}_2 - r\langle\omega_2(\theta, \varphi, \lambda, \psi), \omega(\theta, \varphi)\rangle\omega(\theta, \varphi)}^{\mathfrak{w}_2':=}$$

$$\Longrightarrow I[f](\mathfrak{v}) =$$

$$\int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^L [f(\mathfrak{v}_2') f(\mathfrak{w}_2') - f(\mathfrak{v}) f(\mathfrak{w}_2)] k(r\omega_2, \omega) D(\cdots)\, \mathrm{d}r \mathrm{d}\psi \mathrm{d}\lambda \mathrm{d}\varphi \mathrm{d}\theta$$

□

Remark 3.3.10
The applied transformations and the usage of the rotation matrices may look artificially complicated. In the following lemmas and theorems we can see that the rotation matrix representation can be used to apply a simplification similar to Euler, Tait-Bryan angles respectively extrinsic and intrinsic two dimensional rotations. This simplification pans out to be very helpful to apply an elegant automorphism group based completion of the following discretization. The last transformation within the rotated coordinate system seems to be essential to obtain the minimal symmetry properties of the DVM operator $A^{\bullet;\bullet}_{\bullet,\bullet}$ in the end of this section. Moreover this Ansatz can be seen as a direct generalization of the two dimensional case.

Theorem 3.3.11 (Farey discretization of the collision operator)
Let $f \in C^s(\mathbb{R}^3 \to \mathbb{R}), k \in C^s(\mathbb{R}^3 \times S^2 \to \mathbb{R})$ and G be the automorphism group corresponding to the uniform grid. The transformed collision operator

$$I[f](\mathfrak{v}) = \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^{2\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^L h(\theta, \varphi, \lambda, \psi, r) D(r, \theta, \varphi, \psi)\, \mathrm{d}r \mathrm{d}\varphi \mathrm{d}\theta \mathrm{d}\psi \mathrm{d}\lambda$$

with

$$\begin{aligned} h(\theta, \varphi, \lambda, \psi, r) &:= [f(\mathfrak{v}_2') f(\mathfrak{w}_2') - f(\mathfrak{v}) f(\mathfrak{w}_2)] k(r\omega_2, \omega) \\ D(r, \theta, \varphi, \psi) &:= r^2 r_{\mathfrak{x}}(\theta, \varphi) r_{\mathfrak{y}}(\theta, \varphi) r_{\mathfrak{z}}(\theta, \varphi) \cos(\varphi) \cos(\psi)\,, \end{aligned}$$

can be approximated by

$$\tilde{I}[f](\mathfrak{v}) = \sum_{(i,k)\in B} \sum_{(j,l)\in C_{i,k}} \alpha_{ijkl} \Delta v_{ijkl} \sum_{q=0}^{\lfloor L/\Delta v_{ijkl} \rfloor} \sum_{\alpha,\beta \in G} \begin{bmatrix} g(q) h(\theta_i, \varphi_{i,j}, \lambda_k, \psi_{k,l}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_i, \varphi_{i,j}, \psi_{k,l}) \end{bmatrix}$$

with

$$\begin{aligned}
\mathfrak{w}_2(i,j,k,l,q,\alpha,\beta) &:= \mathfrak{v} + q\Delta v\big(\alpha[\mathfrak{x}',\mathfrak{y}',\mathfrak{z}']\big)\beta P_{k,l}\,,\\
\mathfrak{v}_2'(i,j,k,l,q,\alpha,\beta) &:= \mathfrak{v} + q\Delta v(\beta P_{k,l})_1\alpha P_{i,j}\,,\\
\mathfrak{w}_2'(i,j,k,l,q,\alpha,\beta) &:= \mathfrak{v} + \mathfrak{w}_2 - \mathfrak{v}_2'\,,\\
P_{a,b} &:= t_{a,b}\begin{pmatrix} q_a \\ p_a \\ \frac{p_{a,b}}{q_{a,b}}q_a \end{pmatrix} = r_{a,b}\omega(\theta_a,\varphi_{a,b}), \quad r_{a,b} = \|P_{a,b}\|, \text{ for } t_{a,b} \text{ see } 3.3.2\,,
\end{aligned}$$

$$\mathfrak{x}' := P_{i,j},\ \mathfrak{y}' := t_{i,j}\begin{pmatrix} -p_i \\ q_i \\ 0 \end{pmatrix},\ \mathfrak{z}' := t_{i,j}\begin{pmatrix} -q_i^2\frac{p_{i,j}}{q_{i,j}} \\ -p_iq_i\frac{p_{i,j}}{q_{i,j}} \\ q_i^2+p_i^2 \end{pmatrix},$$

$$\begin{aligned}
r_{\mathfrak{x}} := \|\mathfrak{x}'\|, \quad r_{\mathfrak{y}} &:= \|\mathfrak{y}'\|, \quad r_{\mathfrak{z}} := \|\mathfrak{z}'\|\,,\\
B := N\times M, \quad C_{i,k} &:= N_i \times M_k, \quad \alpha_{ijkl} := \alpha_i\alpha_k\alpha_{i,j}\alpha_{k,l}\,,\\
\Delta v_{ijkl} &:= \Delta v\|[\mathfrak{x}',\mathfrak{y}',\mathfrak{z}']P_{k,l}\|,\ r_q := q\cdot\Delta v_{k,l} = q\Delta v r_{k,l}\,.
\end{aligned}$$

This approximation yields an error bound of

$$\begin{aligned}
e\ &< 2^{s+1}3^{\frac{s}{2}+1}Lc(\Delta v)^s n^{2s}m^sK_r\\
&\quad + 576\pi^2\left(K_{\theta,\varphi,\lambda}\frac{\ln(m)+1}{m^2} + 2K_{\theta,\varphi,\lambda,\psi}\frac{1}{m}\right)\\
&\quad + 192\left(K_\theta\frac{\ln(n)+1}{n^2} + 2K_{\theta,\varphi}\frac{1}{n}\right)\,,
\end{aligned}$$

where $K_\theta, K_{\theta,\varphi}, K_{\theta,\varphi,\lambda}, K_{\theta,\varphi,\lambda,\psi}, K_r$ are some constants depending only on f and its derivatives and s, c correspond to the used Newton-Cotes formula, s being the error order and c corresponding to some error constants.

Proof:
Analog to the proof of theorem 3.2.3 we define the occurring integrals

$$\begin{aligned}
H_1(\theta,\varphi,\lambda,\psi) &:= \int_0^L h(\theta,\varphi,\lambda,\psi,r)r^2\mathrm{d}r\\
H_2(\theta,\varphi) &:= \int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H_1(\theta,\varphi,\lambda,\psi)\overbrace{r_{\mathfrak{x}}(\theta,\varphi)r_{\mathfrak{y}}(\theta,\varphi)r_{\mathfrak{z}}(\theta,\varphi)\cos(\psi)}^{\tilde{D}(\theta,\varphi,\psi):=}\mathrm{d}\psi\mathrm{d}\lambda\,,
\end{aligned}$$

and throughout this proof we define that objects possessing one index $\bullet_i$ correspond to the Farey sequence $\tilde{\mathfrak{F}}_n$, objects possessing two subscripts $\bullet_{ij}$ correspond to the Farey sequence $\tilde{\mathfrak{F}}_{i,n}$ (see 3.3.2). Now we take a closer look at the associated approximation errors:

$$e_3 := \left|\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H_2(\theta,\varphi)\cos(\varphi)\mathrm{d}\theta\mathrm{d}\varphi - \sum_{i=1}^{N}\sum_{j=1}^{N_i}\sum_{x=1}^{48}\alpha_i\alpha_{i,j}H_2(\Phi_x(\theta_i,\varphi_{i,j}))\cos(\varphi_{i,j})\right|\,,$$

$$
\begin{aligned}
&D_{i,j,k,l} := D(1, \theta_i, \varphi_{i,j}, \psi_{k,l})\,,\\
&e_2(i,j,x) := \left| H_2(\theta_i, \varphi_{i,j}) - \sum_{k=1}^{M} \sum_{l=1}^{M_i} \sum_{y=1}^{48} \alpha_k \alpha_{k,l} H_1(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l})) D_{i,j,k,l} \right|\,,\\
&e_2 := \sum_{i=1}^{N} \sum_{j=1}^{N_i} \sum_{x=1}^{48} \alpha_i \alpha_{i,j} e_2(i,j,x)\,,\\
&e_1 \begin{pmatrix} i,j,k, \\ l,x,y \end{pmatrix} := \left| H_1 \begin{pmatrix} \Phi_x(\theta_i, \varphi_{i,j}), \\ \Phi_y(\lambda_k, \psi_{k,l}) \end{pmatrix} - \Delta v_{ijkl} \sum_{q=1}^{\lfloor L/\Delta v_{ijkl} \rfloor} \left(h \begin{pmatrix} \Phi_x(\theta_i, \varphi_{i,j})\,, \\ \Phi_y(\lambda_k, \psi_{k,l}), r_q \end{pmatrix} g(q) r_q^2 \right) \right|\,,\\
&e_1 := \sum_{i=1}^{N} \sum_{j=1}^{N_i} \sum_{k=1}^{M} \sum_{l=1}^{M_i} \sum_{x,y=1}^{48} \alpha_i \alpha_{i,j} \alpha_k \alpha_{k,l} e_1(i,j,k,l,x,y)\,.
\end{aligned}
$$

Using 3.3.4 for $e_3, e_2(i,j,x)$ we get

$$
\begin{aligned}
e_3 \leq& 192 \left(2K_{\theta,\varphi} \frac{1}{n} + K_\theta \frac{\ln(n)+1}{n^2} \right)\,,\\
e_2(i,j,x) \leq& 192 \left(2K_{\theta,\varphi,\lambda,\psi} \frac{1}{m} + K_{\theta,\varphi,\lambda} \frac{\ln(m)+1}{m^2} \right) \qquad \Longrightarrow\\
e_2 =& \sum_{i=1}^{N} \sum_{j=1}^{N_i} \sum_{x=1}^{48} \alpha_i \alpha_{i,j} e_2(i,j,x)\\
\leq& \sum_{i=1}^{N} \sum_{j=1}^{N_i} \alpha_i \alpha_{i,j} 48 \cdot 192 \left(2K_{\theta,\varphi,\lambda,\psi} \frac{1}{m} + K_{\theta,\varphi,\lambda} \frac{\ln(m)+1}{m^2} \right)\\
\overset{\substack{3.1.4\\3.3.3}}{\leq}& \ \frac{\pi}{4} \frac{\pi}{4} 48 \cdot 192 \left(2K_{\theta,\varphi,\lambda,\psi} \frac{1}{m} + K_{\theta,\varphi,\lambda} \frac{\ln(m)+1}{m^2} \right)\\
=& \ 576\pi^2 \left(2K_{\theta,\varphi,\lambda,\psi} \frac{1}{m} + K_{\theta,\varphi,\lambda} \frac{\ln(m)+1}{m^2} \right)\,,\\
\text{with } K_\theta :=& \sup_{\theta \in [0,2\pi]} \left| \frac{\partial \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H_2(\theta,\varphi) \mathrm{d}\varphi}{\partial \theta} \right|\,,\\
K_{\theta,\varphi} :=& \sup_{\substack{\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \theta \in [0, 2\pi]}} \left| \frac{\partial H_2(\theta,\varphi)}{\partial \varphi} \right|\,,\\
K_{\theta,\varphi,\lambda} :=& \sup_{\substack{\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \theta, \lambda \in [0, 2\pi]}} \left| \frac{\partial \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} H_1(\theta,\varphi,\lambda,\psi) \cos(\varphi) D(\varphi,\lambda,\psi) \mathrm{d}\psi}{\partial \lambda} \right|\,,
\end{aligned}
$$

$$K_{\theta,\varphi,\lambda,\psi} := \sup_{\substack{\varphi,\,\psi\,\in\,[-\frac{\pi}{2},\,\frac{\pi}{2}]\\ \theta,\,\lambda\,\in\,[0,\,2\pi]}} \left|\frac{\partial H_1(\theta,\varphi,\lambda,\psi)\cos(\varphi)D(\varphi,\lambda,\psi)}{\partial\psi}\right| .$$

And to get a grip on the remaining error we need to take a closer look at the step size for the innermost integration. For this we need some identities following from the proof of 3.3.2 and the transformation 3.3.9 :

$$\begin{aligned}
&\mathfrak{w}_2 = \mathfrak{v} + r_q\omega(\lambda_k,\psi_{k,l}), \quad \mathfrak{v}_2' = \mathfrak{v} + r_q\langle\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}),\omega(\theta_i,\varphi_{i,j})\rangle\omega(\theta_i,\varphi_{i,j})\,,\\
&\mathfrak{w}_2' = \mathfrak{w}_2 - r_q\langle\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}),\omega(\theta_i,\varphi_{i,j})\rangle\omega(\theta_i,\varphi_{i,j})\,,\\
&P_{i,j} := \begin{pmatrix} t_{i,j}\begin{pmatrix} q_i\\ p_i\end{pmatrix}\\ \frac{t_{i,j}q_i}{q_{i,j}}p_{i,j}\end{pmatrix} = \|P_{i,j}\|\omega(\theta_i,\varphi_{i,j}), \quad r_{i,j} := \|P_{i,j}\| = t_{i,j}\sqrt{q_i^2+p_i^2+\frac{q_i^2}{q_{i,j}^2}p_{i,j}^2}\,,\\
&c_{i,j} := \frac{t_{i,j}q_i}{q_{i,j}} \in \mathbb{N}\,.
\end{aligned}$$

As we can see $P_{i,j}$ is the grid point corresponding to $\omega(\theta_i,\varphi_{i,j})$. It is a bit more complicated to obtain the gridpoint corresponding to ω_2. To simplify this we look at ω_2 as spherical coordinates within the rotated and rescaled coordinate system with the axis

$$\begin{aligned}
\mathfrak{x}' &:= r_{\mathfrak{x}}(\theta_i,\varphi_{i,j})\omega(\theta_i,\varphi_{i,j}) &&= t_{i,j}\begin{pmatrix} q_i\\ p_i\\ \frac{p_{i,j}}{q_{i,j}}q_i\end{pmatrix}, \quad r_{\mathfrak{x}}(\theta_i,\varphi_{i,j}) := \|\mathfrak{x}'\|\,,\\
\mathfrak{y}' &:= r_{\mathfrak{y}}(\theta_i,\varphi_{i,j})R_{\mathfrak{z}}(\theta_i)\mathfrak{y} &&= t_{i,j}\begin{pmatrix} -p_i\\ q_i\\ 0\end{pmatrix}, \quad r_{\mathfrak{y}}(\theta_i,\varphi_{i,j}) := \|\mathfrak{y}'\|\,,\\
\mathfrak{z}' &:= r_{\mathfrak{z}}(\theta_i,\varphi_{i,j})R_{\mathfrak{y}'}(-\varphi_{i,j})\mathfrak{z} &&= t_{i,j}\begin{pmatrix} -q_i^2\frac{p_{i,j}}{q_{i,j}}\\ -p_iq_i\frac{p_{i,j}}{q_{i,j}}\\ q_i^2+p_i^2\end{pmatrix}, \quad r_{\mathfrak{z}}(\theta_i,\varphi_{i,j}) := \|\mathfrak{z}'\|\,,
\end{aligned}$$

$$\text{where } \mathfrak{x} := (1,0,0)^T, \mathfrak{y} := (0,1,0)^T, \mathfrak{z} := (0,0,1)^T, \text{ for comparison see 3.3.9}\,.$$

These axis-vectors have the important feature that they end on grid points. The explicit representation of these vectors (explicit in the sense that we get a direct representation without angles θ,φ) can be obtained by a direct calculation of the rotations (using 3.3.7) and subsequent trigonometric transformations. Now we can "simply" write

$$\begin{aligned}
\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}) &= [\mathfrak{x}',\mathfrak{y}',\mathfrak{z}']\,\omega(\lambda_k,\psi_{k,l})\\
&\Longrightarrow P_{i,j,k,l} := r_{k,l}\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}) = [\mathfrak{x}',\mathfrak{y}',\mathfrak{z}']\,r_{k,l}\omega(\lambda_k,\psi_{k,l}) = [\mathfrak{x}',\mathfrak{y}',\mathfrak{z}']\,P_{k,l}
\end{aligned} \tag{3.3.7}$$

$$\begin{aligned}
&= t_{i,j}t_{k,l}\begin{pmatrix} q_kq_i - p_ip_k - \frac{p_{i,j}p_{k,l}}{q_{i,j}q_{k,l}}q_kq_i^2 \\ q_kp_i + p_kq_i - \frac{p_{i,j}p_{k,l}}{q_{i,j}q_{k,l}}q_kq_ip_i \\ \frac{p_{i,j}}{q_{i,j}}q_iq_k + \frac{p_{k,l}}{q_{k,l}}q_k(q_i^2+p_i^2) \end{pmatrix} \\
\Longrightarrow r_{i,j,k,l} := t_{i,j}t_{k,l}&\sqrt{\begin{bmatrix} \frac{p_{i,j}^2}{q_{i,j}^2}q_i^2q_k^2 + \frac{p_{k,l}^2}{q_{k,l}^2}p_i^4q_k^2 + 2\frac{p_{k,l}^2}{q_{k,l}^2}p_i^2q_i^2q_k^2 + \frac{p_{k,l}^2}{q_{k,l}^2}q_k^2q_i^4 + q_k^2p_i^2 + \\ p_k^2q_i^2 + \frac{p_{k,l}^2p_{i,j}^2}{q_{k,l}^2q_{i,j}^2}p_i^2q_i^2q_k^2 + q_k^2q_i^2 + p_i^2p_k^2 + \frac{p_{i,j}^2p_{k,l}^2}{q_{i,j}^2q_{k,l}^2}q_i^4q_k^2 \end{bmatrix}} \\
&\Longrightarrow \langle \omega(\theta_i,\varphi_{i,j}), P_{i,j,k,l}\rangle = \langle \omega(\theta_i,\varphi_{i,j}), t_{k,l}q_k\mathfrak{x}'\rangle = t_{k,l}q_k\|\mathfrak{x}'\| = t_{k,l}q_kr_{i,j}, \qquad (3.3.8) \\
&\qquad \text{because } \mathfrak{x}'\perp\mathfrak{y}'\perp\mathfrak{z}' \wedge \mathfrak{x}' \parallel \omega(\theta_i,\varphi_{i,j})\,.
\end{aligned}$$

Additionally we want that the velocities $\mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2'$ lie on our grid, so our first guess for r_q is:

$$r_q := q\cdot\Delta v\cdot r_{k,l} = q\Delta v_{k,l}.$$

So let us take a look if this results into $\mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2'$ lying on the grid:

$$\begin{aligned}
\mathfrak{w}_2 &= \mathfrak{v} + r_q\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}) = \mathfrak{v} + q\Delta v r_{k,l}\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}) && (3.3.9)\\
&= \mathfrak{v} + q\Delta v P_{i,j,k,l} \in \mathfrak{V},\ \text{by (3.3.7)}\,, && (3.3.10)\\
\Longrightarrow \mathfrak{v}_2' &= \mathfrak{v} + q\Delta v r_{k,l}\langle\omega(\theta_i,\varphi_{i,j}),\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l})\rangle\omega(\theta_i,\varphi_{i,j}) && (3.3.11)\\
&= \mathfrak{v} + q\Delta v t_{k,l}q_kP_{i,j} \in \mathfrak{V} \text{ by (3.3.8)} && (3.3.12)\\
\Longrightarrow \mathfrak{w}_2' &= \mathfrak{v} + r_q\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}) - r_q\langle\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l}),\omega(\theta_i,\varphi_{i,j})\rangle\omega(\theta_i,\varphi_{i,j}) && (3.3.13)\\
&= \mathfrak{v} + q\Delta v P_{i,j,k,l} - q\Delta v t_{k,l}q_kP_{i,j} \in \mathfrak{V}\,. && (3.3.14)
\end{aligned}$$

To obtain the step size for the Newton-Cotes formula we have to remember that the step size corresponds to

$$\Delta v_{ijkl} = \|\mathfrak{w}_2 - \mathfrak{v}\| = \Delta v\|P_{i,j,k,l}\| = \Delta v r_{i,j,k,l}\,.$$

Now we define

$$\tilde{K}_r(i,j,k,l) := \max_{\|\mathfrak{w}_2-\mathfrak{v}\|\in[0,L]}\left|\frac{\partial^s h(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l},r)r^2r_{\mathfrak{x}}r_{\mathfrak{y}}r_{\mathfrak{z}}\cos(\varphi_{i,j})\cos(\psi_{k,l})}{\partial\|\mathfrak{w}_2-\mathfrak{v}\|^s}\right|,$$

and here we can see that the actual calculation of this derivative becomes very complicated, because the step length $r_{i,j,k,l} = \mathfrak{w} - \mathfrak{v}$ is only implicitly given due to $|\omega_2| \neq 1$ in most cases. Nonetheless this problem can be easily tackled by calculating $\lambda_k, \psi_{k,l}$ according to $P_{i,j,k,l}$, (3.3.1), (3.3.2) and replacing $\omega_2(\theta_i,\varphi_{i,j},\lambda_k,\psi_{k,l})$ by $\omega(\lambda_{i,j,k,l},\psi_{i,j,k,l})$ as well as replacing the last transformation in 3.3.9 with a transformation into spherical coordinates. This would result in an explicit representation of the step size within

$\mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2'$ allowing the calculation of the above derivative. We suppress this calculation and alternative representations, because this approach results into even more trigonometric calculations without further insight and we do not need this representation nor do we "really" calculate this derivative in the following work. An interested reader can do this by using a CAS. Using a composite, closed Newton-Cotes formula with an error order of s leads (analog to 3.2.3) to

$$\begin{aligned}
e_1(i,j,k,l) &= \lceil L/\Delta v_{ijkl} \rceil \Delta v_{ijkl} c \Delta v_{ijkl}^s \tilde{K}_r(i,j,k,l) \\
&< \frac{3}{2} Lc(\Delta v)^s r_{i,j,k,l}^s \tilde{K}_r \\
&= \frac{3}{2} Lc(\Delta v)^s t_{i,j}^s t_{k,l}^s \left[\begin{array}{l} \frac{p_{i,j}^2}{q_{i,j}^2} q_i^2 q_k^2 + \frac{p_{k,l}^2}{q_{k,l}^2} p_i^4 q_k^2 + 2\frac{p_{k,l}^2}{q_{k,l}^2} p_i^2 q_i^2 q_k^2 + \\ \frac{p_{k,l}^2}{q_{k,l}^2} q_k^2 q_i^4 + q_k^2 p_i^2 + p_k^2 q_i^2 + \frac{p_{k,l}^2 p_{i,j}^2}{q_{k,l}^2 q_{i,j}^2} p_i^2 q_i^2 q_k^2 + \\ q_k^2 q_i^2 + p_i^2 p_k^2 + \frac{p_{i,j}^2 p_{k,l}^2}{q_{i,j}^2 q_{k,l}^2} q_i^4 q_k^2 \end{array} \right]^{\frac{s}{2}} \tilde{K}_r \\
&\leq \frac{3}{2} Lc(\Delta v)^s \left\lfloor \frac{n}{q_i} \right\rfloor^s \left\lfloor \frac{m}{q_k} \right\rfloor^s \left[\begin{array}{l} q_i^2 q_k^2 + q_i^4 q_k^2 + 2q_i^4 q_k^2 + \\ q_k^2 q_i^4 + q_k^2 q_i^2 + q_k^2 q_i^2 + q_i^4 q_k^2 + \\ q_k^2 q_i^2 + q_i^2 q_k^2 + q_i^4 q_k^2 \end{array} \right]^{\frac{s}{2}} \tilde{K}_r \\
&= \frac{3}{2} Lc(\Delta v)^s \left\lfloor \frac{n}{q_i} \right\rfloor^s \left\lfloor \frac{m}{q_k} \right\rfloor^s \left(5q_i^2 q_k^2 + 6q_i^4 q_k^2\right)^{\frac{s}{2}} \tilde{K}_r \\
\implies e_1 &= \sum_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} \alpha_i \alpha_{i,j} \alpha_k \alpha_{k,l} \frac{3}{2} Lc(\Delta v)^s \left\lfloor \frac{n}{q_i} \right\rfloor^s \left\lfloor \frac{m}{q_k} \right\rfloor^s \left(5q_i^2 q_k^2 + 6q_i^4 q_k^2\right)^{\frac{s}{2}} \tilde{K}_r \\
&\overset{\substack{\text{3.1.4(iii)} \\ \text{3.3.3}}}{<} \sum_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} \frac{4\left(mq_k \left\lfloor \frac{m}{q_k} \right\rfloor \sqrt{q_k^2 + p_k^2}\right)^{-1}}{nq_i \left\lfloor \frac{n}{q_i} \right\rfloor \sqrt{q_i^2 + p_i^2}} \frac{3}{2} Lc(\Delta v)^s \left\lfloor \frac{n}{q_i} \right\rfloor^s \left\lfloor \frac{m}{q_k} \right\rfloor^s \begin{pmatrix} 5q_i^2 q_k^2 + \\ 6q_i^4 q_k^2 \end{pmatrix}^{\frac{s}{2}} \tilde{K}_r \\
&\leq \sum_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} 6\frac{1}{nq_i^2 \left\lfloor \frac{n}{q_i} \right\rfloor \left\lfloor \frac{m}{q_k} \right\rfloor mq_k^2} Lc(\Delta v)^s \left\lfloor \frac{n}{q_i} \right\rfloor^s \left\lfloor \frac{m}{q_k} \right\rfloor^s \begin{pmatrix} 5q_i^2 q_k^2 + \\ 6q_i^4 q_k^2 \end{pmatrix}^{\frac{s}{2}} \tilde{K}_r \\
&= \sum_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} 6\frac{(5 + 6q_i^2)^{\frac{s}{2}}}{nq_i mq_k} Lc(\Delta v)^s \left(q_i^{s-1} \left\lfloor \frac{n}{q_i} \right\rfloor^{s-1} \right) \left(q_k^{s-1} \left\lfloor \frac{m}{q_k} \right\rfloor^{s-1} \right) \tilde{K}_r \\
&\leq \sum_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} 6\frac{(5 + 6q_i^2)^{\frac{s}{2}}}{q_i q_k} Lc(\Delta v)^s n^{s-2} m^{s-2} \tilde{K}_r \\
&\leq \sum_{(i,k)\in B} 6\frac{(5 + 6q_i^2)^{\frac{s}{2}}}{q_i q_k} Lc(\Delta v)^s n^{s-1} m^{s-1} \overbrace{\max_{\substack{(i,\,k)\,\in\,B \\ (j,\,l)\,\in\,C_{ik}}} (\tilde{K}_r(i,j,k,l))}^{K_r :=}
\end{aligned}$$

$$
\begin{aligned}
&= 6Lc(\Delta v)^s n^{s-1} m^{s-1} K_r \sum_{i=1}^{N} \frac{(5+6q_i^2)^{\frac{s}{2}}}{q_i} \sum_{k=1}^{M} \frac{1}{q_k} \\
&= 6Lc(\Delta v)^s n^{s-1} m^{s-1} K_r \sum_{q=1}^{n} \sum_{p=1}^{q} \frac{(5+6q^2)^{\frac{s}{2}}}{q} \sum_{\tilde{q}=1}^{m} \sum_{\tilde{p}=1}^{\tilde{q}} \frac{1}{\tilde{q}} \\
&= \overbrace{6Lc(\Delta v)^s n^{s-1} m^s K_r}^{\tilde{c}:=} \sum_{q=1}^{n} (5+6q^2)^{\frac{s}{2}} < 6^{\frac{s}{2}} \tilde{c} \sum_{q=1}^{n} \sqrt{(q^2+1)^s} \\
&= 6^{\frac{s}{2}} \tilde{c} \sum_{q=1}^{n} \sqrt{\sum_{k=0}^{s} \binom{s}{k} (q^2)^{s-k}} < 6^{\frac{s}{2}} \tilde{c} \sum_{q=1}^{n} \sqrt{2^s q^{2s}} < 6^{\frac{s}{2}} \tilde{c}\ 2^{\frac{s}{2}} n^{s+1} \\
&= 6^{\frac{s}{2}+1} 2^{\frac{s}{2}} Lc(\Delta v)^s n^{2s} m^s K_r = 2^{s+1} 3^{\frac{s}{2}+1} Lc(\Delta v)^s n^{2s} m^s K_r
\end{aligned}
$$

Adding the three errors together we arrive at

$$
\begin{aligned}
e = e_1 + e_2 + e_3 = {} & 2^{s+1} 3^{\frac{s}{2}+1} Lc(\Delta v)^s n^{2s} m^s K_r \\
& + 576\pi^2 \left(K_{\theta,\varphi,\lambda} \frac{\ln(m)+1}{m^2} + 2K_{\theta,\varphi,\lambda,\psi} \frac{1}{m} \right) \\
& + 192 \left(K_\theta \frac{\ln(n)+1}{n^2} + 2K_{\theta,\varphi} \frac{1}{n} \right) .
\end{aligned}
$$

The last thing we want to achieve is to get rid of the coordinate transformations $\Phi_\bullet$ in favor of the corresponding matrices $M_\bullet \in G$, see proof of 3.3.4. So we take a closer look at $h(\theta, \varphi, \lambda, \psi, r)$ and use its structure together with alternative representations of $\mathfrak{v}_2, \mathfrak{w}_2, \mathfrak{w}_2'$.
Looking at ω,ω_2 from 3.3.9 using 3.3.6, 3.3.7 we can see that

$$
\begin{aligned}
&\omega(\theta,\varphi) = R_{R_{\mathfrak{z}}(\theta)\mathfrak{y}}(-\varphi) R_{\mathfrak{z}}(\theta)\mathfrak{x} \text{ corresponds to } \mathfrak{x} \xrightarrow{R_{\mathfrak{z}}(\theta)} \tilde{\mathfrak{x}} \xrightarrow{R_{\mathfrak{z}}(\theta) R_{\mathfrak{y}}(-\varphi) R_{\mathfrak{z}}(\theta)^{-1}} \mathfrak{x}' = \omega(\theta,\varphi) \\
&\Longrightarrow \omega(\theta,\varphi) = R_{\mathfrak{z}}(\theta) R_{\mathfrak{y}}(-\varphi)\mathfrak{x} \ .
\end{aligned}
$$

As we have seen in the proof of 3.3.4 the transformations $\Phi_\bullet$ correspond to permutations $M_\bullet$ of the coordinates after the rotation, so we obtain

$$
\begin{aligned}
&\omega(\Phi_x(\theta,\varphi)) = M_x \omega(\theta,\varphi) = M_x R_{R_{\mathfrak{z}}(\theta)\mathfrak{y}}(-\varphi) R_{\mathfrak{z}}(\theta)\mathfrak{x} \text{ corresponds to} \\
&\mathfrak{x} \xrightarrow{R_{\mathfrak{z}}(\theta)} \tilde{\mathfrak{x}} \xrightarrow{R_{\mathfrak{z}}(\theta) R_{\mathfrak{y}}(-\varphi) R_{\mathfrak{z}}(\theta)^{-1}} \mathfrak{x}' = \omega(\theta,\varphi) \xrightarrow{M_x} M_x \omega(\theta,\varphi) \\
&\quad \Longrightarrow M_x \omega(\theta,\varphi) = M_x R_{\mathfrak{z}}(\theta) R_{\mathfrak{y}}(-\varphi)\mathfrak{x} =: \omega(\theta,\varphi,M_x) \ .
\end{aligned}
$$

These representations via intrinsic and extrinsic rotations seem to be obvious but this becomes more complicated when we look at ω_2. At this point we begin with the diagram obtained through 3.3.9 and 3.3.4:

$$
\begin{aligned}
&D := \operatorname{diag}(r_{\mathfrak{x}}, r_{\mathfrak{y}}, r_{\mathfrak{z}})\,,\\
&\mathfrak{x} \xrightarrow{R_{\mathfrak{z}}(\theta)} \mathfrak{x}_2 \xrightarrow{R_{\mathfrak{z}}(\theta_i)R_{\mathfrak{y}}(-\varphi_{i,j})R_{\mathfrak{z}}(\theta_i)^{-1}} \mathfrak{x}_3 \xrightarrow{R_{\mathfrak{z}}(\theta_i)R_{\mathfrak{y}}(-\varphi_{i,j})D(R_{\mathfrak{z}}(\theta_i)R_{\mathfrak{y}}(-\varphi_{i,j}))^{-1}} P_{i,j}\\
&P_{i,j} \xrightarrow{M_x} \mathfrak{x}_4 = \overbrace{M_x R_{\mathfrak{z}}(\theta_i) R_{\mathfrak{y}}(-\varphi_{i,j}) D}^{B:=}\mathfrak{x}_1 \xrightarrow{BR_{\mathfrak{z}}(\lambda_k)B^{-1}} \mathfrak{x}_5\\
&\mathfrak{x}_5 \xrightarrow{BR_{\mathfrak{z}}(\lambda_k)R_{\mathfrak{y}}(-\psi_{k,l})(BR_{\mathfrak{z}}(\lambda_k))^{-1}} \mathfrak{x}_6 \xrightarrow{BM_yB^{-1}} \mathfrak{x}_7 = \omega_2(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}))\\
\Longrightarrow\ &\mathfrak{x}_7 = BM_yB^{-1}\mathfrak{x}_6\\
&= BM_yB^{-1}BR_{\mathfrak{z}}(\lambda_k)R_{\mathfrak{y}}(-\psi_{k,l})[BR_{\mathfrak{z}}(\lambda_k)]^{-1}\mathfrak{x}_5\\
&= BM_yB^{-1}BR_{\mathfrak{z}}(\lambda_k)R_{\mathfrak{y}}(-\psi_{k,l})[BR_{\mathfrak{z}}(\lambda_k)]^{-1}BR_{\mathfrak{z}}(\lambda_k)B^{-1}B\mathfrak{x}_1\\
&= BM_yR_{\mathfrak{z}}(\lambda_k)R_{\mathfrak{y}}(-\psi_{k,l})\mathfrak{x}_1\\
&= M_xR_{\mathfrak{z}}(\theta_i)R_{\mathfrak{y}}(-\varphi_{i,j})DM_yR_{\mathfrak{z}}(\lambda_k)R_{\mathfrak{y}}(-\psi_{k,l})\mathfrak{x}_1\\
&=: \omega_2(\theta, \varphi, \lambda, \psi, M_x, M_y) = \omega_2(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}))\,.
\end{aligned}
$$

Now we have to argue why $r_{k,l}\omega_2, r_{k,l}\langle\omega,\ \omega_2\rangle\omega$ should lie on the grid. We know that the transformations to $P_{i,j}$ correspond to a transformation from the original grid to the sub grid given by the axis $\mathfrak{x}', \mathfrak{y}', \mathfrak{z}'$, where the distances between neighboring grid points on these axis are given by $r_{\mathfrak{x}}, r_{\mathfrak{y}}, r_{\mathfrak{z}}$. The transformation $P_{i,j} \to \mathfrak{x}_4$ is a simple permutation of the coordinates (with possible sign change). The transformations from $\mathfrak{x}_4 \longrightarrow \mathfrak{x}_7$ correspond to the same transformations used in $\mathfrak{x}_1 \longrightarrow \mathfrak{x}_4$ considering that we use the base transformation B to do this within the established sub grid. Multiplying the result with $r_{k,l}$ gives a point on the sub grid established by $\mathfrak{x}', \mathfrak{y}', \mathfrak{z}'$. The last thing we need to clarify is the implication of these transformations M_x, M_y for the scalar product (3.3.8), $\langle\omega, r_{k,l}\omega_2\rangle$. We know that M_x corresponds to a permutation of the old coordinate axis (with possible sign changes) and that M_y corresponds to a permutation of the new coordinate axis (with possible sign changes). This can be used to obtain an alternative representation of ω_2:

$$
\begin{aligned}
&\omega_2(\theta_i, \varphi_{i,j}, \lambda_k, \psi_{k,l}, M_x, M_y) = \big(M_x[\mathfrak{x}', \mathfrak{y}', \mathfrak{z}']\big)M_y\omega(\lambda_k, \psi_{k,l})\\
\Longrightarrow &\langle\omega(\theta_i, \varphi_{i,j}, M_x), r_{k,l}\omega_2(\theta_i, \varphi_{i,j}, \lambda_k, \psi_{k,l}, M_x, M_y)\rangle\\
&= \Big\langle M_x\omega(\theta_i, \varphi_{i,j}), \big(M_x[\mathfrak{x}', \mathfrak{y}', \mathfrak{z}']\big)M_y r_{k,l}\omega(\lambda_k, \psi_{k,l})\Big\rangle\\
&= \Big\langle M_x\omega(\theta_i, \varphi_{i,j}), M_x\mathfrak{x}'(M_yP_{k,l})_1\Big\rangle\\
&= \|\mathfrak{x}'\| \cdot (M_yP_{k,l})_1 = r_{i,j}(M_yP_{k,l})_1\,.
\end{aligned}
$$

This gives a representation for $\mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2'$ incorporating the coordinate transformations Φ_x, Φ_y:

$$\mathfrak{w}_2(i, j, k, l, q, x, y) = \mathfrak{v} + q\Delta v r_{k,l}\omega_2(\theta_i, \varphi_{i,j}, \lambda_k, \psi_{k,l}, M_x, M_y)$$

$$\begin{aligned}
&= \mathfrak{v} + q\Delta v\big(M_x[\mathfrak{x}', \mathfrak{y}', \mathfrak{z}']\big)M_y P_{k,l} \in \mathfrak{V} \\
&= \mathfrak{w}_2(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}), r_q) \\
\mathfrak{v}_2'(i,j,k,l,q,x,y) &= \mathfrak{v} + q\Delta v r_{k,l} \left\langle \omega(\theta_i, \varphi_{i,j}, M_x), \omega_2 \left[\begin{array}{c} \theta_i, \varphi_{i,j}, \\ \lambda_k, \psi_{k,l}, \\ M_x, M_y \end{array} \right] \right\rangle \omega(\theta_i, \varphi_{i,j}, M_x) \\
&= \mathfrak{v} + q\Delta v(M_y P_{k,l})_1 M_x P_{i,j} \in \mathfrak{V} \\
&= \mathfrak{v}_2'(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}), r_q) \\
\mathfrak{w}_2'(i,j,k,l,q,x,y) &= \mathfrak{v} + \mathfrak{w}_2 - \mathfrak{v}_2' = \mathfrak{w}_2'(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}), r_q)\ .
\end{aligned}$$

The application of 3.3.4 together with the above results gives the final result regarding the form of the discretization. So last but not least we can rewrite h:

$$h(\Phi_x(\theta_i, \varphi_{i,j}), \Phi_y(\lambda_k, \psi_{k,l}), r_q) = h(\theta_i, \varphi_{i,j}, \lambda_k, \psi_{k,l}, M_x, M_y)\,. \qquad \square$$

Theorem 3.3.12 (Convergence order in terms of Δv)
Assuming that $\Delta v \in \mathbb{R}^+, s \in \mathbb{N}$ are given constants satisfying

$$\frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \frac{\pi^{\frac{2}{3}}}{2^{\frac{2}{3}}} \left(\frac{\pi^{\frac{4}{3}} 3^3 2^{\frac{32}{3}}}{L^2 c^2}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \frac{1}{2^3 3^{\frac{3}{2}} \pi^{\frac{2}{3}}} \left(\frac{2^{18} 3^{\frac{11}{2}} \pi^{\frac{8}{3}}}{L^3 c^3}\right)^{\frac{1}{3s+1}}} \geq s\,.$$

and choosing n, m according to

$$n = \lceil n(\Delta v) \rceil = \left\lceil \frac{1}{3^{\frac{1}{2}} \pi^{\frac{2}{3}} 2^{\frac{1}{3}}} \left(\frac{3^{\frac{3}{2}} \pi^{\frac{2}{3}} 2^{5+\frac{1}{3}}}{Lc}\right)^{\frac{1}{3s+1}} (\Delta v)^{-\frac{s}{3s+1}} \right\rceil, \quad m = \left\lceil 3\pi^2 n(\Delta v) \right\rceil\,,$$

the convergence order of the discretization 3.3.11 is

$$|I[f](\mathfrak{v})\tilde{I} - [f](\mathfrak{v})| < e(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right)\,.$$

As in the two dimensional case 3.2.4, the main conclusion is that if n, m grow sufficiently slow with and in comparison to $\frac{1}{\Delta v}$ than the whole approximation converges with the order given above. The first requirement ensures that at least s points lie on every line that is used for the innermost integration. A lower bound for the minimal number of points $\tilde{s}$ on any given line is

$$\tilde{s} \geq \frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \frac{\pi^{\frac{2}{3}}}{2^{\frac{2}{3}}} \left(\frac{\pi^{\frac{4}{3}} 3^3 2^{\frac{32}{3}}}{L^2 c^2}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \frac{1}{2^3 3^{\frac{3}{2}} \pi^{\frac{2}{3}}} \left(\frac{2^{18} 3^{\frac{11}{2}} \pi^{\frac{8}{3}}}{L^3 c^3}\right)^{\frac{1}{3s+1}}} \in \mathcal{O}\big((\Delta v)^{-\frac{1}{3s+1}}\big)\,.$$

Proof:
To reduce the anxiety of another proof similar painful as 3.2.4: this one is way more

understandable, because we loose nothing in using the estimate $\frac{\ln(n)}{n^2} < \frac{1}{n}$. Starting with a simplification of the error e:

$$
\begin{aligned}
e &= 2^{s+1}3^{\frac{s}{2}+1}Lc(\Delta v)^s n^{2s} m^s K_r + 576\pi^2 \left(K_{\theta,\varphi,\lambda}\frac{\ln(m)+1}{m^2} + 2K_{\theta,\varphi,\lambda,\psi}\frac{1}{m}\right)\\
&+ 192\left(K_\theta \frac{\ln(n)+1}{n^2} + 2K_{\theta,\varphi}\frac{1}{n}\right)\\
&\le 2^{s+1}3^{\frac{s}{2}+1}Lc(\Delta v)^s \cdot n^{2s} m^s \cdot K_r + 576\pi^2 \left(K_{\theta,\varphi,\lambda} + 2K_{\theta,\varphi,\lambda,\psi}\right)\frac{1}{m}\\
&+ 192(K_\theta + 2K_{\theta,\varphi})\frac{1}{n}\\
&\le K\left[\overbrace{2^{s+1}3^{\frac{s}{2}+1}Lc}^{c_3:=}(\Delta v)^s \cdot n^{2s} m^s + \overbrace{1728\pi^2}^{c_2:=}\frac{1}{m} + \overbrace{576}^{c_1:=}\frac{1}{n}\right],
\end{aligned}
$$

with $K = \max\{K_\theta, K_{\theta,\varphi}, K_\lambda, K_{\lambda,\psi}, K_l\}$. Here we use the same ansatz as in 3.2.4. Despite the fact that we ignore the better convergence of one of the integrals in spherical coordinates, we force the three integrations (two over a sphere and one along a line) to deliver approximately the same error, only disturbed by the differences between the $K_\bullet$:

$$c_1\frac{1}{n} = c_2\frac{1}{m} = c_3(\Delta v)^s n^{2s} m^s\,.$$

Now everything becomes very simple:

$$
\begin{aligned}
&c_1\frac{1}{n} = c_2\frac{1}{m} \iff m(n) = \frac{c_2}{c_1}n\,,\\
\Longrightarrow &c_1\frac{1}{n} = c_3(\Delta v)^s n^{2s} m^s = c_3(\Delta v)^s n^{2s}\frac{c_2^s}{c_1^s}n^s = c_3\frac{c_2^s}{c_1^s}(\Delta v)^s n^{3s}\\
\iff &n^{3s+1} = \frac{c_1^{s+1}}{c_3 c_2^s}(\Delta v)^{-s} \iff n = \left(\frac{c_1^{s+1}}{c_3 c_2^s}\right)^{\frac{1}{3s+1}}(\Delta v)^{-\frac{s}{3s+1}}\\
\iff &\frac{1}{n} = \left(\frac{c_1^{s+1}}{c_3 c_2^s}\right)^{-\frac{1}{3s+1}}(\Delta v)^{\frac{s}{3s+1}} \qquad\qquad \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right)\\
\Longrightarrow &m(\Delta v) = \frac{c_2}{c_1}\left(\frac{c_1^{s+1}}{c_3 c_2^s}\right)^{\frac{1}{3s+1}}(\Delta v)^{-\frac{s}{3s+1}} = \left(\frac{c_2^{2s+1}}{c_3 c_1^{2s}}\right)^{\frac{1}{3s+1}}(\Delta v)^{-\frac{s}{3s+1}}\\
\iff &\frac{1}{m} = \left(\frac{c_2^{2s+1}}{c_3 c_1^{2s}}\right)^{-\frac{1}{3s+1}}(\Delta v)^{\frac{s}{3s+1}} \qquad\qquad \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right)\\
\Longrightarrow &c_3(\Delta v)^s n^{2s} m^s = c_3(\Delta v)^s\left(\left(\frac{c_1^{s+1}}{c_3 c_2^s}\right)^{\frac{1}{3s+1}}(\Delta v)^{-\frac{s}{3s+1}}\right)^{2s}\left(\left(\frac{c_2^{2s+1}}{c_3 c_1^{2s}}\right)^{\frac{1}{3s+1}}(\Delta v)^{-\frac{s}{3s+1}}\right)^{s}\\
&= \text{const}\cdot(\Delta v)^s(\Delta v)^{-\frac{2s^2}{3s+1}}(\Delta v)^{-\frac{s^2}{3s+1}} = \text{const}\cdot(\Delta v)^{\frac{s(3s+1)}{3s+1}-\frac{2s^2}{3s+1}-\frac{s^2}{3s+1}}
\end{aligned}
$$

$$
\begin{aligned}
&= \text{const} \cdot (\Delta v)^{\frac{s}{3s+1}} \qquad\qquad \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right) \\
\Longrightarrow & K\left(c_1 \frac{1}{n} + c_2 \frac{1}{m} + c_3 (\Delta v)^s n^{2s} m^s\right) \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right) \\
\Longrightarrow & e \in \mathcal{O}\left((\Delta v)^{\frac{s}{3s+1}}\right) .
\end{aligned}
$$

The last thing we have to look at is how many points are at least on any line for the innermost integration. This determines what the largest s (order of the Newton-Cotes formula) can be. We know that the minimal number of points $\tilde{s}$ corresponds to

$$
\begin{aligned}
\forall i,j,k,l : \frac{L}{\Delta r_{i,j,k,l}} &= \frac{L}{\Delta v t_{i,j} t_{k,l} \sqrt{\begin{bmatrix} \frac{p_{i,j}^2}{q_{i,j}^2} q_i^2 q_k^2 + \frac{p_{k,l}^2}{q_{k,l}^2} p_i^4 q_k^2 + 2\frac{p_{k,l}^2}{q_{k,l}^2} p_i^2 q_i^2 q_k^2 + \\ \frac{p_{k,l}^2}{q_{k,l}^2} q_k^2 q_i^4 + q_k^2 p_i^2 + p_k^2 q_i^2 + \frac{p_{k,l}^2 p_{i,j}^2}{q_{k,l}^2 q_{i,j}^2} p_i^2 q_i^2 q_k^2 + \\ q_k^2 q_i^2 + p_i^2 p_k^2 + \frac{p_{i,j}^2 p_{k,l}^2}{q_{i,j}^2 q_{k,l}^2} q_i^4 q_k^2 \end{bmatrix}}} \\
&> \frac{L}{\Delta v \sqrt{\begin{bmatrix} n^2 m^2 + n^4 m^2 + 2n^2 m^2 + n^4 m^2 + n^2 m^2 + \\ n^2 m^2 + n^4 m^2 + n^2 m^2 + n^2 m^2 + n^4 m^2 \end{bmatrix}}} \\
&= \frac{L}{\Delta v \sqrt{7n^2 m^2 + 4n^4 m^2}} > \frac{L}{\Delta v(\sqrt{7} nm + 2n^2 m)} \\
&= \frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \left(\frac{c_2^{s+1}}{c_3^2 c_1^{s-1}}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \left(\frac{c_1^2}{c_3^3 c_2^{s-1}}\right)^{\frac{1}{3s+1}}} \\
&= \frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \frac{\pi^{\frac{2}{3}}}{2^{\frac{2}{3}}} \left(\frac{\pi^{\frac{4}{3}} 3^3 2^{\frac{32}{3}}}{L^2 c^2}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \frac{1}{2^3 3^{\frac{3}{2}} \pi^{\frac{2}{3}}} \left(\frac{2^{18} 3^{\frac{11}{2}} \pi^{\frac{8}{3}}}{L^3 c^3}\right)^{\frac{1}{3s+1}}} \\
&\in \mathcal{O}\big((\Delta v)^{-\frac{1}{3s+1}}\big) .
\end{aligned}
$$

For the application of a Newton-Cotes formula of order s we need at most s points on every line. So a requirement for the application is given by the following condition that must be fulfilled by Δv and s

$$
\frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \frac{\pi^{\frac{2}{3}}}{2^{\frac{2}{3}}} \left(\frac{\pi^{\frac{4}{3}} 3^3 2^{\frac{32}{3}}}{L^2 c^2}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \frac{1}{2^3 3^{\frac{3}{2}} \pi^{\frac{2}{3}}} \left(\frac{2^{18} 3^{\frac{11}{2}} \pi^{\frac{8}{3}}}{L^3 c^3}\right)^{\frac{1}{3s+1}}} \geq s . \qquad \square
$$

Lemma 3.3.13 (Discretization as a DVM and its properties)
Assuming that

$$k(\mathfrak{w}_2 - \mathfrak{v}_i, \omega) = \tilde{k}(\|\mathfrak{w}_2 - \mathfrak{v}_i\|, \angle(\mathfrak{w}_2 - \mathfrak{v}_i, \omega)) = \hat{k}(r_q\|\omega_2\|, \angle(\omega_2, \omega))$$

possesses the same symmetries as the grid (which means independence of $\alpha, \beta \in G$ in the second argument of $\hat{k}$), the discretization 3.3.11 can be transformed into a DVM:

$$\begin{aligned}\tilde{I}[f](\mathfrak{v}_i) &= \sum_{(x,k)\in B}\ \sum_{(j,l)\in C_{x,k}} \alpha_{xjkl}\Delta v_{xjkl} \sum_{q=0}^{\lfloor L/\Delta v_{x,j,k,l}\rfloor}\ \sum_{\alpha,\beta\in G} \begin{bmatrix} g(q)h(\theta_x, \varphi_{x,j}, \lambda_k, \psi_{k,l}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_x, \varphi_{x,j}, \psi_{k,l}) \end{bmatrix} \\ &= \sum_{j,k,l} A_{i,j}^{k,l}\big(f(\mathfrak{v}_k)f(\mathfrak{v}_l) - f(\mathfrak{v}_i)f(\mathfrak{v}_j)\big)\,,\end{aligned}$$

with

$$A_{i,j}^{k,l} := \begin{cases} A(a,b,c,d,e), & \text{if } \begin{array}{c} \exists a,b,c,d \in \overline{N}, e \in N_{abcd}, \alpha, \beta \in G : \\ (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')\begin{pmatrix} a,b,c,d, \\ e,\alpha,\beta \end{pmatrix} \end{array} \\ 0, & \text{else} \end{cases} ,$$

$$\begin{aligned} A(a,b,c,d,e) &:= \begin{bmatrix} \alpha_{abcd}\Delta v_{abcd} g(e) D(r_e, \theta_a, \varphi_{a,b}, \psi_{c,d}) s_{abcde} \cdot \\ k(r_e\omega_2(\theta_a, \varphi_{a,b}, \lambda_c, \psi_{c,d}), \omega(\theta_a, \varphi_{a,b})) \end{bmatrix} \\ \overline{N} &:= \left\{ (a,b,c,d) \in \mathbb{N}^4 \;\middle|\; \begin{array}{l} a \in \{1,\dots,N\}, b \in \{1,\dots,N_a\}, \\ c \in \{1,\dots,M\}, d \in \{1,\dots,M_c\} \end{array} \right\} , \\ N_{abcd} &:= \{0, \dots, \lfloor L/\Delta v_{a,b,c,d} \rfloor\} , \\ s_{abcde} &:= \left|\left\{ (\alpha,\beta) \;\middle|\; (\alpha,\beta) \in G^2, (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')(a,b,c,d,e,\alpha,\beta) \right\}\right| . \end{aligned}$$

This DVM fulfills the minimal properties 2.1.2.4 and possesses no artificial collision invariants on normal grids.

Proof:
Similar to the two dimensional case 3.2.9 we begin this proof with a closer look at the existence of a one to one correspondence between (i,j,k,l) and (a,b,c,d,e,α,β). In the end we can see that such a correspondence is non-existent, but we realize that different (a,b,c,d,e,α,β) corresponding to the same (i,j,k,l) only differ in α, β. Due to this we get the situation that the coefficients $A(a,b,c,d,e,\alpha,\beta)$ corresponding to the index (i,j,k,l) are equal and can be merged into $A_{i,j}^{k,l}$. We begin with a naive definition of $A_{i,j}^{k,l}$ as

$$\tilde{A}_{i,j}^{k,l} := \begin{cases} \tilde{A}(a,b,c,d,e,\alpha,\beta), & \text{if } \begin{array}{c} \exists a,b,c,d \in \overline{N}, c \in N_{abcd}, \beta, \gamma \in G : \\ (\mathfrak{v}_i, \mathfrak{v}_j, \mathfrak{v}_k, \mathfrak{v}_l) = (\mathfrak{v}_i, \mathfrak{w}_2, \mathfrak{v}_2', \mathfrak{w}_2')\begin{pmatrix} a,b,c,d, \\ e,\alpha,\beta \end{pmatrix} \end{array} \\ 0, & \text{else} \end{cases} ,$$

$$\tilde{A}(a,b,c,d,e,\alpha,\beta) := \left[\begin{array}{l} \alpha_{abcd}\Delta v_{abcd}\, g(e) D(r_e,\theta_a,\varphi_{a,b},\psi_{c,d})\cdot \\ k(r_e\omega_2(\theta_a,\varphi_{a,b},\lambda_c,\psi_{c,d},\alpha,\beta),\omega(\theta_a,\varphi_{a,b},\alpha)) \end{array} \right] ,$$

by simply collecting all coefficients corresponding to an multi index (a,b,c,d,e,α,β). This approach would be sufficient iff there exists a one to one correspondence between the multi indexes and the (i,j,k,l). Now we can use the same argumentation as in the second part of the proof of 3.2.9. We realize that by construction

$$\forall (a,b,c,d,e) \neq (\tilde{a},\tilde{b},\tilde{c},\tilde{d},\tilde{e}) : (\mathfrak{w}_2,\mathfrak{v}_2')(a,b,c,d,e) \neq (\mathfrak{w}_2,\mathfrak{v}_2')(\tilde{a},\tilde{b},\tilde{c},\tilde{d},\tilde{e}) ,$$

and so we see that changes in (a,b,c,d,e) lead to different collision pairs $\mathfrak{v}_i,\mathfrak{v}_j,\mathfrak{v}_k,\mathfrak{v}_l$ and to different coefficients $\tilde{A}_{i,j}^{k,l}$. So for every (i,j,k,l) we have at most one (a,b,c,d,e). Considering the reflections and rotations in the automorphism group we get the insight that we have at least a two to one correspondence in the general case, given by

$$\begin{aligned} &\forall \alpha_1,\beta_1 \in G : \alpha_2 := -I\cdot\alpha_1, \beta_2 := -I\beta_1 \\ &\Longrightarrow \mathfrak{w}_2(\alpha_1,\beta_1) = \mathfrak{w}_2(\alpha_2,\beta_2),\ \mathfrak{v}_2'(\alpha_1,\beta_1) = \mathfrak{v}_2'(\alpha_2,\beta_2),\ \mathfrak{w}_2'(\alpha_1,\beta_1) = \mathfrak{w}_2'(\alpha_2,\beta_2) . \end{aligned}$$

And the same argumentation as in 3.2.9 for three dimensions gives the result that we also get four to one, 8 to 1, 12 to 1 and 16 to 1 correspondences for special values of $\theta,\varphi,\lambda,\psi$. As in the two dimensional case these multiple correspondences differ only in α,β so all coefficients relating to the same (i,j,k,l) are equal iff we assume that the structure of k gives

$$k(\mathfrak{w}_2-\mathfrak{v}_i,\omega) = \tilde{k}(\|\mathfrak{w}_2-\mathfrak{v}_i\|,\angle(\mathfrak{w}_2-\mathfrak{v}_i,\omega)) = \hat{k}(r_e\|\omega_2\|,\angle(\omega_2,\omega)) ,$$

and that k possesses at least the grid symmetries - making it (in the second argument, the first one is clear) independent of $\alpha,\beta\in G$. From this arises the necessity to count the number of occurrences

$$s_{ijkl} := \left|\left\{\text{index}\,\middle|\,\text{index} \in \overline{N}\times N_{abcd}\times G^2, (\mathfrak{v}_i,\mathfrak{v}_j,\mathfrak{v}_k,\mathfrak{v}_l) = (\mathfrak{v}_i,\mathfrak{w}_2,\mathfrak{v}_2',\mathfrak{w}_2')(\text{index})\right\}\right| ,$$

and to multiply it with $\tilde{A}$:

$$A_{i,j}^{k,l} := \tilde{A}_{i,j}^{k,l}\cdot s_{ijkl} .$$

The same argumentation as in the two dimensional case gives

$$s_{ijkl} = s_{abcde} := |\{(\alpha,\beta)\,|(\mathfrak{v}_i,\mathfrak{v}_j,\mathfrak{v}_k,\mathfrak{v}_l) = (\mathfrak{v}_i,\mathfrak{w}_2,\mathfrak{v}_2',\mathfrak{w}_2')(a,b,c,d,e,\alpha,\beta)\}| ,$$

eliminating the need to look at the symmetries of s as soon as we have proved that the necessary symmetries depend only on α,β. This is our next step. We prove these symmetries with calculations similar to the two dimensional case. For this we need some preliminary considerations and results from the proof of 3.3.11:

$$\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\beta) = e\Delta v r_{c,d}\omega_2(\theta_a,\varphi_{a,b},\lambda_c,\psi_{c,d},\alpha,\beta) = e\Delta v(\alpha[\mathfrak{x}',\mathfrak{y}',\mathfrak{z}'])\beta P_{c,d}$$

$$\begin{aligned}
\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(\alpha,\beta) &= \overrightarrow{\mathfrak{w}_2'\mathfrak{w}_2}(\alpha,\beta)\\
&= e\Delta v r_{c,d}\langle\omega(\theta_a,\varphi_{a,b},\alpha),\omega_2(\theta_a,\varphi_{a,b},\lambda_c,\psi_{c,d},\alpha,\beta)\rangle\omega(\theta_a,\varphi_{a,b},\alpha)\\
&= q\Delta v(\beta P_{c,d})_1\alpha P_{a,b}\\
\overrightarrow{\mathfrak{w}_2'\mathfrak{v}_2'}(\alpha,\beta) &= e\Delta v r_{c,d}\omega_2(\theta_a,\varphi_{a,b},\lambda_c,\psi_{c,d},\alpha,\gamma\beta) = e\Delta v(\alpha[\mathfrak{x}',\mathfrak{y}',\mathfrak{z}'])\gamma\beta P_{c,d} \qquad (3.3.15)\\
&= \overrightarrow{\mathfrak{w}_2'\mathfrak{w}_2} - \overrightarrow{\mathfrak{v}\mathfrak{w}_2} + \overrightarrow{\mathfrak{v}\mathfrak{v}_2'} = 2\overrightarrow{\mathfrak{v}\mathfrak{v}_2'} - \overrightarrow{\mathfrak{v}\mathfrak{w}_2}\,,\\
\text{with } \gamma &:= \begin{pmatrix}1&0&0\\0&-1&0\\0&0&-1\end{pmatrix}\in G\,. \qquad (3.3.16)
\end{aligned}$$

The insight (3.3.15) follows from geometrical considerations. Knowing that ω_2 corresponds to spherical coordinates (in λ,ψ) on a sub grid it becomes clear, that the reflection of ω_2 on the axis $\mathfrak{x}'$ ($=\omega(\theta_a,\varphi_{a,b})$ results into $\overrightarrow{\mathfrak{w}_2'\mathfrak{v}_2'}$. The following diagram and considerations analog to the derivation of the representation of ω_2 through extrinsic rotations within the proof of 3.3.11 give:

$$\begin{aligned}
&D := \operatorname{diag}(r_{\mathfrak{x}},r_{\mathfrak{y}},r_{\mathfrak{z}})\,,\\
&\mathfrak{x}\xrightarrow{R_{\mathfrak{z}}(\theta)}\mathfrak{x}_2\xrightarrow{R_{\mathfrak{z}}(\theta_a)R_{\mathfrak{y}}(-\varphi_{a,b})R_{\mathfrak{z}}(\theta_a)^{-1}}\mathfrak{x}_3\xrightarrow{R_{\mathfrak{z}}(\theta_a)R_{\mathfrak{y}}(-\varphi_{a,b})D[R_{\mathfrak{z}}(\theta_a)R_{\mathfrak{y}}(-\varphi_{a,b})]^{-1}}P_{a,b}\\
&P_{a,b}\xrightarrow{\alpha}\mathfrak{x}_4=\overbrace{\alpha R_{\mathfrak{z}}(\theta_a)R_{\mathfrak{y}}(-\varphi_{a,b})D}^{B:=}\mathfrak{x}_1\xrightarrow{BR_{\mathfrak{z}}(\lambda_c)B^{-1}}\mathfrak{x}_5\\
&\mathfrak{x}_5\xrightarrow{BR_{\mathfrak{z}}(\lambda_c)R_{\mathfrak{y}}(-\psi_{c,d})[BR_{\mathfrak{z}}(\lambda_c)]^{-1}}\mathfrak{x}_6\xrightarrow{B\beta B^{-1}}\mathfrak{x}_7\xrightarrow{B\gamma B^{-1}}\mathfrak{x}_8\\
\Longrightarrow\ &\mathfrak{x}_8=\alpha R_{\mathfrak{z}}(\theta_a)R_{\mathfrak{y}}(-\varphi_{a,b})D\gamma\beta R_{\mathfrak{z}}(\lambda_c)R_{\mathfrak{y}}(-\psi_{c,d})\mathfrak{x}_1\\
&\quad=\omega_2(\theta,\varphi,\lambda,\psi,\alpha,\gamma\cdot\beta)\,.
\end{aligned}$$

And the form of γ in (3.3.16) is given by the demand that this operator reflects a point on the x - axis ($\mathfrak{x}'$ as can be seen in the diagram). From the above representations we get the following identities

$$\begin{aligned}
&\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\beta)=\overrightarrow{\mathfrak{w}_2'\mathfrak{v}_2'}(\alpha,\gamma\beta)=\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\gamma^2\beta)\,,\\
&\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\beta)=-\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,\beta)=-\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,-\beta)=\overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,-\beta)\,,\\
&\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(\alpha,\beta)=-\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(-\alpha,\beta)=-\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(\alpha,-\beta)=\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(-\alpha,-\beta)=\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}(\alpha,\gamma\beta)\,.
\end{aligned}$$

This and (3.3.15) can be used to prove the symmetries $A_{i,j}^{k,l}=A_{k,l}^{i,j}=A_{j,i}^{l,k}$ by proving that the corresponding velocities differ only in the operators α,β, which do not change the corresponding coefficients $A(a,b,c,d,e,\alpha,\beta)$:

$$\begin{pmatrix}\mathfrak{v}_i\\\mathfrak{v}_j\\\mathfrak{v}_k\\\mathfrak{v}_l\end{pmatrix}=\begin{pmatrix}\mathfrak{v}_i\\\mathfrak{v}_i+\overrightarrow{\mathfrak{v}\mathfrak{w}_2}\\\mathfrak{v}_i+\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}\\\mathfrak{v}_i+\overrightarrow{\mathfrak{v}\mathfrak{w}_2}-\overrightarrow{\mathfrak{v}\mathfrak{v}_2'}\end{pmatrix}$$

$$
\begin{aligned}
\Longrightarrow \begin{pmatrix}\mathfrak{v}_k\\ \mathfrak{v}_l\\ \mathfrak{v}_i\\ \mathfrak{v}_j\end{pmatrix} &= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} + \overrightarrow{\mathfrak{v}\mathfrak{w}_2} - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} + \overrightarrow{\mathfrak{v}\mathfrak{w}_2} \end{pmatrix} \\
&= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - 2\overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) \\ \mathfrak{v}_k - 2\overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(\alpha,\beta) + \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) \end{pmatrix} \\
&= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{w}'_2\mathfrak{v}'_2}(\alpha,\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) \\ \mathfrak{v}_k - \overrightarrow{\mathfrak{w}'_2\mathfrak{v}'_2}(\alpha,\beta) + \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(\alpha,\beta) \end{pmatrix} \\
&= \begin{pmatrix} \mathfrak{v}_k \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,\gamma\beta) \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(-\alpha,\gamma\beta) \\ \mathfrak{v}_k + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,\gamma\beta) - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(-\alpha,\gamma\beta) \end{pmatrix} \\
\Longrightarrow \begin{pmatrix}\mathfrak{v}_j\\ \mathfrak{v}_i\\ \mathfrak{v}_l\\ \mathfrak{v}_k\end{pmatrix} &= \begin{pmatrix} \mathfrak{v}_j \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_2} \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_2} + \overrightarrow{\mathfrak{v}\mathfrak{w}_2} - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} \\ \mathfrak{v}_j - \overrightarrow{\mathfrak{v}\mathfrak{w}_2} + \overrightarrow{\mathfrak{v}\mathfrak{v}'_2} \end{pmatrix} \\
&= \begin{pmatrix} \mathfrak{v}_j \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,\beta) \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(-\alpha,\beta) \\ \mathfrak{v}_j + \overrightarrow{\mathfrak{v}\mathfrak{w}_2}(-\alpha,\beta) - \overrightarrow{\mathfrak{v}\mathfrak{v}'_2}(-\alpha,\beta) \end{pmatrix} .
\end{aligned}
$$

The non negativity of $A_{i,j}^{k,l}$ is given by the fact that the factors A consists of are non negative. The lack of artificial collision invariants can be obtained by using 2.1.2.10 and realizing that $n \geq 1, m \geq 1$ leads to the inclusion of all squares (in planes parallel to the x-y, x-z, y-z plane in the velocity space) with a diameter of $2\Delta v$, $\sqrt{2}\Delta v$ as in the two dimensional case 3.2.9. □

Remark 3.3.14

As in the 2D case it is not possible to transform this DVM into an eLGpM, because in general it does not possess the necessary property (2.2.1) that is needed for theorem 2.2.2 . This condition is violated as soon as the order of the Farey sequences n, m gets larger than one. Due to this we need an according adjustment of the discretization to make it compatible with the eLGpMs.

Lemma 3.3.15 (Farey approximation as an eLGpM)
An eLGpM based on the Farey approximation (using $\mathfrak{F}_n, \mathfrak{F}_\mathfrak{m}$) can be written as

$$I_{\mathrm{LGpM}}[f](\mathfrak{v}_i) = \sum_{\mathfrak{c}\in\mathfrak{C}} \sum_{[\mathfrak{v}]\in\tilde{S}^{\mathfrak{V}}_{i,\mathfrak{c}}} \sum_{[\varphi]\in\tilde{G}} \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} \left(\prod_{\varphi'\in[\varphi]} f(\mathfrak{c}+\varphi'(\mathfrak{v}-\mathfrak{c})) - \prod_{\varphi'\in H} f(\mathfrak{c}+\varphi'(\mathfrak{v}_i-\mathfrak{c})) \right)$$

with

$$\mathfrak{C} := \mathfrak{V} \cup \mathfrak{V}_{\frac{1}{2}},\ \mathfrak{v}_k(\mathfrak{c},\mathfrak{v},\varphi) := \mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}),\ \mathfrak{v}_j(\mathfrak{c},i) := 2\mathfrak{c}-\mathfrak{v}_i\,,$$
$$\tilde{S}^{\mathfrak{V}}_{i,\mathfrak{c}} := S^{\mathfrak{V}}_{ij}/\sim_{\mathfrak{c}},\ \tilde{G} := G/\sim_H,\ \alpha^{\varphi,\mathfrak{v}}_{\mathfrak{c},\mathfrak{v}_i} := \frac{2\overline{A^k_{i,j}}}{|\{\varphi'\in G|\mathfrak{c}+\varphi(\mathfrak{v}-\mathfrak{c}) = c+\varphi'(\mathfrak{v}-\mathfrak{c})\}|}\,,$$
$$\overline{A^{k,l}_{i,j}} := A^{k,l}_{i,j}\cdot\overline{\Delta v_{ij}},\quad \overline{\Delta v_{ij}} := \begin{cases} 1, & \text{if } \frac{\overrightarrow{\mathfrak{v}_i\mathfrak{v}_j}}{2c} \in \mathfrak{C} \\ 2, & \text{else} \end{cases},\quad c = \max\left\{\tilde{c}\in\mathbb{N}\,\middle|\,\frac{\overrightarrow{\mathfrak{v}_i\mathfrak{v}_j}}{\tilde{c}} \in \mathfrak{V}\right\}$$

where $A^{k,l}_{i,j}$ comes from 3.3.13. The convergence order is the same as in 3.3.12, the error boundary is increased by a factor of 2^s and this LGpM has no artificial collision invariants.

Proof:
Analog to the 2D case 3.2.11. □

3.4 High order schemes

At this point let us take a look at the approach so far. We use the ansatz proposed by Rogier and Schneider apply another transformation of the collision operator, use a Newton-Cotes formula for the approximation of the innermost integral and Farey sequences for the approximation of the angular integrals. This approximation uses the Farey angles as grid points and the Farey arcs as the corresponding step sizes. This approach results into a discretization with a convergence order of up to 1 (in 2D) and we can transform this into a DVM that possesses the minimal requirements guaranteeing the correct collisional invariants (no artificial ones), equilibrium solutions and the H - theorem. Panferov and Heintz as well as Michel and Schneider [MS00, PH99] constructed similar discretizations that possess the same properties (order 1 and DVM with minimal requirements). In the last subsection we showed that this can be generalized into three dimensions satisfying the same minimal requirements and giving the same properties except the convergence order which only goes up to a maximum of $\frac{1}{3}$. Other discretization approaches for the extension of Farey sequences into three dimensions like [MS00, PH99] reach an order of 1. It remains to be seen if these approaches are generally better, because in this subsection we look at the question "Can we chose the integration weights $\alpha_\bullet$ corresponding to the Farey angles in such a way that we obtain a higher convergence order of the discretization ?". The answer is yes. To give an idea about the reachable convergence we use the simplest idea and apply standard quadrature formulas on the grid points given by the Farey angles. Fortunately the symmetries necessary to obtain the minimal requirements are independent of the $\alpha_\bullet$ due to the used transformation of the collision operator, our choice to discretize the minimal symmetry region S_1 and the successive application of the automorphism group. To minimize the overhead that is involved in the necessary modification of 3.2.3, 3.2.4, 3.3.11, 3.3.12, we only redo the necessary parts and refer to these proofs for anything else. At this points we use the developed framework (a framework that works parallel and is comparable to the ones used by [MS00,PH99]) to construct "high" order (order > 1) discretizations of the collision operator retaining the exact conservation of the moments by taking advantage of the explicit representation of the grid symmetries through the symmetry regions or the automorphism group. As far as the author knows the detailed inspection of this idea and its implementation seems to be new.

Corollary 3.4.1 (High order quadrature in 2D)
If we use quadrature formulas over Farey angles for the angular integrals in 3.2.3, we obtain the same discretization, where the only difference is the definition of $\alpha_{i,n}, \alpha_{j,m}$:

$$\alpha_i^{(2)} := \begin{cases} \int_{\theta_{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor s+1}}^{\theta_N} \prod\limits_{\substack{k=0 \\ k \neq i+s-N}}^{s} \frac{\theta - \theta_{N-s+k}}{\theta_i - \theta_{N-s+k}} \mathrm{d}\,\theta, & \text{if } i \geq N-s \\ 0, & \text{else} \end{cases},$$

$$\alpha_i := \alpha_i^{(1)} + \alpha_i^{(2)}, \qquad \tilde{N} := N-1\,,$$

$$\alpha_i^{(1)} := \begin{cases} \int_{\theta_{s\left(\left\lceil \frac{i}{s} \right\rceil -1\right)+1}}^{\theta_{\left\lceil \frac{i}{s} \right\rceil s+1}} L_{[i-1 \bmod(s)],s}^{\left\lceil \frac{i}{s} \right\rceil}(\theta)\mathrm{d}\,\theta, & \text{if } [i-1 \bmod(s)] \neq 0 \vee \left\lceil \frac{i}{s} \right\rceil = 1 \\ \int_{\theta_{s\left(\left\lceil \frac{i}{s} \right\rceil -1\right)+1}}^{\theta_{\left\lceil \frac{i}{s} \right\rceil s+1}} L_{s,s}^{\left\lceil \frac{i}{s} \right\rceil -1}(\theta) + L_{0,s}^{\left\lceil \frac{i}{s} \right\rceil}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \left\lceil \frac{i}{s} \right\rceil \notin \left\{1, \left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1\right\} \end{cases} \\ \int_{\theta_{s\left(\left\lceil \frac{i}{s} \right\rceil -1\right)+1}}^{\theta_{\left\lceil \frac{i}{s} \right\rceil s+1}} L_{s,s}^{\left\lceil \frac{i}{s} \right\rceil -1}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \left\lceil \frac{i}{s} \right\rceil = \left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1 \end{cases} \\ \int_{\theta_{s\left(\left\lceil \frac{i}{s} \right\rceil -1\right)+1}}^{\theta_{\left\lceil \frac{i}{s} \right\rceil s+1}} L_{0,s}^{\left\lceil \frac{i}{s} \right\rceil}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \left\lceil \frac{i}{s} \right\rceil = 1 \end{cases} \\ 0, & \text{else} \end{cases},$$

where $\alpha_{j,m}$ can be defined analog to $\alpha_{i,n}$ by substituting i with j and N with M. Here t corresponds to the number of used points for the interpolation (polynomial degree s plus one). The discretization 3.2.3 with these integration weights possesses an upper error bound of

$$e < \frac{K_\theta}{t!}\left(\left\lfloor \frac{n^2}{s} \right\rfloor + 1\right)\left(\frac{s}{n}\right)^{t+1} + \frac{\pi}{4}\frac{K_\lambda}{t!}\left(\left\lfloor \frac{m^2}{s} \right\rfloor + 1\right)\left(\frac{s}{m}\right)^{t+1}$$

$$+\frac{3\pi^2}{32} cL(\Delta v)^r K_l 2^r n^r m^r \,,$$

where K_θ, K_λ, K_l are some constants depending on f, k and r, c as in 3.2.3. Assuming that $\Delta v \in \mathbb{R}^+, r \in \mathbb{N}, t \in \mathbb{N}_{>1}$ are given constants satisfying

$$\frac{L}{2\Delta v \tilde{n}(\Delta v, r)\tilde{m}(\Delta v, r)} > r, \;\wedge\; t \le N, M \,,$$

with

$$\tilde{n}(\Delta v, r, s) = \left(\frac{c_3}{c_1}\right)^{\frac{-1}{s+2r}} (\Delta v)^{\frac{-r}{s+2r}} \left(\frac{c_1}{c_2}\right)^{\frac{-r}{s(s+2r)}} \,, \quad \tilde{m}(\Delta v, r, s) = \left(\frac{c_1}{c_2}\right)^{\frac{-1}{s}} n$$

(constants can be found in the proof) and choosing n, m according to

$$n = \lceil \tilde{n}(\Delta v, r, s) \rceil, \; m = \lceil \tilde{m}(\Delta v, r, s) \rceil \,,$$

the convergence order of this discretization is

$$\mathcal{O}\left((\Delta v)^{\frac{rs}{2r+s}}\right) \,.$$

The minimal number of points $\tilde{r}$ on any given line associated with the approximation of the innermost integral grows asymptotically as is given by

$$\tilde{r}(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{-s}{2r+s}}\right) \,.$$

Proof:
For this proof we use the same conventions and abbreviations as in the original proof 3.2.3. The only difference that occurs is the calculation of the three errors. Now we approximate the angular integrals by integration over interpolation polynomials of order t. For this we need the definition of the Lagrange polynomials and an estimate for successive Farey angles:

$$L_{j,s}^{(i)}(\theta) := \prod_{k=0,k\neq j}^{s} \frac{\theta - \theta_{s(i-1)+1+k}}{\theta_{s(i-1)+1+j} - \theta_{s(i-1)+1+k}} \,,$$

$$\begin{aligned}\theta_{is+1} - \theta_{(i-1)s+1} &= \arctan\left(\frac{p_{is+1,n}}{q_{is+1,n}}\right) - \arctan\left(\frac{p_{(i-1)s+1,n}}{q_{(i-1)s+1,n}}\right) \overset{(3.3.3)}{<} \frac{p_{is+1,n}}{q_{is+1,n}} - \frac{p_{(i-1)s+1,n}}{q_{(i-1)s+1,n}} \\ &= \sum_{j=(i-1)s+1}^{is} \frac{p_{j+1,n}}{q_{j+1,n}} - \frac{p_{j,n}}{q_{j,n}} \le \sum_{j=(i-1)s+1}^{is} \frac{1}{n} = \frac{s}{n} \,.\end{aligned}$$

Now we calculate the error of a quadrature formula using a polynomial of order $s = t_1 - 1$ with discretization points θ_i and $N = \tilde{N} + 1$ points. Unfortunately we have to assume $\frac{\tilde{N}}{s} \notin \mathbb{N}$, which complicates the normally simple calculations. We begin with the error of the outermost integral:

$$\begin{aligned}
e_3 :=& \int_0^{\frac{\pi}{4}} H_2(\theta)\mathrm{d}\theta - \sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \int_{\theta_{(i-1)s+1}}^{\theta_{is+1}} \sum_{j=0}^{s} H_2(\theta_{(i-1)s+1+j}) L_{j,s}^{(i)}(\theta)\mathrm{d}\,\theta \\
&- \int_{\theta_{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor s+1}}^{\theta_N} \sum_{j=0}^{s} H_2(\theta_{N-s+j}) \prod_{k=0,k\neq j}^{s} \frac{\theta - \theta_{N-s+k}}{\theta_{N-s+j} - \theta_{N-s+k}} \mathrm{d}\,\theta \\
&= \sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \int_{\theta_{(i-1)s+1}}^{\theta_{is+1}} H_2(\theta) - \sum_{j=0}^{s} H_2(\theta_{(i-1)s+1+j}) L_{j,s}^{(i)}(\theta)\mathrm{d}\,\theta \\
&+ \int_{\theta_{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor s+1}}^{\theta_N} H_2(\theta) - \sum_{j=0}^{s} H_2(\theta_{N-s+j}) \prod_{k=0,k\neq j}^{s} \frac{\theta - \theta_{N-s+k}}{\theta_{N-s+j} - \theta_{N-s+k}} \mathrm{d}\,\theta \\
&\leq \sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \int_{\theta_{(i-1)s+1}}^{\theta_{is+1}} \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \prod_{k=0}^{s} \left(\theta - \theta_{(i-1)s+1+k}\right) \mathrm{d}\,\theta \\
&+ \int_{\theta_{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor s+1}}^{\theta_N} \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \prod_{k=0}^{s} \left(\theta - \theta_{N-s+k}\right) \mathrm{d}\,\theta \\
&< \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \int_{\theta_{(i-1)s+1}}^{\theta_{is+1}} \left(\theta_{is+1} - \theta_{(i-1)s+1}\right)^{t_1} \mathrm{d}\,\theta \\
&+ \int_{\theta_{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor s+1}}^{\theta_N} \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \left(\theta_N - \theta_{N-s}\right)^{t_1} \mathrm{d}\,\theta \\
&< \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \left(\sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \left(\frac{s}{n}\right)^{t_1+1} + \left(\frac{s}{n}\right)^{t_1+1} \right) \\
&= \frac{\sup\limits_{\theta \in \left[0, \frac{\pi}{4}\right]} \left| H_2^{(t_1)}(\theta) \right|}{t_1!} \left(\left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1 \right) \left(\frac{s}{n}\right)^{t_1+1}, \text{ and}
\end{aligned}$$

$$e_3 := \int_0^{\frac{\pi}{4}} H_2(\theta)\mathrm{d}\theta - \sum_{i=1}^{\left\lfloor \frac{\tilde{N}}{s} \right\rfloor} \int_{\theta_{s(i-1)+1}}^{\theta_{is+1}} \sum_{j=0}^{s} H_2(\theta_{s(i-1)+1+j}) L_{j,s}^{(i)}(\theta)\mathrm{d}\,\theta$$

$$
\begin{aligned}
&- \int_{\theta_{\lfloor \frac{\tilde{N}}{s} \rfloor s+1}}^{\theta_N} \sum_{j=0}^{s} H_2(\theta_{N-s+j}) \prod_{k=0,k\neq j}^{s} \frac{\theta - \theta_{N-s+k}}{\theta_{N-s+j} - \theta_{N-s+k}} \mathrm{d}\,\theta \\
&= \int_0^{\frac{\pi}{4}} H_2(\theta)\mathrm{d}\theta - \sum_{i=1}^{N} \alpha_i H_2(\theta_i), \text{ with} \\
\alpha_i^{(2)} &:= \begin{cases} \int_{\theta_{\lfloor \frac{\tilde{N}}{s} \rfloor s+1}}^{\theta_N} \prod\limits_{\substack{k=0 \\ k \neq i+s-N}}^{s} \frac{\theta - \theta_{N-s+k}}{\theta_i - \theta_{N-s+k}} \mathrm{d}\,\theta, & \text{if } i \geq N-s \\ 0, & \text{else} \end{cases} \\
\alpha_i &:= \alpha_i^{(1)} + \alpha_i^{(2)}
\end{aligned}
$$

and

$$
\alpha_i^{(1)} := \begin{cases}
\int_{\theta_{s\left(\lceil \frac{i}{s} \rceil -1\right)+1}}^{\theta_{\lceil \frac{i}{s} \rceil s+1}} L_{[i-1 \bmod(s)],s}^{\lceil \frac{i}{s} \rceil}(\theta)\mathrm{d}\,\theta, & \text{if } [i-1 \bmod(s)] \neq 0 \vee \lceil \frac{i}{s} \rceil = 1 \\
\int_{\theta_{s\left(\lceil \frac{i}{s} \rceil -1\right)+1}}^{\theta_{\lceil \frac{i}{s} \rceil s+1}} L_{s,s}^{\lceil \frac{i}{s} \rceil -1}(\theta) + L_{0,s}^{\lceil \frac{i}{s} \rceil}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \lceil \frac{i}{s} \rceil \notin \left\{1, \left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1\right\} \end{cases} \\
\int_{\theta_{s\left(\lceil \frac{i}{s} \rceil -1\right)+1}}^{\theta_{\lceil \frac{i}{s} \rceil s+1}} L_{s,s}^{\lceil \frac{i}{s} \rceil -1}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \lceil \frac{i}{s} \rceil = \left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1 \end{cases} \\
\int_{\theta_{s\left(\lceil \frac{i}{s} \rceil -1\right)+1}}^{\theta_{\lceil \frac{i}{s} \rceil s+1}} L_{0,s}^{\lceil \frac{i}{s} \rceil}(\theta)\mathrm{d}\,\theta, & \text{if } \begin{cases} [i-1 \bmod(s)] = 0 \\ \wedge \lceil \frac{i}{s} \rceil = 1 \end{cases} \\
0, & \text{else}
\end{cases} .
$$

This complicated looking definition of $\alpha_i^{(1)}$ is generally very simple. The first case corresponds to the inner points used for the quadratures, the second case to the points where different polynomials begin and end (we use a composite rule) and the next two cases to the begin and the end of the regular quadrature. Then $\alpha^{(2)}$ corresponds to the case where we have some points left which must be specially treated . Fortunately we can reuse this for the error of the next integral:

$$
e_2(i) := H_2(\theta_i) - \sum_{j=1}^{M} \alpha_{j,m} H_1(\lambda_j, \theta_i)\,,
$$

where $\alpha_{j,m}$ is defined analog to α_i, n (above).

$$
\begin{aligned}
\Longrightarrow e_2(i) &\leq \frac{\sup\limits_{\lambda \in \left[0,\frac{\pi}{4}\right]} \left| \frac{\partial^{t_2} H_1(\lambda,\theta_i)}{\partial \lambda^{t_2}} \right|}{t_2!} \left(\left\lfloor \frac{\tilde{M}}{s_2} \right\rfloor + 1 \right) \left(\frac{s_2}{m} \right)^{t_2+1}, \\
e_2 &:= \sum_{i=1}^{N} \alpha_i e_2(i)
\end{aligned}
$$

$$
\begin{aligned}
&\le \overbrace{\frac{\sup\limits_{\lambda,\theta\in\left[0,\frac{\pi}{4}\right]}\left|\frac{\partial^{t_1}H_1(\lambda,\theta)}{\partial\lambda^{t_1}}\right|}{t_2!}\left(\left\lfloor\frac{\tilde{M}}{s_2}\right\rfloor+1\right)\left(\frac{s_2}{m}\right)^{t_2+1}}^{a:=}\sum_{i=1}^{N}\alpha_i \\
&= a\sum_{i=1}^{\left\lfloor\frac{\tilde{N}}{s}\right\rfloor}\int_{\theta_{s(i-1)+1}}^{\theta_{is+1}}\sum_{j=0}^{s}L_{j,s}^{(i)}(\theta)\mathrm{d}\,\theta \\
&+ a\int_{\theta_{\left\lfloor\frac{\tilde{N}}{s}\right\rfloor s+1}}^{\theta_N}\sum_{j=0}^{s}\prod_{k=0,k\neq j}^{s}\frac{\theta-\theta_{N-s+k}}{\theta_{N-s+j}-\theta_{N-s+k}}\mathrm{d}\,\theta \\
&= a\left(\sum_{i=1}^{\left\lfloor\frac{\tilde{N}}{s}\right\rfloor}\int_{\theta_{s(i-1)+1}}^{\theta_{is+1}}1\mathrm{d}\,\theta+\int_{\theta_{\left\lfloor\frac{\tilde{N}}{s}\right\rfloor s+1}}^{\theta_N}1\mathrm{d}\,\theta\right) \\
&= a(\theta_N-\theta_1)=\frac{\pi}{4}a\,.
\end{aligned}
$$

And the same procedure can be used in the calculation of the last error (for $e_1(i,j)$ see 3.2.3) :

$$
\begin{aligned}
|e_1| &\le \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_i\alpha_j e_1(i,j) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_i\alpha_j c\frac{3}{2}L(\Delta v)^r(p_{i,n}^2+q_{i,n}^2)^{\frac{r}{2}}(p_{j,m}^2+q_{j,m}^2)^{\frac{r}{2}}\sup_{l\in[0,L]}\left|\frac{\partial^r h(l,\lambda_j,\theta_i)}{\partial l^r}\right| \\
&\le \frac{3}{2}cL(\Delta v)^r\overbrace{\sup_{\substack{l\in[0,L]\\ \theta,\lambda\in\left[0,\frac{\pi}{4}\right]}}\left|\frac{\partial^r h(l,\lambda,\theta)}{\partial l^r}\right|}^{=:K_l}\sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_i\alpha_j(p_{i,n}^2+q_{i,n}^2)^{\frac{r}{2}}(p_{j,m}^2+q_{j,m}^2)^{\frac{r}{2}} \\
&< \frac{3}{2}cL(\Delta v)^r K_l 2^r n^r m^r\sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_i\alpha_j=\frac{3\pi^2}{32}cL(\Delta v)^r K_l 2^r n^r m^r\,.
\end{aligned}
$$

Adding these three errors together and defining

$$
K_\lambda := \sup_{\theta,\lambda\in\left[0,\frac{\pi}{4}\right]}\left|\frac{\partial^{t_1}H_1(\lambda,\theta)}{\partial\lambda^{t_1}}\right|,\quad K_\theta := \sup_{\theta\in\left[0,\frac{\pi}{4}\right]}\left|H_2^{(t_1)}(\theta)\right|\,,
$$

we obtain

$$\begin{aligned} e := \left| I[f](\mathfrak{v}) - \tilde{I}[f](\mathfrak{v}) \right| < &\frac{K_\theta}{t_1!} \left(\left\lfloor \frac{\tilde{N}}{s} \right\rfloor + 1 \right) \left(\frac{s}{n} \right)^{t_1+1} \\ &+ \frac{\pi}{4} \frac{K_\lambda}{t_2!} \left(\left\lfloor \frac{\tilde{M}}{s_2} \right\rfloor + 1 \right) \left(\frac{s_2}{m} \right)^{t_2+1} \\ &+ \frac{3\pi^2}{32} cL(\Delta v)^r K_l 2^r n^r m^r \, . \end{aligned} \tag{3.4.1}$$

Now we need to get an estimation for N to determine what convergence order could be obtained through this approach. We know that $N = |\mathfrak{F}_n|$ and it is well known that

$$|\mathfrak{F}_n| = 1 + \sum_{m=1}^{n} \varphi(m)$$

where $\varphi(m)$ corresponds to Euler's totient function. This function counts the positive integers less than or equal to m that are relatively prime to m. This directly gives $\varphi(m) \leq m$ leading to

$$|\mathfrak{F}_n| \leq 1 + \sum_{m=1}^{n} m = 1 + \frac{n^2+n}{2} \leq 1 + n^2 \, .$$

Using this in the corresponding error estimations we get

$$\begin{aligned} e < &\frac{K_\theta}{t_1!} \left(\left\lfloor \frac{n^2}{s} \right\rfloor + 1 \right) \left(\frac{s}{n} \right)^{t_1+1} + \frac{\pi}{4} \frac{K_\lambda}{t_2!} \left(\left\lfloor \frac{m^2}{s_2} \right\rfloor + 1 \right) \left(\frac{s_2}{m} \right)^{t_2+1} \\ &+ \frac{3\pi^2}{32} cL(\Delta v)^r K_l 2^r n^r m^r \\ &< K \left(\overbrace{2 \frac{s^{t_1+1}}{t_1!}}^{c_1:=} n^{-t_1+1} + \overbrace{\frac{2\pi}{4} \frac{s_2^{t_2+1}}{t_2!}}^{c_2:=} m^{-t_2+1} + \overbrace{\frac{3\pi^2}{32} cL2^r}^{c_3:=} (\Delta v)^r n^r m^r \right) \, , \\ &\text{because } \left\lfloor \frac{n^2}{s} \right\rfloor + 1 \overset{s \geq 1}{\leq} 2n^2 \, . \end{aligned}$$

Now we use the same ansatz as in the last 2 convergence order proofs 3.2.4, 3.3.12 and we assume $t_1 = t_2$ for the sake of simplicity:

$$c_1 n^{-t+1} = c_2 m^{-t+1} = c_3 (\Delta v)^r n^r m^r \Longrightarrow m = \left(\frac{c_1}{c_2} \right)^{\frac{1}{-t+1}} n$$

$$\Longrightarrow c_1 n^{-t+1} = c_3 (\Delta v)^r n^r \left(\left(\frac{c_1}{c_2} \right)^{\frac{1}{-t+1}} n \right)^r = c_3 (\Delta v)^r n^r \left(\frac{c_1}{c_2} \right)^{\frac{r}{-t+1}} n^r$$

$$\begin{aligned}
\Longleftrightarrow n^{1-t-2r} &= \frac{c_3}{c_1}(\Delta v)^r \left(\frac{c_1}{c_2}\right)^{\frac{r}{-t+1}} \\
\Longleftrightarrow n &= \left(\frac{c_3}{c_1}\right)^{\frac{1}{1-t-2r}} (\Delta v)^{\frac{r}{1-t-2r}} \left(\frac{c_1}{c_2}\right)^{\frac{r}{(-t+1)(1-t-2r)}} \in \mathcal{O}\left((\Delta v)^{\frac{r}{1-t-2r}}\right) \\
\Longrightarrow m &= \left(\frac{c_1}{c_2}\right)^{\frac{1}{-t+1}+\frac{r}{(-t+1)(1-t-2r)}} \left(\frac{c_3}{c_1}\right)^{\frac{1}{1-t-2r}} (\Delta v)^{\frac{r}{1-t-2r}} \in \mathcal{O}\left((\Delta v)^{\frac{r}{1-t-2r}}\right) \\
\Longrightarrow c_1 n^{-t+1} &= \text{const} \cdot \left((\Delta v)^{\frac{r}{1-t-2r}}\right)^{-t+1} = \text{const} \cdot (\Delta v)^{\frac{t-1}{2+\frac{t-1}{r}}} \\
\Longrightarrow c_2 m^{-t+1} &= \text{const} \cdot \left((\Delta v)^{\frac{r}{1-t-2r}}\right)^{-t+1} = \text{const} \cdot (\Delta v)^{\frac{t-1}{2+\frac{t-1}{r}}} \\
\Longrightarrow c_3 (\Delta v)^r n^r m^r &= \text{const} \cdot (\Delta v)^r \left((\Delta v)^{\frac{r}{1-t-2r}}\right)^r \cdot \left((\Delta v)^{\frac{r}{1-t-2r}}\right)^r \\
&= \text{const} \cdot (\Delta v)^{\frac{t-1}{2+\frac{t-1}{r}}} \\
\Longrightarrow e &< c_1 n^{-t+1} + c_2 m^{-t+1} + c_3 (\Delta v)^r n^r m^r = \mathcal{O}\left((\Delta v)^{\frac{t-1}{2+\frac{t-1}{r}}}\right) .
\end{aligned}$$

The number of minimal points on any given line can be taken from an analog adaption of the proof of 3.2.4. □

Remark 3.4.2
The completion of this approximation can be taken from 3.2.5, because it can be done in the exact same way. The change of the integration weights α_i, α_j does not change the fact, that the corresponding DVM possesses the minimal properties 2.1.2.4 giving the result that the DVM has no artificial collision invariants, the correct equilibrium solutions and the H - theorem. According to our ansatz these minimal properties are only influenced by the transformation of the collision integral, the successive approximation above minimal symmetry regions $S_\bullet$ and the following application of the automorphism group. The original Farey discretization according to theorem 3.2.4 corresponds to choosing $s = 2$. Another interesting point is that choosing a fixed s we obtain $\lim\limits_{r\to\infty} (\Delta v)^{\frac{rs}{2r+s}} = (\Delta v)^{\frac{s}{2}}$ and choosing a fixed r we get $\lim\limits_{s\to\infty} (\Delta v)^{\frac{rs}{2r+s}} = (\Delta v)^r$. So we are free to construct high order deterministic approximations of the collision operator satisfying the demand to have the correct collisional invariants, equilibrium solutions and the H - theorem.

The same can be done in the three dimensional case.

Corollary 3.4.3 (High order quadrature in 3D)
If we use quadrature formulas over Farey angles for the angular integrals in 3.3.11, we obtain the same discretization, where the only difference is the definition of $\alpha_i, \alpha_{i,j}$, here α_i is defined as in the last corollary and $\alpha_{i,j}$ is given as

$$\alpha_{i,j}^{(2)} := \begin{cases} \int_{\varphi_{i,\left\lfloor \frac{\tilde{N}_i}{\underline{t}} \right\rfloor \underline{t}+1}}^{\varphi_{i,N_i}} \prod_{\substack{k\,=\,0\\ k\,\neq\, i+\underline{t}-N_i}}^{\underline{t}} \frac{\varphi-\varphi_{i,N_i-\underline{t}+k}}{\varphi_{i,j}-\varphi_{i,N_i-\underline{t}+k}} \mathrm{d}\,\varphi, & \text{if } j \geq N_i - \underline{t} \\ 0, & \text{else} \end{cases} ,$$

$$\alpha_{i,j} := \alpha_{i,j}^{(1)} + \alpha_{i,j}^{(2)} ,$$

$$\alpha_{i,j}^{(1)} := \begin{cases} \int_{\varphi_{i,\underline{t}\left(\left\lceil \frac{j}{\underline{t}} \right\rceil -1\right)+1}}^{\varphi_{i,\left\lceil \frac{j}{\underline{t}} \right\rceil \underline{t}+1}} L_{[i-1 \bmod(\underline{t})],\underline{t}}^{\left\lceil \frac{j}{\underline{t}} \right\rceil,i}(\varphi) \mathrm{d}\,\varphi, & \text{if } [j-1 \bmod(\underline{t})] \neq 0 \vee \left\lceil \frac{j}{\underline{t}} \right\rceil = 1 \\ \int_{\varphi_{i,\underline{t}\left(\left\lceil \frac{j}{\underline{t}} \right\rceil -1\right)+1}}^{\varphi_{i,\left\lceil \frac{j}{\underline{t}} \right\rceil \underline{t}+1}} L_{\underline{t},\underline{t}}^{\left\lceil \frac{j}{\underline{t}} \right\rceil -1,i}(\varphi) + L_{0,\underline{t}}^{\left\lceil \frac{j}{\underline{t}} \right\rceil,i}(\varphi) \mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil \frac{j}{\underline{t}} \right\rceil \notin \left\{1, \left\lfloor \frac{\tilde{N}_i}{\underline{t}} \right\rfloor + 1\right\} \end{cases} \\ \int_{\varphi_{i,\underline{t}\left(\left\lceil \frac{j}{\underline{t}} \right\rceil -1\right)+1}}^{\varphi_{i,\left\lceil \frac{j}{\underline{t}} \right\rceil \underline{t}+1}} L_{\underline{t},\underline{t}}^{\left\lceil \frac{j}{\underline{t}} \right\rceil -1,i}(\varphi) \mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil \frac{j}{\underline{t}} \right\rceil = \left\lfloor \frac{\tilde{N}_i}{\underline{t}} \right\rfloor + 1 \end{cases} \\ \int_{\varphi_{i,\underline{t}\left(\left\lceil \frac{j}{\underline{t}} \right\rceil -1\right)+1}}^{\varphi_{i,\left\lceil \frac{j}{\underline{t}} \right\rceil \underline{t}+1}} L_{0,\underline{t}}^{\left\lceil \frac{j}{\underline{t}} \right\rceil,i}(\varphi) \mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil \frac{j}{\underline{t}} \right\rceil = 1 \end{cases} \\ 0, & \text{else} \end{cases} ,$$

$$L_{j,\underline{t}}^{(l,i)}(\varphi) := \prod_{k=0,k\neq j}^{\underline{t}} \frac{\varphi - \varphi_{i,\underline{t}(l-1)+1+k}}{\varphi_{i,\underline{t}(l-1)+1+j} - \varphi_{i,\underline{t}(l-1)+1+k}} .$$

Here $\underline{t} := t-1$ and t corresponds to the number of used points for the interpolation polynomial used in the approximation of the angular integrals. The discretization 3.3.11 with these integration weights possesses an upper error bound of

$$e < 96 \cdot (K_\theta + \frac{\pi}{4} K_{\theta,\varphi}) \frac{(t-1)^{t+1}}{t!} n^{1-t} + 288 \cdot \pi^2 \cdot (K_{\theta,\varphi,\lambda} + \frac{\pi}{4} K_{\theta,\varphi,\lambda,\psi}) \frac{(t-1)^{t+1}}{t!} m^{1-t} \\ + \frac{27\pi^4}{2} Lc3^{\frac{s}{2}} 2^s K_r (\Delta v)^s n^{2s} m^s ,$$

where $K_\theta, K_{\theta,\varphi}, K_{\theta,\varphi,\lambda}, K_{\theta,\varphi,\lambda,\psi}, K_r$ are some constants depending only on f, k and its derivatives and s, c correspond to the used Newton-Cotes formula, s being the error order and c corresponding to some error constants. The order of the interpolation polynomials used for the quadrature of the angular integrals is given by $\underline{t}$. Assuming that $\Delta v \in \mathbb{R}^+, s \in \mathbb{N}_{>0}$ are given constants satisfying

$$\frac{L}{\sqrt{7}(\Delta v)^{\frac{s+1}{3s+1}} \left(\frac{c_2^{s+1}}{c_3^2 c_1^{s-1}}\right)^{\frac{1}{3s+1}} + 2(\Delta v)^{\frac{1}{3s+1}} \left(\frac{c_1^2}{c_3^3 c_2^{s-1}}\right)^{\frac{1}{3s+1}}} \geq s .$$

with

$$\tilde{n}(\Delta v, s, \underline{t}) = \left(\frac{c_3}{c_1}\right)^{\frac{-1}{\underline{t}+3s}} (\Delta v)^{\frac{-s}{\underline{t}+3s}} \left(\frac{c_1}{c_2}\right)^{\frac{s}{\underline{t}(\underline{t}+3s)}}, \quad \tilde{m}(\Delta v, s, \underline{t}) = \left(\frac{c_1}{c_2}\right)^{\frac{-1}{\underline{t}}} n$$

(constants can be found in the proof) and choosing n, m according to

$$n = \lceil \tilde{n}(\Delta v, s, \underline{t}) \rceil, \ m = \lceil \tilde{m}(\Delta v, s, \underline{t}) \rceil ,$$

the convergence order of this discretization is

$$\mathcal{O}\left((\Delta v)^{\frac{s\underline{t}}{3s+\underline{t}}}\right) .$$

The minimal number of points $\tilde{r}$ on any given line associated with the approximation of the innermost integral grows asymptotically as is given by

$$\tilde{r}(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{-\underline{t}}{3s+\underline{t}}}\right) .$$

Proof:
For this proof we use the definitions from the proof of 3.3.11, 3.3.12 and the findings within the proof of the last corollary. We use the same procedure as above, this means we use standard quadrature formulas based on polynomial interpolation for the approximation of the angular integrals and Newton-Cotes formulas for the innermost integral, prove the resulting error boundary and use this to obtain a convergence order depending on the order of the Newton-Cotes formula and the degree used in the polynomial interpolations. To do this we need an estimate for successive angles that occur in the expansion of Farey sequences into the third dimension:

$$\begin{aligned} \varphi_{i,jt} - \varphi_{i,(j-1)t+1} &= \arctan\left(\frac{p_{i,jt}}{q_{i,jt}} \frac{q_i}{\sqrt{q_i^2+p_i^2}}\right) - \arctan\left(\frac{p_{i,(j-1)t+1}}{q_{i,(j-1)t+1}} \frac{q_i}{\sqrt{q_i^2+p_i^2}}\right) \\ &< \frac{p_{i,jt}}{q_{i,jt}} \frac{q_i}{\sqrt{q_i^2+p_i^2}} - \frac{p_{i,(j-1)t+1}}{q_{i,(j-1)t+1}} \frac{q_i}{\sqrt{q_i^2+p_i^2}} = \left(\frac{p_{i,jt}}{q_{i,jt}} - \frac{p_{i,(j-1)t+1}}{q_{i,(j-1)t+1}}\right) \frac{q_i}{\sqrt{q_i^2+p_i^2}} \\ &\leq \frac{t-1}{n} \frac{q_i}{\sqrt{q_i^2+p_i^2}} \leq \frac{t-1}{n} . \end{aligned}$$

Now we apply the same quadrature formulas as in the last proof and create an alternative to 3.3.2. As before anything within this proposition holds true except the definition of the αs and the error boundary. Using the definitions from 3.3.2 and the same quadrature as in the last proof we instantly get:

$$e := \left| \sum_{i=1}^{N} \alpha_i \sum_{j=1}^{N_i} \alpha_{i,j} H(\theta_i, \varphi_{i,j}) - \int_0^{\frac{\pi}{4}} \int_0^{\arctan(\sin(\theta))} H(\theta, \varphi) \mathrm{d}\varphi \mathrm{d}\theta \right|$$

$$
\begin{aligned}
&< \sup_{\theta\in[0,2\pi]} \left|H^{(t)}(\theta)\right| \frac{1}{t!}\left(\left\lfloor\frac{N-1}{t-1}\right\rfloor+1\right)\left(\frac{t-1}{n}\right)^{t+1} \\
&+ \frac{\pi}{4} \sup_{\substack{\theta\,\in\,[0,\,2\pi] \\ \varphi\,\in\,\left[-\frac{\pi}{2},\,\frac{\pi}{2}\right]}} \left|\frac{\partial^t H(\theta,\varphi)}{\partial\varphi^t}\right| \frac{1}{t!}\left(\left\lfloor\frac{N_i-1}{t-1}\right\rfloor+1\right)\left(\frac{t-1}{n}\right)^{t+1} ,
\end{aligned}
$$

with the same α_i as in the last proof, $\tilde{N}_i := N_i - 1, \underline{t} := t-1$ and the integration weights

$$
\alpha^{(2)}_{i,j} := \begin{cases} \int_{\varphi_{i,\left\lfloor\frac{\tilde{N}_i}{\underline{t}}\right\rfloor\underline{t}+1}}^{\varphi_{i,N_i}} \prod\limits_{\substack{k\,=\,0 \\ k\,\neq\, i\,+\,\underline{t}\,-\,N_i}}^{\underline{t}} \frac{\varphi-\varphi_{i,N_i-\underline{t}+k}}{\varphi_{i,j}-\varphi_{i,N_i-\underline{t}+k}}\mathrm{d}\,\varphi, & \text{if } j \geq N_i-\underline{t} \\ 0, & \text{else} \end{cases} ,
$$

$$
\alpha_{i,j} := \alpha^{(1)}_{i,j} + \alpha^{(2)}_{i,j} ,
$$

$$
\alpha^{(1)}_{i,j} := \begin{cases}
\int_{\varphi_{i,\underline{t}\left(\left\lceil\frac{j}{\underline{t}}\right\rceil-1\right)+1}}^{\varphi_{i,\left\lceil\frac{j}{\underline{t}}\right\rceil\underline{t}+1}} L^{\left\lceil\frac{j}{\underline{t}}\right\rceil,i}_{[i-1 \bmod(\underline{t})],\underline{t}}(\varphi)\mathrm{d}\,\varphi, & \text{if } [j-1 \bmod(\underline{t})] \neq 0 \vee \left\lceil\frac{j}{\underline{t}}\right\rceil = 1 \\
\int_{\varphi_{i,\underline{t}\left(\left\lceil\frac{j}{\underline{t}}\right\rceil-1\right)+1}}^{\varphi_{i,\left\lceil\frac{j}{\underline{t}}\right\rceil\underline{t}+1}} L^{\left\lceil\frac{j}{\underline{t}}\right\rceil-1,i}_{\underline{t},\underline{t}}(\varphi) + L^{\left\lceil\frac{j}{\underline{t}}\right\rceil,i}_{0,\underline{t}}(\varphi)\mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil\frac{j}{\underline{t}}\right\rceil \notin \left\{1, \left\lfloor\frac{\tilde{N}_i}{\underline{t}}\right\rfloor+1\right\} \end{cases} \\
\int_{\varphi_{i,\underline{t}\left(\left\lceil\frac{j}{\underline{t}}\right\rceil-1\right)+1}}^{\varphi_{i,\left\lceil\frac{j}{\underline{t}}\right\rceil\underline{t}+1}} L^{\left\lceil\frac{j}{\underline{t}}\right\rceil-1,i}_{\underline{t},\underline{t}}(\varphi)\mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil\frac{j}{\underline{t}}\right\rceil = \left\lfloor\frac{\tilde{N}_i}{\underline{t}}\right\rfloor+1 \end{cases} \\
\int_{\varphi_{i,\underline{t}\left(\left\lceil\frac{j}{\underline{t}}\right\rceil-1\right)+1}}^{\varphi_{i,\left\lceil\frac{j}{\underline{t}}\right\rceil\underline{t}+1}} L^{\left\lceil\frac{j}{\underline{t}}\right\rceil,i}_{0,\underline{t}}(\varphi)\mathrm{d}\,\varphi, & \text{if } \begin{cases} [j-1 \bmod(\underline{t})] = 0 \\ \wedge \left\lceil\frac{j}{\underline{t}}\right\rceil = 1 \end{cases} \\
0, & \text{else}
\end{cases} ,
$$

$$
L^{(l,i)}_{j,\underline{t}}(\varphi) := \prod_{k=0,k\neq j}^{\underline{t}} \frac{\varphi-\varphi_{i,\underline{t}(l-1)+1+k}}{\varphi_{i,\underline{t}(l-1)+1+j}-\varphi_{i,\underline{t}(l-1)+1+k}} .
$$

Calculations analog to the proof of the last corollary give:

$$
\begin{aligned}
&\left|\int_0^{\frac{\pi}{4}}\int_0^{\arctan(\sin(\theta))} H(\theta,\varphi)\cos(\varphi)\mathrm{d}\varphi\mathrm{d}\theta - \sum_{i=1}^{N}\alpha_{i,n}\sum_{j=1}^{N_i}\alpha_{i,j}H(\theta_i,\varphi_{i,j})\cos(\varphi_{i,j})\right| \\
&< K_\theta\frac{1}{t!}\left(\left\lfloor\frac{\tilde{N}}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{n}\right)^{t+1} + \frac{\pi}{4}K_{\theta,\varphi}\frac{1}{t!}\left(\left\lfloor\frac{\tilde{N}_i}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{n}\right)^{t+1} .
\end{aligned}
$$

With the same $K_\theta, K_{\theta,\varphi}$ as in 3.3.4. So we can directly engage a generalized quadrature alternative to 3.3.11. For this we use the definitions given in the proof of 3.3.11 and take a look at the occurring errors (the rest of the proof remains the same) :

$$
e_3(x) < K_\theta\frac{1}{t!}\left(\left\lfloor\frac{\tilde{N}}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{n}\right)^{t+1} + \frac{\pi}{4}K_{\theta,\varphi}\frac{1}{t!}\left(\left\lfloor\frac{\tilde{N}_i}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{n}\right)^{t+1} ,
$$

$$
\begin{aligned}
e_2(i,j,x,y) &< K_{\theta,\varphi,\lambda}\frac{1}{t!}\left(\left\lfloor\frac{\tilde{M}}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{m}\right)^{t+1}+\frac{\pi}{4}K_{\theta,\varphi,\lambda,\psi}\frac{1}{t!}\left(\left\lfloor\frac{\tilde{M}_i}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{m}\right)^{t+1}\\
&\leq \left(K_{\theta,\varphi,\lambda}+\frac{\pi}{4}K_{\theta,\varphi,\lambda,\psi}\right)\frac{1}{t!}\left(\left\lfloor\frac{\tilde{M}}{\underline{t}}\right\rfloor+1\right)\left(\frac{\underline{t}}{m}\right)^{t+1} =: \tilde{e}_2(x,y)\,,\\
e_2(x,y) &= \sum_{i=1}^{N}\sum_{j=1}^{N_i}\alpha_i\alpha_{i,j}e_2(i,j,x,y) \leq \sum_{i=1}^{N}\alpha_i\overbrace{\arctan(\sin(\theta_i))}^{\leq\frac{\pi}{4}}\tilde{e}_2(x,y)\\
&\leq \frac{\pi^2}{16}\tilde{e}_2(x,y)\,,\\
e_1(x,y) &= \sum_{\substack{(i,\,k)\,\in\,C\\(j,\,l)\,\in\,C_{ik}}}\alpha_i\alpha_{i,j}\alpha_k\alpha_{k,l}\frac{3}{2}Lc(\Delta v)^s\left\lfloor\frac{n}{q_i}\right\rfloor^s\left\lfloor\frac{m}{q_k}\right\rfloor^s\left(5q_i^2q_k^2+6q_i^4q_k^2\right)^{\frac{s}{2}}\tilde{K}_r\\
&\leq \sum_{\substack{(i,\,k)\,\in\,C\\(j,\,l)\,\in\,C_{ik}}}\alpha_i\alpha_{i,j}\alpha_k\alpha_{k,l}\frac{3}{2}Lc(\Delta v)^s(5+6q_i^2)^{\frac{s}{2}}n^sm^s\tilde{K}_r\\
&< \sum_{\substack{(i,\,k)\,\in\,C\\(j,\,l)\,\in\,C_{ik}}}\alpha_i\alpha_{i,j}\alpha_k\alpha_{k,l}\frac{3}{2}Lc(\Delta v)^s6^{\frac{s}{2}}(1+q_i^2)^{\frac{s}{2}}n^sm^s\tilde{K}_r\\
&\leq \sum_{\substack{(i,\,k)\,\in\,C\\(j,\,l)\,\in\,C_{ik}}}\alpha_i\alpha_{i,j}\alpha_k\alpha_{k,l}\frac{3}{2}Lc(\Delta v)^s6^{\frac{s}{2}}(2q_i^2)^{\frac{s}{2}}n^sm^s\tilde{K}_r\\
&\leq \sum_{\substack{(i,\,k)\,\in\,C\\(j,\,l)\,\in\,C_{ik}}}\alpha_i\alpha_{i,j}\alpha_k\alpha_{k,l}\frac{3}{2}Lc(\Delta v)^s3^{\frac{s}{2}}2^sn^{2s}m^s\tilde{K}_r\\
&\leq \frac{3\pi^4}{2^9}Lc(\Delta v)^s3^{\frac{s}{2}}2^sn^{2s}m^sK_r\,.
\end{aligned}
$$

Analog to the end of the last proof we obtain

$$
\begin{aligned}
e_3 &< 96\cdot\left(K_\theta+\frac{\pi}{4}K_{\theta,\varphi}\right)\frac{(t-1)^{t+1}}{t!}n^{1-t} && = c_1\left(K_\theta+\frac{\pi}{4}K_{\theta,\varphi}\right)n^{1-t}\\
e_2 &< 288\pi^2\cdot\left(K_{\theta,\varphi,\lambda}+\frac{\pi}{4}K_{\theta,\varphi,\lambda,\psi}\right)\frac{(t-1)^{t+1}}{t!}m^{1-t} && = c_2\left(K_{\theta,\varphi,\lambda}+\frac{\pi}{4}K_{\theta,\varphi,\lambda,\psi}\right)m^{1-t}\,,\\
e_1 &< \frac{27\pi^4}{2}Lc3^{\frac{s}{2}}2^sK_r(\Delta v)^sn^{2s}m^s && = c_3K_r(\Delta v)^sn^{2s}m^s\,,
\end{aligned}
$$

and take a look at

$$
\begin{aligned}
&e_1+e_2+e_3 < K(c_1n^{1-t}+c_2m^{1-t}+c_3(\Delta v)^sn^{2s}m^s)\\
&\implies c_1n^{1-t}=c_2m^{1-t}=c_3(\Delta v)^sn^{2s}m^s\\
&\qquad\implies m=\left(\frac{c_1}{c_2}\right)^{\frac{1}{1-t}}n
\end{aligned}
$$

$$
\begin{aligned}
\Longrightarrow n &= \left(\frac{c_3}{c_1}\right)^{\frac{1}{1-t-3s}} (\Delta v)^{\frac{s}{1-t-3s}} \left(\frac{c_1}{c_2}\right)^{\frac{s}{(1-t)(1-t-3s)}} \in \mathcal{O}\left((\Delta v)^{\frac{s}{1-t-3s}}\right) \\
\Longrightarrow c_1 n^{1-t} &= \text{const} \cdot \left((\Delta v)^{\frac{s}{1-t-3s}}\right)^{1-t} = \text{const} \cdot (\Delta v)^{\frac{s(t-1)}{3s+t-1}} \\
\Longrightarrow c_3 (\Delta v)^s n^{2s} m^s &= \text{const} \cdot (\Delta v)^s \left((\Delta v)^{\frac{s}{1-t-3s}}\right)^{2s} \left((\Delta v)^{\frac{s}{1-t-3s}}\right)^{s} \\
&= \text{const} \cdot (\Delta v)^{\frac{s(t-1)}{3s+t-1}} \\
\Longrightarrow e_1 + e_2 + e_3 &\in \mathcal{O}\left((\Delta v)^{\frac{s(t-1)}{3s+t-1}}\right) .
\end{aligned}
$$

The number of minimal points on any given line can be taken from an analog adaption of the proof of 3.3.12. □

Remark 3.4.4 (Interpolation region)

(i) At this point we can construct discretization schemes with "arbitrary" high convergence orders. This sounds good, but we have to take two things into consideration. The first problem we need to be aware of is that high order Lagrange interpolations can lead to negative integration weights resulting into an unstable scheme. This can possibly be avoided by applying other (more complex) interpolation approaches. The second problem arises from the question: "What is the minimal size of the velocity space to achieve a specific convergence order ?". The answer to this question is unsatisfactory, because at this point one can calculate (not a trivial calculation) that 22801 points in the 2D velocity space are necessary to reach a convergence order of 2. The following corollaries aim at a reduction of the necessary points by extending the interpolation (quadratures) onto regions of maximal size retaining the minimal properties of a DVM, but sacrificing some symmetries of the discretization.

(ii) The vigilant reader may has realized that the last corollary suffers from a fundamental flaw. It is obvious (see figure 3.5) that there exist θ_i (for example θ_0 in S_1) for which $|\mathfrak{F}_{i,n}| = 1$. This generally forbids to use quadrature formulas in $\varphi_\bullet$ or $\psi_\bullet$. Fortunately this little bug is easily curable by putting some symmetry regions together to form larger symmetry regions. We avoid a full discussion of this at this point, because we solve this problem in the corollaries below.

Corollary 3.4.5 (Extension of the interpolation region in 2D)
A discretization of the form

$$
\hat{I}[f](\mathfrak{v}) = \sum_{i=1}^{\hat{N}} \alpha_i \sum_{j=1}^{\hat{M}} \alpha_j \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\substack{\varphi_\alpha \,\in\, \{-id, id\} \\ \varphi_\beta \,\in\, \{id, \varphi_\gamma\}}} g(k) \left(\tilde{h}\left(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta\right)\right) ,
$$

with the angles

$$
\hat{\mathfrak{F}}_n := \mathfrak{F}_n \cup \left(\frac{\pi}{2} - \mathfrak{F}_n\right) \cup \left(\frac{\pi}{2} + \mathfrak{F}_n\right) \cup \left(\pi - \mathfrak{F}_n\right), \quad \left|\hat{\mathfrak{F}}_n\right| = \hat{N} = 4N - 3 , \tag{3.4.2}
$$

$$\hat{\mathfrak{F}}_m := \mathfrak{F}_m \cup \left(\frac{\pi}{2} - \mathfrak{F}_m\right) \cup \left(\frac{\pi}{2} + \mathfrak{F}_m\right) \cup (\pi - \mathfrak{F}_m)\,, \quad \left|\hat{\mathfrak{F}}_m\right| = \hat{M} = 4M - 3\,,$$

and the weights as given in 3.4.1 (exchanging N, M with $\hat{N}, \hat{M}$) can be transformed into a DVM and an eLGpM, possesses the minimal requirements and one can use nearly 4 times more points for the quadrature formulas than the aforementioned discretizations, significantly decreasing the number of necessary points in the velocity space. The used discretization points corresponding to $\hat{\mathfrak{F}}_n, \hat{\mathfrak{F}}_m$ can be calculated by

$$\begin{aligned}
\mathfrak{w}_3(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + k\Delta v \left(\varphi_\alpha \varphi_{x(i)} \left[P_{y(i)}, P^\perp_{y(i)}\right]\right) \varphi_\beta \varphi_{x(j)} P_{y(j)}\,,\\
\mathfrak{v}'_3(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + k\Delta v(\varphi_\beta \varphi_{x(j)} P_{y(j)})_1 \varphi_\alpha \varphi_{x(i)} P_{y(i)}\,,\\
\mathfrak{w}'_3(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta) &= \mathfrak{v} + \mathfrak{w}_3 - \mathfrak{v}'_3, \qquad P_i := \begin{pmatrix} q_i \\ p_i \end{pmatrix}, \quad P_i^\perp := \begin{pmatrix} -p_i \\ q_i \end{pmatrix}\,,
\end{aligned}$$

$$x(i) := \begin{cases} 1, & \text{if } 0 < i \leq N \\ 2, & \text{if } N < i \leq 2N-1 \\ 3, & \text{if } 2N-1 < i \leq 3N-2 \\ 4, & \text{if } 3N-2 < i \leq 4N-3 \end{cases}\,,$$

$$y(i) := \begin{cases} i, & \text{if } i \leq N \\ 2N - i, & \text{if } N \leq i \leq 2N-1 \\ i - 2N + 2, & \text{if } 2N-1 \leq i \leq 3N-2 \\ 4N - 2 - i, & \text{if } 3N-2 \leq i \leq 4N-3 \end{cases}\,,$$

according to 3.2.8. The corresponding functions for j are given by exchanging i and j as well as N and M in x, y.

Proof:
At first we have to identify which symmetry regions need to be separated in order to retain the minimal properties of a DVM. So we take a second look at the proof of 3.2.9. In this proof it can be seen that we only need the symmetry regions created by the operator $-$identity for θ and φ_γ for λ, because the symmetries corresponding to these operators are needed to obtain the necessary symmetries of the Operator $A^{\bullet,\bullet}_{\bullet,\bullet}$. This means that we can reduce the 8 different Symmetry regions to two symmetry regions with the property that φ_γ and $-$identity completely map one of these regions onto the other. From this argumentation follows that the largest symmetry regions are given by

$$\hat{S}_1 = S_1 \cup S_2 \cup S_3 \cup S_4 \text{ and } \hat{S}_2 = S_5 \cup S_6 \cup S_7 \cup S_8\,,$$

compare figure 3.7 and 3.5. This corresponds to the two operator sets

$$\begin{aligned}
G_1 &:= \left\{\varphi_1 := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \varphi_2 := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \varphi_3 := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \varphi_4 := \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\right\}\,,\\
G_2 &:= -id \cdot G_1\,,
\end{aligned}$$

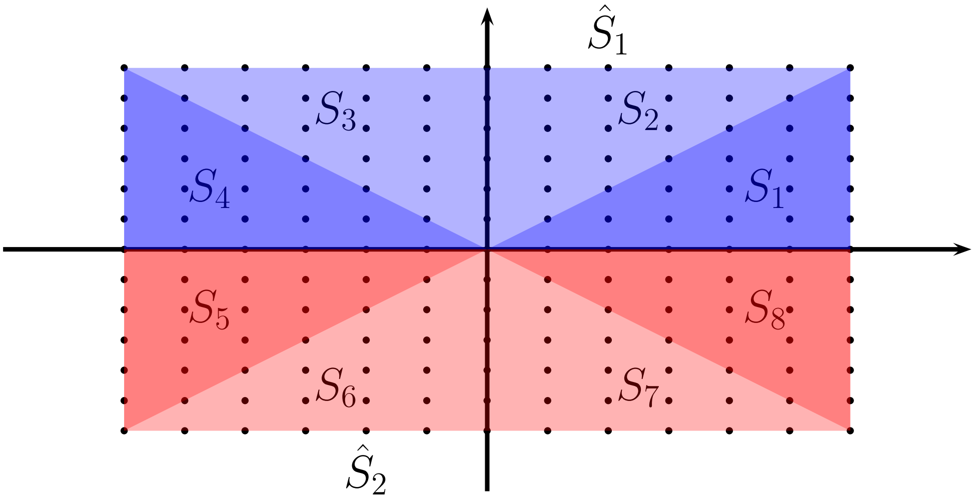


Figure 3.7: *Largest symmetry regions giving minimal requirements*

dividing G into the two equivalence classes $[id], [-id]$. The usable angles for θ, λ are now given by

$$\hat{\mathfrak{F}}_n := \mathfrak{F}_n \cup \left(\frac{\pi}{2} - \mathfrak{F}_n\right) \cup \left(\frac{\pi}{2} + \mathfrak{F}_n\right) \cup (\pi - \mathfrak{F}_n)\,, \quad \left|\hat{\mathfrak{F}}_n\right| = 4N - 3\,,$$
$$\hat{\mathfrak{F}}_m := \mathfrak{F}_m \cup \left(\frac{\pi}{2} - \mathfrak{F}_m\right) \cup \left(\frac{\pi}{2} + \mathfrak{F}_m\right) \cup (\pi - \mathfrak{F}_m)\,, \quad \left|\hat{\mathfrak{F}}_m\right| = 4M - 3\,.$$

These symmetry regions are given for the approximation above λ, because these are the only ones with the property that φ_γ (reflection on x-axis) maps these regions onto each other. We could choose other regions for the approximation in θ, because here we only need the symmetry corresponding to $-id$. So we could use any symmetry axis of the given grid to divide the grid into two symmetry regions. In order to simplify things we choose the same symmetry regions for both approximations. These considerations become more complex and meaningful in the next corollary doing the same in three dimensions. The usage of these regions result into a discretization of the form

$$\hat{I}[f](\mathfrak{v}) = \sum_{i=1}^{\hat{N}} \alpha_i \sum_{j=1}^{\hat{M}} \alpha_j \Delta v_{ij} \sum_{k=0}^{\lfloor L/\Delta v_{ij} \rfloor} \sum_{\substack{\varphi_\alpha \in \{-id, id\} \\ \varphi_\beta \in \{id, \varphi_\gamma\}}} g(k) \left(\tilde{h}\left(l_k, \lambda_j, \theta_i, \varphi_\alpha, \varphi_\beta\right)\right)\,.$$

Here we can use the simplifications given in 3.2.8 to calculate the used discretization points. This discretization retains only the symmetries $(-I, \varphi_g)$ necessary to obtain a DVM with the minimal requirements (see 3.2.9) and gives almost 4 times more angular points for the interpolation in the quadrature formulas. All other results can be taken from the proof of 3.4.1 with minimal modifications. □

Corollary 3.4.6 (Extension of the interpolation region in 3D)
A discretization of the form

$$\begin{aligned}\tilde{I}[f](\mathfrak{v}) = &\sum_{(i,k)\in\hat{B}}\sum_{(j,l)\in\hat{C}_{i,k}} \alpha^{1,1}_{ijkl} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l}\rfloor}\sum_{\alpha,\beta\in A_3}\begin{bmatrix} g(q)h(\theta_i^{(1)},\varphi_{i,j}^{(1)},\lambda_k^{(1)},\psi_{k,l}^{(1)},r_q,\alpha,\beta)\\ \cdot D(r_q,\theta_i^{(1)},\varphi_{i,j}^{(1)},\psi_{k,l}^{(1)})\end{bmatrix}\\
+ &\sum_{(i,k)\in\hat{B}}\sum_{(j,l)\in\hat{C}_{i,k}} \alpha^{1,2}_{ijkl} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l}\rfloor}\sum_{\alpha\in A_3,\beta\in A_2}\begin{bmatrix} g(q)h(\theta_i^{(1)},\varphi_{i,j}^{(1)},\lambda_k^{(2)},\psi_{k,l}^{(2)},r_q,\alpha,\beta)\\ \cdot D(r_q,\theta_i^{(1)},\varphi_{i,j}^{(1)},\psi_{k,l}^{(2)})\end{bmatrix}\\
+ &\sum_{(i,k)\in\hat{B}}\sum_{(j,l)\in\hat{C}_{i,k}} \alpha^{2,1}_{ijkl} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l}\rfloor}\sum_{\alpha\in A_1,\beta\in A_3}\begin{bmatrix} g(q)h(\theta_i^{(2)},\varphi_{i,j}^{(2)},\lambda_k^{(1)},\psi_{k,l}^{(1)},r_q,\alpha,\beta)\\ \cdot D(r_q,\theta_i^{(2)},\varphi_{i,j}^{(2)},\psi_{k,l}^{(1)})\end{bmatrix}\\
+ &\sum_{(i,k)\in\hat{B}}\sum_{(j,l)\in\hat{C}_{i,k}} \alpha^{2,2}_{ijkl} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l}\rfloor}\sum_{\alpha\in A_1,\beta\in A_2}\begin{bmatrix} g(q)h(\theta_i^{(2)},\varphi_{i,j}^{(2)},\lambda_k^{(2)},\psi_{k,l}^{(2)},r_q,\alpha,\beta)\\ \cdot D(r_q,\theta_i^{(2)},\varphi_{i,j}^{(2)},\psi_{k,l}^{(2)})\end{bmatrix},\\
A_1 := \{id,-id\},\qquad & A_2 := \{id,\gamma\},\qquad A_3 := A_1\cup A_2\,,\end{aligned}$$

using the angles corresponding to the regions given in figure 3.8

$$\begin{aligned}&\hat{\mathfrak{F}}^1_n := \mathfrak{F}_n \cup \left(\frac{\pi}{2}-\mathfrak{F}_n\right),\quad \hat{\mathfrak{F}}^1_{i,n} := \begin{cases}\mathfrak{F}_{i,n}\cup -\mathfrak{F}_{i,n}, & \text{if } i\le N\\ \mathfrak{F}_{2N-1-i,n}\cup -\mathfrak{F}_{2N-1-i,n}, & \text{if } i> N\end{cases},\\
&\hat{\mathfrak{F}}^3_n := \mathfrak{F}_n \cup (-\mathfrak{F}_n)\,,\quad \hat{\mathfrak{F}}^3_{i,n} := \begin{cases}\mathfrak{F}_{N-i,n}\cup -\mathfrak{F}_{N-i,n}, & \text{if } i\le N\\ \mathfrak{F}_{i,n}\cup -\mathfrak{F}_{i,n}, & \text{if } i> N\end{cases},\\
&|\hat{\mathfrak{F}}_n| = 2N-1,\qquad |\hat{\mathfrak{F}}_{i,n}| = 2N_i-1\,,\end{aligned}$$

and standard quadrature formulas based on Lagrangian polynomials, can be transformed into a DVM and an eLGpM, possesses the minimal requirements and one can use nearly 2 times more points for the quadrature formulas than the aforementioned discretizations, significantly decreasing the number of necessary points in the velocity space. Here we used a slightly changed definition of $\mathfrak{F}_{i,n}$ (see proof) guaranteeing that there exist enough points for quadrature formulas.

Proof:
As already mentioned in 3.4.4 we can not use quadrature formulas over S_1 (as given in figure 3.5), due to the lack of discretization points in φ for some specific values of θ, for example $\theta = 0$. This results into new (larger) smallest symmetry regions that are usable for quadrature formulas. These are obviously $S_1 \cup S_6, S_2 \cup S_3, S_4 \cup S_5$, because by applying a discretization using the Farey angles $\mathfrak{F}_n$ these symmetry regions give N points for the quadrature in θ and at least n points for the quadrature in φ. Now we are searching for the largest symmetry regions which result into a DVM possessing the minimal requirements. As can be seen in 3.3.13, these symmetry regions must possess

the property that the operators $-id$ and γ (point reflection on zero and reflection on the x-axis) map every symmetry region completely onto another one. The optimal decomposition of the velocity space into symmetry regions would be one dividing the space into two regions alongside a symmetry plane in such a way that the symmetries corresponding to $-id$ and γ survive and that we can use quadrature formulas above these regions. Unfortunately this is not possible within our framework, because a single quadrature in φ above $S_1 \cup S_6 \cup S_5$ is not possible due to the fact that $\forall \varphi \in \mathfrak{F}_1 : \varphi < \frac{\pi}{4}$. Or in other words: there exist θ for which no new φ appear when using S_5 in addition to S_6, S_1. Dividing the velocity space into symmetry regions $\hat{S}$ consisting of multiple smallest symmetry regions $S_\bullet$ in such a way that the new regions consist of the same number of angles in θ and φ and with respect to the symmetries that need to be retained $(-id, \gamma)$ we obtain the three regions $\hat{S}_1, \hat{S}_2, \hat{S}_3$ which are given in figure 3.8.

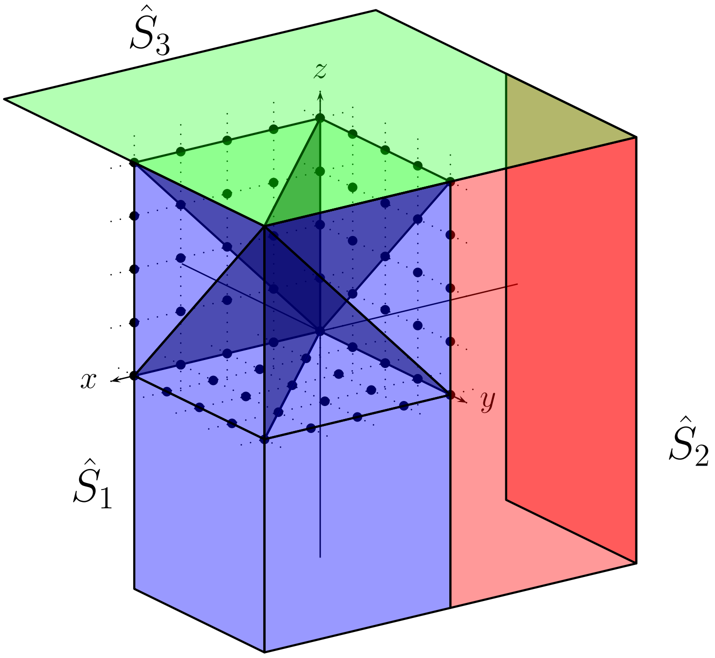


Figure 3.8: *Largest symmetry regions giving the minimal requirements and usable angles for quadrature formulas*

Here every symmetry region $\hat{S}_\bullet$ consists of 8 smallest regions $S_\bullet$ and possesses $2N-1$ angles in θ and at least $2n-1$ angles in φ. As one can see we break with the tradition to use quadrature regions which can be mapped onto each other by using elements of the automorphism group. So at this point we loose a rather large amount of symmetries within our discretization, but at the same time we specifically retain $-id, \gamma$. The usable angles for the three regions are given by

$$\hat{\mathfrak{F}}_n^1 := \mathfrak{F}_n \cup \left(\frac{\pi}{2} - \mathfrak{F}_n\right), \quad \hat{\mathfrak{F}}_{i,n}^1 := \begin{cases} \mathfrak{F}_{i,n} \cup -\mathfrak{F}_{i,n}, & \text{if } i \leq N \\ \mathfrak{F}_{2N-1-i,n} \cup -\mathfrak{F}_{2N-1-i,n}, & \text{if } i > N \end{cases},$$

$$\hat{\mathfrak{F}}_n^3 := \mathfrak{F}_n \cup (-\mathfrak{F}_n)\,, \quad \hat{\mathfrak{F}}_{i,n}^3 := \begin{cases} \mathfrak{F}_{N-i,n} \cup -\mathfrak{F}_{N-i,n}, & \text{if } i \leq N \\ \mathfrak{F}_{i,n} \cup -\mathfrak{F}_{i,n}, & \text{if } i > N \end{cases}.$$

The discretization points for the second angle are given by

$$\tilde{\mathfrak{F}}_{i,n} := \left(\left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j}\right) \middle| \begin{array}{l} p_{i,j} \leq q_{i,j}, p \in \mathbb{N}_0, t \in \left\{1, \ldots, \left\lfloor \frac{n}{q_i} \right\rfloor\right\}, q = t \cdot q_i : \\ 0 \leq p \leq t \cdot q_i \ \wedge \ \gcd(p_{i,j}, q_{i,j}) = 1 \ \wedge \\ \left(\frac{p_{i,j}}{q_{i,j}}, t_{i,j}\right) = \left(\frac{p}{q}, t\right) \ \wedge \ \forall j > 1 : \frac{p_{i,j-1}}{q_{i,j-1}} < \frac{p_{i,j}}{q_{i,j}} \\ \wedge \text{ when multiple } t \text{ s are possible choose the smallest one} \end{array} \right)$$

$$= \left(\left(\frac{p_{i,1}}{q_{i,1}}, t_{i,1}\right), \ldots, \left(\frac{p_{i,N_i}}{q_{i,N_i}}, t_{i,N_i}\right) \right) = \left((F_{i,1}, t_{i,1}), \ldots, (F_{i,N_i}, t_{i,N_i})\right), \quad N_i := |\tilde{\mathfrak{F}}_{i,n}|$$

$$\varphi_{i,j} := \arctan\left(F_{i,j} \frac{q_i}{\sqrt{q_i^2 + p_i^2}}\right), \ \mathfrak{F}_{i,n} := \{\varphi_{i,j} | j = 1, \ldots, N_i\},$$

(compare 3.3.2) and can now be used to apply quadrature formulas. The integral corresponding to $\hat{S}_2$ can be transformed to have the domain of integration $\hat{S}_1$ resulting into the usability of the same angles. The integral corresponding to $\hat{S}_3$ can be transformed in such a way that the domain of integration rotates 90° around the y-axis where the angles $\hat{\mathfrak{F}}_n^3, \hat{\mathfrak{F}}_{i,n}^3$ can be used. This results into a discretization of the form

$$\begin{aligned} \tilde{I}[f](\mathfrak{v}) = & \sum_{(i,k)\in\hat{B}} \sum_{(j,l)\in\hat{C}_{i,k}} \alpha_{ijkl}^{1,1} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l} \rfloor} \sum_{\alpha,\beta\in A_3} \left[\begin{array}{c} g(q) h(\theta_i^{(1)}, \varphi_{i,j}^{(1)}, \lambda_k^{(1)}, \psi_{k,l}^{(1)}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_i^{(1)}, \varphi_{i,j}^{(1)}, \psi_{k,l}^{(1)}) \end{array} \right] \\ & + \sum_{(i,k)\in\hat{B}} \sum_{(j,l)\in\hat{C}_{i,k}} \alpha_{ijkl}^{1,2} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l} \rfloor} \sum_{\alpha\in A_3,\beta\in A_2} \left[\begin{array}{c} g(q) h(\theta_i^{(1)}, \varphi_{i,j}^{(1)}, \lambda_k^{(2)}, \psi_{k,l}^{(2)}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_i^{(1)}, \varphi_{i,j}^{(1)}, \psi_{k,l}^{(2)}) \end{array} \right] \\ & + \sum_{(i,k)\in\hat{B}} \sum_{(j,l)\in\hat{C}_{i,k}} \alpha_{ijkl}^{2,1} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l} \rfloor} \sum_{\alpha\in A_1,\beta\in A_3} \left[\begin{array}{c} g(q) h(\theta_i^{(2)}, \varphi_{i,j}^{(2)}, \lambda_k^{(1)}, \psi_{k,l}^{(1)}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_i^{(2)}, \varphi_{i,j}^{(2)}, \psi_{k,l}^{(1)}) \end{array} \right] \\ & + \sum_{(i,k)\in\hat{B}} \sum_{(j,l)\in\hat{C}_{i,k}} \alpha_{ijkl}^{2,2} L_{i,j,k,l} \sum_{q=0}^{\lfloor L/\Delta v_{i,j,k,l} \rfloor} \sum_{\alpha\in A_1,\beta\in A_2} \left[\begin{array}{c} g(q) h(\theta_i^{(2)}, \varphi_{i,j}^{(2)}, \lambda_k^{(2)}, \psi_{k,l}^{(2)}, r_q, \alpha, \beta) \\ \cdot D(r_q, \theta_i^{(2)}, \varphi_{i,j}^{(2)}, \psi_{k,l}^{(2)}) \end{array} \right], \\ & A_1 := \{id, -id\}, \qquad A_2 := \{id, \gamma\}, \qquad A_3 := A_1 \cup A_2\,. \end{aligned}$$

This complicated looking discretization comes from the situation that we have to integrate over the symmetry regions for two spherical integrals (one in θ, φ and one in λ, ψ), so we have the combinations $(\hat{S}_1 \cup \hat{S}_2, \hat{S}_1 \cup \hat{S}_2)$, $(\hat{S}_1 \cup \hat{S}_2, \hat{S}_3)$, $(\hat{S}_3, \hat{S}_1 \cup \hat{S}_2)$, $(\hat{S}_3, \hat{S}_3)$ for the domains of integration resulting into these four sums. Here we have $\theta_i^{(1)} \in \hat{\mathfrak{F}}_n^1, \lambda_k^{(1)} \in \hat{\mathfrak{F}}_m^1, \varphi_{i,j}^{(1)} \in \hat{\mathfrak{F}}_{i,n}^1, \psi_{k,l}^{(1)} \in \hat{\mathfrak{F}}_{k,m}^1$, $\theta_i^{(2)} \in \hat{\mathfrak{F}}_n^3, \lambda_k^{(2)} \in \hat{\mathfrak{F}}_m^3, \varphi_{i,j}^{(2)} \in \hat{\mathfrak{F}}_{i,n}^3, \psi_{k,l}^{(2)} \in \hat{\mathfrak{F}}_{k,m}^3$ and the αs correspond to the coefficients of the quadrature formulas applied and can be calculated analog to 3.4.3. This discretization possesses the necessary symmetries to result into a DVM possessing the minimal requirements and we can use $2 \cdot 2 \cdot 2 \cdot 2 = 16$ times more points for the quadrature formulas (two for every scalar angular integral involved). □

Remark 3.4.7

(i) The discretization points as well as the interpolation weights $\alpha_\bullet$ can be easily calculated by an adaption of 3.3.11,3.4.3 analog to the two dimensional case 3.4.5.

(ii) Now we have nearly reached the "final" version of our discretization scheme alongside the corresponding DVM and eLGpM, ready to be tested within the next chapter. We want to point out, that the integration weights $\alpha_\bullet$ can be precomputed for every used Farey sequence with arbitrary precision. So the calculation of these weights does not slows down the resulting algorithm for the approximation of the collision operator.

(iii) We also want to point out that the symmetry regions can be further extended, especially over the symmetry axis necessary to obtain the minimal requirements if we choose the quadrature rule in such a way that the weights retain the necessary symmetry. For example we could use

$$\hat{\mathfrak{F}}_n \cup \left(\pi + \hat{\mathfrak{F}}_n\right)$$

in the two dimensional discretization as long as the corresponding weights

$$(\alpha_1, \ldots, \alpha_{\hat{N}}, \ldots, \alpha_{2\hat{N}-1})$$

are symmetric with respect to $\alpha_{\hat{N}}$.

4 Numerical analysis

In this chapter we want to investigate some properties of the developed discretizations. So we take a look at the convergence orders for some given velocity lattices and we verify the exact conservation of the moments by numerically solving stationary problems and calculating the reached convergence orders and moment changes.

4.1 Minimal velocity space sizes

We begin with a theoretical review of the question: "What is the minimal size of the velocity space to obtain a specific convergence order ?". The answer of this question will give us the last necessary tool to construct discretizations with given convergence orders.

Corollary 4.1.1 (Minimal size to reach specific convergence in 2D)
Using the discretization 3.4.5 to reach a convergence order of

$$o = \frac{rs}{2r+s}\,,$$

by applying a Newton-Cotes formula of order r (using $\tilde{r}$ points for the interpolation polynomials in the quadrature) and quadrature formulas with polynomials of order s (thus using $t = s+1$ points for them) the minimal size of the velocity space is given by

$$\begin{aligned} |\mathfrak{V}| &= (1 + 2 \cdot (\tilde{r} - 1) \cdot 2n^2)^2\,, \\ n &= \max\left\{ \left(\frac{10(t-1)^{t+1}}{3\pi c t! 2^r} \right)^{\frac{1}{s+2r}} (\Delta v)^{\frac{-r}{s+2r}}, \min\left\{ n \in \mathbb{N} \,\middle|\, t \leq 4 \sum_{k=1}^{n} \varphi(k) - 3 \right\} \right\}\,. \end{aligned} \tag{4.1.1}$$

Here c corresponds to the error constant of the used Newton-Cotes formula and we assume that we approximate both angular integrals using a Farey sequence of order n (implying $m = n$). Moreover $\varphi(n), n \in \mathbb{N}$ is Euler's totient function.

Proof:
Using the discretization 3.4.5 and redoing the proof of 3.4.1 with $\hat{N} = 4N - 3, \hat{M} = 4M - 3$ points for the interpolation results into the error estimation

$$\begin{aligned} e < {} & \frac{K_\theta}{t!} \left(\left\lfloor \frac{\hat{N} - 1}{s} \right\rfloor + 1 \right) \left(\frac{s}{n} \right)^{t+1} + \pi \frac{K_\lambda}{t!} \left(\left\lfloor \frac{\hat{M} - 1}{s} \right\rfloor + 1 \right) \left(\frac{s}{m} \right)^{t+1} \\ & + \frac{3\pi^2}{2} c L (\Delta v)^r K_l 2^r n^r m^r\,, \end{aligned}$$

compare (3.4.1). Here t corresponds to the number of points used for the interpolation, so $t \geq 2$ and s to the order of the polynomial $s = t - 1$. For the following estimations we use

$$\begin{aligned}
K_l &:= \sup_{\substack{l \in [0, L] \\ \theta, \lambda \in [0, 2\pi]}} \left| \frac{\partial^r h(l, \lambda, \theta)}{\partial l^r} \right| , \\
L\hat{K}_\lambda &:= L \sup_{\substack{l \in [0, L] \\ \theta, \lambda \in [0, 2\pi]}} \left| \frac{\partial^t h(l, \lambda, \theta)}{\partial \lambda^t} \right| > \sup_{\theta,\lambda \in [0,2\pi]} \left| \frac{\partial^t \int_0^L h(l, \lambda, \theta) \mathrm{d}l}{\partial \lambda^t} \right| = K_\lambda , \\
L\pi\hat{K}_\theta &:= L\pi \sup_{\substack{l \in [0, L] \\ \theta, \lambda \in [0, 2\pi]}} \left| \frac{\partial^t h(l, \lambda, \theta)}{\partial \theta^t} \right| > \sup_{\theta \in [0,2\pi]} \left| \frac{\partial^t \int_0^\pi \int_0^L h(l, \lambda, \theta) \mathrm{d}l \mathrm{d}\lambda}{\partial \theta^t} \right| = K_\theta , \\
\hat{N} &= 4|\mathfrak{F}_n| - 3 < 4 + 2n^2 + 2n - 3, \qquad \hat{M} = 4|\mathfrak{F}_m| - 3 < 4 + 2m^2 + 2n - 3 ,
\end{aligned}$$

giving

$$\begin{aligned}
e &< L\pi\hat{K}_\theta \frac{s^{t+1}}{t!} \left(\left\lfloor \frac{2n^2 + 2n}{s} \right\rfloor + 1 \right) \frac{1}{n^{t+1}} \\
&+ L\pi\hat{K}_\lambda \frac{s^{t+1}}{t!} \left(\left\lfloor \frac{2m^2 + 2m}{s} \right\rfloor + 1 \right) \frac{1}{m^{t+1}} \\
&+ \frac{3\pi^2}{2} cL(\Delta v)^r K_l 2^r n^r m^r \\
&< L\pi \left(\hat{K}_\theta \overbrace{5\frac{s^{t+1}}{t!}}^{c_1:=} \frac{1}{n^s} + \hat{K}_\lambda \overbrace{5\frac{s^{t+1}}{t!}}^{c_2:=} \frac{1}{m^s} + K_l \overbrace{\frac{3\pi}{2} c 2^r}^{c_3:=} (\Delta v)^r n^r m^r \right) ,
\end{aligned}$$

because

$$\left(\left\lfloor \frac{2n^2 + 2n}{s} \right\rfloor + 1 \right) \overset{s \geq 1}{\leq} \left(4n^2 + 1 \right) \leq 5n^2 .$$

Now we do the rest analog to the proof of 3.4.1, giving the same result:
Assuming that $\Delta v \in \mathbb{R}^+, r \in \mathbb{N}, t \in \mathbb{N}_{>1}$ are given constants satisfying

$$\frac{L}{2\Delta v \tilde{n}(\Delta v, r) \tilde{m}(\Delta v, r)} > r, \ \wedge \ t \leq \hat{N}, \hat{M} ,$$

with

$$\tilde{n}(\Delta v, r) = \left(\frac{c_3}{c_1} \right)^{\frac{-1}{s+2r}} (\Delta v)^{\frac{-r}{s+2r}}, \quad \tilde{m}(\Delta v, r) = n \tag{4.1.2}$$

and choosing n, m according to

$$n = \lceil \tilde{n}(\Delta v, r) \rceil, \ m = \lceil \tilde{m}(\Delta v, r) \rceil ,$$

the convergence order of this discretization is

$$\mathcal{O}\left((\Delta v)^{\frac{rs}{2r+s}}\right) .$$

The minimal number of points $\tilde{r}$ on any given line associated with the approximation of the innermost integral grows asymptotically as is given by

$$\tilde{r}(\Delta v) \in \mathcal{O}\left((\Delta v)^{\frac{-s}{2r+s}}\right) .$$

Our aim is to calculate the minimum size of the velocity space to reach a specific convergence order. So let es assume that we use a Newton-Cotes formula of order r (resulting into the usage of at least $\tilde{r}$ points on every line) and quadrature formulas with polynomials of order s (resulting into the usage of at least $t = s + 1$ angles), so that we potentially reach a convergence order of $\frac{-s}{2r+s}$. Now our used Farey sequences have to possess the property

$$n \geq \left(\frac{c_3}{c_1}\right)^{\frac{-1}{s+2r}} (\Delta v)^{\frac{-r}{s+2r}}, \qquad m \geq n .$$

Moreover we need at least t angles per angular integral and $\tilde{r}$ points on every resulting line. This gives the additional restriction

$$n \geq \min\left\{ n \in \mathbb{N} \,\middle|\, t \leq 4\left|\mathfrak{F}_n\right| - 3 = 4N - 3 = 4\sum_{k=1}^{n} \varphi(k) - 3 \right\}, \qquad m \geq n .$$

At this points we determined the necessary orders of the Farey sequences to use. From this we can deduce the necessary size of the velocity space. The approximation of the outermost integral (over θ) restricts the rest of the approximation to live on the sub lattice

$$\left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{Z}^2 \,\middle|\, \begin{pmatrix} x \\ y \end{pmatrix} = \tilde{x} P_i + \tilde{y} P_i^{\perp},\ \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \in \mathbb{Z}^2 \right\} .$$

Now the application of another Farey sequence of order m results into using the points

$$P_{i,j} = q_j P_i + p_j P_i^{\perp} = \begin{pmatrix} q_i q_j - p_i p_j \\ p_i q_j + q_i p_j \end{pmatrix} ,$$

for the approximation. So we need at least $\tilde{r} \cdot (n^2 + m^2)$ points in every coordinate (including zero) resulting in a minimal grid size of

$$(1 + 2 \cdot (\tilde{r} - 1) \cdot (n^2 + m^2))^2 .$$

□

Remark 4.1.2 (Interpretation of minimal velocity space size)

(i) To understand the implications of the last corollary we calculate the convergence orders as well as the corresponding minimal velocity space sizes. For this we assume that $L = 15$ (the only free parameter). The results can be found in table 4.1. Some specific sweet spots, characterized by a maximal convergence order for a given velocity space size are:

$r\diagdown t$	2		3		4		5		6		7	
	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$
2	0.40	25	0.66	25	0.86	289	1.00	289	1.11	289	1.20	289
4	0.44	81	0.80	81	1.09	81	1.33	81	1.54	1089	1.71	1089
6	0.46	289	0.86	289	1.20	289	1.50	289	1.76	4225	2.00	4225

Table 4.1: *convergence order o and minimal size of velocity space $|\mathfrak{V}|$ above order of Newton Cotes formula r and points used for angular quadratures t*

$$
\begin{aligned}
&|\mathfrak{V}| = 25 \Longrightarrow r = 2 \wedge t = 5 \Longrightarrow o = 1\,,\\
&|\mathfrak{V}| = 81 \Longrightarrow r = 4 \wedge t = 5 \Longrightarrow o = 1 + \frac{1}{3}\,,\\
&|\mathfrak{V}| = 289 \Longrightarrow r = 2 \wedge t = 9 \Longrightarrow o = 1 + \frac{1}{3}; \qquad r = 6 \wedge t = 5 \Longrightarrow o = 1.5\,,\\
&|\mathfrak{V}| = 1089 \Longrightarrow r = 4 \wedge t = 9 \Longrightarrow o = 2\,.
\end{aligned}
$$

Here it remains to be seen if the corresponding high order polynomials suffer from oscillations or lead to negative integration weights.

(ii) The minimal size of the velocity space greatly depends on the necessary order of the used Farey sequences. This order is determined by the maximum of the two variables

$$
\left(\frac{8(t-1)^{t+1}}{3\pi c t! 2^r}\right)^{\frac{1}{s+2r}} (\Delta v)^{\frac{-r}{s+2r}}\,, \quad \min\left\{ n \in \mathbb{N} \,\middle|\, t \leq 4\sum_{k=1}^{n} \varphi(k) - 3 \right\}\,.
$$

The second one corresponds to the necessity of having at least enough angles for the angular quadrature formulas and the first one comes from the approach that we determine the convergence order by assuming that the errors of the three integral approximations are approximately the same. So as long as the error of the Newton-Cotes formula (integration along a line) remains large enough only the second variable determines the order of the Farey sequence and by this the size of the necessary velocity space.

Remark 4.1.3 (Minimal velocity space size in 3D)

(i) Analog considerations to the two dimensional case give:
Using the discretization 3.4.6 to reach a convergence order of

$$o = \frac{s\underline{t}}{3s + \underline{t}},$$

by applying a Newton-Cotes formula of order s (using $\tilde{s}$ points for the interpolation polynomials in the quadrature) and quadrature formulas with polynomials of order $\underline{t}$ (thus using $t = \underline{t} + 1$ points for them) the minimal size of the velocity space is given by

$$|\mathfrak{V}| = (1 + 2 \cdot (\tilde{s} - 1) \cdot (n^2 + 2n^3))^3 ,$$
$$n = \max\left\{ \left(\frac{8(t-1)^{t+1}}{3\pi 3^{\frac{s}{2}} ct! 2^s} \right)^{\frac{1}{\underline{t}+3s}} (\Delta v)^{\frac{-s}{\underline{t}+3s}} , \min\{n \in \mathbb{N} \,|t \leq 2n+1\} \right\} .$$

Here c corresponds to the error constant of the used Newton-Cotes formula and we assume that we approximate both angular integrals using a Farey sequence of order n (implying $m = n$).

(ii) As in the two dimensional case we calculate the convergence orders as well as the corresponding minimal velocity space sizes. The results can be found in table 4.2. We can see that we need a huge amount of points in the velocity space to even reach a convergence order of 1. This result can be substantially improved if we extend the quadrature regions by applying quadrature rules with the necessary symmetries, see remark 3.4.7(iii).

$s \diagdown t$	2		3		4		5	
	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$
2	0.29	343	0.50	343	0.66	68921	0.80	68921
4	0.31	2197	0.57	2197	0.80	$> 10^5$	1.00	$> 10^5$
6	0.32	15625	0.60	15625	0.86	$> 10^5$	1.09	$> 10^5$

Table 4.2: *convergence order o and minimal size of velocity space $|\mathfrak{V}|$ above order of Newton Cotes formula r and points used for angular quadratures t*

4.2 Implementation and test of the discretization

4.2.1 Adjustments of the discretization

In this section we take a look at the approximation of the collision operator for a test problem. For all following calculations we choose the mass distribution

$$f(\mathfrak{v}) := \frac{1}{2\pi} \cdot e^{-\frac{\|\mathfrak{v}\|^2}{2}} + \frac{1}{2\pi} \cdot e^{-\frac{\|\mathfrak{v}+E\|^2}{2}}, \quad E := \begin{pmatrix} 2 \\ 3 \end{pmatrix} .$$

And we restrict the region (integration region) to

$$A := [-7.5, 7.5] \times [-7.5, 7.5], \qquad L := \sqrt{2} \cdot 15 .$$

The distribution function is given in figure 4.1. Our convergence results are generally asymptotic results, so it is foreseeable that we need a huge amount of discretization points to obtain the proven convergence rates. Due to this we restrict our approximation of the collision integral to the point $\mathfrak{v}_0 = (0,0)^T$, because otherwise the calculation time would exceed the amount of time we are willing to spend for a numerical verification. This implies that we are looking at an approximation of (compare 3.2.1)

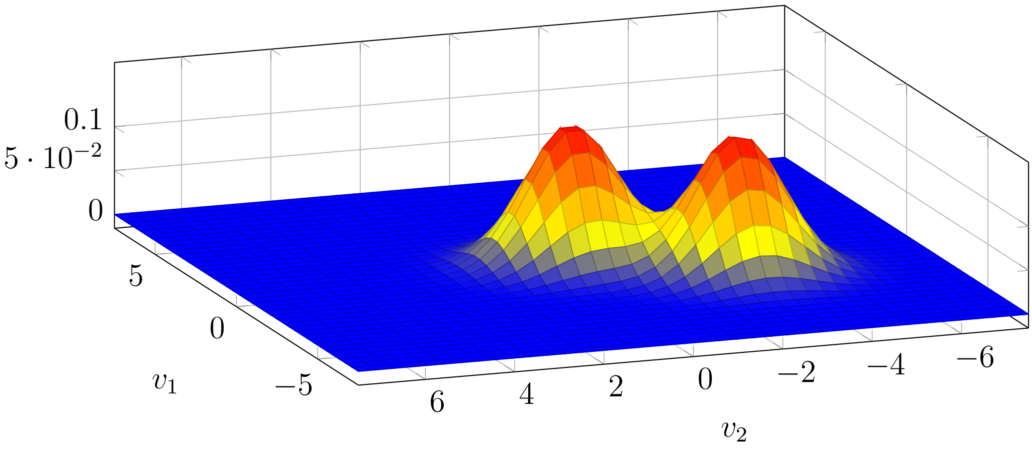


Figure 4.1: *Distribution function used for numerical calculations*

$$I[f](\mathfrak{v}_0) = \int_0^{2\pi} \int_0^{2\pi} \int_0^{L} \left[f\left(\mathfrak{v}_2'\right) f\left(\mathfrak{w}_2'\right) - f\left(\mathfrak{v}\right) f\left(\mathfrak{w}_2\right) \right] l \, \mathrm{d}l \, \mathrm{d}\lambda \, \mathrm{d}\theta \; ,$$

with

$$\mathfrak{v}_2' := \mathfrak{v} + l\langle \omega(\theta+\lambda), \omega(\theta) \rangle \omega(\theta), \qquad \mathfrak{w}_2' := \mathfrak{v} + l\langle \omega(\theta+\lambda), \omega^\perp(\theta) \rangle \omega^\perp(\theta) \; ,$$
$$\mathfrak{w}_2 := \mathfrak{v} + l\omega(\theta+\lambda) \; .$$

We calculate a reference solution for the following error calculation. Before we start this calculation we have to take into account that our numerical algorithm ignores possible cut off errors, so our reference solution should do the same. To do this we restrict f onto the region A by setting it to zero outside of this region:

$$\tilde{f}(\mathfrak{v}) := \mathbb{1}_A(\mathfrak{v}) \cdot f(\mathfrak{v}) \, .$$

An adaptive Clenshaw–Curtis quadrature yields

$$I[f](\mathfrak{v}_0) \approx -0.537037185 =: I_{\mathrm{cc}} \, .$$

Here we applied the standard algorithm in Octave using a relative error of 10^{-8} and an adaptive Gaussian quadrature to verify the result. The main difference between such rules and our approach is that Clenshaw-Curtis needs a function (integrand) for which it can calculate arbitrary discretization points whereas our approach works on a uniform discretization of the velocity space. Before we proceed we recollect the former results by giving a rough step by step guide for the creation of a discretization in the form of an algorithm.

Algorithm 4.2.1.1

(i) Start: input values

  a) L_{num}: desired number of discretization points in one dimension (we use a quadratic grid)

  b) r: order of the Newton-Cotes formula

  c) s: order of the polynomial used for angular quadrature

  d) L: length of one edge of the quadratic integration domain (the discretization parameter is now given by $\Delta v = \frac{L}{L_{\mathrm{num}}-1}$)

(ii) Calculation of the necessary order of the Farey sequence n (see (4.1.1)) to obtain the proved convergence behavior and possible correction of L_{num} in order to have enough angles. We also calculate the velocity grid $\mathfrak{V}$ as well as a predefined mass distribution on this grid $f : \mathfrak{V} \to \mathbb{R}^+$.

(iii) Calculation of all usable angles based on the Farey angles, of all used discretization points (and corresponding collision pairs $\mathfrak{v}, \mathfrak{w}, \mathfrak{v}', \mathfrak{w}'$), of the discretization weights $\alpha_\bullet$, determination of the "smallest" grid point $\|\mathfrak{v}\| \to \min$ on every line given through the farey angles in order to determine the step size of the Newton Cotes formula and finally the calculation of the integration weights corresponding to the Newton Cotes formula. We calculate all according to 3.4.5 and only in the point $\mathfrak{o}$.

(iv) Now we condense the collision pairs by kicking double occurrences of (i, j, k, l) as well as (i, j, l, k) (using the symmetry $A^{k,l}_{i,j} = A^{l,k}_{i,j}$). This can be done by neglecting the additional occurrences and summing the corresponding integration weights.

(v) At this point we have obtained a discretization of the collision operator that can be used for all further evaluations of this operator. This scheme is defined by the used discretization points as well as the integration weights $A_{i,j}^{k,l}$. So we can calculate the collision operator by

$$J[f](\mathfrak{v}_i) = \sum_{j,k,l} A_{i,j}^{k,l}(f(\mathfrak{v}_k)f(\mathfrak{v}_l) - f(\mathfrak{v}_i)f(\mathfrak{v}_j)), \ \mathfrak{v}_i := \mathfrak{o}\,.$$

Using this algorithm to create different discretizations (depending on Δv or L_{num}) we take a look at the error evolution for decreasing Δv and for different discretizations. We start with $t = 2, r = 2$ giving a convergence order of 0.4. In the following discussions we refer to the results of our approximation as $I_{r,t}$, so in this case $I_{2,2}$. A visualization of the relative error

$$e\left((\Delta v)^{-1}\right) := \left|\frac{I_{cc} - I_{2,2}\left((\Delta v)^{-1}\right)}{I_{cc}}\right|$$

can be found in figure 4.2. We have split the whole plot in order to improve the readability. The number of points used for the discretization can be obtained via

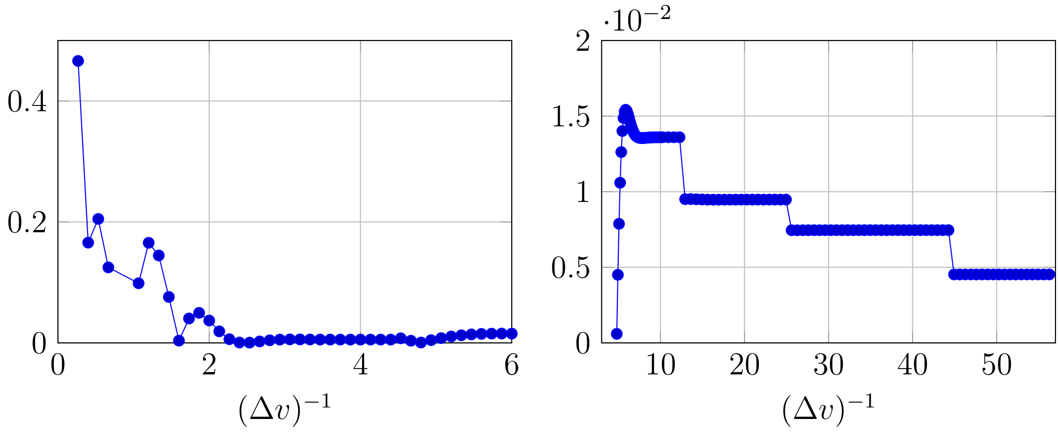


Figure 4.2: *Relative error e*

$$|\mathfrak{V}| = (15 \cdot (\Delta v)^{-1} + 1)^2\,.$$

The plot on the left hand gives the error for rough discretizations and the second plot for very fine ones, giving an impression of the asymptotic behavior. We can see several effects in these two plots, the first one is some kind of error nullifying at $(\Delta v)^{-1} = 4.8$ where we get a relative error of $5.98 \cdot 10^{-4}$. The error development in the left plot is mainly influenced by the error generated by the Newton-Cotes rule for the integration over the straight lines whereas the asymptotic error is dominated by the error generated by the approximation of the angular integrals. This can be clearly observed in the right hand side figure. Here the error looks like a step function, and the edges of the steps correspond to the points where the order of the used Farey sequences increases due to the requirement (4.1.1). Now we want to calculate the numeric convergence order by fitting the function

$$\tilde{e}(\Delta v) := c \cdot (\Delta v)^a \tag{4.2.1}$$

through the error data. This corresponds to the least square optimization problem

$$\arg\min_{a \in \mathbb{R}} \sum_{i=b}^{d} \left(e(i) - e(b) \cdot \left(\frac{\Delta v(i)}{\Delta v(b)} \right)^a \right)^2 . \tag{4.2.2}$$

This corresponds to the calculation of a function $\tilde{e}$ that goes through $e(b)$ and that minimizes the quadratic difference to the calculated errors $e(i), i > b$. Now we have the free parameters b and d corresponding to the start and end index for our calculation. We are interested in the asymptotic error, so we want to use all error values that are larger than our start index b. This means we always chose d as large as possible (in this case 143). Determining the start index b is a bit more complex, because we want to set it in such a way that we obtain the main error behavior. Now we could do some more or less justifiable assumptions and choose b in such a way that we exactly obtain a convergence order of 0.4, but instead of doing this we calculate a "convergence order function"

$$o(j) := \arg\min_{a \in \mathbb{R}} \sum_{i=j}^{d} \left(e(i) - e(j) \cdot \left(\frac{\Delta v(i)}{\Delta v(j)} \right)^a \right)^2$$

and take a look at the result in figure 4.3. Here the red line corresponds to the theoretical result of a convergence order of 0.4. And as we can see, beginning from the point where the asymptotic behavior kicks in (around 5) the estimated convergence order o stays above 0.4. Indicating that the theoretical result holds true.

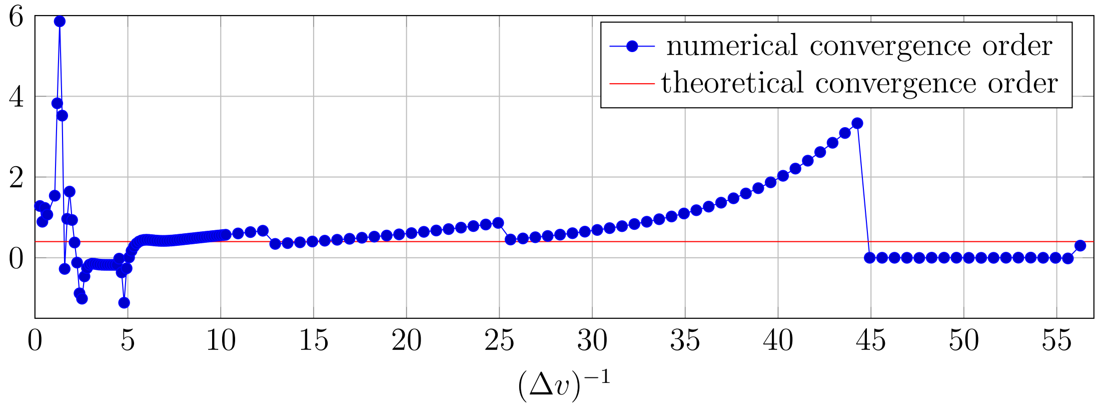


Figure 4.3: *Plot of the "convergence order function" $o(j)$*

Remark 4.2.1.2

(i) Now we take a little break and think about the problems in the determination of the numerical convergence order. The proof of the convergence order is based on the assumption that we can choose n (the order of the Farey sequence) in such a way that the three errors created by the three integral approximations are approximately the same. But unfortunately that is only possible if $n \in \mathbb{R}$. This is obviously not the case (we have to choose $n \in \mathbb{N}$), which led to the round up of n according to (4.1.1). So it seems appropriate to calculate the (asymptotic) numerical convergence by using only the points $\frac{1}{\Delta v}$ where the order of the Farey sequence increases, because at these points the assumption is nearly met, and the above figure containing o indicates that the numerical convergence order only increases in between these points (due to the approach of the next step at which the error significantly decreases).

(ii) We have often referred to this chapter when we spoke about possible oscillations and negative integration weights due to high order interpolating polynomials. Now we have to admit that even second order polynomials for the interpolation in the angular quadratures lead to negative integration weights when the order of the Farey sequence increases. This poses a real problem, because the order of the used Farey sequences increases automatically with decreasing Δv due to (4.1.1). The reason for this lies in the very unevenly spaced discretization points given by the Farey sequence. For example the first and second element of the Farey sequence have always the largest distance. Looking at a Farey sequence of order n this distance is given by $\frac{1}{n} - \frac{0}{n} = \frac{1}{n}$, whereas the second and third element have always the smallest occurring distance $\frac{1}{n-1} - \frac{1}{n} = \frac{1}{n(n-1)}$. At this point there is a number of possibilities to approach this problem. Because we want to stick to simple Lagrange interpolation within the framework of this work we choose the simple way and abandon the Farey sequence in favor of the easier angles

$$\mathfrak{G}_n := \left\{ \arctan(a) \,\middle|\, a = \frac{p}{n}, p \in \{0, \ldots, n\} \right\} ,$$

which are a subset of the Farey angles. Fortunately this simplifies all calculation and it increases the convergence order of the resulting discretization, as can be seen in the next theorem. Moreover this approach gives positive integration weights for polynomials up to order 6, independent of the chosen n.

(iii) It is possible to obtain stable higher order discretizations by using higher order polynomials (higher than 6). The main idea behind this is that one uses more than $s + 1$ points to obtain a polynomial of order s by using a least square minimization between the polynomial and the given points. This can be done in such a way that positive integration weights can always be obtained, even for arbitrary high polynomials. This approach for uniform discretizations (Newton

Cotes like) can be found in [Huy09] and the references therein, another approach (on non uniform grids) can be found in [Gra12].

Corollary 4.2.1.3 (Final discretization)
A discretization as in 3.4.5, but with the usage of the angles

$$\begin{aligned}\hat{\mathfrak{G}}_n &:= \mathfrak{G}_n \cup \left(\frac{\pi}{2} - \mathfrak{G}_n\right) \cup \left(\frac{\pi}{2} + \mathfrak{G}_n\right) \cup (\pi - \mathfrak{G}_n), |\hat{\mathfrak{G}}_n| = 4n + 1\\ \mathfrak{G}_n &:= \left\{\arctan(a) \middle| a = \frac{p}{n}, p \in \{0, \dots, n\}\right\} ,\end{aligned}$$

possesses a convergence order of

$$e \in \mathcal{O}\left((\Delta v)^{\frac{rt}{2r+t}}\right)$$

if we apply a Newton Cotes formula of order r (using $\tilde{r}$ points), a quadrature rule with polynomials of order $s = t - 1$ (for the angular integrals) and choose the used angles as well as the minimal size of the velocity space according to

$$\begin{aligned}|\mathfrak{V}| &= (1 + 2 \cdot (\tilde{r} - 1) \cdot 2n^2)^2 ,\\ n &= \max\left\{\left(\frac{(s)^t}{3\pi c t! 2^{r-4}}\right)^{\frac{1}{t+2r}} (\Delta v)^{\frac{-r}{t+2r}} , \left\lceil\frac{t-1}{4}\right\rceil\right\} .\end{aligned}$$

Here we assume that we use the same discretization angles for the angular integrations ($m = n$).

Proof:
Starting with the convergence order we need to take a look at the occurring errors. For this we take the calculations from the proof of 3.4.1 and realize that the only change lies in the change of N now defined as $N := 4n + 1,\ \tilde{N} := 4n$. So we can take the same proof, giving :

$$\begin{aligned}|e_3| &< \frac{1}{t!}\overbrace{\sup_{\theta\in[0,\pi]}\left|H_2^{(t)}(\theta)\right|}^{K_\theta:=}\left(\left\lfloor\frac{\tilde{N}}{s}\right\rfloor + 1\right)\left(\frac{s}{n}\right)^{t+1} = \frac{1}{t!}K_\theta\left(\left\lfloor\frac{4n}{s}\right\rfloor + 1\right)\left(\frac{s}{n}\right)^{t+1}\\ &\leq \frac{1}{t!}K_\theta\left(\frac{8n}{s}\right)\left(\frac{s}{n}\right)^{t+1} = \frac{8s^t}{t!}K_\theta n^{-t}\\ |e_2| &< \pi\frac{1}{t!}\overbrace{\sup_{\lambda,\theta\in[0,\pi]}\left|\frac{\partial^t H_1(\theta,\lambda)}{\partial\lambda^t}\right|}^{K_\lambda:=}\left(\left\lfloor\frac{\tilde{N}}{s}\right\rfloor + 1\right)\left(\frac{s}{n}\right)^{t+1} \leq \pi\frac{8s^t}{t!}K_\lambda n^{-t}\\ |e_1| &< \frac{3\pi^2}{2}cL(\Delta v)^r K_l 2^r n^r m^r .\end{aligned}$$

The same considerations as in 4.1.1 give

$$|e| < L\pi \left((K_\theta + K_\lambda) \overbrace{\frac{8s^t}{t!}}^{c_1=c_2:=} n^{-t} + \overbrace{\frac{3\pi}{2}}^{c_3:=} c2^r K_l (\Delta v)^r n^r n^r \right) .$$

Now we apply the same approach as in the end of 3.4.1 (but this time a shortened version):

$$\begin{aligned} c_1 n^{-t} = c_3 (\Delta v)^r n^{2r} \Longleftrightarrow n = \left(\frac{c_1}{c_3}\right)^{\frac{1}{2r+t}} (\Delta v)^{\frac{-r}{2r+t}} = \left(\frac{s^t}{3\pi 2^{r-4} c t!}\right)^{\frac{1}{2r+t}} (\Delta v)^{\frac{-r}{2r+t}} \\ \Longrightarrow |e| \in \mathcal{O}\left(\left[(\Delta v)^{\frac{-r}{2r+t}} \right]^{-t} + (\Delta v)^r \left[(\Delta v)^{\frac{-r}{2r+t}} \right]^{2r} \right) = \mathcal{O}\left((\Delta v)^{\frac{rt}{2r+t}} \right) . \end{aligned}$$

And the second condition for n is given by the necessity

$$t \leq N = 4n + 1 \Longleftarrow n = \left\lceil \frac{t-1}{4} \right\rceil .$$

□

Remark 4.2.1.4

(i) For the following calculations we use the same algorithm as described in 4.2.1.1, but we need to slightly change steps (ii),(iii). We now use 4.2.1.3 to calculate n, L_{num} in step (ii) and we use the angles $\mathfrak{G}_n$ in step (iii).

(ii) Using the above discretization as well as $L = 15$ the table 4.1 changes to table 4.3. And now we can say that the corresponding discretizations remain stable due to positive integration weights. Moreover the same can be done in the three dimensional case, also increasing the final convergence order.

$r \diagdown t$	2		3		4		5		6	
	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$	o	$\lvert\mathfrak{V}\rvert$
2	0.67	25	0.86	25	1.00	289	1.11	289	1.2	289
4	0.8	81	1.09	81	1.33	81	1.54	81	1.71	1089
6	0.86	289	1.2	289	1.50	289	1.76	289	2.00	4225

Table 4.3: *convergence order o and minimal size of velocity space $|\mathfrak{V}|$ above order of Newton Cotes formula r and points used for angular quadratures t*

Now we reach the last theorem / corollary within this work. We finally take a look at the computational complexity of our discretizations.

Corollary 4.2.1.5 (Computational complexity)
Assuming that we use 4.2.1.3 for a discretization of the collision operator in two dimensions we obtain a computational complexity in

$$\mathcal{O}\left((\Delta v)^{-3-\frac{2r}{t+2r}}\right),$$

where t corresponds to the order of the polynomials used for the angular quadrature and r to the order of the applied Newton Cotes formula. Assuming that we use 3.4.6, but with the simpler angles used in 4.2.1.3 we obtain a computational complexity in

$$\mathcal{O}\left((\Delta v)^{-4-\frac{4s}{t+3s}}\right)$$

where t corresponds to the order of the polynomials used for the angular quadrature and s to the order of the applied Newton Cotes formula.
These complexities correspond to the evolution of the number of collision pairs

$$\left|\left\{(i,j,k,l)\,\middle|\,A_{i,j}^{k,l} \neq 0\right\}\right| .$$

Proof:
We begin with the two dimensional case. We assume that the quadratic discretization domain (in which the support of f lies) possesses a side length of L. This implies that the maximum number of discretization points on a line within this domain and the total number of discretization points is given by

$$L_{\text{num}} = \frac{L}{\Delta v} + 1, \qquad |\mathfrak{V}| = L_{\text{num}}^2 .$$

Moreover we use n angles in the first half of the first quadrant, resulting into a total of

$$N = 2 \cdot (4n-3)$$

angles. Here we have $4n-3$ angles in the upper half plane and the same number in the lower one. We use a Newton Cotes formula of order r for the innermost integration and a polynomial quadrature formula with polynomials of order $t-1$ which gives

$$n \in \mathcal{O}\left((\Delta v)^{-\frac{r}{t+2r}}\right),$$

according to 4.2.1.3. Now we simply need to put these ingredients together to get an upper bound of the number of collision pairs. We obtain this bound by multiplying the maximal number of points on a line with the number of angles in θ (N) and the number of angles in φ (N) as well as the number of points on the grid:

$$\begin{aligned}\left|\left\{(i,j,k,l)|A_{i,j}^{k,l} \neq 0\right\}\right| &< L_{\text{num}} \cdot N^2 \cdot |\mathfrak{V}| \\ &\in \mathcal{O}\left((\Delta v)^{-1}\right) \cdot \mathcal{O}\left((\Delta v)^{-\frac{2r}{t+2r}}\right) \cdot \mathcal{O}\left((\Delta v)^{-2}\right)\end{aligned}$$

$$= \mathcal{O}\left((\Delta v)^{-3-\frac{2r}{t+2r}}\right) .$$

The same argumentation in the three dimensional case yields (where s corresponds to the order of the Newton-Cotes formula)

$$\begin{aligned}\left|\left\{(i,j,k,l)|A_{i,j}^{k,l} \neq 0\right\}\right| &< L_{\text{num}} \cdot N^4 \cdot |\mathfrak{V}| \\ &\in \mathcal{O}\left((\Delta v)^{-1}\right) \cdot \mathcal{O}\left((\Delta v)^{-\frac{4s}{t+3s}}\right) \cdot \mathcal{O}\left((\Delta v)^{-3}\right) \\ &= \mathcal{O}\left((\Delta v)^{-4-\frac{4s}{t+3s}}\right) . \qquad \square\end{aligned}$$

Remark 4.2.1.6

(i) The message of the above result is that we have a computational complexity around $\mathcal{O}\left((\Delta v)^{-3-\frac{2}{3}}\right)$ in two and around $\mathcal{O}\left((\Delta v)^{-5}\right)$ in three dimensions. In two dimensions this seems to be less than one would expect (4) and especially in three dimensions this result seems to be one polynomial order better (5 instead of 6) than one would expect, compare [IR02] or the introduction of [BR00,PH99]. Here it seems to be interesting that the usage of the Farey angles (roughly corresponding to all available angles) results into a quadratic growths of the angles giving complexities of $\mathcal{O}\left((\Delta v)^{-4-\frac{1}{3}}\right)$, $\mathcal{O}\left((\Delta v)^{-6}\right)$ in 2 resp. 3 dimensions. So we can deduce that the usage of the fewer and simpler angles $\mathcal{G}_n$ results into a higher convergence, lower computational complexity and more stable discretizations (regarding negative integration weights).

(ii) It is interesting to mention that the computational complexity of a discretization only depends on the n (number of used angles within one half of a quadrant) determined by

$$n = \max\left\{\left(\frac{(s)^t}{3\pi ct!2^{r-4}}\right)^{\frac{1}{t+2r}} (\Delta v)^{\frac{-r}{t+2r}} , \left\lceil\frac{t-1}{4}\right\rceil\right\} .$$

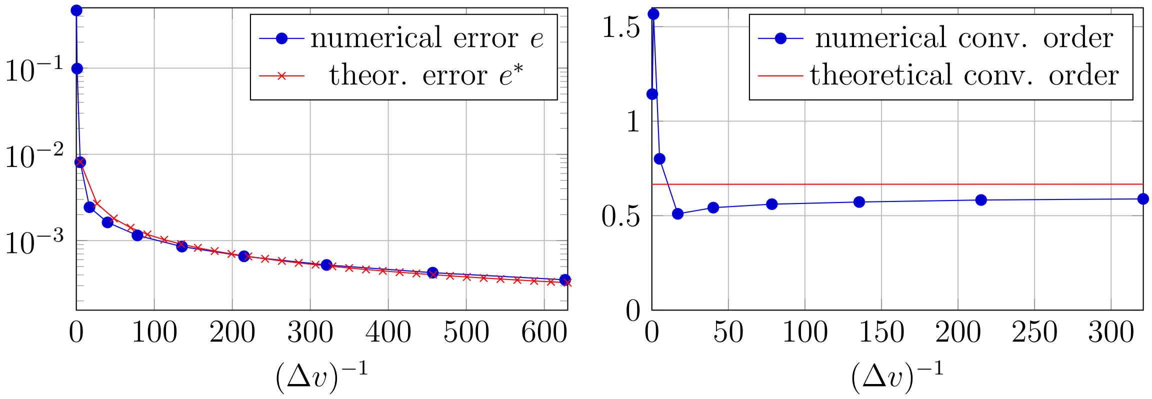
For example we can take a look at table 4.4 . Here we see the number of collision pairs (direct proportional to the necessary number of floating point operations) for two discretizations constructed according to 4.2.1.3. Here we see that these two discretizations (created according to 4.2.1.3) use the same number of collision pairs if n is the same for the discretizations. Now it is easy to compute that the n for $t = 2, r = 2$ grows as $(\Delta v)^{\frac{-1}{3}}$ and the other one as $(\Delta v)^{\frac{-4}{11}}$. So even if it looks in the table as if the higher order discretization generally needs less or the same computational effort this situation changes when $L_{\text{num}} \approx \frac{1}{\Delta v}$ gets sufficiently large.

L_{num}	$r = 2, s = 1$		$r = 4, s = 2$	
	number of collision pairs	n	number of collision pairs	n
9	288	1	288	1
17	6500	2	2176	1
19	9298	2	3078	1
33	53128	2	53128	2

Table 4.4: *computational complexity in numbers of collision pairs*

4.2.2 Numerical validation

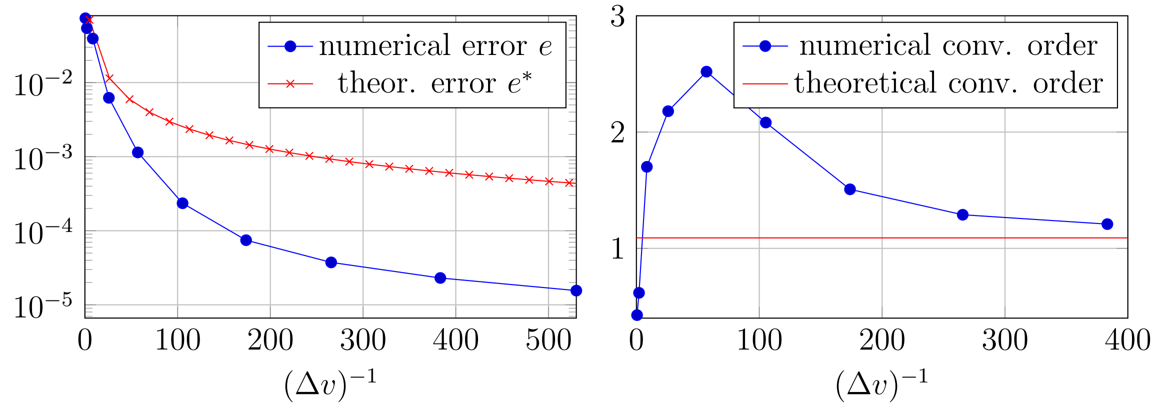
Now we redo the calculation of $I_{2,2}$ with remark 4.2.1.2 in mind and by applying a discretization based on 4.2.1.3. So this time we calculate the errors only at the points where the n changes and in order to obtain the asymptotic behavior for very large $|\mathfrak{V}|$. The result for $t = 2, r = 2$ can be found in 4.4a, this time with a visualization of the theoretical error development in the asymptotic regime. The theoretical error is given by $e^*\left((\Delta v)^{-1}\right) = a \cdot (\Delta v)^{\frac{2}{3}}$, where we have chosen a in such a way that $e(5) = e^*(5)$. As before we do some fitting for the function (4.2.1) by calculating (4.2.2) for different start indexes, resulting into figure 4.4b. Now we have the situation that figure 4.4a indicates that our convergence order of $\frac{2}{3}$ seems to be correct whereas the message of 4.4b seems to be unclear. At this point we have to use the asymptotic joker card. We see that in figure 4.4b the order monotonically increases, beginning from $(\Delta v)^{-1} = 20$. So it seems like a slow convergence of the approximated convergence order $o(j)$ towards $\frac{2}{3}$. Now it would be interesting to look at the development for larger $(\Delta v)^{-1}$ to verify this conjecture, but even the calculation of the used 11 points was time consuming (more than a day after algorithmic optimization in Octave), where the largest used velocity grid consists of $88'378'801$ points. For the next example we redo the applied



(a) *numerical and theor. relative error*

(b) *plot of $o(j)$*

Figure 4.4: *calculation results for $I_{2,2}$*



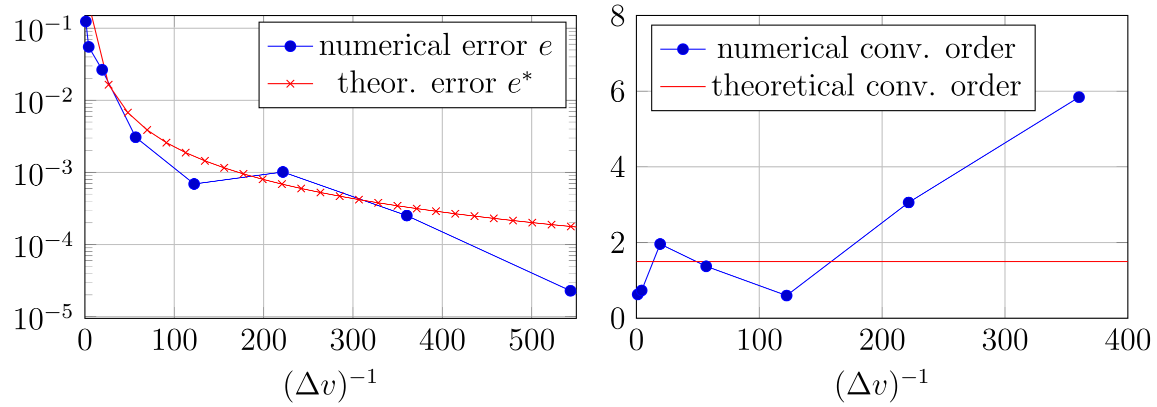
(a) *numerical and theor. relative error*

(b) *plot of $o(j)$*

Figure 4.5: *calculation results for $I_{4,3}$*

interpretation for $t = 3, r = 4$ giving $I_{4,3}$. As before we have plotted $e^* = a \cdot (\Delta v)^{1.09}$ and the numerical error in figure 4.5a. Here it looks like the discretization possesses a significantly better convergence order than estimated. Interestingly the visualization of the function $o(j)$ (4.5b) shows again that the convergence order asymptotically converges towards the estimated value of 1.09.

In the next example we look at $t = 4, r = 6$ resulting into a convergence order of 1.5 In figure 4.6a we see that the error e generally remains below $e^* = a \cdot (\Delta v)^{1.5}$, but



(a) *numerical and theor. relative error*

(b) *plot of $o(j)$*

Figure 4.6: *calculation results for $I_{6,4}$*

figure 4.6b indicates that we need to calculate more values to get an idea about the asymptotic numerical behavior. Unfortunately this is not feasible due to the computational complexity. Finally we want to mention that we also derived complete discretizations for velocity spaces up to $|\mathfrak{V}| = 2401$. Here we calculated the change of the mass, momentum and energy in order to reassure us that the minimal requirements are met and that the theoretical considerations are correct. The relative error in mass, momentum and energy lies between 10^{-14} and 10^{-16}, indicating that the mass, momentum and energy are in fact conserved, because these relative errors correspond to the used machine precision (double). It is interesting to mention that the application of a Clenshaw Curtis rule on a grid with 81 velocities generates a relative error around 1%, independent of the used precision for the rule. Finally we compare the three discretizations in figure 4.7. Here we see that the error for small $(\Delta v)^{-1}$ exhibits another error behavior than in the asymptotic region. For small Δv the error substantially grows when the order of the scheme gets increased resulting into a "break even" point where the accuracy of the higher order scheme becomes better than the accuracy of a lower order scheme. The observed effect is partially created by the factors

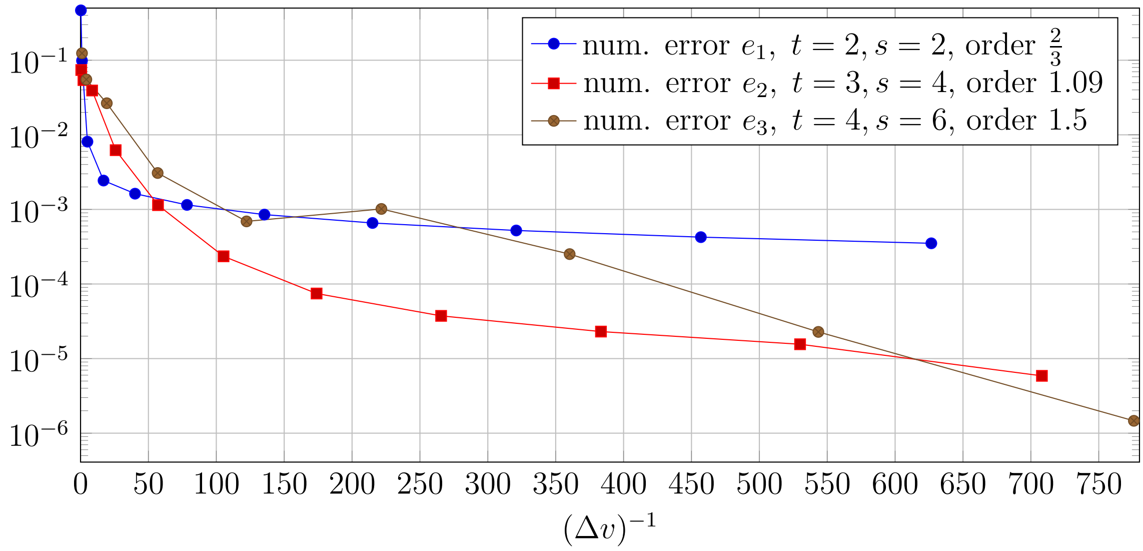


Figure 4.7: *calculation results for* $I_{2,2}, I_{4,3}, I_{6,4}$

$$\frac{(t-1)^t}{t!}, \qquad 2^r$$

within the angular and Newton-Cotes errors (see proof of 4.2.1.3 or 3.4.1). These factors grow exponentially with t, r, because

$$\frac{(t-1)^t}{t!} \geq \frac{(t-1)^t}{\sqrt{2\pi} t^{t+\frac{1}{2}} e^{-t}} = \left(\frac{t-1}{t}\right)^t \cdot \frac{1}{\sqrt{2\pi} e^{-t} t^{\frac{1}{2}}} \overset{t\geq 2}{>} \frac{1}{e\sqrt{2\pi}} \frac{e^t}{t^{\frac{1}{2}}}.$$

Here we used a lower bound for $t!$ given by Stirling's approximation in the first step and the fact that $\left(\frac{t-1}{t}\right)^t$ is a monotonously increasing function for $t \geq 1$ that goes asymptotically to $\frac{1}{e}$. So the "error constant" of our discretizations grows exponentially with the number of points used for the polynomial interpolation in the quadrature formulas. This is a standard result regarding quadrature formulas in which one uses

Lagrange or Hermite interpolation due to the interpolation error that introduces an error constant of the form $\frac{t^t}{t!}$, for example see proof of 3.4.1. Moreover we see in figure 4.7 that the error e_2 behaves exceptionally good, but gets defeated by e_3 on extremely large grids. Now that we have analyzed the asymptotic behavior of our discretizations that comes into play in homogeneous problems we look at the behavior in regions typically used in inhomogeneous problems. Due to the computational complexity one is typically limited to use around 1089 velocities or less. So in figure 4.8 we take a short look at the error development for quadratic velocity spaces with up to $1089 = 33^2$ velocities. Here we see that the most primitive discretization performs almost consistently better than higher order discretizations on small grids. Moreover in this region we are not able to see any convergence behavior in e_3, which makes the fact that the relative error almost stays at 7% somehow astonishing. For the moment the message seems to be that it is not clear which of these schemes should be used in discretizations for inhomogeneous problems, but we should use higher order schemes for homogeneous problems where a high precision is required. This figure also implies that the simplest scheme corresponding to $e_1, I_{2,2}$ gives an error below 10% for velocity spaces with more than $17^2 = 289$ velocities and below 5% for spaces with more than $23^2 = 529$ velocities as long as the initial distribution is reasonably smooth and velocity resolved. Moreover the behavior of $e_3, I_{6,4}$ implies that we should stick to this discretization for 17^2 to 21^2 velocities, because this discretization has the smallest number of collision pairs, due to $n = 1$ in the whole figure. These conclusions have to be taken with some caution, because we only look at one example in a region where we can not see any asymptotic behavior, so these last observations are only exemplarily.

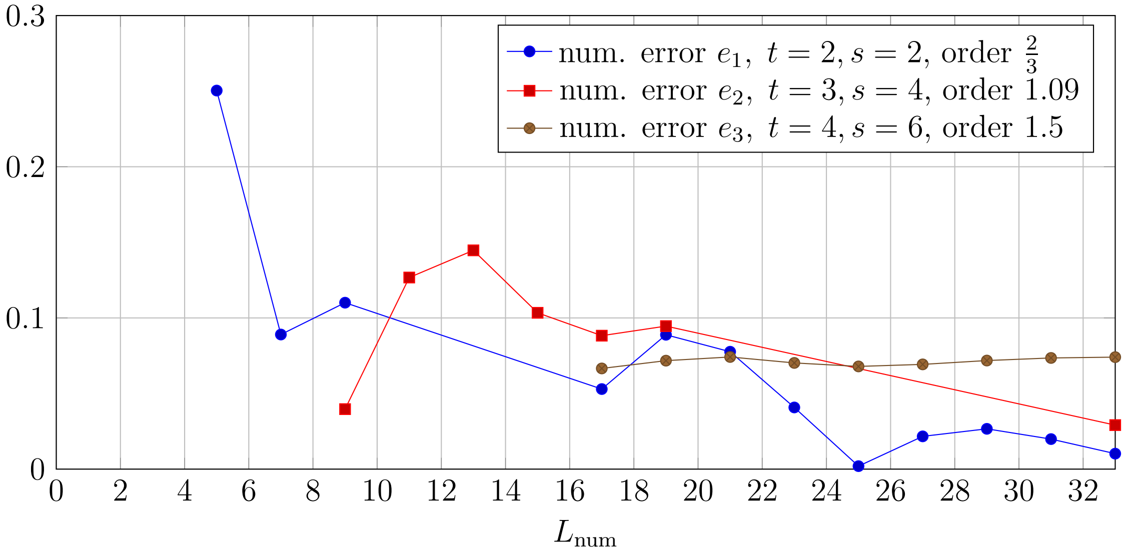


Figure 4.8: *calculation results for* $I_{2,2}, I_{4,3}, I_{6,4}$

5 Implementation and high performance computing

In this chapter we discuss our parallelized implementation of a Boltzmann solver based on LGpMs, whereas this implementation can easily be generalized to the point where it is able to use arbitrary discretizations of the collision operator as long as one supplies a C++ routine that constructs the operator A of a DVM. Here it is not our aim to go into great detail about the implementation specifics in C,CUDA or the GNU compiler collection. But we want to give an overview about the general approach as well as optimization (parallelization) strategies which lead to a high utilization of the used processors. Assuming that one was able to construct a satisfying discretization of the collision operator the next problem that occurs in the numerical treatment of the Boltzmann equation is the following system of equations

$$(\partial_t + v_i \cdot \nabla_{\mathfrak{x}}) f_i = \alpha \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} (f_k f_l - f_i f_j) =: J_i(\mathbf{f}),\ i \in M_{\mathfrak{V}}, \mathbf{f} \in \mathbb{R}_+^{|M_{\mathfrak{V}}|} . \tag{5.0.1}$$

5.1 Numerical Methods

Because the calculation of $\mathbf{J}$ (the right hand side of (5.0.1)) has high computational costs and because this was our first attempt in parallelization, we wanted to use simple algorithms with a minimal number of calculations of $\mathbf{J}$. This led to a first order (with minimal modifications second order) operator splitting method for (5.0.1) that gives the two equations

$$\partial_t f_i(\mathfrak{x}, t) = -v_i \cdot \nabla_{\mathfrak{x}} f_i(\mathfrak{x}, t) \tag{5.1.1}$$

$$\partial_t f_i(\mathfrak{x}, t) = \alpha \sum_{j,k,l \in M_{\mathfrak{V}}} A_{i,j}^{k,l} \big(f_k(\mathfrak{x}, t) f_l(\mathfrak{x}, t) - f_i(\mathfrak{x}, t) f_j(\mathfrak{x}, t) \big) = J_i(\mathbf{f}) , \tag{5.1.2}$$

that have to be solved consecutively. We now discretize a bounded domain Ω of the position space $\mathbb{R}^n_{\mathbf{x}}$, $n = 2, 3$ in an equidistant manner,

$$\mathfrak{X} := \{\mathfrak{x} | x_j \in \Delta x \cdot \mathbb{Z}, j = 1, \ldots, n, \mathfrak{x} \in \Omega \subset \mathbb{R}^n_{\mathbf{x}}\} \ , \Delta x \in \mathbb{R}^+ .$$

The collision equation (5.1.2) is independent of the position variable $\mathfrak{x}$, thus giving a straightforward way of parallelization, because this equation can be numerically solved in parallel for every point in the position space $\mathfrak{X}$. We solve the transport equation (5.1.1) with a first order finite difference (upwind) scheme over the equidistant position space. Because the discretization of the velocity space typically does not fit onto the discretization of the position space an additional step of linear interpolation is necessary. This upwind solver can be described as a series of linear interpolations, this gives (in two dimensions):

$$
\begin{aligned}
\lambda_i &:= \Delta t_T \cdot \frac{\max\{\mathfrak{v}_{i,1}, \mathfrak{v}_{i,2}\}}{\Delta x} && (5.1.3)\\
f_i(\xi_i, t) &:= (1-\alpha) f_i(\mathfrak{x}_a(\mathfrak{v}_i), t) + \alpha f_i(\mathfrak{x}_b(\mathfrak{v}_i), t) && (5.1.4)\\
f_i(\mathfrak{x}, t+\Delta t) &= \big(1-\lambda_i\big) f_i(\mathfrak{x}, t) + \lambda_i f_i(\xi_i, t) && (5.1.5)
\end{aligned}
$$

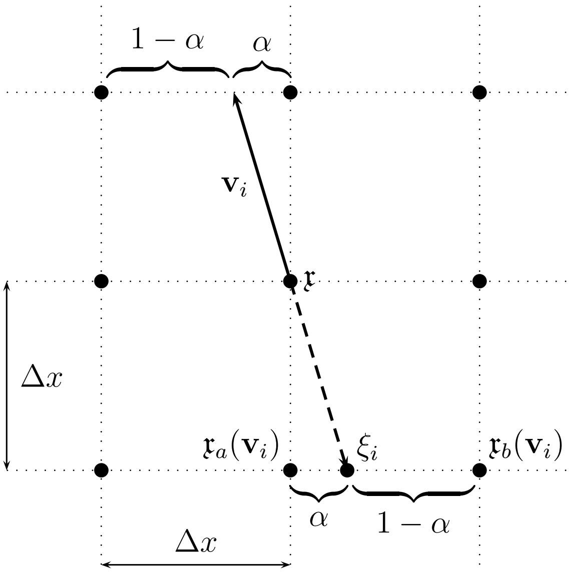


Figure 5.1: *position space grid*

Figure 5.1 shows how we calculate the mass distribution f that corresponds to the velocity $\mathbf{v}_i$ in the position space point $\mathbf{x}$. At first (5.1.4) we have to interpolate the distribution f_i at the point ξ_i that lies in the opposite direction of $\mathbf{v}_i$. After that a second interpolation (5.1.5),(5.1.3) calculates how much mass flows from point ξ_i to $\mathbf{x}$ within a time of Δt_T. With such a scheme we obtain a stability-bound for the time step based on the Courant-Friedrichs-Lewy-condition:

$$\Delta\, t_T \leq \frac{\Delta x}{\max\{\|\mathfrak{v}\|_\infty, \mathfrak{v} \in \mathfrak{V}\}} \qquad (5.1.6)$$

guaranteeing, that the mass can not flow further than one cell within one time step.

Analogous considerations lead to a similar scheme in three dimensions we also implemented. We solve the collision equation (5.1.2) with a time adaptive second order explicit Runge-Kutta method:

$$
\begin{aligned}
k_{i,1} &= \Delta t_C J_i f(\mathfrak{x}, t)\\
k_{i,2} &= \Delta t_C J_i \left[f + 0.5 k_{i,1}, f + 0.5 k_{i,1}\right](\mathfrak{x}, t)\\
f_i(\mathfrak{x}, t + \Delta t_C) &= f_i(\mathfrak{x}, t) + k_2 \; .
\end{aligned}
$$

Our error estimator is the difference between the explicit Euler and a second order Runge-Kutta scheme:

$$e = \overbrace{\sum_{i \in M_{\mathfrak{V}}} \frac{|k_{i,1} - k_{i,2}|}{\max(f_i(\mathfrak{x}, t), f_i(\mathfrak{x}, t + \Delta t))} \cdot \frac{1}{|M_{\mathfrak{V}}|}}^{\text{relative error over velocity space}} \cdot \frac{\varepsilon}{100} \, ,$$

here ε is the desired relative error in %. If the error is greater than one, the calculation gets rejected and the time step Δt_C gets divided by 2. Otherwise the next time step length is determined in such a way, that the estimated error lies around 10% of the desired error, in formulas:

$$\Delta t_{C,\text{new}} = \begin{cases} 0.5 \cdot \Delta t_C, & \text{if } e > 1 \\ \Delta t_C \cdot \begin{cases} 0.1 \cdot e^{-1}, & \text{if } e \geq 0.05 \\ 2, & \text{if } e < 0.05 \end{cases}, & \text{if } e \leq 1 \end{cases} .$$

For such a method we generally get a stability bound around

$$\Delta t_C \leq \frac{1}{|\lambda_{\max}(t)|}, \tag{5.1.7}$$

as long as the solution f is sufficiently near to an equilibrium solution, here λ_{max} is the biggest eigenvalue of the linearization of the ODE (5.1.2) at time t with respect to its absolute value. In our numerical experiments it turned out, that (5.1.6) only posed a real restriction at high and transitional Knudsen numbers, whereas (5.1.7) dominated cases with low Knudsen numbers. We were able to simulate phenomena like vortex shedding of a compressible mono-atomic gas with our model (and with only the above equations, no other modifications are needed), so a further investigation in low Knudsen numbers in conjunction with implicit schemes could lead to interesting results. Since we only have the mass distribution function f we need to calculate the macroscopic properties of the gas. Let n be the desired dimensionality and $V := (\Delta x)^n$ the volume that corresponds to the position space points. With that we get the

- *mass density* $\frac{\rho}{V}$, *momentum* $\mathbf{m} = (m_i)_{i=1}^n$ and *kinetic energy* E of ρ via

$$\rho := \sum_{\mathfrak{v} \in \mathfrak{V}} f(\mathfrak{v}), \qquad m_i := \sum_{\mathfrak{v} \in \mathfrak{V}} v_i f(\mathfrak{v}), \qquad E := \frac{1}{2} \sum_{\mathfrak{v} \in \mathfrak{V}} \|\mathfrak{v}\|_2^2 f(\mathfrak{v});,$$

- *stress tensor* $S = (s_{ij})_{i,j=1}^n$ with

$$s_{ij} := \sum_{\mathfrak{v} \in \mathfrak{V}} v_i v_j f(\mathfrak{v}) - \frac{1}{\rho} m_i m_j \ ,$$

- *hydrostatic pressure* p with

$$p := \frac{1}{nV} \cdot \operatorname{tr}(S) = \frac{1}{nV} \cdot \sum_{i=1}^{n} s_{ii} \ ,$$

- *temperature* T with

$$T := \frac{pV}{\frac{\rho}{m_e} \cdot k_B} \ ,$$

  where k_B represents the Boltzmann constant and m_e the mean mass of a particle in the gas.

5.2 Technical basics

Using the aforementioned numerical schemes we can approximate the number of floating point operations necessary to solve a small sized problem. Assuming that we use $33^2 = 1089$ points in the velocity space, $960 \times 320 = 307200$ points in the position space and apply the approximation 4.2.1.3 with $t = 4, r = 4$ we obtain 53128 collision pairs. Here we have already reduced the number of collision pairs that must be calculated per evaluation of J by a factor of 4, here we use the symmetry properties of A to calculate $f_k f_l - f_i f_j$ only once and then add it to f_i, f_j and subtract it from f_k, f_l thus saving 60% of the calculations. One such collision pair results into 8 floating point operations, 3 multiplications and 5 additions. A simulation of such a space inhomogeneous case typically needs around 50 Runge Kutta steps (each step needs two evaluations of $\mathbb{J}$) per time step and $20'000$ time steps overall. This leads to $2.6113 \cdot 10^{17}$ floating point operations or 261.13 peta FLOP. Even the newest CPUs (with around 200 GFLOPs) would need 15 days to solve this problem if the used code is highly parallelized (using all cores as well as the advanced vector extensions - AVX). Without such a parallelization this computation would be around $4 \cdot 8$ times slower, under the assumption that the processor possesses 4 cores and can use 256 bit vector operations (calculation with 8 floats in parallel). So without parallelization this problem would need 483 days on the same processor. This example clearly says that parallelization is mandatory if we want to do numerical simulations. We also see that the collision step is much more expensive than any other part of the algorithm. That means that the performance of our resulting program will mainly be bound by the FLOPS of the used processor and not by the memory bandwidth. At large velocity spaces the collision time step needs more than 95% of the program run time. Because of that we focus on the parallelization of this part of the algorithm. We investigate parallelization on CPUs and GPUs to find out which architecture is better suited for our problem. One of the main findings of that comparison is that a parallelized CPU implementation (that should use the vector processing extensions of recent CPUs to be efficient) and a parallelized GPU implementation are very similar and exhibit very similar parallelization strategies. In order to parallelize our numerical schemes we need an idea about an algorithm that allows parallelization and the processor architecture we are aiming at (partially in order to minimize the code difference between a CPU and a GPU implementation).

5.2.1 Algorithm - an overview

We now want to give a brief overview about the necessary data structures and the part of the implementation that corresponds to the numerics. We have a discretization of the position space $\mathfrak{X}$ and a discretization of the velocity space $\mathfrak{V}$. Let us assume we have m points in the position space and n points in the velocity space. Our Algorithm now needs 8 arrays $A, B, C, m_1, m_2, m_3, m_4, m_5$. Array A holds the information about the position and velocity space that means A is a linear array with $n \cdot m$ elements. The information is stored in the following way:

$$A = \big(f_1(\mathfrak{x}_1), \ldots, f_n(\mathfrak{x}_1), f_1(\mathfrak{x}_2), \ldots, f_n(\mathfrak{x}_2), \ldots, f_1(\mathfrak{x}_m), \ldots, f_n(\mathfrak{x}_m)\big) \ . \tag{5.2.1}$$

B has the same dimensions as A and it is also used to save the grid informations, array C holds the collision pairs. The other arrays m_1 to m_5 are used to store the macroscopic values $\rho, \mathbf{m}, E, p, T$. Figure 5.2 shows how the algorithm generally works, it consists of four main steps: the transport, the collision, the communication step and the calculation of the macroscopic values. At the beginning array A must be initialized, after that we have an $A{-}B$ memory access pattern. That means the transport transfers the Data from array A to array B and the collision transfers it from B back to A, after this an additional communication step can occur before the next transport step starts. Communication steps occur in the case of periodic boundary conditions where we need to transfer the corresponding information between border points. After a fixed number of time steps the algorithm calculates the macroscopic values and saves them in the corresponding arrays. The saving of these values to the hard disc and the next of $A - B$ iteration happens in parallel.

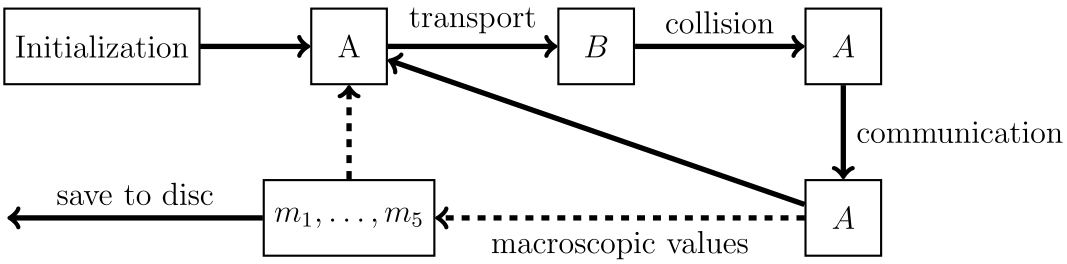


Figure 5.2: *workflow of the algorithm*

5.2.2 Processor Architecture

Figure 5.3 shows a direct comparison of the GPU and the CPU architecture. A GPU consists of a number n of *multi processors* (MP_i) whereas the CPU (as a many core Processor) consists of m independent cores (C_i) and on high end Intel®CPUs the *Hyper-Threading Technology* (HTT) is available. That means that every core of the CPU can execute two threads to reduce memory access wait times, summing up to $2 \cdot m$ virtual processors that can be used to fully utilize the Intel®CPU. The GPU multi processors consist of q *scalar processors* (SP_i), each can do one scalar operation (so q parallel operations) and a single core of a CPU on the other hand can use *single instruction multiple data* (SIMD) instructions (via SSE - *Streaming SIMD Extensions* or AVX - *Advanced Vector Extensions*) to do up to p scalar operations (S_i) in parallel.

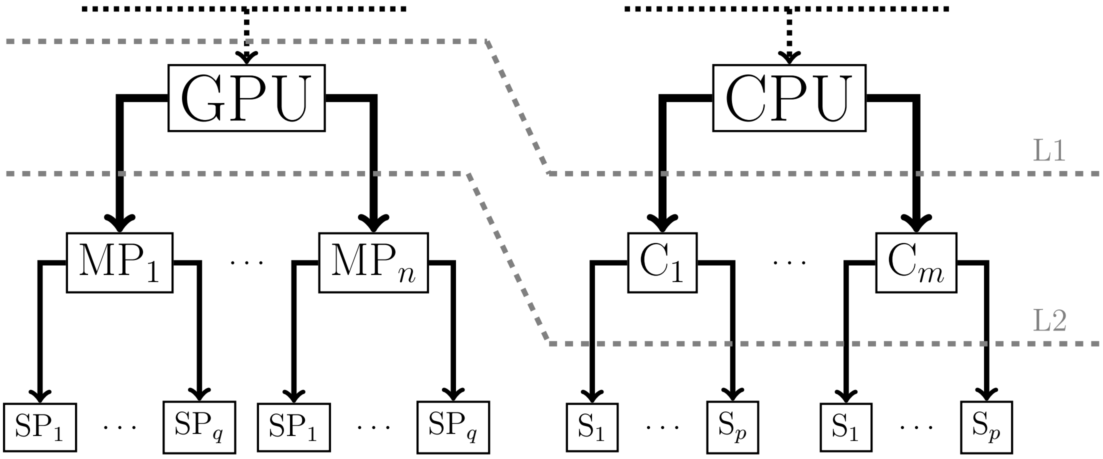


Figure 5.3: *architecture comparison GPU vs CPU*

Due to this hardware structure we need two levels of parallelization. The first level (L1) is a parallelization across multiple independent processors (cores or GPUs) that can communicate with each other. The second level (L2) is a SIMD parallelization where the same operations are executed in parallel over a set of data.

Level 1:
Let's assume we have m GPUs or cores. In the first parallelization level we divide the position space grid into m rectangular sub-grids. If we use the notions from Figure 5.2 we now get the two grid sets $(A_i)_{i=1}^m, (B_i)_{i=1}^m$ and the macroscopic values $(m_{i,1}, \dots, m_{i,5})_{i=1}^m$. So if we substitute $A, B, m_1, \dots, m_5$ with $A_i, B_i, m_{i,1}, \dots, m_{i,5}$ in Figure 5.2 this work flow also holds true for the parallelized version. With the addition that, in the communication step, the used processors transfer the borders of their Grid A_i into memory buffers that are associated to the processors that work on neighboring Grids. In the end of such a communication step the processors synchronize (wait until all copied the data to the buffers) and then they update the borders of their Grid A_i with these memory buffers and start the next iteration.

Level 2:
Now we look at a single core or a single GPU. To use the SIMD approach we have to divide the sub-grid A_i into r-point blocks in such a way that the r position space points in such a block can be calculated in parallel. That means, that r is a hardware dependent variable and that the memory storage of these r-point blocks must be adapted to the memory access pattern restrictions of the used processor.

CPU: We apply the SIMD approach on the CPU in such a way that we calculate the differences corresponding to collision pairs $(f_k f_l - f_i f_j)$ for four different position space points in parallel ($r = 4$). So we have to store

$$(f(\mathfrak{v}_1)_k, f(\mathfrak{v}_2)_k, f(\mathfrak{v}_3)_k, f(\mathfrak{v}_4)_k) =: sse_k$$

into one SSE register so that we can simply calculate $sse_k sse_l - sse_i sse_j := sse_{sol}$ to get the four difference values into one SSE register (sse_{sol}). To efficiently store data in a SSE register, the data to store must be four consecutive four byte values in main memory. And these 128 bit long memory region must be 128 bit aligned in main memory. That means, that the leading address of the four byte array must be divisible by 8. So the adaption of the former memory storage within A (see (5.2.1)) now looks like

$$(f_1(\mathfrak{x}_i), \ldots, f_1(\mathfrak{x}_{i+3}), f_2(\mathfrak{x}_i), \ldots, f_2(\mathfrak{x}_{i+3}), \ldots, f_n(\mathfrak{x}_i), \ldots, f_n(\mathfrak{x}_{i+3}))\,.$$

This approach corresponds to SSE, for AVX we can do the same, but with $r = 8, 16$ due to the $256, 512$ bit long AVX registers. For this to work we need to reorder the data of these r points prior to the SIMD calculations for this block. We do this locally, meaning that we reorder one block, do the calculations, reverse the reordering before we write back to RAM and then reorder the next block. Fortunately this also guarantees (as much as it can be guaranteed) that the used (and actually necessary) data gets completely transferred into the processor cache prior to the calculations and nothing else. This effectively corresponds to cache blocking and is possibly the reason for the super linear speedup that can be observed.

GPU: On the GPU we simply have to transfer all memory that corresponds to the r-point block into shared memory (Multiprocessor cache). The access pattern we have to obey says, that we have to transfer consecutive linear memory regions from the graphics card memory into the shared memory. This condition is automatically met. So no data reordering is necessary. There are two main differences to the CPU core parallelization. The first is that the GPU can handle n r-point blocks in parallel, because it simply has n multiprocessors. And the second is that such a multiprocessor has to have at least 64 scalar operations it can do in parallel. This is needed to fully utilize the 32 scalar processors per multiprocessor. Unfortunately the shared memory (48 KiB) isn't large enough to hold all the necessary information for 64 different position space points (considering velocity space discretizations with up to 1000 velocities). Because of that the GPU isn't generally calculating r-point blocks in parallel but it calculates the r times number of collision pairs differences $f_k f_l - f_i f_j$ in parallel. Now r has to be chosen in such a way that some hardware restrictions are met for an efficient computation (maximum amount of available registers per multiprocessor, amount of shared memory, ...).

As far as comparability between CPUs and GPUs is concerned one of the main messages of this architecture comparison is:
The general structure of GPUs and CPUs (see Figure 5.3) is not as different as one could think, so a CPU implementation should at least use all cores of the CPU and

SSE to be comparable with a GPU implementation, because otherwise we potentially loose between 95 % and 60 % of the CPU peak performance. This would distort a speedup factor for the CPU / GPU comparison by a factor between 2.5 and 20.

5.2.3 Additional Optimizations

(i) We maximize the available FLOPS of the CPU through the utilization of the main CPU capabilities (HTT,SSE) and of the GPU through the creation of a hardware and discretization dependent automatic optimization algorithm which calculates the GPU kernel (a kernel is a function that operates on the GPU and does things in parallel) launch parameters in such a way that every Multiprocessor can calculate two r-point blocks (if $|\mathfrak{V}|$ is too big it can happen, that $r = 1$ and that the MP can only calculate one block at a time) in parallel and that every multiprocessor can calculate at least 320 scalar operations in parallel.

(ii) We optimized (minimized) the necessary bandwidth from the main memory to the processor caches. The major part $(90 - 99\%)$ of the bandwidth that was needed during a collision time step was due to read accesses on the array C (see section Algorithm - Overview). The collision pairs don't fit into the Processor caches and therefore they must be looked up every time a corresponding difference $f_i f_l - f_k f_l$ gets evaluated. On the CPU we naturally cut the needed bandwidth down by a factor of 4/8, because we used SSE/AVX (possibly contributing to the super linear speedup). For one SSE calculation of a collision difference that is dedicated to four different position space points we only accessed C one instead of four times. At the GPU we cut the needed bandwidth down by a factor of two with the same approach. At the GPU the scalar processors now calculate a collision difference for two different position space points consecutively, so the collision pair (i, j, k, l) can be reused one time. That optimization lead to 21.5% bandwidth usage (of the benchmarked maximum of this GPU) during the collision step, indicating that the scalar processors are not slowed down by insufficient memory bandwidth.

(iii) This optimization only involves the GPU version. Within the shared memory (MP cache) we can get so called memory collisions that stall the computation for all participating SPs. This can happen because the shared memory consists of 32 memory banks which are ordered in such a way that 32 consecutive four byte elements in the shared memory are in 32 different banks. If it happens that more than one SP of a MP wants to access an element in the same memory bank, the access gets serialized. This can happen when the 32 SPs calculate 32 collision differences $f_k f_l - f_i f_j$, because these SPs must access the elements $f_{i_1}, \ldots, f_{i_{32}}$ in parallel. Thus we can avoid shared memory collisions by reordering the collision pairs (i, j, k, l) within the array C and by the use of the symmetry properties of the Operator A, these properties allow us to use one of the other three representations of the collision pair (i,j,k,l) that are given by 2.1.2.4 (i)b. The efficiency of this approach increases with the number of collision pairs i.e. the number of velocities $|\mathfrak{V}|$. For a grid of medium size, like $|\mathfrak{V}| = 123$, this decreased the overall number of serialization due to shared memory collisions and atomic operations by 85.5% leading to a 50% decrease of the calculation time.

(iv) Last but not least we want to point at a number of minor (or obvious) optimizations. On a GPU we have a relatively complex memory hierarchy consisting of global memory, shared memory, constant memory, texture memory and local memory. The NVIDIA programming guide [NVI14] explains very well which memory region should be used for a specific purpose. We have not said a word about the parallelization of the transport time step, because the transport step only accounts for less than 5% of the computation time of our algorithm, at least for sufficiently large velocity spaces (at 9 velocities it accounts for 50 % on the CPU). But because of the low memory bandwidth between the RAM and the graphics cards memory (global memory) we implemented a GPU parallelized version of the transport time step that is heavily memory throughput bound (speedup between 5 and 10 for different velocity space discretizations). This enabled us to hold the main data grids A and B only in the graphic cards main memory so we are able to completely avoid the transfer of these grids to the host computer RAM. The only data that needs to be transferred back from the graphics card to the RAM during the computation are the macroscopic values in the arrays m_1 to m_5. That are only 6 floating point values per position space point, a fraction of the $|\mathfrak{V}|$ values per position space point in the arrays A, B.

Appendix A.3 explains how the information about bandwidth, register, instruction, SSE and thread usage was obtained.

5.3 Parallelization results

5.3.1 Methodology

Hardware

The used hardware consists of a Core I7 960 CPU (with HTT and SSE) and a Geforce GTX 580 graphic cards. The theoretic capabilities of the used CPU and GPU are shown in table 5.1, the used legend corresponds to figure 5.3. The GFLOPs and bandwidth comparison leads to a theoretic speedup factor around 15 with a theoretical performance per price ratio of 10.6. The CPU resp. GPU were launched in October 2009 resp. November 2010. A corresponding high end Intel®CPU from 2010 (I7 970, released in June) should be around 50%[1] faster than the used model but had the same launch price as the used GPU. So the used CPU has the same performance per price

Processor	C/MP	S/SP	GFLOPs	Bandwidth GB/s	Price €
Core I7 960	4	4	105.1[1]	12.8[2]	240
GF 110 (GTX580)	16	32	1581[3]	192.4[3]	363

Table 5.1: *CPU / GPU comparison*

ratio as the direct competitor of the GPU.

Compiler / Software, Benchmark Problem
For the following experiments and CPU / GPU comparisons we used the GCC compiler version 4.5.3 with the optimization flags *-march=native -O3* and parallelized through the usage of pthreads for the CPU part and the *CUDA* (Compute Unified Device Architecture) toolkit version 4.1 with the optimization flags *-O3 -arch sm_20* for the GPU part of the program. All calculations were done in single precision. Beside the main program, which does the calculations, we developed a *graphical user interface* (GUI) that uses OpenGL for a visualization and the interactive creation of position spaces and discretization of the velocity space. It is possible to create three dimensional position spaces with complex objects inside it. The GUI also possesses the capability of data inspection of the macroscopic values, automatic picture and video creation, calculation of streak and stream lines and three dimensional visualization via visualization of the macroscopic values on planes in the position space. For the speed comparison between CPU and GPU we timed the collision, transport, communication step and the calculation of macroscopic values and summed them up. We used the problem of a three dimensional, mach one gas flow through a pipe with Knudsen numbers in the range [0.04,0.2] and position space grids that contain between 65536 and 1048576 points.

5.3.2 Results

The wall-times for the following discussion can be found in appendix A.2. In figure 5.4 we see a comparison of four different versions of the algorithm in such a way that we see the speedup factor over the size of the discretized velocity space. The first thing we have to mention is that the speedups generally increase with the size of the velocity space (at least between 9 and 57 velocities). That is due to the fact that our algorithm is memory bandwidth bound for v-space grids with less than 50 velocities. This change from memory bandwidth bound to FLOP bound comes from the quadratic growth of the number of collision pairs with the number of velocities used to discretize the velocity space. In figure 5.4 it is shown that a SSE parallelized version of our algorithm, that uses only one core of the CPU, is around three times as fast as a not

[1] Intel®specifications [Int11]
[2] Theoretical peak transfer rate of the used DDR3-1600 RAM
[3] NVIDIA®Kepler whitepaper [NVI12] p. 6

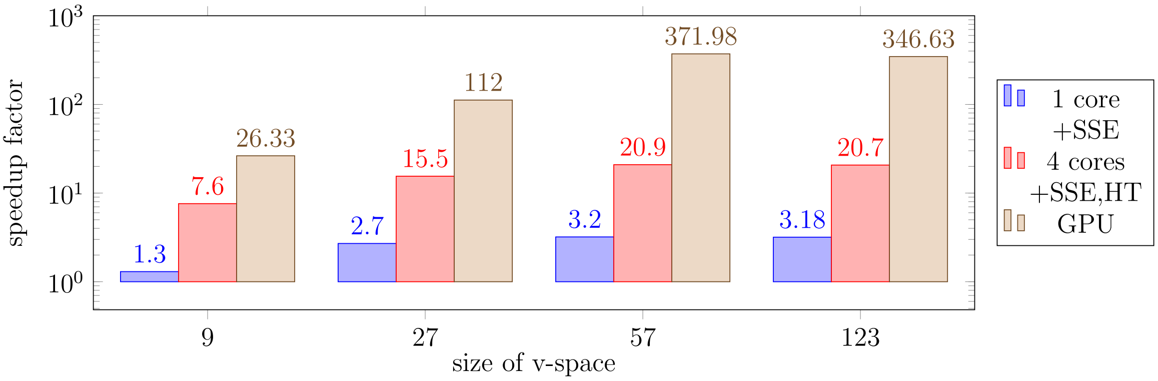


Figure 5.4: *speed comparison of not parallelized CPU vs SSE CPU vs parallelized SSE CPU vs GPU version*

parallelized version. If we use the SSE together with the utilization of all four cores and hyper-threading (eight virtual cores, which can be utilized through the usage more than 4 threads) on this CPU we get an additional speedup of around 6.5 thus giving us a total speedup of around 20 for an efficient CPU parallelized implementation. This lies above the estimated speedup of $4 \cdot 4 = 16$, where we should get a speedup around 4 due to SSE and another speedup around 4 due to the 4 used cores. It is not entirely clear why this super linear speedup occurs, but it possibly has to do with the implicit cache blocking and the usage of Hyper Threading as well as an implicit load balancing due to the larger number of threads. As we can see the GPU is another three to 18 times faster than the whole CPU.

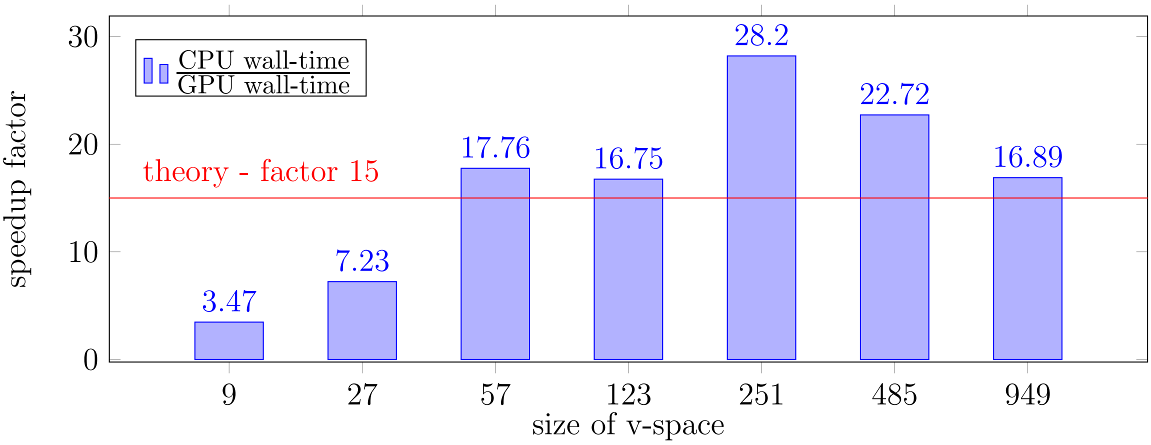


Figure 5.5: *speed comparison of fully parallelized CPU vs GPU version*

Figure 5.5 shows the comparison of a fully parallelized CPU vs a GPU implementation on a wider number of velocity spaces. As we can see the GPU implementation is much better on big velocity spaces and reaches speedup factors above the theoretical speedup of 15. That is due to parallelization level two for the GPU, the additional optimization

(iii) and the fact, that it is not possible (in a simple way) to explicitly program and use the CPU caches whereas we can easily optimize the GPU cache usage for our problem. The main parallelization on the GPU is focused on a huge amount of collision pairs, where shared memory collisions can be avoided through reordering of the collision pairs and where every SP has to calculate many operations that correspond to collision pairs. Since our aim was to create an efficient implementation for big velocity spaces, we are satisfied with these results. An implementation that focuses on small velocity spaces would look very different, because the problem is memory throughput bound on small spaces. If we want to compare the GPU with a single threaded, not optimized (in the context of parallelization) CPU version we can look at the speedup factor at $|\mathfrak{V}| = 251$ and multiply it with the CPU speedup factor of 20 giving a total speedup of around 600 (which could maybe be reduced by further optimization of the CPU version for this specific lattice).

Remark 5.3.2.1

(i) An important thing we have to mention here is that a GPU implementation is much more complicated than a simple (not parallelized) CPU implementation and because of this the development of such programs simply need more time. So a GPU implementation should be compared with a parallelized and optimized CPU implementation that was developed with a comparable or at least an adequate amount of time. Unfortunately only a fraction of the scientific GPU programmers is following this idea and because of this there are always publications in which the authors claim to get speedups of around two magnitudes or more compared to recent CPUs. As one can easily see, parallelization on one Intel®processor already gives speedups of more than one magnitude. Because of this we made a realistic comparison of GPUs and CPUs, that means we compared a parallelized and optimized CPU implementation that uses SSE with an optimized (regarding memory access restrictions and bandwidth usage) GPU implementation (figure 5.5).

(ii) The physical phenomena that have been successfully simulated through this approach include Rayleigh-Bènard convection, Kàrmàn vortex streets around obstacles, shock fronts produced by supersonic flows and the Knudsen pump in two and three dimensions. A more detailed description of the latter 3, together with simulation results and visualization of the results can be found in [Bre12].

(iii) We have shown that typical parallelization strategies used to create parallelized CPU implementations can partially be used to develop efficient GPU implementations. Finally we have calculated a 10.6 times better performance per initial price ratio (assuming a speedup of 16) for a GF110 GPU over an Intel Core I7 9xx series CPU on sufficiently large velocity spaces. This result should typically hold true for FLOP bound algorithms that are well suited for parallelization.

A Appendix

A.1 Algorithms belonging to chapter 2

Algorithm A.1.1 (Kernel computation to determine artificial collision invariants) This algorithm belongs to 2.1.2.8. This is a simple and inefficient implementation that goes through all permutations of length 4 in the velocity space.

```
function kernel = kernel(dim)
#this function calculates the dimension of the space of collision invariants,
#usage: kernel(dimension), where dimension equals to the dimensionality
#of the velocity space.

i=1;
# creating velocity space
for x=-1:1:1
  for y=-1:1:1
    if dim == 3
      for z=-1:1:1
        V(i,:)=[x,y,z];
        i=i+1;
      end
    else
      V(i,:)=[x,y];
      i=i+1;
    end
  end
end
Mzt=zeros(1,size(V)(1));

#calculating all possible collision pairs
c = 1;
for i=1:1:size(V)(1)
  for j=1:1:size(V)(1)
    for k=1:1:size(V)(1)
      for l=1:1:size(V)(1)
        vec_m=(V(i,:)+V(j,:))/2;
        vec_d2=V(i,:)-V(j,:);
        vec_d1=V(k,:)-V(l,:);
        if (
        # diameter of the squares is 2 or sqrt(2)
          (abs(norm(vec_d2)-2)<1e-6 || abs(norm(vec_d2)-sqrt(2))<1e-6) &&
        # the diagonals are orthogonal
          abs(vec_d2 * vec_d1') < 1e-6 &&
        # and the center of the diagonals are equal
          norm((V(i,:)+V(j,:))/2 - (V(k,:)+V(l,:))/2) < 1e-6
        # the length of the diagonals is equal
          && abs(norm(vec_d1) - norm(vec_d2))<1e-6
        # the edges of the square are pairwise different
          && i != j && i != k && i != l && j !=k && j != l && k != l
        )
        Mzt(c,k)=1;
        Mzt(c,l)=1;
        Mzt(c,i)=-1;
        Mzt(c,j)=-1;
        c = c + 1;
        end
      end
    end
  end
end
# dropping doppelgänger from the set of collision pairs
for i=1:1:size(Mzt)(1)
  j=i+1;
  while j<=size(Mzt)(1)
    if Mzt(i,:)==Mzt(j,:) || Mzt(i,:)==-Mzt(j,:)
      Mzt(j,:)=[];
    else
      j=j+1;
    end
  end
end
Mzt
printf("above is the matrix corresponding to the possible collision pairs.\n")
printf("size of this matrix: %dx%d.\n“,size(Mzt)(1),size(Mzt)(2))
printf("dimension of the kernel is %d\n“,size(V)(1)-rank(Mzt))
```

A.2 Wall-times for benchmarks

All time measurements in table A.1 are given in seconds. The difference between many equal runs was less than 1%, so these values can be considered as the mean wall time of the Calculations. These large time frames were choosen, because the CPU version needed a long “burn in” time before it reached its real (and constant) performance.

Velocities	1 T no SSE	1 T SSE	16 T SSE	GPU	Grid Points
9	17689	13628	2328	671	1024x512
27	249914	92654	16126	2231	256x64x64
57	308673	96477	14734	829	128x64x64
123	408180	128218	19721	1177	128x64x64
251	-	-	37568	1332	64x64x64
485	-	-	22754	1001	64x32x32
949	-	-	65415	3873	64x32x32

Table A.1: *mean wall-time of CPU and GPU benchmarks, 1 T equals 1 thread*

A.3 Methods used for performance analysis

CPU

On the CPU we measured the wall-time for program runs with different numbers of threads used to do the calculations. This was simple to achieve, because we designed our program in such a way, that the number of threads is one of the input parameters of the program. When it comes to SSE usage the situation is more complex. If one compiles a program (assuming gcc compiler) with the optimization flag *-O3* the compiler automatically tries to vectorize (as part of the optimization) and thus uses SSE instructions. But without the usage of keywords like *const* and *restrict* and an already vector operation adapted code, the automatic vectorization doesn’t yield any speed improvement (at least in our case).

That means if we speak about a program version without SSE instructions, we speak about an efficient C implementation without specific compiler hints or code restructuring aiding automatic vectorization. The SSE version of the code uses explicit SSE instructions and needed a redesign of the data structures and the code within the optimized functions, in our case only the collision step. The difference between the SSE and the non SSE code are approximately 250 lines of C++ code, including a wrapper class for the SSE instructions and registers to overload the common operators and simplify the code development.

GPU

On the GPU we simply used the NVIDIA Visual Profiler that is included in the CUDA toolkit (the executable is called nvv in CUDA 4.x and nvvp in CUDA 5.x). This program can be used to profile any CUDA binary without the need of specific compiler options or recompilation and it automatically handles the necessary logfiles and displays the information in a readable and understandable way. It can be used to collect and display all necessary hardware and software counters (for that it starts the program multiple times and collects the log files) and it provides automatic calculation of performance metrics like “Branch Efficiency” or “Achieved Occupancy”.

References

[Bab08] Hans Babovsky. Kinetic models on orthogonal groups and the simulation of the Boltzmann equation. In T. Abe, editor, *Proceedings of* 26^{th} *International Symposium on Rarefied Gas Dynamics*, volume 1084 of *AIP Conference Proceedings*, pages 415–420, 2008. Cited on pages 2 and 5.

[Bab09] Hans Babovsky. A numerical model for the Boltzmann equation with applications to micro flows. *Comput. Math. Appl.*, 58(4):791–804, August 2009. Cited on pages 2, 5, 19, 20, 21, and 29.

[Bab11a] Hans Babovsky. Kinetic Lattice Group Models: Structure and Numerics. In 27^{th} *International Symposium on Rarefied Gas Dynamics*, volume 1333 of *American Institute of Physics Conference Series*, pages 63–68, May 2011. Cited on pages 2 and 29.

[Bab11b] Hans Babovsky. Numerical Simulation of the Boltzmann Equation: Deterministic vs. Monte Carlo Schemes. *PAMM*, 11(1):759–760, 2011. Cited on page 2.

[Bab12] H. Babovsky. "Small" kinetic models for transitional flow simulations. In M. Mareschal and A. Santos, editors, 28^{th} *International Symposium on Rarefied Gas Dynamics*, volume 1501 of *American Institute of Physics Conference Series*, pages 272–278, November 2012. Cited on page 2.

[Bab14] Hans Babovsky. Discrete kinetic models in the fluid dynamic limit. *Computers & Mathematics with Applications*, 67(2):256 – 271, 2014. Mesoscopic Methods for Engineering and Science (Proceedings of ICMMES-2012, Taipei, Taiwan, 23–27 July 2012). Cited on page 2.

[BG03] N. Bellomo and R. Gatignol, editors. *Lecture Notes on the Discretization of the Boltzmann Equation*, volume 63 of *Series on Advances in Mathematics for Applied Sciences*. World Scientific, 2003. Cited on page 2.

[BPS95] Alexandre V. Bobylev, Andrzej Palczewski, and Jacques Schneider. On approximation of the boltzmann equation by discrete veclocity models. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 320:639–644, 1995. Cited on pages 2 and 5.

[BR97] A V Bobylev and S Rjasanow. Difference scheme for the boltzmann equation based on the fast fourier transform. *European J. Mech. B Fluids*, 16:293–306, 1997. Cited on page 2.

[BR00] A. V. Bobylev and S. Rjasanow. Numerical solution of the boltzmann equation using a fully conservative difference scheme based on the fast fourier transform. *Transport Theory and Statistical Physics*, 29(3-5):289–310, 2000. Cited on pages 2 and 128.

[Bre12] S. Brechtken. Lattice group models: GPU acceleration and numerics. In M. Mareschal and A. Santos, editors, 28^{th} *International Symposium on Rarefied Gas Dynamics*, volume 1501 of *American Institute of Physics Conference Series*, pages 239–246, November 2012. Cited on page 144.

[Bue96] C. Buet. A discrete-velocity scheme for the boltzmann operator of rarefied gas dynamics. *Transport Theory and Statistical Physics*, 25(1):33–60, 1996. Cited on pages 2 and 5.

[BV12] A. V. Bobylev and M. C. Vinerean. Symmetric extensions of normal discrete velocity models. In M. Mareschal and A. Santos, editors, 28^{th} *International Symposium on Rarefied Gas Dynamics*, volume 1501 of *American Institute of Physics Conference Series*, pages 254–261, November 2012. Cited on page 2.

[CGL03] H. Cabannes, R. Gatignol, and L.S. Luo. *The Discrete Boltzmann Equation: Theory and Applications.* University of California, College of engineering, nov 2003. Cited on page 2.

[CIP94] Carlo Cercignani, Reinhard Illner, and Mario Pulvirenti. *The Mathematical Theory of Dilute Gases.* Springer series in Applied Mathematical Sciences. Springer-Verlag, New York, 1994. Cited on page 5.

[CSB87] J. H. Conway, N. J. A. Sloane, and E. Bannai. *Sphere-packings, Lattices, and Groups.* Springer-Verlag New York, Inc., New York, NY, USA, 1987. Cited on page 19.

[FKW06] Laura Fainsilber, Pär Kurlberg, and Bernt Wennberg. Lattice points on circles and discrete velocity models for the boltzmann equation. *SIAM J. Math. Anal*, 37:1903–1922, 2006. Cited on page 2.

[FMP06] F. Filbet, C. Mouhot, and L. Pareschi. Solving the boltzmann equation in n log2n. *SIAM Journal on Scientific Computing*, 28(3):1029–1053, 2006. Cited on page 2.

[FR03] Francis Filbet and Giovanni Russo. High order numerical methods for the space non-homogeneous boltzmann equation. *J. Comput. Phys.*, 186(2):457–480, April 2003. Cited on page 2.

[Gra12] Màrio M. Graca. Quadrature as a least-squares and minimax problem. arXiv1206.0281v1, `http://arxiv.org/abs/1206.0281v1`, 2012. Cited on page 125.

[Huy09] Daan Huybrechs. Stable high-order quadrature rules with equidistant points. *J. Computational Applied Mathematics*, 231(2):933–947, 2009. Cited on page 125.

[HW60] G. H. Hardy and Edward Maitland Wright. *An introduction to the theory of numbers.* Clarendon Press Oxford, 4th ed., 2nd (corr.) impression. edition, 1960. Cited on pages 31 and 33.

[Int11] Intel core i7-900 desktop processor series. `http://download.intel.com/support/processors/corei7/sb/core_i7-900_d.pdf` [Acccessed: 2013-04-23], September 2011. Cited on page 142.

[IR02] I. Ibragimov and S. Rjasanow. Numerical solution of the boltzmann equation on the uniform grid. *Computing*, 69(2):163–186, 2002. Cited on pages 2 and 128.

[IW93] Reinhard Illner and Wolfgang Wagner. A random discrete velocity model and approximation of the boltzmann equation. *Journal of Statistical Physics*, 70(3-4):773–792, 1993. Cited on page 2.

[MS00] Philippe Michel and Jacques Schneider. Approximation simultanée de réels par des nombres rationnels et noyau de collision de l'équation de Boltzmann. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 330(9):857–862, May 2000. Cited on pages 2, 5, 31, 44, 62, 95, and 96.

[NVI12] Whitepaper nvidia geforce gtx 680. `http://international.download.nvidia.com/webassets/en_US/pdf/GeForce-GTX-680-Whitepaper-FINAL.pdf` [Acccessed: 2013-04-23], 2012. Cited on page 142.

[NVI14] Nvidia cuda c programming guide. `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html` [Acccessed: 2014-11-14], August 2014. Cited on page 141.

[OLBC10] Frank W. Olver, Daniel W. Lozier, Ronald F. Boisvert, and Charles W. Clark. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. `http://dlmf.nist.gov/4.13`. Cited on page 45.

[PDCD93] S. Ponce Dawson, S. Chen, and G. D. Doolen. Lattice boltzmann computations for reaction-diffusion equations. *The Journal of Chemical Physics*, 98(2):1514–1523, 1993. Cited on page 1.

[PH99] Vladislav A. Panferov and Alexei G. Heintz. A New Consistent Discrete-Velocity Model for the Boltzmann Equation. *Math. Methods Appl. Sci*, 25:571–593, 1999. Cited on pages 2, 5, 31, 44, 95, 96, and 128.

[PI88] Tadeusz Płatkowski and Reinhard Illner. Discrete velocity models of the Boltzmann equation: A survey on the mathematical aspects of the theory. *SIAM Review*, 30(2):213–255, June 1988. Cited on pages 2, 5, and 8.

[PS98] A Palczewski and J Schneider. Existence, stability, and convergence of solutions of discrete velocity models to the boltzmann equation. *J. Statist. Phys*, (91):1–2, 1998. Cited on pages 2 and 5.

[PSB97] Andrzej Palczewski, Jacques Schneider, and Alexandre V. Bobylev. A consistency result for a discrete-velocity model of the boltzmann equation. *SIAM J. Numer. Anal.*, 34(5):1865–1883, October 1997. Cited on pages 2 and 39.

[Raa04] D Raabe. Overview of the lattice boltzmann method for nano- and microscale fluid dynamics in materials science and engineering. *Modelling and Simulation in Materials Science and Engineering*, 12(6):R13, 2004. Cited on page 1.

[RS94] Francois Rogier and Jacques Schneider. A direct method for solving the boltzmann equation. *Transport Theory and Statistical Physics*, 23(1-3):313–338, 1994. Cited on pages 2, 5, 31, 33, 34, 39, and 62.

[RSW98] Sergej Rjasanow, Thomas Schreiber, and Wolfgang Wagner. Reduction of the number of particles in the stochastic weighted particle method for the boltzmann equation. *Journal of Computational Physics*, 145(1):382 – 405, 1998. Cited on page 2.

[RW98] S. Rjasanow and W. Wagner. A generalized collision mechanism for stochastic particle schemes approximating boltzmann-type equations. *Computers & Mathematics with Applications*, 35(1–2):165 – 178, 1998. Cited on page 2.

[RW07] Sergej Rjasanow and Wolfgang Wagner. Stochastic weighted particle method, theory and numerical examples. *Bulletin of the Institute of Mathematics Academia Sinica (New Series)*, 2(2):461 – 493, 2007. Cited on page 2.

[SBH91] Sauro Succi, Roberto Benzi, and Francisco Higuera. The lattice boltzmann equation: A new tool for computational fluid-dynamics. *Physica D: Nonlinear Phenomena*, 47(1–2):219 – 230, 1991. Cited on page 1.

[TR04] Nils Thuerey and U. Ruede. Free Surface Lattice-Boltzmann fluid simulations with and without level sets. *Proc. of Vision, Modelling, and Visualization VMV*, pages 199–207, 2004. Cited on page 1.

[Wag95] Wolfgang Wagner. Approximation of the boltzmann equation by discrete velocity models. *Journal of Statistical Physics*, 78(5-6):1555–1570, 1995. Cited on page 2.

List of Symbols

$\mathbb{N}, \mathbb{R}$	the natural and real numbers
$\mathbb{R}^n_{\mathfrak{r}}, \mathbb{R}^n_{\mathfrak{v}}$	the n-dimensional position and velocity space
$\mathbb{R}_t$	the time space
$\begin{bmatrix}\bullet\pm\\ \vdots\\ \bullet\end{bmatrix}$	$= \bullet \pm \ldots \pm \bullet$, alternative to represent long sums
$\mathtt{0}, \mathtt{1}, \ldots$	$= \begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix}, \begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}, \ldots$
$\mathfrak{v}, \mathfrak{w}, \mathbf{v}, \mathbf{w}$	fraktur and bold characters are elements of $\mathbb{R}^n, n = 2, 3$
$\mathfrak{v} = \begin{pmatrix}v_1\\ v_2\\ v_3\end{pmatrix}$	indexed normal characters are components of vectors
$\mathfrak{v}^2$	$:= \langle \mathfrak{v}, \mathfrak{v} \rangle$
a, b, c, d	normal characters are scalars
$f, f_i, \mathbf{f}_i$	a function $f : \mathbb{R}^n_{\mathfrak{v}} \to \mathbb{R}^+_{\{0\}}$, a value of a function $f_i := f(\mathfrak{v}_i)$ and a function of a function set, most likely a basis vector of a function space, only used to avoid any possibility of confusion
A	$:= \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$ the set of angles for the completion of the two dimensional approximation
$A^{\bullet,\bullet}_{\bullet,\bullet}$	the coefficient tensor corresponding to DVMs
$A^{\bullet}_{\bullet,\bullet}$	the reduced coefficient tensor, using that the last index can be represented through the first 3
$B_r(\mathfrak{a})$	$= \{\mathfrak{v} \in \mathbb{R}^n \mid \parallel \mathfrak{v} - \mathfrak{a} \parallel \leq r\}$, ball with radius r around $\mathfrak{a}$
G	the automorphism group of a given lattice
H	the subgroup of G containing only $id, -id$
$H[\bullet]$	the H-functional corresponding to the H-operator and the H-theorem

I the collision operator in the Boltzmann - equation

J discretization of the collision operator I

L_{ij} $:= \lceil L \diagup \Delta v_{ij} \rceil \Delta v_{ij}$, length of the domain for the innermost integration in 2D

M the set of all collision pairs, 4 points are a collision pair if the corresponding operator A or α is nonzero

$M_{\mathfrak{V}}$ index set of the set $\mathfrak{V}$

M_i collision pairs containing $\mathfrak{v}_i$

P_i $:= \begin{pmatrix} q_i \\ p_i \end{pmatrix}$

P_{ij} $:= \begin{pmatrix} q_i q_j - p_i p_j \\ p_i q_j + p_j q_i \end{pmatrix}$

R_α rotation matrix in two dimensions with angle α around zero

$R_\alpha(\mathfrak{b})$ rotation matrix in three dimensions with angle α around $\mathfrak{b}$

$R_{\alpha,\beta}$ rotation matrix in three dimensions with angles α, β around zero

S^{n-1} the n-dimensional unit sphere

$S_r(\mathfrak{a})$ $= \{\mathfrak{v} \in \mathbb{R}^n \mid \ \parallel \mathfrak{v} - \mathfrak{a} \parallel = r\} = \partial B_r(\mathfrak{a})$, sphere around $\mathfrak{a}$ with radius r

S_{ij} $:= S_{ij}^{\mathfrak{V}}$

S_{ij}^B $= \{\mathfrak{v} \in B \mid \ \parallel \mathfrak{v} - \frac{\mathfrak{v}_i + \mathfrak{v}_j}{2} \parallel = \parallel \frac{\mathfrak{v}_i - \mathfrak{v}_j}{2} \parallel\}$, sphere with a diagonal from $\mathfrak{v}_i$ to $\mathfrak{v}_j$, unless stated otherwise we assume the Euclidean norm

Δv_{ij} $:= \Delta v r_{ij} = \Delta v \sqrt{(p_i^2 + q_i^2)(p_j^2 + q_j^2)}$, stepsize of the approximation of the innermost integration in 2D

$\mathfrak{F}_n$ arctan of the Farey sequence of order n

$\mathfrak{V}$ $= \overline{\mathfrak{V}} \cap B_r(\mathbf{0}), \quad r \in \mathbb{R}^+$, the finite discretization of the velocity space

$\mathfrak{V}_{\frac{1}{2}}$ $= \left\{\mathfrak{a} \mid \exists \mathfrak{v}, \mathfrak{w} \in \mathfrak{V} : \parallel \overrightarrow{\mathfrak{v}\mathfrak{w}} \parallel = \sqrt{n} \Delta v \wedge \mathfrak{a} = \frac{\mathfrak{v} + \mathfrak{w}}{2}\right\}$

$\alpha_{\bullet,\bullet}^{\bullet,\bullet}$ the coefficient operator corresponding to eLGpMs

$\alpha_{\bullet,\bullet}^{\bullet}$ the coefficient operator corresponding to LGpMs

$\alpha_{i,n}$	the Farey arcs corresponding to $\mathfrak{F}_n$, $i \in \{0, \dots, N\}$
$\mathbb{I}$	the space of collisional invariants
$\mathbb{1}_A(a)$	$= \begin{cases} 1, & \text{if } a \in A \\ 0, & \text{else} \end{cases}$, the indicator function over A
$\mathfrak{C}$	the set of discrete sphere center points for a LGpM, in this work commonly $\mathfrak{V} \cup \mathfrak{V}_{\frac{1}{2}}$, if not stated otherwise
$\omega(\bullet)$	parametrization of the unit sphere in 2 dimensions
$\omega(\bullet, \bullet)$	parametrization of the unit sphere in 3 dimensions
$\overline{\mathfrak{V}}$	$= \Delta v \mathbb{Z}^n, \quad \Delta v \in \mathbb{R}^+$, the infinite discretization of the velocity space
$\tilde{\mathfrak{F}}_n$	the Farey sequence of order n
$\mathfrak{v}'_1, \mathfrak{w}'_1$	standard representation of the post collision velocities
$\mathfrak{v}'_2, \mathfrak{w}'_2, \mathfrak{w}_2$	representation of the velocities after the transformations in 2 resp. 3 dimensions
$\mathfrak{v}'_3, \mathfrak{w}'_3, \mathfrak{w}_3$	representation of the velocities after the transformations in 2 resp. 3 dimensions using elements of the automorphism group for the completion of the approximation
e_i	the i-th unit vector of the space under consideration, 1 at the i-th position, zero otherwise
r_i	$:= \sqrt{p_i^2 + q_i^2}$
r_{ij}	$:= \sqrt{(p_i^2 + q_i^2)(p_j^2 + q_j^2)}$
MVT	mean value theorem