ESAIM: M2AN 55 (2021) 2567–2608
https://doi.org/10.1051/m2an/2021065

**ESAIM: Mathematical Modelling and Numerical Analysis**
www.esaim-m2an.org

# A NEW ENTROPY-VARIABLE-BASED DISCRETIZATION METHOD FOR MINIMUM ENTROPY MOMENT APPROXIMATIONS OF LINEAR KINETIC EQUATIONS [⋆]

Tobias Leibner[*] and Mario Ohlberger

**Abstract.** In this contribution we derive and analyze a new numerical method for kinetic equations based on a variable transformation of the moment approximation. Classical minimum-entropy moment closures are a class of reduced models for kinetic equations that conserve many of the fundamental physical properties of solutions. However, their practical use is limited by their high computational cost, as an optimization problem has to be solved for every cell in the space-time grid. In addition, implementation of numerical solvers for these models is hampered by the fact that the optimization problems are only well-defined if the moment vectors stay within the realizable set. For the same reason, further reducing these models by, *e.g.*, reduced-basis methods is not a simple task. Our new method overcomes these disadvantages of classical approaches. The transformation is performed on the semi-discretized level which makes them applicable to a wide range of kinetic schemes and replaces the nonlinear optimization problems by inversion of the positive-definite Hessian matrix. As a result, the new scheme gets rid of the realizability-related problems. Moreover, a discrete entropy law can be enforced by modifying the time stepping scheme. Our numerical experiments demonstrate that our new method is often several times faster than the standard optimization-based scheme.

**Mathematics Subject Classification.** 35L40, 35F61, 65M08, 65M70, 82C70.

Received July 8, 2020. Accepted October 4, 2021.

## 1. Introduction

Kinetic equations play an important role in many physical applications. One of the earliest and most prominent examples is the Boltzmann equation which was derived by the Austrian physicist Ludwig Boltzmann in 1872 [9] and still forms the basis for the kinetic theory of rarefied gases. The Boltzmann equation or similar kinetic equations proved to be applicable not only to classical gases but also to electron transport in solids and plasmas, neutron transport in nuclear reactors, photon transport in superfluids and radiative transfer, among others [13, 35, 39–41]. More recently, kinetic equations were also derived in the context of biological modelling, *e.g.*, for studying cell movement or wolf migration [10, 27, 30].

While analytic solutions can be derived in some special cases [22], usually kinetic equations have to be solved numerically. Due to their high dimensionality, directly solving kinetic equations with standard discretizations

*Keywords and phrases.* Moment models, minimum entropy, kinetic transport equation, model order reduction, realizability.

[⋆] *Supplementary Online Material is only available in electronic form at https://doi.org/10.1051/m2an/2021065/olm.*

Fachbereich Mathematik und Informatik, WWU Münster, Einsteinstrasse 62, 48149 Münster, Germany.
[*]Corresponding author: tobias.leibner@wwu.de; tobias.leibner@googlemail.com

© The authors. Published by EDP Sciences, SMAI 2021

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

(*e.g.*, finite difference methods) is often infeasible or restricted to very small grid sizes. For that reason, a variety of specialized approximate methods have been developed, many of which belong to the class of moment methods. Instead of computing the whole kinetic density function, moment approximations choose a set of weight functions (usually polynomials up to some order) on the velocity space and only track the weighted velocity averages (called moments) of the kinetic density with respect to these functions. This is usually done by performing a Galerkin projection of the original kinetic equation to the linear span of the weight functions. In general, the resulting moment equations are not closed and thus an ansatz for the velocity distribution has to be made. Choosing a linear combination of the weight functions gives the widely used $\mathrm{P}_N$ closure [39], where $N$ is the degree of the highest-order moments in the model. The $\mathrm{P}_N$ closure results in linear equations, is simple to implement and often gives reasonable results. However, it does not guarantee non-negativity of the approximated kinetic density. This sometimes leads to physically meaningless solutions, as the $\mathrm{P}_N$ solutions can, *e.g.*, contain negative values for the local particle density.

The so-called minimum-entropy moment models $\mathrm{M}_N$ [18, 45] avoid these problems by choosing the ansatz function such that it minimizes an entropy functional which usually models the (negative) physical entropy. The resulting closed system of equations is hyperbolic and dissipates the chosen entropy [38]. However, numerically solving the $\mathrm{M}_N$ equations requires the solution of a non-linear optimization problem at every point on the space-time grid. Although the optimization problems can be solved in parallel [2, 29, 34, 52], the computational cost for high moment orders still is prohibitively high in practical applications. Another drawback of the entropy-based moment closures is that the optimization problem is solvable only for so-called realizable moment vectors, *i.e.*, vectors that actually are moments of a positive density function. As explicit descriptions of the set of realizable moment vectors are usually not available, discretizations (especially of higher order) often struggle to keep the approximate solutions realizable [1, 14, 46, 53, 57, 61, 65].

A partial remedy for the high computational cost of the minimum entropy models could be additional model reduction, for example *via* reduced basis methods [47]. These methods generate a reduced description of the (discretized) equations first and then use this reduced model to perform the actual computations. In some cases, *e.g.*, if a given kinetic equation has to be solved many times for different parameters, this reduces overall computation time by several orders of magnitude. Generating the reduced model is usually done by constructing a low-dimensional linear subspace from solution trajectories and then projecting the problem to this subspace. This has been successfully done for the $\mathrm{P}_N$ models [31]. In the context of minimum-entropy moment models, however, this procedure is problematic as it does not preserve realizability, which may render the reduced model useless as it does not admit a solution.

Checking realizability is much easier when using piecewise linear bases instead of the standard polynomial basis on the whole velocity space [19, 20, 49, 59, 60, 62, 63]. In addition, the computational cost is significantly lower for these models. However, solving the optimization problems is still costly compared to linear models and maintaining realizability still requires additional limiters [63].

Another approach to fix the realizability issues is to introduce a regularization of the optimization problem [4]. The regularized problem admits a solution also for moments vectors that are not realizable and maintains most of the desirable properties of the original problem, at the cost of an additional approximation error (which, however, can be controlled by the regularization parameter). However, this approach still requires the solution of the (regularized) minimum entropy problem in each cell of the space-time grid.

In this paper, we will present a new discretization scheme for the minimum-entropy moment equations based on a transformation of the semi-discretized equations to entropy variables. The new scheme replaces the non-linear optimization problems by matrix inversions and inherently guarantees realizability. As a consequence, it avoids many of the problems described above. In addition, the new scheme is often significantly faster than the untransformed scheme and shows improved parallel scaling. Moreover, a discrete entropy law can be enforced for the new scheme by using a relaxed Runge–Kutta method. On the downside, adaptive timestepping is strictly needed for the transformed scheme. Moreover, numerically singular Hessian matrices will result in a failure of the scheme if no additional regularization is employed. However, we did not encounter such a situation during our extensive numerical experiments (despite the fact that the untransformed reference scheme had to use regularization in several of the tests).

This paper is organized as follows. First, in Section 2 we shortly recall the necessary background on minimum entropy moment models. In Section 3, the new scheme is presented and analysed. In Section 4, we give an outline of our implementation which is then used for the extensive numerical investigations in Section 5.

## 2. Minimum-entropy moment models

### 2.1. Kinetic transport equation

We consider the linear transport equation

$$\partial_t\psi + \mathbf{\Omega}\cdot\nabla_{\mathbf{x}}\psi + \sigma_a\psi = \sigma_s\,\mathcal{C}(\psi) + Q, \tag{2.1a}$$

which describes the density of particles with speed $\mathbf{\Omega}\in\mathcal{S}^2$ at position $\mathbf{x}\in X\subseteq\mathbb{R}^3$ and time $t\in T=[0,t_{\text{end}}]$ under the events of scattering (proportional to $\sigma_s(t,\mathbf{x})\geq 0$), absorption (proportional to $\sigma_a(t,\mathbf{x})\geq 0$) and emission (proportional to $Q(t,\mathbf{x},\mathbf{\Omega})\geq 0$). The equation is supplemented with initial condition and Dirichlet boundary conditions:

$$\psi(0,\mathbf{x},\mathbf{\Omega}) = \psi_{t=0}(\mathbf{x},\mathbf{\Omega}) \qquad \text{for } \mathbf{x}\in X, \mathbf{\Omega}\in\mathcal{S}^2, \tag{2.1b}$$

$$\psi(t,\mathbf{x},\mathbf{\Omega}) = \psi_b(t,\mathbf{x},\mathbf{\Omega}) \qquad \text{for } t\in T, \mathbf{x}\in\partial X, \mathbf{n}\cdot\mathbf{\Omega}<0, \tag{2.1c}$$

where $\psi_{t=0}$ and $\psi_b$ are given functions and $\mathbf{n}$ is the outward unit normal vector in $\mathbf{x}\in\partial X$. For simplicity, we will consider isotropic scattering

$$\mathcal{C}(\psi)(t,\mathbf{x},\mathbf{\Omega}) = \frac{1}{|\mathcal{S}^2|}\int\limits_{\mathcal{S}^2}\psi\Big(t,\mathbf{x},\tilde{\mathbf{\Omega}}\Big)\,\mathrm{d}\tilde{\mathbf{\Omega}} - \psi(t,\mathbf{x},\mathbf{\Omega}), \tag{2.2}$$

isotropic time-independent source $Q(t,\mathbf{x},\mathbf{\Omega}) = Q(\mathbf{x})$ and time-independent scattering $\sigma_s(t,\mathbf{x}) = \sigma_s(\mathbf{x})$ and absorption $\sigma_a(t,\mathbf{x}) = \sigma_a(\mathbf{x})$.

Parameterizing $\mathbf{\Omega}$ in spherical coordinates we obtain

$$\mathbf{\Omega} = \left(\sqrt{1-\mu^2}\cos(\varphi), \sqrt{1-\mu^2}\sin(\varphi), \mu\right)^T =: (\Omega_x,\Omega_y,\Omega_z)^T, \tag{2.3}$$

where $\varphi\in[0,2\pi]$ is the azimuthal and $\mu\in[-1,1]$ the cosine of the polar angle.

As a one-dimensional simplification, we will also consider the models in slab geometry, which is a projection of the sphere onto the $z$-axis [64]. The transport equation under consideration then has the form

$$\partial_t\psi + \mu\partial_z\psi + \sigma_a\psi = \sigma_s\,\mathcal{C}(\psi) + Q, \qquad t\in T, z\in X, \mu\in[-1,1]. \tag{2.4}$$

### 2.2. The moment approximation

In the following, $V$ will always denote the angular domain, *i.e.*, $V=[-1,1]$ in slab geometry and $V=\mathcal{S}^2$ in the three-dimensional case, and $\mathbf{\Omega}$ will denote the corresponding angular variable. Moreover, we will use angle brackets to denote integration over $V$, *i.e.*,

$$\langle f\rangle := \int\limits_V f(\mathbf{\Omega})\,\mathrm{d}\mathbf{\Omega} \quad \text{for all } f\in L^1(V).$$

Due to the high-dimensionality, directly discretizing and solving (2.1) *via* standard numerical schemes is usually not viable. We will thus consider moment approximations of (2.1). These models transfer the kinetic equation to a coupled system of PDEs for weighted velocity averages (moments) of the solution.

**Definition 2.1.** The vector of functions $\mathbf{b}\colon V \to \mathbb{R}^n$ consisting of $n$ linearly-independent *basis functions* $b_l \in L^1(V)$, $l \in \{0, \dots n-1\}$, is called a *moment basis*. The *moments* $\mathbf{u}_{\mathbf{b},\psi} = (u_0, \dots, u_{n-1})^T \in \mathbb{R}^n$ with respect to the basis $\mathbf{b}$ of a given density function $\psi \in L^1(V)$ are then defined by

$$\mathbf{u}_{\mathbf{b},\psi} := \langle \mathbf{b}\psi \rangle, \tag{2.5}$$

where the integration is performed component-wise. Furthermore, the vector $\mathbf{u}_{\mathbf{b}}^{\text{iso}} := \langle \mathbf{b} \rangle$ is called the *isotropic moment.*

**Definition 2.2.** The quantity $\rho_\psi := \langle \psi \rangle \in \mathbb{R}$ is called the *local particle density* of the function $\psi$. If we assume that there exists a vector $\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \in \mathbb{R}^n$ such that

$$\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{b} \equiv 1, \tag{2.6}$$

we have

$$\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{u}_{\mathbf{b},\psi} = \left\langle \boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{b}\psi \right\rangle = \langle \psi \rangle = \rho_\psi.$$

Hence, we define the local particle density of the moment vector $\mathbf{u} \in \mathbb{R}^n$ with respect to the basis $\mathbf{b}$ as

$$\rho_{\mathbf{b}}(\mathbf{u}) := \boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{u}. \tag{2.7}$$

**Remark 2.3.** The vector $\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}}$ exists for all bases regarded in this paper (see Sect. 2.4).

In the following, if basis $\mathbf{b}$ or density function $\psi$ are clear from the context, we will usually omit the corresponding subscripts.

Equations for the moments $\mathbf{u}$ can be obtained by multiplying (2.1) with $\mathbf{b}$ and integrating over $V$, yielding

$$\langle \mathbf{b}\partial_t \psi \rangle + \langle \mathbf{b}(\boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} \psi) \rangle + \sigma_a \langle \mathbf{b}\psi \rangle = \sigma_s \langle \mathbf{b}\, \mathcal{C}(\psi) \rangle + \langle \mathbf{b}Q \rangle.$$

Collecting known terms, and interchanging integration and differentiation where possible, the moment system has the form

$$\partial_t \mathbf{u} + \langle \mathbf{b}(\boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} \psi) \rangle + \sigma_a \mathbf{u} = \sigma_s \langle \mathbf{b}\, \mathcal{C}(\psi) \rangle + \langle \mathbf{b}Q \rangle. \tag{2.8}$$

For isotropic collision operator (2.2), the scattering term becomes

$$\langle \mathbf{b}\, \mathcal{C}(\psi) \rangle = \left\langle \mathbf{b} \left( \frac{\langle \psi \rangle}{|V|} - \psi \right) \right\rangle = \frac{\langle \psi \rangle}{|V|} \langle \mathbf{b} \rangle - \mathbf{u} = \frac{\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{u}}{|V|} \langle \mathbf{b} \rangle - \mathbf{u} = \left( \mathbf{G}_{\mathbf{b}}^{\text{iso}} - \mathbf{I} \right) \mathbf{u} \tag{2.9}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the unit matrix and

$$\mathbf{G}_{\mathbf{b}}^{\text{iso}} = \frac{1}{|V|} \langle \mathbf{b} \rangle \left( \boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \right)^T \in \mathbb{R}^{n \times n} \tag{2.10}$$

is the matrix mapping the moment vector $\mathbf{u}$ to the isotropic moment vector with the same density

$$\mathbf{G}_{\mathbf{b}}^{\text{iso}} \mathbf{u} = \mathbf{u}_{\mathbf{b}}^{\text{iso}} \cdot \frac{\rho_{\mathbf{b}}(\mathbf{u})}{|V|}.$$

Consequently, for isotropic scattering, equation (2.8) simplifies to

$$\partial_t \mathbf{u} + \langle \mathbf{b}(\boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} \psi) \rangle = \left( \sigma_s \mathbf{G}_{\mathbf{b}}^{\text{iso}} - \sigma_t \mathbf{I} \right) \mathbf{u} + \langle \mathbf{b}Q \rangle \tag{2.11}$$

where $\sigma_t = \sigma_a + \sigma_s$ is the *total cross section.*

However, even in the isotropic case, the transport term $\langle \mathbf{b}(\boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} \psi) \rangle$ usually cannot be given explicitly in terms of $\mathbf{u}$. For non-isotropic scattering operator, the same applies to the scattering term. Therefore, additional assumptions have to be made to close the unknown terms. A common approach is to replace $\psi$ in (2.11) by a moment-dependent ansatz $\hat{\psi}_{\mathbf{u},\mathbf{b}}$, resulting in a closed system of non-linear equations for $\mathbf{u}$:

$$\partial_t \mathbf{u} + \sum_{k=1}^{d} \partial_{x_k} \mathbf{f}_k(\mathbf{u}) = \mathbf{s}(\mathbf{x}, \mathbf{u}), \tag{2.12}$$

where

$$\mathbf{f}_k(\mathbf{u}) = \left\langle \Omega_k \mathbf{b} \hat{\psi}_{\mathbf{u},\mathbf{b}} \right\rangle \tag{2.13}$$

and

$$\mathbf{s}(\mathbf{x}, \mathbf{u}) = \left( \sigma_s \mathbf{G}_{\mathbf{b}}^{\mathrm{iso}} - \sigma_t \mathbf{I} \right) \mathbf{u} + \langle \mathbf{b} Q \rangle. \tag{2.14}$$

**Remark 2.4.** We will always assume that the ansatz exactly reproduces the moments, *i.e.*, $\left\langle \mathbf{b} \hat{\psi}_{\mathbf{u},\mathbf{b}} \right\rangle = \mathbf{u}$. Note that this may not be fulfilled by regularized moment approximations as regarded, *e.g.*, in [4].

It remains to specify the basis functions and the ansatz density $\hat{\psi}_{\mathbf{u},\mathbf{b}}$. In the following, we will often omit the $\mathbf{b}$-dependency of the ansatz function and only write $\hat{\psi}_{\mathbf{u}}$ if the basis is clear from the context.

## 2.3. Minimum-entropy closure

For the minimum entropy closure [21, 38, 44, 45], we choose a strictly convex and twice continuously differentiable *entropy density function* $\eta \colon D \subset \mathbb{R} \to \mathbb{R}$ and demand that the ansatz function minimizes the entropy functional

$$\mathcal{H}(\psi) = \langle \eta(\psi) \rangle \tag{2.15}$$

under the moment constraints

$$\langle \mathbf{b} \psi \rangle = \mathbf{u}. \tag{2.16}$$

Here, the minimum is simply taken over all functions $\psi = \psi(\mathbf{\Omega})$ such that $\mathcal{H}(\psi)$ is well-defined, *i.e.*,

$$\hat{\psi}_{\mathbf{u}} = \hat{\psi}_{\mathbf{u},\mathbf{b},\eta} = \underset{\{\psi \,|\, \mathrm{Range}(\psi) \subset D,\, \eta(\psi) \in L^1(V),\, \langle \mathbf{b}\psi \rangle = \mathbf{u}\}}{\operatorname{argmin}} \langle \eta(\psi) \rangle. \tag{2.17}$$

This problem, which must be solved over the space-time mesh, is typically solved through its strictly convex finite-dimensional dual,

$$\boldsymbol{\alpha}_{\mathbf{b},\eta}(\mathbf{u}) := \underset{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^n}{\operatorname{argmin}} \langle \eta_*(\mathbf{b} \cdot \tilde{\boldsymbol{\alpha}}) \rangle - \mathbf{u} \cdot \tilde{\boldsymbol{\alpha}}, \tag{2.18}$$

where $\eta_*$ is the Legendre dual of $\eta$. The first-order necessary conditions for the multipliers $\boldsymbol{\alpha}_{\mathbf{b},\eta}(\mathbf{u})$ show that the solution to (2.17), if it exists, has the form

$$\hat{\psi}_{\mathbf{u},\mathbf{b},\eta} = \eta_*'(\mathbf{b} \cdot \boldsymbol{\alpha}_{\mathbf{b},\eta}(\mathbf{u})), \tag{2.19}$$

where $\eta_*'$ is the derivative of $\eta_*$.

As in [29, 38, 59], for sake of simplicity, we focus on *Maxwell–Boltzmann entropy*

$$\eta(\psi) = \psi \log(\psi) - \psi, \tag{2.20}$$

which is used for non-interacting, classical particles as in an ideal gas. Thus, $D = (0, \infty)$ and (2.17) becomes

$$\hat{\psi}_{\mathbf{u}} = \underset{\left\{\psi \,|\, \psi \in L^1_+(V),\, \langle \mathbf{b}\psi \rangle = \mathbf{u}\right\}}{\operatorname{argmin}} \langle \eta(\psi) \rangle, \tag{2.21}$$

where

$$L^1_+ := \left\{ \psi \in L^1(V) \,\middle|\, \psi > 0 \text{ almost everywhere} \right\} \tag{2.22}$$

is the space of positive integrable functions. Further, we have $\eta_*(p) = \eta_*'(p) = \eta_*''(p) = \exp(p)$ and thus the minimum entropy ansatz (2.19) becomes $\hat{\psi}_{\mathbf{u}} = \exp(\mathbf{b} \cdot \boldsymbol{\alpha}_{\mathbf{b},\eta}(\mathbf{u}))$.

**Remark 2.5.** In principle, the new scheme described in Section 3 could be used in the same way with other physically relevant entropies, *e.g.*, the Bose–Einstein entropy

$$\eta_{\mathrm{BE}}(\psi) = \psi \log(\psi) - (1+\psi)\log(1+\psi).$$

In this case, the ansatz distribution is given by

$$\hat{\psi}_{\mathbf{u}} = (\eta_{\mathrm{BE}})'_*(\boldsymbol{\alpha}_{\mathrm{BE}}(\mathbf{u}) \cdot \mathbf{b}) = \frac{\exp(\boldsymbol{\alpha}_{\mathrm{BE}}(\mathbf{u}) \cdot \mathbf{b})}{1 - \exp(\boldsymbol{\alpha}_{\mathrm{BE}}(\mathbf{u}) \cdot \mathbf{b})}$$

with $\boldsymbol{\alpha}_{\mathrm{BE}} = \boldsymbol{\alpha}_{\mathbf{b},\eta_{\mathrm{BE}}}$. To ensure positivity, we thus have to keep the multipliers $\boldsymbol{\alpha}_{\mathrm{BE}}$ in the basis-dependent set

$$\{\boldsymbol{\alpha} \in \mathbb{R}^n \mid \boldsymbol{\alpha} \cdot \mathbf{b} < 0\},$$

*i.e.*, other than in the Maxwell–Boltzmann case, we again have to deal with realizability (compare Sect. 2.5), also in the transformed scheme. Depending on the basis, this might significantly complicate the implementation.

Using the entropy-based closure, the moment system (2.12) is hyperbolic and dissipates the chosen entropy [58] (for $\sigma_a = Q = 0$)

$$\partial_t \mathcal{H}\left(\hat{\psi}_{\mathbf{u}}\right) + \sum_{k=0}^{d-1} \partial_{x_k} \left\langle \Omega_k \eta\left(\hat{\psi}_{\mathbf{u}}\right)\right\rangle \leq 0, \tag{2.23}$$

*i.e.*, $\mathcal{H}\left(\hat{\psi}_{\mathbf{u}}\right) = \left\langle \eta\left(\hat{\psi}_{\mathbf{u}}\right)\right\rangle = \langle \eta \circ \eta'_*(\boldsymbol{\alpha}(\mathbf{u}) \cdot \mathbf{b})\rangle$ and $\left\langle \boldsymbol{\Omega} \eta\left(\hat{\psi}_{\mathbf{u}}\right)\right\rangle = \langle \boldsymbol{\Omega}\, \eta \circ \eta'_*(\boldsymbol{\alpha}(\mathbf{u}) \cdot \mathbf{b})\rangle$ form an entropy–entropy flux pair in the sense of hyperbolic systems.

## 2.4. Basis functions

We will consider three options for the basis functions $\mathbf{b}$: the full moment basis $\mathbf{f}_N$, the hat function basis $\mathbf{h}_n$ and the partial moment basis $\mathbf{p}_n$.

### *2.4.1. Full moment basis*

The full moment basis $\mathbf{f}_N$ is the standard choice and consists of polynomials of up to order $N$, resulting in $n = N + 1$ and $n = (N+1)^2$ basis functions in one and three dimensions, respectively. We will use Legendre polynomials in slab geometry and real spherical harmonics in the full three-dimensional setting. In one dimension, the isotropic moment is $\mathbf{u}^{\mathrm{iso}}_{\mathbf{f}_N} = \langle \mathbf{f}_N \rangle = (2, 0, 0, \ldots, 0)^T$ and the multiplier $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{f}_N}$ can be chosen as $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{f}_N} = (1, 0, 0, \ldots, 0)^T$. In three dimensions, we have $\mathbf{u}^{\mathrm{iso}}_{\mathbf{f}_N} = (\sqrt{4\pi}, 0, 0, \ldots, 0)^T = \boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{f}_N}$.

**Definition 2.6.** The minimum-entropy moment models using the $\mathbf{f}_N$ will be called $\mathrm{M}_N$ models, where $N$ is the maximal polynomial order of the basis functions.

### *2.4.2. First-order finite-element bases*

Models using the full moment basis show optimal (spectral) convergence for smooth problems. For non-smooth problems, however, instead of increasing the polynomial order $N$, it might be better to keep $N$ fixed and regard piecewise polynomials on increasingly refined partitions of the domain. We will here restrict ourselves to piecewise linear bases ($N = 1$) which avoid many of the performance and realizability problems of the classical polynomial models [59, 63]. In the following, we will shortly state the definitions of the first-order bases. For a more detailed introduction see [59].

To define the first-order bases, we choose a partition $\mathcal{P}$ dividing the velocity domain $V$ into intervals (slab geometry) or spherical triangles (three dimensions). Let $n_v$ and $n_e$ be the number of nodes (vertices) and elements (intervals or spherical triangles) of this partition, respectively.

The first basis of interest, the hat function basis $\mathbf{h}_n$, consists of $n = n_v$ continuous basis functions $h_l$ which, similar to the linear basis typically used in the continuous finite element method, fulfill the partition of unity property, *i.e.*, $\sum_{l=0}^{n-1} h_l \equiv 1$, and the Lagrange property, *i.e.*, each basis function evaluates to 1 at one node of the partition and to 0 at all other nodes.

The partial moment basis $\mathbf{p}_n$, on the other hand, is defined in analogy to the discontinuous finite element method and consists of the $n = 2n_e$ or $n = 4n_e$ (in one and three dimensions, respectively) basis functions for the space of piecewise linear functions on $\mathcal{P}$ that may be discontinuous between elements of the partition.

More precisely, in slab geometry, we will always choose the partition $\mathcal{P}$ as the equidistant partition of $V = [-1,1]$ into $k$ intervals $I_i = [\mu_i, \mu_{i+1}]$ given by the set of $k+1$ angular "grid" points $-1 = \mu_0 < \mu_1 = -1 + \frac{2}{k} < \cdots < \mu_{k-1} = -1 + (k-1)\cdot\frac{2}{k} < \mu_k = 1$. Given this partition, the continuous piecewise linear basis functions $\mathbf{h}_n = (h_0, \dots, h_{n-1})^T$ (hat functions) are defined as

$$h_l(\mu) = \mathbb{1}_{I_{l-1}} \frac{\mu - \mu_{l-1}}{\mu_l - \mu_{l-1}} + \mathbb{1}_{I_l} \frac{\mu - \mu_{l+1}}{\mu_l - \mu_{l+1}}, \tag{2.24}$$

where $\mathbb{1}_{I_i}(\mu)$ is the indicator function on the interval $I_i$ (with $I_{-1} \equiv I_k \equiv 0$) and $n = k+1$ is the number of basis functions. The isotropic moment is $\mathbf{u}^{\text{iso}}_{\mathbf{h}_n} = \left(\frac{1}{k}, \frac{2}{k}, \dots, \frac{2}{k}, \frac{1}{k}\right)^T$ and we have $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{h}_n} = (1,\dots,1)^T$ due to the partition of unity property.

The partial moment basis in slab geometry is given by $\mathbf{p}_n = (p_0, \dots, p_{n-1}) = \left(\mathbf{p}_{I_0}^T, \dots, \mathbf{p}_{I_{k-1}}^T\right)^T$ with

$$\mathbf{p}_{I_i}(\mu) = \begin{cases} (1,\mu)^T & \text{if } \mu \in \operatorname{int}(I_i), \\ (0,0)^T & \text{if } \mu \in [-1,1] \setminus I_i. \end{cases}$$

Here, $n = 2k$ is again the number of basis functions and $\operatorname{int}(I_i)$ is the interior of $I_i$. The isotropic moment is $\mathbf{u}^{\text{iso}}_{\mathbf{p}_n} = \frac{1}{2}\left(\frac{1}{k}, \mu_1^2 - \mu_0^2, \frac{1}{k}, \mu_2^2 - \mu_1^2, \dots\right)^T$ and $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{p}_n} = (1,0,1,0,\dots)^T$.

In three dimensions, the triangulation $\mathcal{P}$ will be obtained by dyadic refinement of the octants of the sphere $V = \mathcal{S}^2$, *i.e.*, the coarsest triangulation contains the eight spherical triangles obtained by projecting the octahedron with vertices $\{(\pm1,0,0)^T, (0,\pm1,0)^T, (0,0,\pm1)^T\}$ to the sphere and finer partitions are obtained by iteratively subdividing each spherical triangle into four new ones, adding vertices at the midpoints of the triangle edges. After $r$ refinements, we thus obtain $n_v(r) = 4^{r+1} + 2$ vertices and $n_e(r) = 2 \cdot 4^{r+1}$ spherical triangles.

To get a three-dimensional equivalent of the continuous hat function basis (2.24), we consider basis functions defined using spherical barycentric coordinates [12, 36, 51]. On each spherical triangle $\widehat{K} \in \mathcal{P}$ all elements of $\mathbf{h}_n$ are defined to be zero except for the three basis functions associated with the vertices of $\widehat{K}$. Denoting these vertices as $\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{\Omega}_3 \in \mathcal{S}^2$, the values of the corresponding basis functions $h_1, h_2, h_3$ are defined by requiring that

$$h_l(\boldsymbol{\Omega}_m) = \delta_{lm} \quad \text{(Lagrange property)}$$

for $l, m \in \{1,2,3\}$, and that, for every point $\boldsymbol{\Omega} \in \widehat{K}$,

$$h_1(\boldsymbol{\Omega}) + h_2(\boldsymbol{\Omega}) + h_3(\boldsymbol{\Omega}) = 1 \quad \text{(partition of unity)}$$

and

$$\boldsymbol{\Omega} \in \widehat{K} \text{ is the Riemannian center of mass with weights } h_l(\boldsymbol{\Omega}) \text{ and nodes } \boldsymbol{\Omega}_l.$$

As in one dimension, the resulting basis functions are non-negative. Due to the partition of unity property we again have $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{h}_n} = (1,\dots,1)^T$.

The three-dimensional discontinuous partial moment basis $\mathbf{p}_n$ is chosen analogously to the one-dimensional case as

$$\mathbf{p}_n = \left(\left(\mathbf{p}_{\widehat{K}_0}\right)^T, \dots, \left(\mathbf{p}_{\widehat{K}_{n_e-1}}\right)^T\right)^T$$

with

$$\mathbf{p}_{\widehat{K}}(\mathbf{\Omega}) = \begin{cases} (1, \Omega_x, \Omega_y, \Omega_z)^T & \text{if } \mathbf{\Omega} \in \operatorname{int}\left(\widehat{K}\right), \\ (0,0)^T & \text{if } \mathbf{\Omega} \in \mathcal{S}^2 \setminus \widehat{K}, \end{cases}$$

for any spherical triangle $\widehat{K} \in \mathcal{P} = \left\{\widehat{K}_i \,|\, i = 0, \dots, n_e - 1\right\}$. Here, $\operatorname{int}\left(\widehat{K}\right)$ is the interior of $\widehat{K}$. The unit multiplier is given by $\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{p}_n} = (1,0,0,0,1,0,0,0,\dots)^T$.

**Definition 2.7.** The minimum-entropy moment models using the $\mathbf{h}_n$ and $\mathbf{p}_n$ basis will be called HFM$_n$ and PMM$_n$ models, respectively, where $n$ is the number of moments (or equivalently, the number of basis functions, *i.e.*, the length of the vectors of functions $\mathbf{h}_n$ and $\mathbf{p}_n$).

## 2.5. Realizability

Since the dual problem (2.18) is strictly convex, a solution exists if and only if the first-order necessary conditions (compare (2.19)) are fulfilled, *i.e.*, (2.17) is solvable for moment vectors in the ansatz set

$$\mathcal{A}_{\mathbf{b},\eta} := \{\langle \mathbf{b}\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle \,|\, \boldsymbol{\alpha} \in \mathbb{R}^n\}. \tag{2.25}$$

For Maxwell–Boltzmann entropy, it can be shown [33, 59] that this set is equal to the *positively realizable set*

$$\mathcal{R}^+_{\mathbf{b}} := \left\{\mathbf{u} \in \mathbb{R}^n \,|\, \exists \psi \in L^1_+(V) \text{ such that } \mathbf{u} = \langle \mathbf{b}\psi\rangle\right\} \tag{2.26}$$

and that the map $\boldsymbol{\alpha}_{\mathbf{b},\eta} \colon \mathcal{R}^+_{\mathbf{b}} \to \mathbb{R}^n$ given by (2.18) is a diffeomorphism with inverse map

$$\mathbf{u}_{\mathbf{b},\eta} \colon \mathbb{R}^n \to \mathcal{R}^+_{\mathbf{b}}, \quad \mathbf{u}_{\mathbf{b},\eta}(\boldsymbol{\alpha}) := \langle \mathbf{b}\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle. \tag{2.27}$$

Vectors $\mathbf{u} \in \mathcal{R}^+_{\mathbf{b}}$ will be called *realizable*. Similar to the ansatz function, we will often omit one or all of the subscripts of $\mathbf{u}$ and $\boldsymbol{\alpha}$ if the corresponding dependencies are clear from the context. Motivated by the mapping (2.27), in the following, we will also refer to the multipliers $\boldsymbol{\alpha}$ as the *entropy variables* (or transformed variables) and to the realizable moments $\mathbf{u}$ as standard or original variables.

The realizable set is a convex cone that, depending on the choice of basis $\mathbf{b}$, may have a complicated structure. For example, a moment vector $\mathbf{u}$ is realizable with respect to the full-moment basis $\mathbf{f}_N$ if some ($\mathbf{u}$-dependent) Hankel matrices are positive definite [16]. This criterion is hard to test in practice, especially for large polynomial order $N$. As a consequence, given a moment vector $\mathbf{u} \in \mathbb{R}^n$, it can be very difficult to check whether $\mathbf{u}$ is realizable and even more difficult to compute a projection to the realizable set. This is a major problem for numerical solvers which have to ensure that the approximate solutions stay realizable during the whole solution process since otherwise the minimum entropy optimization problems are ill-posed.

In contrast, the realizability conditions for the piecewise linear bases are quite simple [59]. In particular, a moment vector is realizable with respect to $\mathbf{h}_n$ if and only if all its entries are positive [58, 59]:

$$\mathcal{R}^+_{\mathbf{h}_n} := \{\mathbf{u} \in \mathbb{R}^n \,|\, u_i > 0 \quad \text{for all } i \in \{0, \dots, n-1\}\}. \tag{2.28}$$

In this case, distinguishing realizable from non-realizable vectors is easy. Still, as we will see in the next section, also for the piecewise linear bases we have to take some extra measures (in particular, restrict the time step size) to ensure that the numerical solutions are always realizable. Moreover, for higher-order numerical schemes or reduced order-models, maintaining realizability at all times (without introducing large errors) is still challenging, also for the hat function models.

**Remark 2.8.** Realizability is further complicated by the fact that we usually cannot solve the velocity integrals analytically and have to approximate them by a numerical quadrature $\mathcal{Q}$. For the full moments, this can have a severe impact on the realizable set [1, 2], *i.e.*, the numerically realizable set $\mathcal{R}^{\mathcal{Q}}_{\mathbf{b}}$ obtained by replacing the integral in (2.26) by its quadrature approximation significantly differs from the realizable set $\mathcal{R}^+_{\mathbf{b}}$. In the following, for notational simplicity, we will neglect the quadrature-related complications and assume that the integrals are evaluated exactly.

### 2.6. Standard finite volume discretization

We consider two discretization schemes for the moment equations (2.12), a standard finite volume scheme presented in this section and a new scheme based on the identification (2.27) between realizable set and $\mathbb{R}^N$ (see Sect. 3). For simplicity, we will restrict ourselves to first-order schemes.

The reference scheme is a standard first-order finite volume scheme. Let $\mathcal{G} = \{T_i \,|\, i \in I_{\mathcal{G}} = \{0, \ldots, n_{\mathbf{x}} - 1\}\}$ be a numerical grid for the spatial domain $X$ with $n_{\mathbf{x}}$ elements such that

$$X = \bigcup_i T_i$$

and define

$$\overline{\mathbf{u}}_i(t) = \frac{1}{|T_i|} \int_{T_i} \mathbf{u}(t, \mathbf{x}) \, \mathrm{d}\mathbf{x}.$$

Integrating (2.12) over a grid cell $T_i$ and dividing by $|T_i|$ gives

$$\partial_t \overline{\mathbf{u}}_i + \frac{1}{|T_i|} \int_{T_i} \sum_{k=0}^{d-1} \partial_{x_k} \mathbf{f}_k(\mathbf{u}) \mathrm{d}\mathbf{x} = \frac{1}{|T_i|} \int_{T_i} \mathbf{s}(\mathbf{x}, \mathbf{u}).$$

Using the midpoint rule to approximate the source term

$$\frac{1}{|T_i|} \int_{T_i} \mathbf{s}(\mathbf{x}, \mathbf{u}) \, \mathrm{d}\mathbf{x} = \mathbf{s}(\mathbf{x}_i, \overline{\mathbf{u}}_i) + \mathcal{O}(\Delta x^2),$$

where $\mathbf{x}_i$ is the centre of grid cell $T_i$, we arrive at

$$\partial_t \overline{\mathbf{u}}_i + \frac{1}{|T_i|} \int_{T_i} \sum_{k=0}^{d-1} \partial_{x_k} \mathbf{f}_k(\mathbf{u}) \mathrm{d}\mathbf{x} = \mathbf{s}(\mathbf{x}_i, \overline{\mathbf{u}}_i).$$

By applying the divergence theorem, we obtain

$$\partial_t \overline{\mathbf{u}}_i + \frac{1}{|T_i|} \sum_{j \in \mathcal{N}(i)} \int_{S_{ij}} \mathbf{F}(\mathbf{u}) \mathbf{n}_{ij} = \mathbf{s}(\mathbf{x}_i, \overline{\mathbf{u}}_i),$$

where $\mathcal{N}(i)$ is the set of all indices of neighbors of $T_i$, $S_{ij} = T_i \cap T_j$ is the interface between grid cells $T_i$ and $T_j$, $\mathbf{n}_{ij}$ is the unit outer normal of $T_i$ on $S_{ij}$ and the flux matrix is given as $\mathbf{F}(\mathbf{u}) = (\mathbf{f}_0(\mathbf{u}), \ldots, \mathbf{f}_{d-1}(\mathbf{u})) \in \mathbb{R}^{n \times d}$.

Replacing the flux term by a numerical flux $\mathbf{g}_{ij}$ on $S_{ij}$, we get the semidiscrete form

$$\partial_t \overline{\mathbf{u}}_i + \frac{1}{|T_i|} \sum_j \mathbf{g}_{ij}(\overline{\mathbf{u}}_i, \overline{\mathbf{u}}_j) = \mathbf{s}(\overline{\mathbf{u}}_i). \tag{2.29}$$

In principle, we could use any numerical flux for hyperbolic equations, *e.g.*, the Lax–Friedrichs flux. We will, however, use a numerical flux which is specifically designed for the equations under consideration. Define the two half integrals

$$\langle \cdot \rangle_{+,\mathbf{n}} = \int\limits_{V^{+,\mathbf{n}}} \cdot \; \mathrm{d}\boldsymbol{\Omega} \qquad \text{and} \qquad \langle \cdot \rangle_{-,\mathbf{n}} = \int\limits_{V^{-,\mathbf{n}}} \cdot \; \mathrm{d}\boldsymbol{\Omega},$$

where

$$V^{+,\mathbf{n}} = \{\boldsymbol{\Omega} \in V \,|\, \boldsymbol{\Omega} \cdot \mathbf{n} > 0\}, \quad V^{-,\mathbf{n}} = \{\boldsymbol{\Omega} \in V \,|\, \boldsymbol{\Omega} \cdot \mathbf{n} < 0\}.$$

In the following, we will omit the normal vector if it is clear from the context and write, *e.g.*, $\langle\cdot\rangle_+$ instead of $\langle\cdot\rangle_{+,\mathbf{n}}$ in these cases. The kinetic flux is defined as [20,23,29,61]

$$\mathbf{g}_{ij}^{kin}(\mathbf{u}_i,\mathbf{u}_j) = \left(\left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\mathbf{u}_i}\mathbf{b}\right\rangle_{+,\mathbf{n}_{ij}} + \left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\mathbf{u}_j}\mathbf{b}\right\rangle_{-,\mathbf{n}_{ij}}\right)|S_{ij}|. \tag{2.30}$$

Using the kinetic flux, the semidiscrete form (2.29) becomes

$$\partial_t\overline{\mathbf{u}}_i + \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\left(\left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\overline{\mathbf{u}}_i}\mathbf{b}\right\rangle_+ + \left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\overline{\mathbf{u}}_j}\mathbf{b}\right\rangle_-\right) = \mathbf{s}(\mathbf{x}_i,\overline{\mathbf{u}}_i). \tag{2.31}$$

We will then use an explicit one-step scheme for the time discretization. For example, an explicit Euler discretization gives the fully discrete form

$$\begin{aligned}\overline{\mathbf{u}}_i^{\kappa+1} &= \overline{\mathbf{u}}_i^{\kappa} - \Delta t\left(\sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\left(\left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}}\mathbf{b}\right\rangle_+ + \left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\overline{\mathbf{u}}_j^{\kappa}}\mathbf{b}\right\rangle_-\right) - \mathbf{s}(\mathbf{x}_i,\overline{\mathbf{u}}_i^{\kappa})\right)\\ &= \overline{\mathbf{u}}_i^{\kappa} - \Delta t\mathbf{u}_i^{\upharpoonleft}\left(\overline{\mathbf{u}}_0^{\kappa},\overline{\mathbf{u}}_1^{\kappa},\dots,\overline{\mathbf{u}}_{n_{\mathbf{x}}-1}^{\kappa}\right),\end{aligned} \tag{2.32}$$

where $\overline{\mathbf{u}}_i^{\kappa}$ is the approximation of $\overline{\mathbf{u}}_i$ at time step $\kappa$ and we defined

$$\mathbf{u}_i^{\upharpoonleft}\left(\mathbf{u}_0^{\kappa},\mathbf{u}_1^{\kappa},\dots,\mathbf{u}_{n_{\mathbf{x}}-1}^{\kappa}\right) := \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\left(\left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\mathbf{u}_i^{\kappa}}\mathbf{b}\right\rangle_+ + \left\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\hat{\psi}_{\mathbf{u}_j^{\kappa}}\mathbf{b}\right\rangle_-\right) - \mathbf{s}(\mathbf{x}_i,\mathbf{u}_i^{\kappa}). \tag{2.33}$$

The scheme (2.32) requires the solution of the minimization problem (2.21) in every time step on each grid cell. The initial values thus have to be realizable and we have to limit the time step $\Delta t$ to ensure that the scheme yields realizable moments.

**Theorem 2.9.** *The numerical scheme* (2.32) *using a structured cubic grid with equally-sized grid cells with edge length $\Delta x$ is realizability-preserving under the CFL-like condition*

$$\Delta t < \frac{1}{\sigma_t^{\max} + \frac{\sqrt{d}}{\Delta x}} \tag{2.34}$$

*where $\sigma_t^{\max} = \max_{\mathbf{x}\in X}\sigma_t(\mathbf{x})$.*

*Proof.* We will generalize the proof of Corollary 3.17 from [58] to several dimensions. Let $\overline{\mathbf{u}}_i^{\kappa}$ be realizable. By (2.31), (2.14) and the definition of $\mathbf{G}_{\mathbf{b}}^{\mathrm{iso}}$ (compare (2.9)), we have

$$\begin{aligned}\overline{\mathbf{u}}_i^{\kappa+1} &= \left\langle\mathbf{b}\left(\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}} - \Delta t\left(\sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\left((\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^+\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}} + (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^-\hat{\psi}_{\overline{\mathbf{u}}_j^{\kappa}}\right) - \frac{\sigma_s}{|V|}\left\langle\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}}\right\rangle - Q + \sigma_t\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}}\right)\right)\right\rangle\\ &=: \left\langle\mathbf{b}\psi_i^{\kappa+1}\right\rangle,\end{aligned}$$

where $(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^+ = \max(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij},0)$ and $(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^- = \min(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij},0)$. We have to show that $\psi_i^{\kappa+1}$ is positive for all $\boldsymbol{\Omega}\in V$ under the time step restriction (2.34). Neglecting non-negative terms (remember that $\sigma_s$, $\sigma_t$ and $Q$ are assumed to be non-negative), we arrive at

$$\psi_i^{\kappa+1} \geq \left(1 - \Delta t\left(\sigma_t + \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^+\right)\right)\hat{\psi}_{\overline{\mathbf{u}}_i^{\kappa}}. \tag{2.35}$$

For our uniform equidistant grid, we have $\frac{|S_{ij}|}{|T_i|} = \frac{1}{\Delta x}$ for all $(i,j)$ and

$$\sum_{j\in\mathcal{N}(i)} (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})^+ = \|\boldsymbol{\Omega}\|_1 \le \sqrt{d}\|\boldsymbol{\Omega}\|_2 \le \sqrt{d},$$

where we used that $\|\boldsymbol{\Omega}\|_2 \le 1$ for all $\boldsymbol{\Omega} \in V$ holds true both for $V = [-1,1]$ and for $V = \mathcal{S}^2$. Consequently, equation (2.35) becomes

$$\psi_i^{\kappa+1} \ge \left(1 - \Delta t\left(\sigma_t + \frac{\sqrt{d}}{\Delta x}\right)\right)\hat{\psi}_{\overline{\mathbf{u}}_i^\kappa},$$

which is positive if (2.34) holds. □

The time step restriction due to the cross-section $\sigma_t$ can be avoided, *e.g.*, by using implicit-explicit methods where the source term is treated implicitly [55–57,63]. As in [63], we will use a second-order Strang splitting scheme for the split system

$$\partial_t\overline{\mathbf{u}}_i = -\frac{1}{|T_i|}\sum_{j\in\mathcal{N}(i)} \mathbf{g}_{ij}^{kin}(\overline{\mathbf{u}}_i,\overline{\mathbf{u}}_j), \tag{2.36a}$$

$$\partial_t\overline{\mathbf{u}}_i = \mathbf{s}(\mathbf{x},\overline{\mathbf{u}}_i), \tag{2.36b}$$

*i.e.*, in each time step from $t$ to $t+\Delta t$ we first solve the (linear) source system (2.36b) analytically (see Sect. 4) up to the time $t+\frac{\Delta t}{2}$, then we use the result as input to solve the hyperbolic part (2.36a) with a full timestep $\Delta t$, then we again advance (2.36b) analytically for a half time step $\frac{\Delta t}{2}$. As the source system is solved analytically (and thus preserves realizability), we only have to ensure that realizability is preserved when solving the hyperbolic part. We can thus avoid the time step restriction due to the cross-section $\sigma_t$.

**Corollary 2.10.** *The splitting scheme based on* (2.36) *is realizability-preserving under the CFL-like condition*

$$\Delta t < \frac{\Delta x}{\sqrt{d}}. \tag{2.37}$$

Note that we assumed in the proof of Theorem 2.9 that the optimization problems are solved exactly (by using the exact ansatz functions $\hat{\psi}_{\overline{\mathbf{u}}_i^\kappa}$). In practice, we have to solve these problems numerically which inevitably introduces some numerical errors. Fortunately, we can account for these inexact solutions by using a slightly tighter time step restriction as long as we can control the relative error in the ansatz function (compare [2,3,63]).

**Corollary 2.11.** *Let $\varepsilon_\gamma \in (0,1)$. If the numerical solution $\widetilde{\boldsymbol{\alpha}}(\mathbf{u})$ of the minimum-entropy problem* (2.18) *fulfils*

$$\gamma(\boldsymbol{\Omega}) := \frac{\hat{\psi}_{\mathbf{u}}(\boldsymbol{\Omega})}{\eta_*'(\mathbf{b}(\boldsymbol{\Omega})\cdot\widetilde{\boldsymbol{\alpha}}(\mathbf{u}))} \ge 1-\varepsilon_\gamma \tag{2.38}$$

*for all* $\mathbf{u}$ *for which the problem has to be solved during the finite volume scheme, then Corollory 2.10 still holds if the time step restrictions are tightened to*

$$\Delta t < \frac{1-\varepsilon_\gamma}{\sqrt{d}}\Delta x. \tag{2.39}$$

We can ensure that (2.38) holds by appropriately choosing the stopping criterion for the Newton scheme that is used to solve the minimum entropy optimization problems (see Sect. 4). We will always use $\varepsilon_\gamma = 0.1$ which means that we scale the time step restrictions (2.37) by a safety factor of 0.9.

If we advance the hyperbolic system (2.36a) in time by strong-stability preserving Runge–Kutta schemes [25] the CFL condition (2.37) still holds as these schemes consist of convex combinations of forward Euler steps. In particular, we will use Heun's method in all tests which is of second order.

## 3. New scheme in transformed variables

We will now present the new scheme in transformed variables which uses the identification between the realizable set $\mathcal{R}_{\mathbf{b}}^{+}$ and $\mathbb{R}^n$ given by the diffeomorphism (2.27).

### 3.1. Semidiscrete formulation

To derive the transformed scheme, note that

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\alpha}}\left(\left\langle \mathbf{b}\hat{\psi}_{\mathbf{u}}\right\rangle\right) \overset{(2.19)}{=} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\alpha}}(\langle \mathbf{b}\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle) = \left\langle \mathbf{b}\mathbf{b}^T\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\right\rangle = \mathbf{H}(\boldsymbol{\alpha}) \tag{3.1}$$

is the (positive definite) Hessian of the objective function in the dual problem (2.18) (compare Sect. 4.1.2). Further, for the flux

$$\mathbf{f}_k(\boldsymbol{\alpha}) = \langle \Omega_k \mathbf{b}\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle \tag{3.2}$$

(compare (2.13)) we have

$$\frac{\mathrm{d}\mathbf{f}_k}{\mathrm{d}\boldsymbol{\alpha}} = \left\langle \Omega_k \mathbf{b}\mathbf{b}^T\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\right\rangle =: \mathbf{J}_k(\boldsymbol{\alpha}). \tag{3.3}$$

In transformed variables, assuming $\mathbf{u}$ is sufficiently smooth, the hyperbolic system of equations (2.12) thus becomes [38]

$$\begin{aligned} \mathbf{s}(\mathbf{x},\mathbf{u}(\boldsymbol{\alpha})) &= \partial_t\mathbf{u}(\boldsymbol{\alpha}) + \sum_{k=0}^{d-1}\partial_{x_k}\mathbf{f}_k(\mathbf{u}(\boldsymbol{\alpha})) = \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\boldsymbol{\alpha}}(\boldsymbol{\alpha})\partial_t\boldsymbol{\alpha} + \sum_{k=0}^{d-1}\frac{\mathrm{d}\mathbf{f}_k}{\mathrm{d}\boldsymbol{\alpha}}(\boldsymbol{\alpha})\partial_{x_k}\boldsymbol{\alpha} \\ &= \mathbf{H}(\boldsymbol{\alpha})\partial_t\boldsymbol{\alpha} + \sum_{k=0}^{d-1}\mathbf{J}_k(\boldsymbol{\alpha})\partial_{x_k}\boldsymbol{\alpha}. \end{aligned} \tag{3.4}$$

A numerical scheme based on the form (3.4) could potentially be much faster than the standard finite volume scheme (2.32) as it avoids solving the non-linear optimization problem and only needs inversion of the positive definite symmetric matrix $\mathbf{H}(\boldsymbol{\alpha})$. However, equation (3.4) is not in conservation form which makes it hard to guarantee that numerical schemes converge to the correct weak solution of (2.12).

On the other hand, if we perform the space discretization first and then transform the semi-discrete equation (2.29) to the new variables, we arrive at

$$\mathbf{H}(\overline{\boldsymbol{\alpha}}_i)\partial_t\overline{\boldsymbol{\alpha}}_i + \frac{1}{|T_i|}\sum_{j\in\mathcal{N}(i)}\mathbf{g}_{ij}(\mathbf{u}(\overline{\boldsymbol{\alpha}}_i),\mathbf{u}(\overline{\boldsymbol{\alpha}}_j)) = \mathbf{s}(\mathbf{x}_i,\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)), \tag{3.5}$$

where $\overline{\boldsymbol{\alpha}}_i = \boldsymbol{\alpha}(\overline{\mathbf{u}}_i)$ are the multipliers corresponding to the finite volume averaged moment $\overline{\mathbf{u}}_i$ *via* the inverse diffeomorphism (2.18).

Since the space discretization is inherited from the conservative form (2.29), we now only have to discretize in time and can expect that the moments of the solution converge to the corresponding solution of the non-transformed scheme. In the following, we will show that this is indeed the case (see Sect. 3.3).

**Remark 3.1.** A similar idea has been used in [52] to efficiently solve an semi-implicit version of the standard finite volume scheme (2.32) with Lax–Friedrichs flux. In fact, the scheme presented in the following is equivalent to using the approach in [52] with explicit time discretization and only performing a single step of the Newton iteration. The authors in [52] restrict their investigation to slab geometry and note that further research is needed to examine the efficiency of their scheme in higher dimensions where the Jacobians of the coupled system are not block-tridiagonal anymore. In contrast, using the fully explicit approach considered here the grid cells decouple and the equations can be solved independently for each grid cell, also in several dimensions.

### 3.2. Entropy stability on the semi-discrete level

On the semidiscrete level, the new scheme is just a variable transformation of the standard scheme. We can thus show that the solutions of (3.5) also inherit a semidiscrete version of the entropy-dissipation property (2.23).

**Theorem 3.2.** *Let* $\{\overline{\boldsymbol\alpha}_i(t)\,|\,i\in I_{\mathcal{G}}\}$ *be a solution of the transformed semidiscrete equation* (3.5) *on a (hyper)rectangular grid. Then we have*

$$\begin{aligned}\partial_t\mathcal{H}\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big) &+ \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\Big\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big)\Big\rangle_+ + \Big\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_j)}\Big)\Big\rangle_-\Big)\\ &\le -\sigma_a\langle(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\rangle + \langle(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})Q\rangle.\end{aligned}\tag{3.6}$$

*Proof.* The negativity of the entropy contribution by the scattering term $\sigma_s\big(\mathbf{G}_{\mathbf{b}}^{\text{iso}}-\mathbf{I}\big)\mathbf{u}$ can be shown exactly as in the continuous case, see, *e.g.*, [58]. The remaining part $-\sigma_a\mathbf{u}+\langle\mathbf{b}Q\rangle$ of the source term directly gives the right-hand side of (3.6) after multiplication by $\overline{\boldsymbol\alpha}_i$ (as done below). In the following, for notational simplicity, we will thus only regard the flux term. To that end, first note that

$$\begin{aligned}\partial_t\mathcal{H}\Big(\hat\psi_{\mathbf{u}(\boldsymbol\alpha)}\Big) &= \partial_t\Big\langle\eta\Big(\hat\psi_{\mathbf{u}(\boldsymbol\alpha)}\Big)\Big\rangle = \partial_t\langle\eta\circ\eta_*'(\boldsymbol\alpha\cdot\mathbf{b})\rangle\\ &= \langle\eta'\circ\eta_*'(\boldsymbol\alpha\cdot\mathbf{b})\eta_*''(\boldsymbol\alpha\cdot\mathbf{b})\mathbf{b}\rangle\cdot\partial_t\boldsymbol\alpha = \langle(\boldsymbol\alpha\cdot\mathbf{b})\eta_*''(\boldsymbol\alpha\cdot\mathbf{b})\mathbf{b}\rangle\cdot\partial_t\boldsymbol\alpha,\end{aligned}\tag{3.7}$$

where we used that $\eta'\circ\eta_*'$ is the identity by definition of the Legendre transform (see, *e.g.*, [50]). Further, the Legendre transform fulfills the relation

$$p\eta_*'(p)-\eta_*(p)=\eta(\eta_*'(p)).\tag{3.8}$$

Multiplying the semi-discrete equation (3.5) by $\overline{\boldsymbol\alpha}_i$, we thus obtain (using the kinetic flux (2.30) and neglecting the source term)

$$\begin{aligned}0 &= \overline{\boldsymbol\alpha}_i\cdot\mathbf{H}(\overline{\boldsymbol\alpha}_i)\partial_t\overline{\boldsymbol\alpha}_i + \overline{\boldsymbol\alpha}_i\cdot\frac{1}{|T_i|}\sum_{j\in\mathcal{N}(i)}\mathbf{g}_{ij}(\mathbf{u}(\overline{\boldsymbol\alpha}_i),\mathbf{u}(\overline{\boldsymbol\alpha}_j))\\
&= \overline{\boldsymbol\alpha}_i\cdot\big\langle\mathbf{b}\mathbf{b}^T\eta_*''(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\big\rangle\partial_t\overline{\boldsymbol\alpha}_i + \overline{\boldsymbol\alpha}_i\cdot\left(\sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta_*'(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\mathbf{b}\rangle_+ + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\mathbf{b}\rangle_-\Big)\right)\\
&= \langle(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\eta_*''(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\mathbf{b}\rangle\cdot\partial_t\overline{\boldsymbol\alpha}_i\\
&\quad+\left(\sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\rangle_+ + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_-\Big)\right)\\
&\overset{(3.7)}{=} \partial_t\mathcal{H}\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big) + \left(\sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\rangle_+ + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_-\right.\\
&\quad\left.+\ \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})((\overline{\boldsymbol\alpha}_i-\overline{\boldsymbol\alpha}_j)\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_-\Big)\right)\\
&\overset{(3.8)}{=} \partial_t\mathcal{H}\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big) + \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\circ\eta_*'(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\rangle_+ + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta_*(\overline{\boldsymbol\alpha}_i\cdot\mathbf{b})\rangle_+\\
&\quad+\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\circ\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_- + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta_*(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_- + \langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})((\overline{\boldsymbol\alpha}_i-\overline{\boldsymbol\alpha}_j)\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol\alpha}_j\cdot\mathbf{b})\rangle_-\Big)\\
&= \partial_t\mathcal{H}\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big) + \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\Big\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_i)}\Big)\Big\rangle_+ + \Big\langle(\boldsymbol\Omega\cdot\mathbf{n}_{ij})\eta\Big(\hat\psi_{\mathbf{u}(\overline{\boldsymbol\alpha}_j)}\Big)\Big\rangle_-\end{aligned}$$

$$
\begin{aligned}
&+ \langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta_*(\overline{\boldsymbol{\alpha}}_i\cdot\mathbf{b})\rangle_+ + \langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})(\eta_*(\overline{\boldsymbol{\alpha}}_j\cdot\mathbf{b}) + ((\overline{\boldsymbol{\alpha}}_i-\overline{\boldsymbol{\alpha}}_j)\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol{\alpha}}_j\cdot\mathbf{b}))\rangle_-\Big)\\
&\geq \partial_t\mathcal{H}\Big(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\Big) + \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\Big\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta\Big(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\Big)\Big\rangle_+ + \Big\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta\Big(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_j)}\Big)\Big\rangle_- + \langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta_*(\overline{\boldsymbol{\alpha}}_i\cdot\mathbf{b})\rangle\Big),
\end{aligned}
$$

where we used for the estimate that the integral $\langle\cdot\rangle_-$ is defined such that $\boldsymbol{\Omega}\cdot\mathbf{n}_{ij}$ is negative and that $\boldsymbol{\alpha}\mapsto\eta_*(\boldsymbol{\alpha}\cdot\mathbf{b})$ is convex (and thus $\eta_*(\overline{\boldsymbol{\alpha}}_j\cdot\mathbf{b}) + ((\overline{\boldsymbol{\alpha}}_i-\overline{\boldsymbol{\alpha}}_j)\cdot\mathbf{b})\eta_*'(\overline{\boldsymbol{\alpha}}_j\cdot\mathbf{b}) \leq \eta_*(\overline{\boldsymbol{\alpha}}_i\cdot\mathbf{b})$). For a hyperrectangular grid, for each interface of grid cell $T_i$ with outer normal $\mathbf{n}_{ij}$ the opposite interface has the same area and outer normal $-\mathbf{n}_{ij}$. Thus, the $\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta_*(\overline{\boldsymbol{\alpha}}_i\cdot\mathbf{b})\rangle$ terms cancel out which finally gives (3.6). □

**Remark 3.3.** The entropy density $\eta$ is strictly convex and thus attains its minimum at $\psi$ with $\eta'(\psi)=0$. Consequently, for $\psi=\hat{\psi}_{\mathbf{u}(\boldsymbol{\alpha})}$, the entropy density is minimal if $0=\eta'\Big(\hat{\psi}_{\mathbf{u}(\boldsymbol{\alpha})}\Big)=\eta'\circ\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})=\boldsymbol{\alpha}\cdot\mathbf{b}$. This explains why absorption decreases the entropy for $\boldsymbol{\alpha}\cdot\mathbf{b}>0$ and increases the entropy for $\boldsymbol{\alpha}\cdot\mathbf{b}<0$ (see the right-hand side of (3.6)). Similarly, a source of particles $Q(\boldsymbol{\Omega})$ with velocity $\boldsymbol{\Omega}$ increases the entropy if $\boldsymbol{\alpha}\cdot\mathbf{b}(\boldsymbol{\Omega})>0$ and decreases it otherwise.

**Remark 3.4.** Since the two schemes are related by the transformation (2.27) on the semidiscrete level, the entropy dissipation law (3.6) also holds for the standard unsplit finite volume scheme if we replace $\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)$ by $\overline{\mathbf{u}}_i$ and $\overline{\boldsymbol{\alpha}}_i$ by $\boldsymbol{\alpha}(\overline{\mathbf{u}}_i)$.

### 3.3. Time discretization

To get a fully discrete numerical scheme, we still have to choose a time discretization for (3.5). To avoid having to solve a large coupled non-linear system of equations in each time step, we will only consider explicit schemes. Using, for example, the explicit Euler scheme and the kinetic flux (2.30), the fully discrete form of (3.5) becomes

$$
\begin{aligned}
\overline{\boldsymbol{\alpha}}_i^{\kappa+1} &= \overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})^{-1}\Bigg(\mathbf{s}(\mathbf{x}_i,\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa}))\\
&\qquad - \sum_{j\in\mathcal{N}(i)}\frac{|S_{ij}|}{|T_i|}\Big(\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta_*'(\overline{\boldsymbol{\alpha}}_i^{\kappa}\cdot\mathbf{b})\mathbf{b}\rangle_+ + \big\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ij})\eta_*'(\overline{\boldsymbol{\alpha}}_j^{\kappa}\cdot\mathbf{b})\mathbf{b}\big\rangle_-\Big)\Bigg)\\
&= \overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})^{-1}\mathbf{u}_i^{\uparrow}\big(\mathbf{u}(\overline{\boldsymbol{\alpha}}_0^{\kappa}),\mathbf{u}(\overline{\boldsymbol{\alpha}}_1^{\kappa}),\dots,\mathbf{u}\big(\overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}^{\kappa}\big)\big) =: \overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})^{-1}\mathbf{u}_i^{\uparrow,\kappa},
\end{aligned}
\tag{3.9}
$$

where the update term $\mathbf{u}_i^{\uparrow}$ has been defined in (2.33). However, a time discretization using fixed time steps might not be a suitable choice for the transformed scheme.

**Example 3.5.** Consider exemplarily the plane-source test (compare Sect. 5) using an equidistant grid with $n_x=240$ elements for the domain $X=[-1.2,1.2]$. We thus have $\Delta x=\frac{2.4}{240}=10^{-2}$. The initial value in this test is a small isotropic vacuum density $\psi_{\mathrm{vac}}=5\cdot10^{-7}$ plus a Dirac delta at $x=0$ which is split into the grid cells $T_{\mathrm{left}}=T_{119}$ and $T_{\mathrm{right}}=T_{120}$ adjacent to 0. The initial values for the finite volume scheme thus are given as

$$
\overline{\mathbf{u}}_{119}^0=\overline{\mathbf{u}}_{120}^0=\frac{1}{\Delta x}\int_0^{\Delta x}\langle\mathbf{b}(\mu)\psi_{t=0}(x,\mu)\rangle\mathrm{d}x=\frac{1}{\Delta x}\int_0^{\Delta x}\langle\mathbf{b}(\mu)(\psi_{\mathrm{vac}}+\delta_0(x))\rangle\mathrm{d}x=\left(\psi_{\mathrm{vac}}+\frac{1}{2\Delta x}\right)\langle\mathbf{b}\rangle\approx 50\langle\mathbf{b}\rangle
$$

and $\overline{\mathbf{u}}_i^0=\psi_{\mathrm{vac}}\langle\mathbf{b}\rangle=5\cdot10^{-7}\langle\mathbf{b}\rangle$ for $i\notin\{119,120\}$. The initial values are thus isotropic with local particle densities $\rho_i:=\rho(\overline{\mathbf{u}}_i^0)$ given as $\rho_{119}=\rho_{120}=\langle 50\rangle=100$ and $\rho_j=\big\langle 5\cdot10^{-7}\big\rangle=10^{-6}$ for $j\notin\{119,120\}$. The multipliers corresponding to the isotropic moment with local particle density $\rho$ are

$$
\boldsymbol{\alpha}_{\mathbf{b}}^{\mathrm{iso}}(\rho)=\eta'\left(\frac{\rho}{\langle 1\rangle}\right)\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}}
\tag{3.10}
$$

with corresponding ansatz function

$$\hat{\psi}^{\text{iso}}_{\mathbf{b}}(\rho) = \eta'_*\big(\boldsymbol{\alpha}^{\text{iso}}_{\mathbf{b}}(\rho)\cdot\mathbf{b}\big) = \eta'_*\left(\eta'\left(\frac{\rho}{\langle 1\rangle}\right)\boldsymbol{\alpha}^{\mathbb{1}}_{\mathbf{b}}\cdot\mathbf{b}\right) = \eta'_*\left(\eta'\left(\frac{\rho}{\langle 1\rangle}\right)\right) = \frac{\rho}{\langle 1\rangle},$$

where we used again that $\eta'$ and $\eta'_*$ are inverse functions. If we now focus on cell $T_{121}$ and ignore the source term $\mathbf{s}(\mathbf{x}_{121}, \mathbf{u}(\overline{\boldsymbol{\alpha}}_{121}))$, the update (3.9) takes the form

$$\begin{aligned}
\overline{\boldsymbol{\alpha}}^1_{121} &= \overline{\boldsymbol{\alpha}}^0_{121} - \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_{121})^{-1}\left(\sum_{j\in\{120,122\}}\frac{1}{\Delta x}\Big(\langle|\mu|\mathbf{b}\eta'_*(\overline{\boldsymbol{\alpha}}_{121}\cdot\mathbf{b})\rangle_+ + \langle-|\mu|\mathbf{b}\eta'_*(\overline{\boldsymbol{\alpha}}_j\cdot\mathbf{b})\rangle_-\Big)\right)\\
&= \overline{\boldsymbol{\alpha}}^0_{121} - \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_{121})^{-1}\left(\sum_{j\in\{120,122\}}\frac{1}{\Delta x}\left(\left\langle|\mu|\mathbf{b}\frac{\rho_{121}}{\langle 1\rangle}\right\rangle_+ + \left\langle-|\mu|\mathbf{b}\frac{\rho_j}{\langle 1\rangle}\right\rangle_-\right)\right)\\
&\approx \overline{\boldsymbol{\alpha}}^0_{121} + \Delta t\mathbf{H}(\overline{\boldsymbol{\alpha}}_{121})^{-1}\left(\frac{1}{\Delta x}\left\langle|\mu|\mathbf{b}\frac{\rho_{120}}{\langle 1\rangle}\right\rangle_-\right),
\end{aligned}$$

where the approximation in the last step is based on the observation that $\rho_{121} = \rho_{122} = 10^{-6}$ are much smaller than $\rho_{120} = 100$ (see above). Inserting the definitions of the Hessian matrix (4.3) and the values of $\Delta x, \rho_{120}$ and $\rho_{121}$ we finally obtain

$$\overline{\boldsymbol{\alpha}}^1_{121} \approx \overline{\boldsymbol{\alpha}}^0_{121} + \Delta t\big\langle\mathbf{b}\mathbf{b}^T\,5\cdot10^{-7}\big\rangle^{-1}\big(100\langle|\mu|\mathbf{b}\,50\rangle_-\big) = \overline{\boldsymbol{\alpha}}^0_{121} + 10^{10}\,\Delta t\mathbf{M}^{-1}\langle|\mu|\mathbf{b}\rangle_-.$$

For the full-moment $\mathrm{M}_N$ models using Legendre polynomials as a basis, the mass matrix $\mathbf{M}$ is the unit matrix and the basis functions $\mathbf{b}$ are approximately of unit order. We thus see that the update term is in the order of $10^{10}$ which is why a very small time step $\Delta t$ has to be chosen initially to limit the time stepping error.

The example shows that the transformed scheme is expected to require very small time steps in some instances, *e.g.*, whenever there are large differences in the particle density between adjacent cells (which is initially the case for all our numerical tests, see Sect. 5). On the other hand, the time step does not have to be restricted to ensure realizability which may allow for time steps that are even larger than those used in the standard scheme in some situations. A time stepping scheme using a fixed time step $\Delta t$ would thus be very inefficient for the new scheme.

Instead, we will use the Runge–Kutta method by Bogacki and Shampine [8] which adaptively chooses the time step according to an embedded error estimate (see Sect. 4.2.1 for details). This way, we can use large time steps where possible without introducing uncontrollable errors in time regions where a small time step is required. If we define

$$\boldsymbol{\alpha}^{\upharpoonright}_i\big(\overline{\boldsymbol{\alpha}}^\kappa_0, \overline{\boldsymbol{\alpha}}^\kappa_1, \ldots, \overline{\boldsymbol{\alpha}}^\kappa_{n_{\mathbf{x}}-1}\big) := \mathbf{H}(\overline{\boldsymbol{\alpha}}^\kappa_i)^{-1}\mathbf{u}^{\upharpoonright}_i\big(\mathbf{u}(\overline{\boldsymbol{\alpha}}^\kappa_0), \ldots, \mathbf{u}(\overline{\boldsymbol{\alpha}}^\kappa_{n_{\mathbf{x}}-1})\big), \tag{3.11}$$

the Runge–Kutta update takes the form

$$\overline{\boldsymbol{\alpha}}^{\kappa+1}_i = \overline{\boldsymbol{\alpha}}^\kappa_i + \Delta t\sum_{p=0}^{s-1}\check{b}_p\boldsymbol{\alpha}^{\upharpoonright}_i\big(\boldsymbol{\beta}^p_0, \ldots, \boldsymbol{\beta}^p_{n_{\mathbf{x}}-1}\big) \tag{3.12a}$$

with stages $\boldsymbol{\beta}^p_i$ given by

$$\boldsymbol{\beta}^p_i = \overline{\boldsymbol{\alpha}}^\kappa_i + \Delta t\sum_{q=0}^{p-1}\check{a}_{pq}\boldsymbol{\alpha}^{\upharpoonright}_i\big(\boldsymbol{\beta}^q_0, \ldots, \boldsymbol{\beta}^q_{n_{\mathbf{x}}-1}\big). \tag{3.12b}$$

Here, $s$ is the number of stages in the Runge–Kutta scheme $\check{a}_{pq}, \check{b}_p$ are the Runge–Kutta coefficients and we neglected the varying time step for notational simplicity.

## 3.4. Convergence properties

The new scheme calculates approximate solutions in transformed ($\boldsymbol{\alpha}$) variables. However, we are usually interested in the solution in original ($\mathbf{u}$) variables. We can easily obtain such a solution by applying the diffeomorphism (2.27) to the transformed solution. However, a priori, it might be possible that this transformation amplifies the discretization errors and destroys the accuracy of the scheme. Luckily, we can show that the order of convergence of the time stepping scheme is preserved by the transformation.

**Theorem 3.6.** *Let* $\{\overline{\boldsymbol{\alpha}}_i(t) \,|\, i \in I_{\mathcal{G}}\}$ *be a solution of the transformed semidiscrete equation* (3.5) *and let* $\{\overline{\mathbf{u}}_i(t) := \mathbf{u}(\overline{\boldsymbol{\alpha}}_i(t)) \,|\, i \in I_{\mathcal{G}}\}$ *be the corresponding solution of the untransformed equation* (2.29)*. Let further* $\{\overline{\boldsymbol{\alpha}}_i^{\kappa}\}$ *be approximations of* $\overline{\boldsymbol{\alpha}}_i$ *at discrete time points* $t_\kappa$ *obtained by a time stepping scheme of order* $r$*, i.e.*

$$\overline{\boldsymbol{\alpha}}_i^{\kappa} = \overline{\boldsymbol{\alpha}}_i(t_\kappa) + \mathbf{r}_i^{\kappa}$$

*with* $\mathbf{r}_i^{\kappa} = \mathcal{O}((\Delta t)^r)$*. If the spectral norm* $\|\mathbf{H}\|$ *of the Hessian is bounded, then the corresponding moments converge with the same order, i.e.,*

$$\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) = \overline{\mathbf{u}}_i(t_\kappa) + \mathcal{O}((\Delta t)^r).$$

*Proof.* Using a zeroth order Taylor approximation with Lagrange form of the remainder, we have

$$\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) = \mathbf{u}(\overline{\boldsymbol{\alpha}}_i(t_\kappa) + \mathbf{r}_i^{\kappa}) = \mathbf{u}(\overline{\boldsymbol{\alpha}}_i(t_\kappa)) + \mathbf{H}\Big(\overline{\boldsymbol{\alpha}}_i^{\xi}\Big)\mathbf{r}_i^{\kappa} = \overline{\mathbf{u}}_i(t_\kappa) + \mathbf{H}\Big(\overline{\boldsymbol{\alpha}}_i^{\xi}\Big)\mathbf{r}_i^{\kappa}$$

with $\overline{\boldsymbol{\alpha}}_i^{\xi} = (1-\xi)\overline{\boldsymbol{\alpha}}_i(t_\kappa) + \xi\mathbf{r}_i^{\kappa}$ for some $\xi \in [0,1]$. Due to the boundedness of $\mathbf{H}$, we further have

$$\Big\|\mathbf{H}\Big(\overline{\boldsymbol{\alpha}}_i^{\xi}\Big)\mathbf{r}_i^{\kappa}\Big\| \le \Big\|\mathbf{H}\Big(\overline{\boldsymbol{\alpha}}_i^{\xi}\Big)\Big\|\|\mathbf{r}_i^{\kappa}\| = \mathcal{O}(\Delta t^r).$$

□

For arbitrary $\boldsymbol{\alpha}$, we cannot expect the upper bound on $\|\mathbf{H}\|$ required by Theorem 3.6 to hold. However, note that

$$\mathbf{w}^T\mathbf{H}(\boldsymbol{\alpha})\mathbf{w} = \Big\langle(\mathbf{b}\cdot\mathbf{w})^2\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\Big\rangle \le \Big(\max_{\boldsymbol{\Omega}\in V}\|\mathbf{b}(\boldsymbol{\Omega})\|^2\Big)\|\mathbf{w}\|^2\langle\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle. \tag{3.13}$$

For Maxwell–Boltzmann entropy, we have $\eta_*'' = \eta_*'$ and thus

$$\langle\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle = \langle\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle = \rho(\mathbf{u}(\boldsymbol{\alpha}))$$

is the local particle density corresponding to the multipliers $\boldsymbol{\alpha}$. Due to the conservation properties, the local particle density remains bounded for the solutions of the moment equations. As a consequence, $\|\mathbf{H}(\boldsymbol{\alpha})\|$ should be bounded for $\boldsymbol{\alpha}$ close enough to a solution of (3.5).

Once we have a $r$-th order time discretization for (3.5), we thus can expect the corresponding moments to converge with the same order in practice, at least for time steps $\Delta t$ small enough. However, the order of convergence of Runge–Kutta schemes is usually shown under the assumption of Lipschitz continuity of the ordinary differential equation's right-hand side (see, *e.g.*, [28], Thm. II.3.4). In our case, we have to show Lipschitz continuity of the functions $\boldsymbol{\alpha}_i^{\uparrow}$. Under some additional assumptions, we indeed obtain Lipschitz continuity for multipliers $\boldsymbol{\alpha}$ in a domain $A$.

**Theorem 3.7.** *The function* $\boldsymbol{\alpha}_i^{\uparrow}$ *is Lipschitz-continuous on* $A^{n_\times}$ *if there exist constants* $c_1, c_2, c_3, c_4, c_5 \in \mathbb{R}^{>0}$ *such that for all* $\boldsymbol{\alpha} \in A \subset \mathbb{R}^n$ *we have*

$$c_1 \le \|\mathbf{H}(\boldsymbol{\alpha})\| \le c_2, \tag{3.14a}$$

$$\rho(\mathbf{u}(\boldsymbol{\alpha})) = \langle\eta_*'(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle \le c_3, \tag{3.14b}$$

$$\langle\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle \le c_4, \tag{3.14c}$$

$$\langle\eta_*'''(\boldsymbol{\alpha}\cdot\mathbf{b})\rangle \le c_5. \tag{3.14d}$$

The proof is technical and can be found in Section A.

For Maxwell–Boltzmann entropy, since $\eta_*' = \eta_*'' = \eta_*''' = \exp$ and by definition of the Hessian (4.3), we see that (3.14b) implies (3.14c), (3.14d) and the upper bound in (3.14a). Moreover, all the assumptions (3.14a)–(3.14d) are fulfilled if $\|\boldsymbol{\alpha}\|$ remains bounded.

**Corollary 3.8.** *For Maxwell–Boltzmann entropy, $\boldsymbol{\alpha}_i^{\uparrow}$ is Lipschitz-continuous on $A^{n_{\mathbf{x}}}$ if $A$ is bounded.*

## 3.5. Regularization

As discussed above, the bound (3.14b) on the local particle density $\rho$ can be expected to hold in practice. Unfortunately, this is not true for the lower bound on $\|\mathbf{H}\|$. Considering again (3.13), we see that we cannot expect the lower bound to hold as long as $\eta_*''$ is not bounded from below. In particular, for Maxwell–Boltzmann entropy, equation (3.14) would require a lower bound on the ansatz density $\hat{\psi}_{\mathbf{u}(\boldsymbol{\alpha}_i)}$. Even if such a lower bound exists for the exact solution, it might be extremely small. Factoring in numerical errors, it is clear that in practice this bound will not always hold. In addition, $\mathbf{H}$ may be very badly conditioned, up to the point that it might be numerically singular.

For the standard scheme, where $\mathbf{H}$ shows up as the Hessian of the objective function in the minimum entropy optimization problem (2.18), we thus use several regularization techniques to ensure that the Newton scheme always converges (see Sect. 4.1.5). While most of these techniques are not easily carried over to the transformed scheme, we here investigate two approaches to regularize the new scheme.

### *3.5.1. Isotropic regularization of the Hessian*

As a first approach, we regularize the Hessian matrix by adding a small multiple of the mass matrix $\mathbf{M} = \left\langle \mathbf{b}\mathbf{b}^T \right\rangle$. Exemplarily, for the explicit Euler scheme, the update formula (3.9) then becomes

$$\overline{\boldsymbol{\alpha}}_i^{\kappa+1} = \overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t \left(\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) + \epsilon\mathbf{M}\right)^{-1} \mathbf{u}_i^{\uparrow,\kappa}. \tag{3.15}$$

This corresponds to adding a small isotropic particle density to the derivative of the ansatz function in the Hessian

$$\mathbf{H}(\boldsymbol{\alpha}) + \epsilon\mathbf{M} = \left\langle \mathbf{b}\mathbf{b}^T \eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b}) \right\rangle + \epsilon\left\langle \mathbf{b}\mathbf{b}^T \right\rangle = \left\langle \mathbf{b}\mathbf{b}^T \left(\eta_*''(\boldsymbol{\alpha}\cdot\mathbf{b}) + \epsilon\right) \right\rangle.$$

and ensures that the lower bound in (3.14a) holds for the regularized matrix.

To analyze the introduced error, note that without regularization the update in the Euler scheme is

$$\begin{aligned} \mathbf{u}\left(\overline{\boldsymbol{\alpha}}_i^{\kappa+1}\right) &= \mathbf{u}\left(\overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t \mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})^{-1}\mathbf{u}_i^{\uparrow,\kappa}\right) = \mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) + \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\boldsymbol{\alpha}}(\overline{\boldsymbol{\alpha}}_i^{\kappa})\Delta t \mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})^{-1}\mathbf{u}_i^{\uparrow,\kappa} + \mathbf{r} \\ &\overset{(3.1)}{=} \mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) + \Delta t \mathbf{u}_i^{\uparrow,\kappa} + \mathbf{r} = \overline{\mathbf{u}}_i^{\kappa+1} + \mathbf{r} \end{aligned} \tag{3.16}$$

with remainder of order $\mathcal{O}((\Delta t)^2)$. Using the regularization (3.15), the local truncation error becomes

$$\mathbf{u}\left(\overline{\boldsymbol{\alpha}}_i^{\kappa+1}\right) = \overline{\mathbf{u}}_i^{\kappa+1} - \Delta t \epsilon \mathbf{M}\left(\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) + \epsilon\mathbf{M}\right)^{-1}\mathbf{u}_i^{\uparrow,\kappa} + \mathbf{r}.$$

Remember that the terms in $\mathbf{u}_i^{\uparrow,\kappa}$, except for the constant source term $\langle \mathbf{b}Q \rangle$, all scale with either $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa})}$ or $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_j^{\kappa})}$, $j \in \mathcal{N}(i)$. Thus, if $\overline{\boldsymbol{\alpha}}_i^{\kappa}$ corresponds to an ansatz density $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa})}$ which is significantly greater than $\epsilon$, the additional error term

$$\Delta t \epsilon \mathbf{M}\left(\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa}) + \epsilon\mathbf{M}\right)^{-1}\mathbf{u}_i^{\uparrow,\kappa} \approx \Delta t \epsilon \mathbf{M}\left(\mathbf{H}(\overline{\boldsymbol{\alpha}}_i^{\kappa})\right)^{-1}\mathbf{u}_i^{\uparrow,\kappa} = \mathcal{O}(\Delta t \epsilon)$$

does not negatively impact the rate of convergence as long as we choose $\epsilon$ as $\mathcal{O}(\Delta t)$. Similarly, if $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa})}$, $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_j^{\kappa})}$ and $Q$ are small (more precisely, $\mathcal{O}(\epsilon)$), the additional error is of order $\mathcal{O}(\Delta t \epsilon \frac{1}{\epsilon}\epsilon) = \mathcal{O}(\Delta t \epsilon)$. However, if $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i^{\kappa})}$ is small, but at least one of $\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_j^{\kappa})}$ and $Q$ is not, we obtain an additional error of $\mathcal{O}(\Delta t \epsilon \frac{1}{\epsilon} 1) = \mathcal{O}(\Delta t)$. In this case, *i.e.*, if the particle density in grid cell $T_i$ is small and there is a strong source or an influx from an neighbor entity $T_i$, the regularization (3.15) might negatively affect the solution. For higher-order time stepping schemes, the error analysis gets much more involved, but we expect similar results, *i.e.*, significant regularization errors only in regions with large differences in particle densities between adjacent cells.

*3.5.2. Direct constraints for the entropy variables*

Instead of modifying the Hessian, we might try to directly enforce the boundedness of the entropy variables required in Corollary 3.8. For example, we could replace all multipliers $\boldsymbol{\alpha}$ with $\|\boldsymbol{\alpha}\| > \alpha_{\max}$ for some $\alpha_{\max} \in \mathbb{R}^{>0}$ by the closest multiplier $\boldsymbol{\alpha}_{\text{reg}}$ with $\|\boldsymbol{\alpha}_{\text{reg}}\| \leq \alpha_{\max}$.

We tested this approach in several numerical tests and found that, for general bases $\mathbf{b}$, the regularization either introduced large errors into the solution (for $\alpha_{\max}$ small) or did not improve the numerical stability (for large $\alpha_{\max}$).

A notable exception is the hat function basis $\mathbf{h}_n$. Since the hat functions are non-negative, remembering that the ansatz density has the form $\exp(\boldsymbol{\alpha} \cdot \mathbf{h}_n)$ (for Maxwell–Boltzmann entropy), we see that large positive entries of $\boldsymbol{\alpha}$ correspond to large ansatz densities and large (in absolute values) negative entries correspond to small ansatz densities. Thus, if we enforce a lower bound on the entries of $\boldsymbol{\alpha}$ by replacing all entries $\alpha_l < \alpha_{\min}$ by $\alpha_{\min}$ for some negative $\alpha_{\min} \in \mathbb{R}$, we would expect in general that the introduced error is small since we are only replacing entries corresponding to very small ansatz densities by entries corresponding to slightly larger but still very small densities.

Indeed, in our numerical tests, the error is very small (see Sect. 5.3.2). Note, however, that there might be cases where also this regularization technique leads to significant errors. Consider, *e.g.*, a highly anisotropic particle distributions in one dimension where the density is very low at one boundary of an interval in the velocity partition and very high at the other boundary. Here, replacing the entry corresponding to the very low density might significantly alter the ansatz density in that interval.

For the other bases ($\mathbf{p}_n$ and $\mathbf{f}_N$), a similar regularization technique which replaces only multipliers corresponding to small ansatz densities is not straightforward since for these bases there is no clear correspondence between the sign of $\boldsymbol{\alpha}$ entries and the magnitude of the ansatz density.

## 3.6. Entropy stability on the fully discrete level

In general, we cannot expect that an explicit time discretization preserves the entropy-stability of the semidiscrete scheme. However, we can enforce this property by using a relaxation of the standard Runge–Kutta scheme [48]. To that end, we collect the multipliers for each grid cell in a single vector, *i.e.*, we define

$$\hat{\boldsymbol{\alpha}} := \begin{pmatrix} \overline{\boldsymbol{\alpha}}_0(t) \\ \vdots \\ \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}(t) \end{pmatrix} \quad \text{and} \quad \hat{\boldsymbol{\alpha}}^{\uparrow}(\hat{\boldsymbol{\alpha}}) = \begin{pmatrix} \boldsymbol{\alpha}_0^{\uparrow}(\overline{\boldsymbol{\alpha}}_0, \ldots, \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}) \\ \vdots \\ \boldsymbol{\alpha}_{n_{\mathbf{x}}-1}^{\uparrow}(\overline{\boldsymbol{\alpha}}_0, \ldots, \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}) \end{pmatrix} = \begin{pmatrix} \mathbf{H}(\overline{\boldsymbol{\alpha}}_0)^{-1} \mathbf{u}_0^{\uparrow}(\overline{\boldsymbol{\alpha}}_0, \ldots, \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}) \\ \vdots \\ \mathbf{H}(\overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1})^{-1} \mathbf{u}_{n_{\mathbf{x}}-1}^{\uparrow}(\overline{\boldsymbol{\alpha}}_0, \ldots, \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}) \end{pmatrix}.$$

Further, we define the total entropy as

$$\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}) := \sum_{i \in I_{\mathcal{G}}} \mathcal{H}\Big(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\Big) = \sum_{i \in I_{\mathcal{G}}} \Big\langle \eta\Big(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\Big)\Big\rangle = \sum_{i \in I_{\mathcal{G}}} \langle \eta(\eta_*'(\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b}))\rangle.$$

Then

$$\hat{\mathcal{H}}'(\hat{\boldsymbol{\alpha}}) = \begin{pmatrix} \mathbf{H}(\overline{\boldsymbol{\alpha}}_0)\overline{\boldsymbol{\alpha}}_0 \\ \vdots \\ \mathbf{H}(\overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1})\overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}, \end{pmatrix}$$

(where we again used that $\eta' \circ \eta_*'$ is the identity) and thus

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}(t)) &= \hat{\mathcal{H}}'(\hat{\boldsymbol{\alpha}}(t)) \cdot \hat{\boldsymbol{\alpha}}^{\uparrow}(\hat{\boldsymbol{\alpha}}(t)) \\ &= \sum_{i \in I_{\mathcal{G}}} \overline{\boldsymbol{\alpha}}_i \cdot \mathbf{u}_i^{\uparrow}(\overline{\boldsymbol{\alpha}}_0, \ldots, \overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}) \end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(Proof of)(3.6)}}{\leq} \sum_{i\in I_{\mathcal{G}}} \Big( \langle (\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b}) Q\rangle - \sigma_a \langle (\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b}) \eta_*'(\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b})\rangle \\
&\qquad - \sum_{j\in\mathcal{N}(i)} \frac{|S_{ij}|}{|T_i|} \left( \left\langle (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij}) \eta\left(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\right)\right\rangle_+ + \left\langle (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij}) \eta\left(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_j)}\right)\right\rangle_- \right)\Bigg) \\
&= \sum_{i\in I_{\mathcal{G}}} \left( \langle (\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b}) Q\rangle - \sigma_a \langle (\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b}) \eta_*'(\overline{\boldsymbol{\alpha}}_i \cdot \mathbf{b})\rangle \right) \\
&\qquad - \sum_{S_{ij}\ \text{boundary interface}} \frac{|S_{ij}|}{|T_i|} \left( \left\langle (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij}) \eta\left(\hat{\psi}_{\mathbf{u}(\overline{\boldsymbol{\alpha}}_i)}\right)\right\rangle_+ + \langle (\boldsymbol{\Omega}\cdot\mathbf{n}_{ij}) \eta(\psi_b(\mathbf{x}_{ij}))\rangle_- \right),
\end{aligned}
\tag{3.17}
$$

*i.e.*, the change in total entropy is bounded by entropy fluxes over the domain boundary and entropy production *via* particle absorption or creation. Here, $\mathbf{x}_{ij}$ is the center of interface $S_{ij}$ and $\psi_b$ is the boundary value of the kinetic equation (2.1c). If we use an explicit Runge–Kutta scheme of the form (3.12) for time discretization of (3.5), we would expect the total entropy to approximately evolve as

$$
\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa+1}) = \hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa}) + \Delta t \sum_{p=0}^{s-1} \breve{b}_p \hat{\mathcal{H}}'\left(\hat{\boldsymbol{\beta}}^p\right) \cdot \hat{\boldsymbol{\alpha}}^{\uparrow}\left(\hat{\boldsymbol{\beta}}^p\right),
\tag{3.18}
$$

where $\hat{\boldsymbol{\beta}}^p = \left(\boldsymbol{\beta}_0^p, \ldots, \boldsymbol{\beta}_{n_{\mathbf{x}}-1}^p\right)^T$ contains the collected Runge–Kutta stages (compare the definition (3.12) of the Runge–Kutta scheme). Though (3.18) does not hold exactly, we can enforce this property by introducing a relaxation into the Runge–Kutta scheme (3.12) and replace (3.12a) by the relaxed version [48]

$$
\boldsymbol{\alpha}_{i,\gamma}^{\kappa+1} = \boldsymbol{\alpha}_i^{\kappa} + \gamma_\kappa \Delta t \sum_{p=0}^{s-1} \breve{b}_p \boldsymbol{\alpha}_i^{\uparrow}\left(\hat{\boldsymbol{\beta}}^p\right),
\tag{3.19}
$$

where $\gamma_\kappa$ is calculated by finding a root of

$$
\begin{aligned}
r(\gamma) &:= \hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}_\gamma^{\kappa+1}) - \hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa}) - \gamma \Delta t \sum_{p=0}^{s-1} \breve{b}_p \hat{\mathcal{H}}'\left(\hat{\boldsymbol{\beta}}^p\right) \cdot \hat{\boldsymbol{\alpha}}^{\uparrow}\left(\hat{\boldsymbol{\beta}}^p\right) \\
&= \sum_{i\in I_{\mathcal{G}}} \left( \langle \eta \circ \eta_*'(\boldsymbol{\alpha}_{i,\gamma}^{\kappa+1} \cdot \mathbf{b})\rangle - \langle \eta \circ \eta_*'(\boldsymbol{\alpha}_i^{\kappa} \cdot \mathbf{b})\rangle - \gamma \Delta t \sum_{p=0}^{s-1} \breve{b}_p \boldsymbol{\beta}_i^p \cdot \mathbf{u}_i^{\uparrow}\left(\hat{\boldsymbol{\beta}}_p\right)\right).
\end{aligned}
\tag{3.20}
$$

If $\hat{\mathcal{H}}$ is convex, $r(\gamma)$ is also convex and has exactly two roots, one at zero and one close to one [48] (for $\Delta t$ small enough). Unfortunately, in our case, $\hat{\mathcal{H}}$ is not necessarily convex since $\eta \circ \eta_*'$ is not necessarily convex. In the Maxwell–Boltzmann case, for example, we have $\eta \circ \eta_*'(p) = \exp(p)(p-1)$ which is convex only for $p > -1$ (and concave for $p < -1$). However, in our numerical experiments, we always found a root of $r(\gamma)$ close to one.

**Remark 3.9.** We could use the same approach to obtain entropy-stability for the standard finite volume scheme. However, in that case, each time we want to evaluate $r(\gamma)$ for a new $\gamma$ we would have to solve the minimum-entropy optimization problem on each grid cell (to evaluate the equivalent of $\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}_\gamma^{\kappa+1})$), which would make the root finding procedure prohibitively expensive.

Additional details on our implementation of the relaxed Runge–Kutta scheme can be found in Section 4.2.1.

## 4. Implementation details

We implemented both schemes in the generic C++ framework `DUNE` [6,7], more specifically in the `DUNE` generic discretization toolbox `dune-gdt` [54] and the `dune-xt`-modules [42,43]. The implementation is available in [37].

## 4.1. Standard finite volume scheme

The implementation for the standard finite volume scheme is taken from [63]. Here, we only shortly recall the relevant parts. Further, we will restrict ourselves to Maxwell–Boltzmann entropy (2.20) such that the ansatz becomes $\hat{\psi}_{\mathbf{u}} = \eta'_*(\boldsymbol{\alpha}(\mathbf{u}) \cdot \mathbf{b}) = \exp(\boldsymbol{\alpha}(\mathbf{u}) \cdot \mathbf{b})$.

### *4.1.1. Analytic solution of the source system*

As mentioned above, we use a second-order splitting approach (see (2.36)) to handle the source term independently of the flux term. Remember that we assume that the scattering is isotropic and that the parameters $\sigma_s$, $\sigma_a$, $Q$ are time-independent. The source system (2.36b) on grid cell $T_i$ thus takes the form

$$\partial_t \overline{\mathbf{u}}_i(t) = \big(\sigma_s(\mathbf{x}_i)\mathbf{G}_{\mathbf{b}}^{\mathrm{iso}} - \sigma_t(\mathbf{x}_i)\mathbf{I}\big)\overline{\mathbf{u}}_i(t) + \langle \mathbf{b} Q \rangle. \tag{4.1}$$

Since (4.1) is linear in $\overline{\mathbf{u}}_i$ with time-independent system matrix, we can solve it explicitly using matrix exponentials and the variation of constants formula. Under the additional assumption that the source is also isotropic, *i.e.*, that $Q$ does not depend on $\boldsymbol{\Omega}$, the solution to (4.1) is (see [63] for a detailed derivation)

$$\overline{\mathbf{u}}_i(t) = \mathrm{e}^{-\sigma_a t}\big(\mathrm{e}^{-\sigma_s t}\overline{\mathbf{u}}_i(0) + \big(1 - \mathrm{e}^{-\sigma_s t}\big)\mathbf{G}_{\mathbf{b}}^{\mathrm{iso}}\overline{\mathbf{u}}_i(0)\big) + \frac{1 - \mathrm{e}^{-\sigma_a t}}{\sigma_a}\langle \mathbf{b} \rangle Q.$$

Note that (4.1.1) can easily be calculated without any matrix operations due to the rank one structure of $\mathbf{G}_{\mathbf{b}}^{\mathrm{iso}}$ (see (2.10)).

### *4.1.2. Solving the optimization problem*

The second part of the standard splitting scheme, the flux system (2.36a), is advanced in time using Heun's method, which is a second-order strong-stability preserving Runge–Kutta scheme [25]. In each stage of the time stepping scheme, we have to solve the optimization problem (2.17) once in each cell. This usually accounts for the majority of computation time which makes it mandatory to pay special attention to the implementation of the optimization algorithm.

Recall that the objective function in the dual problem (2.18) is

$$p(\boldsymbol{\alpha}) = p_{\mathbf{u},\mathbf{b},\eta}(\boldsymbol{\alpha}) = \langle \eta_*(\mathbf{b} \cdot \boldsymbol{\alpha}) \rangle - \mathbf{u} \cdot \boldsymbol{\alpha}.$$

The gradient and the Hessian of $p$ are given by

$$\mathbf{q}(\boldsymbol{\alpha}) = \mathbf{q}_{\mathbf{u},\mathbf{b},\eta}(\boldsymbol{\alpha}) = \nabla_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}) = \langle \mathbf{b}\eta'_*(\mathbf{b} \cdot \boldsymbol{\alpha}) \rangle - \mathbf{u} \tag{4.2}$$

and

$$\mathbf{H}(\boldsymbol{\alpha}) = \mathbf{H}_{\mathbf{b},\eta}(\boldsymbol{\alpha}) = \mathbf{D}_{\boldsymbol{\alpha}}\mathbf{q}(\boldsymbol{\alpha}) = \big\langle \mathbf{b}\mathbf{b}^T \eta''_*(\mathbf{b} \cdot \boldsymbol{\alpha}) \big\rangle, \tag{4.3}$$

respectively. Note that $\eta''_* > 0$ since we assumed that $\eta$ (and thus also $\eta_*$) is strictly convex, and remember that the basis functions contained in $\mathbf{b}$ are linearly independent. As a consequence, the Hessian $\mathbf{H}$ is positive definite (and thus invertible).

To find a minimizer of $p$, we are searching for a root of the gradient $\mathbf{q}$ using Newton's method. The Newton direction $\mathbf{d}(\boldsymbol{\alpha})$ solves

$$\mathbf{H}(\boldsymbol{\alpha})\mathbf{d}(\boldsymbol{\alpha}) = -\mathbf{q}(\boldsymbol{\alpha}).$$

The update takes the form

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \zeta_k \mathbf{d}(\boldsymbol{\alpha}_k)$$

where $\zeta_k$ is determined by a backtracking line search such that

$$p(\boldsymbol{\alpha}_{k+1}) < p(\boldsymbol{\alpha}_k) + \xi\zeta_k \mathbf{q}(\boldsymbol{\alpha}_k) \cdot \mathbf{d}(\boldsymbol{\alpha}_k)$$

with $\xi \in (0,1)$. In our implementation, we always use $\xi = 10^{-3}$.

To avoid numerical problems for moments corresponding to a very small local particle density, before entering the Newton algorithm for the moment vector $\mathbf{u}$ we rescale it to

$$\boldsymbol{\phi} := \frac{\mathbf{u}}{\rho(\mathbf{u})}$$

such that $\rho(\boldsymbol{\phi}) = 1$. If the optimization algorithm for $\boldsymbol{\phi}$ stops at an iterate $\boldsymbol{\beta}$, we return

$$\widetilde{\boldsymbol{\alpha}} = \boldsymbol{\beta} + \boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \log\left(\frac{\rho(\mathbf{u})}{\rho(\mathbf{u}(\boldsymbol{\beta}))}\right) \tag{4.4}$$

(remember that $\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}}$ is the multiplier with the property that $\boldsymbol{\alpha}_{\mathbf{b}}^{\mathbb{1}} \cdot \mathbf{b} \equiv 1$, see Def. 2.2). This ensures that the local particle density is preserved exactly:

$$\rho(\mathbf{u}(\widetilde{\boldsymbol{\alpha}})) = \langle \exp(\mathbf{b} \cdot \widetilde{\boldsymbol{\alpha}}) \rangle = \langle \exp(\mathbf{b} \cdot \boldsymbol{\beta}) \rangle \frac{\rho(\mathbf{u})}{\rho(\mathbf{u}(\boldsymbol{\beta}))} = \rho(\mathbf{u}). \tag{4.5}$$

Given tolerances $\tau \in \mathbb{R}^{>0}$, $\varepsilon_\gamma \in (0,1)$, we will stop the Newton iteration at iterate $\boldsymbol{\beta}$ if

$$(1) \quad \left\|\mathbf{q}_{\boldsymbol{\phi}}(\boldsymbol{\beta})\right\|_2 < \tilde{\tau} := \begin{cases} \frac{\tau}{(1+\|\boldsymbol{\phi}\|_2)\rho(\mathbf{u})+\tau} & \text{if} \quad \mathbf{b} = \mathbf{f}_N \\ \frac{\tau}{(1+\sqrt{n}\|\boldsymbol{\phi}\|_2)\rho(\mathbf{u})+\sqrt{n}\tau} & \text{if} \quad \mathbf{b} \in \{\mathbf{h}_n, \mathbf{p}_n\}, \end{cases} \quad \text{and} \tag{4.6}$$

$$(2) \quad \mathbf{u} - (1-\varepsilon_\gamma)\mathbf{u}(\widetilde{\boldsymbol{\alpha}}) \in \mathcal{R}_{\mathbf{b}}^{+}, \tag{4.7}$$

where $\widetilde{\boldsymbol{\alpha}}$ is obtained from $\boldsymbol{\beta}$ by (4.4) and, as always, $n$ is the number of moments. As shown in [63], the first criterion guarantees that the gradient of the objective function is sufficiently small, *i.e.*, $\|\mathbf{q}_{\mathbf{u}}(\widetilde{\boldsymbol{\alpha}})\|_2 \leq \tau$. The second criterion (4.7) ensures that (2.38) holds and thus that the whole scheme is realizability-preserving although the optimization problems are only solved approximately. In our implementation, we choose the tolerances as $\tau = 10^{-9}$ and $\varepsilon_\gamma = 0.1$.

Checking the second stopping criterion (4.7) might be quite expensive (depending on the basis $\mathbf{b}$). We therefore check this criterion only if additionally

$$1 - \varepsilon_\gamma < \exp(-(\|\mathbf{d}(\boldsymbol{\beta})\|_1 + |\log \rho(\mathbf{u}(\boldsymbol{\beta}))|)) \tag{4.8}$$

holds. This criterion approximately ensures (2.38) (see [3, 58]) but, in general, is much easier to evaluate than (4.7). For the HFM$_n$ models, however, checking realizability is just checking positivity, so in that case we do not need to check (4.8) first.

### *4.1.3. Caching*

We use two types of caching for the standard scheme. First, for each grid cell we store the moment vector $\overline{\mathbf{u}}_i^{\kappa-1}$ from the last time step and the corresponding multiplier $\widetilde{\boldsymbol{\alpha}}_i^{\kappa-1}$ obtained by entropy minimization. In this way we do not have to solve the optimization problem again if the moment vector in that grid cell did not change during the last time step. In addition, we store the last few solutions of the minimization problem with corresponding input moment vectors per thread of execution, so if several grid cells contain the same values, we only have to perform the optimization once and then use the cached values. If we encounter a moment vector that can not be found in the caches, we take the moment vector that is closest to the input vector (in one-norm) and use the corresponding multiplier $\boldsymbol{\alpha}$ as an initial guess.

### *4.1.4. Linear solvers*

In each iteration of the Newton scheme described above and in each time step of the new scheme, we have to apply the inverse of a positive definite Hessian matrix. We assemble the matrices using the quadratures described in Section 4.1.6. Inversion is then done by computing a Cholesky factorization of the assembled matrix. For the full moment models, the Hessian matrices are dense, so we use the `LAPACK` [5] routine `dpotrf` to compute the factorization and then use `dtrsv` to actually invert the linear systems. For the $\mathrm{PMM}_n$ models, the Hessian is block-diagonal (each block corresponds to one interval/triangle of the partition) such that we can perform the Cholesky decomposition independently for each block. For the $\mathrm{HFM}_n$ models in one dimension, the Hessian matrices are tridiagonal, so we can use the specialized `LAPACK` algorithms `dpttrf` and `dpttrs`. In three dimensions, the $\mathrm{HFM}_n$ Hessians are not tridiagonal anymore but still sparse, so we use the sparse `SimplicialLDLT` solver from the `Eigen` library [26].

### *4.1.5. Regularization*

Though the Hessian $\mathbf{H}(\boldsymbol{\alpha})$ is positive definite and thus invertible, it may be very badly conditioned, especially for multipliers $\boldsymbol{\alpha}$ corresponding to moments $\mathbf{u}(\boldsymbol{\alpha})$ close to the boundary of the realizable set. Moreover, in general, the integral in the definition (4.3) of $\mathbf{H}$ can only be calculated approximately using a numerical quadrature (see Sect. 4.1.6). If the quadrature is not sufficiently accurate, the approximate Hessian may have a significantly worse condition or may even be numerically singular.

To improve this situation, a change of basis can be performed after each Newton iteration such that the Hessian at the current iterate becomes the unit matrix in the new basis [3]. We use this procedure in our implementation for all bases except for the hat function bases $\mathbf{h}_n$ where the change of basis would destroy the sparsity of the Hessian [63].

For tests with strong absorption, the local particle density may become very small in parts of the domain. As a consequence, also the entries of the Hessian $\mathbf{H}$ become very small which may cause numerical problems. We thus choose a "vacuum" density $\psi_{\mathrm{vac}}$ with corresponding local particle density $\rho_{\mathrm{vac}} = \langle\psi_{\mathrm{vac}}\rangle$. We then enforce a minimum local particle density of $\rho_{\mathrm{vac}}$ by replacing moments $\mathbf{u}$ with local particle density $\rho(\mathbf{u}) < \rho_{\mathrm{vac}}$ by the isotropic moment with vacuum density $\rho_{\mathrm{vac}}$. Obviously, this approach leads to a violation of the conservation properties of the scheme. However, since we only replace moments with very small local particle densities by moments with slightly larger but still very small densities, the effect should be negligible in practice.

Finally, if the optimizer fails for a moment vector $\mathbf{u}$ (for example, by reaching a maximum number of iterations or being unable to solve for the Newton direction) we use the isotropic-regularization technique from [3], *i.e.*, we replace $\mathbf{u}$ by the regularized moment vector

$$\mathbf{u}^r := (1-r)\mathbf{u} + r\mathbf{G}_{\mathbf{b}}^{\mathrm{iso}}\mathbf{u}. \tag{4.9}$$

and retry the optimization. If the optimizer still fails, we increase $r$ until the optimizer succeeds, which is guaranteed at least for $r = 1$ where $\mathbf{u}^r$ is isotropic. In our implementation, the sequence of regularization parameters $r$ is chosen as $\left\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 0.01, 0.05, 0.1, 0.5, 1\right\}$. As the regularized moment vector $\mathbf{u}^r$ always has the same local particle density as the original moment vector $\mathbf{u}$, this technique does not violate the mass conservation of the scheme but it may potentially completely alter the solution. In practice, regularization is only used rarely and if it is used, a small regularization parameters is usually sufficient.

### *4.1.6. Quadrature rules*

We have to approximate the same integrals for both schemes, so we use the quadratures and quadrature orders that have been determined in [63] for the standard scheme. Using these quadratures, the quadrature error should usually be negligible compared to the moment approximation error [63].

In one dimension, we use Gauss–Lobatto quadratures. These quadratures include the endpoints of the interval in the set of quadrature points which ensures that the numerically realizable set and the analytically realizable set agree for the $\mathrm{HFM}_n$ and $\mathrm{PMM}_n$ models (see [63] for details). For these models, we use a quadrature of order 15 per interval of the partition $\mathcal{P}$. For the full moment $\mathrm{M}_N$ models, we split the domain in the two intervals

$[-1,0]$ and $[0,1]$ that are needed for calculation of the kinetic flux and use a quadrature of order $2N+40$ on each interval.

In three dimensions, for the HFM$_n$ and PMM$_n$ models, we use Fekete quadratures on each spherical triangle of the triangulation $\mathcal{P}$. Similar to the one-dimensional situation, these quadratures include the vertices of the triangle which is again important for realizability considerations [63]. We use a quadrature order of 15 for the HFM$_6$ and PMM$_{32}$ models and a quadrature order of 9 for the other HFM$_n$ and PMM$_n$ models. For the M$_N$ models, we use tensor-product quadrature rules of order $2N+8$ on the octants of the sphere.

### *4.1.7. Implementation of initial and boundary conditions*

The initial values for the finite volume scheme are computed by integration of the kinetic equation's initial values (2.1b):

$$\mathbf{u}_i^0 = \frac{1}{|T_i|}\int_{T_i} \langle \psi_{t=0}(\mathbf{x},\mathbf{\Omega})\mathbf{b}\rangle \mathrm{d}\mathbf{x}$$

Since the initial values in our test cases are isotropic (see Sect. 5), *i.e.*, $\psi_{t=0}(\mathbf{x},\mathbf{\Omega}) = \psi_{t=0}(\mathbf{x})$, we only have to compute the velocity integral of the basis $\langle \mathbf{b}\rangle$. For this integral, we use the same quadratures as in Section 4.1.6 to ensure that the result is numerically realizable. Except for the plane-source and point-source tests, the initial values are constant in each grid cell $T_i$, so we use the midpoint quadrature to evaluate the spatial integral. For the plane-source test, we always use an even number of grid cells and distribute the Dirac delta at $x=0$ into the two adjacent grid cells, *i.e.*, the initial value in these grid cells is set to the constant $\psi_{t=0}|_{T_i}(\mathbf{x}) = \psi_{\text{vac}} + \frac{1}{2\Delta x}$. For the point-source test, we use a Gauss–Legendre tensor product quadrature of order 20 to evaluate the spatial integrals for the initial values.

Boundary conditions for the moment equations are implemented by replacing the ansatz function $\hat{\psi}_{\mathbf{u}_j}$ belonging to a grid cell $T_j$ outside of the computational domain (such cells often called "ghost cells") by the boundary condition $\psi_b$ of the kinetic equation (2.1c) in the computation of the kinetic flux (2.30).

## 4.2. New scheme

For the new scheme, evaluation of quadrature rules and boundary conditions and assembly and inversion of the Hessian matrices is performed exactly in the same way as for the standard scheme (see Sects. 4.1.4, 4.1.6 and 4.1.7. To get the initial values $\left\{\overline{\boldsymbol{\alpha}}_i^0\right\}$ for the new scheme, we solve the minimum entropy problems for the initial moments $\left\{\overline{\mathbf{u}}_i^0\right\}$ which are computed as described in Section 4.1.7 using the Newton scheme from Section 4.1.2.

### *4.2.1. Embedded and relaxed Runge–Kutta schemes*

Our time stepping scheme is outlined in Algorithm 1. As mentioned above (see Sect. 3.3), we use embedded Runge–Kutta methods (see, *e.g.*, [28], Chapt. II.4) to adaptively choose the time step for the new scheme. These schemes include a second set of coefficients $\{\widetilde{b}_p\}_{p=0,\ldots,s-1}$ to obtain a different approximation (usually of lower order) of the solution at the next time step which is used for error estimation. More precisely, in addition to the approximation $\overline{\boldsymbol{\alpha}}_i^{\kappa+1}$ given by (3.12a) we compute a second approximation

$$\widetilde{\boldsymbol{\alpha}}_i^{\kappa+1} = \overline{\boldsymbol{\alpha}}_i^{\kappa} + \Delta t \sum_{p=0}^{s-1} \widetilde{b}_p \boldsymbol{\alpha}_i^{\uparrow}\left(\boldsymbol{\beta}_0^p, \ldots, \boldsymbol{\beta}_{n_{\mathbf{x}}-1}^p\right) \tag{4.10}$$

in each grid cell $T_i$ and regard the error between these two approximations to decide if the time step is appropriate. As an error measure, we use the mixed error (see [28], Chapt. II.4, Eq. (4.11))

$$\operatorname{err}\left(\hat{\boldsymbol{\alpha}}^{\kappa+1}, \widetilde{\boldsymbol{\alpha}}^{\kappa+1}\right) = \max_{l=0,\ldots,M-1} \frac{|\alpha_l - \widetilde{\alpha}_l|}{\tau_{\text{abs}} + \max(\alpha_l, \widetilde{\alpha}_l)\tau_{\text{rel}}}. \tag{4.11}$$

Here, $\tau_{\text{abs}}$ and $\tau_{\text{rel}}$ are absolute and relative error tolerances and $\alpha_l$ and $\widetilde{\alpha}_l$ are the $M = n \cdot n_{\mathbf{x}}$ components of the coefficient vectors

$$\hat{\boldsymbol{\alpha}}^{\kappa+1} = \left(\left(\overline{\boldsymbol{\alpha}}_0^{\kappa+1}\right)^T, \dots, \left(\overline{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}^{\kappa+1}\right)^T\right)^T \quad \text{and} \quad \widetilde{\boldsymbol{\alpha}}^{\kappa+1} = \left(\left(\widetilde{\boldsymbol{\alpha}}_0^{\kappa+1}\right)^T, \dots, \left(\widetilde{\boldsymbol{\alpha}}_{n_{\mathbf{x}}-1}^{\kappa+1}\right)^T\right)^T \tag{4.12}$$

respectively. For simplicity, we will always use $\tau_{\text{abs}} = \tau_{\text{rel}} = \tau$ in the following. We use an automatic step size control which tries to select the time step as large as possible while still maintaining an error (4.11) below one, *i.e.*, at $t = 0$ we start with a very small time step of $\Delta t_0 = 10^{-15}$ and then compute the new time step $\Delta t_{\text{new}}$ from the previous time step $\Delta t$ by (compare [28], Chapt. II.4, Eq. (4.13))

$$\Delta t_{\text{new}}(\Delta t, \text{err}) = \Delta t \cdot \min\left(\max\left(0.8 \cdot \left(\frac{1}{\text{err}}\right)^{\frac{1}{q+1.}}, \frac{1}{5}\right), 5\right) \tag{4.13}$$

where $q$ is the order of the lower-order scheme in the adaptive Runge–Kutta method, 0.8 is a safety factor and the minimum and maximum ensure that the time step does not change too fast. If err $> 1$ or if an exception is thrown during the computation (*e.g.*, if a matrix inversion fails or `inf`s or `NaN`s are detected in the results) we recompute the solutions with halved time step $\frac{\Delta t}{2}$.

**Algorithm 1:** Adaptive time stepping scheme.

```
$\Delta t \leftarrow 10^{-15}$, $t \leftarrow 0$, $\hat{\boldsymbol{\alpha}}^{\kappa} \leftarrow \hat{\boldsymbol{\alpha}}^{0}$;
while $t < t_{\text{end}}$ do
    $e = 1000$; // Make sure the following while-loop is entered
    while $e > 1$ do
        try:
            Compute $\hat{\boldsymbol{\alpha}}^{\kappa+1}$, $\widetilde{\boldsymbol{\alpha}}^{\kappa+1}$ from $\hat{\boldsymbol{\alpha}}^{\kappa}$ by applying (3.12a), (4.10) on each grid cell.
        catch (Linear solver failures or invalid values (infs or NaNs) in $\hat{\boldsymbol{\alpha}}^{\kappa+1}$, $\widetilde{\boldsymbol{\alpha}}^{\kappa+1}$):
            $\Delta t \leftarrow \Delta t/2$;
            continue;
        end
        $\Delta t_{\text{accepted}} \leftarrow \Delta t$;
        $e \leftarrow \text{err}\big(\hat{\boldsymbol{\alpha}}^{\kappa+1}, \widetilde{\boldsymbol{\alpha}}^{\kappa+1}\big)$; // see (4.11)
        $\Delta t \leftarrow \Delta t_{\text{new}}(\Delta t, e)$; // see (4.13)
    end
    $t \leftarrow t + \Delta t_{\text{accepted}}$;
    $\hat{\boldsymbol{\alpha}}^{\kappa} \leftarrow \hat{\boldsymbol{\alpha}}^{\kappa+1}$;
end
```

For implementation of the relaxed Runge–Kutta scheme, we simply compute (compare (3.20))

$$\sum_{i \in I_{\mathcal{G}}} \sum_{p=0}^{s-1} \check{b}_p \boldsymbol{\beta}_i^p \cdot \mathbf{u}_i^{\uparrow}\left(\boldsymbol{\beta}_0^p, \dots, \boldsymbol{\beta}_{n_{\mathbf{x}}-1}^p\right)$$

on the fly while computing our Runge–Kutta scheme. Once the time step is accepted by the adaptive control, we compute

$$\Delta t \sum_{p=0}^{s-1} \check{b}_p \boldsymbol{\alpha}_i^{\uparrow}\left(\boldsymbol{\beta}_0^p, \dots, \boldsymbol{\beta}_{n_{\mathbf{x}}-1}^p\right)$$

(which is needed to compute $\boldsymbol{\alpha}_{i,\gamma}^{\kappa+1}$, see (3.19))) and use a simple bisection algorithm to find the root $\gamma_\kappa$ of $r(\gamma)$ that is close to 1. We then compute $\overline{\boldsymbol{\alpha}}_i^{\kappa+1} = \overline{\boldsymbol{\alpha}}_{i,\gamma_\kappa}^{\kappa+1}$ according to (3.19).

## 5. Numerical experiments

We want to investigate the behaviour of the new scheme in several benchmarks. For that purpose, we use the same test cases as in [63]. Our C++ implementation and the generated data can be found in [37].

In the following, we will briefly restate the test cases. For a more detailed description and plots of (numerical) solutions see [63] and references therein. As the minimum entropy models cannot handle zero densities, we use the small isotropic distribution $\psi_{\text{vac}} = 5 \cdot 10^{-7}$ to approximate a vacuum. Note that we increased the vacuum density slightly compared to [63] to avoid numerical difficulties with very low densities. We use the following test cases:

– *Plane-source.* In this test case, all mass is concentrated in the middle of the computational domain $X = [-1.2, 1.2]$, *i.e.*, we use the isotropic initial distribution

$$\psi_{t=0}(z,\mu) = \psi_{\text{vac}} + \delta(z) \quad \text{for } z \in X.$$

See Section 4.1.7 for details on the implementation of this initial condition. The physical coefficients are set to $\sigma_s \equiv 1$, $\sigma_a \equiv 0$ and $Q \equiv 0$. Vacuum boundary conditions are used.

– *Source-beam.* In this test case, a strongly anisotropic beam enters the computational domain $X = [0, 3]$ from the left. In addition, a source is present in the interval $[1, 1.5]$. More precisely, the approximate vacuum is used as initial condition and boundary condition on the right-hand side, and the left boundary distribution is

$$\psi_b(t, 0, \mu) = \frac{e^{-10^5(\mu-1)^2}}{\left\langle e^{-10^5(\mu-1)^2} \right\rangle}.$$

The parameters are set to

$$\sigma_a(z) = \begin{cases} 1 & \text{if } z \le 2, \\ 0 & \text{else,} \end{cases} \quad \sigma_s(z) = \begin{cases} 0 & \text{if } z \le 1, \\ 2 & \text{if } 1 < z \le 2, \\ 10 & \text{else} \end{cases} \quad Q(z) = \begin{cases} \frac{1}{2} & \text{if } 1 \le z \le 1.5, \\ 0 & \text{else.} \end{cases}$$

– *Point-source.* The point-source test is a smoothed three-dimensional analogue of the plane-source test. The initial condition in the domain $X = [-1, 1]^3$ is

$$\psi_{t=0}(\mathbf{x}, \mathbf{\Omega}) = \psi_{\text{vac}} + \frac{1}{4\pi^4\sigma^3} \exp\left(-\frac{|\mathbf{x}|^2}{\pi\sigma^2}\right),$$

where $\sigma = 0.03$. The parameters are the same as in the plane-source test.

– *Checkerboard.* The checkerboard test case is loosely based on a part of a reactor core [11]. The domain $X = [0, 7]^3$ is split into scattering and absorbing regions, $X = X_s \cup X_a$, where

$$X_a = \left\{ \mathbf{x} = (x, y, z)^T \in [1, 6]^3 \,\middle|\, \begin{array}{l} (\lfloor x \rfloor + \lfloor y \rfloor + \lfloor z \rfloor) \bmod 2 = 1, \\ \mathbf{x} \notin [3, 4]^3 \cup [3, 4] \times [5, 6] \times [3, 4] \end{array} \right\}$$

The parameters are

$$\sigma_s(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in X_s, \\ 0 & \text{else,} \end{cases}, \quad \sigma_a(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in X_s, \\ 10 & \text{else,} \end{cases}, \quad Q(\mathbf{x}) = \begin{cases} \frac{1}{4\pi} & \text{if } \mathbf{x} \in [3, 4]^3, \\ 0 & \text{else.} \end{cases}$$

Vacuum initial and boundary conditions are used.

– *Shadow.* The shadow test case represents an isotropic particle stream that is partially blocked by an absorber, resulting in a shadowed region behind the absorber. The particle stream is given by an isotropic boundary condition with density $\rho = 2$ at $x = 0$. On the other boundaries of the domain $X = [0, 12] \times [0, 4] \times [0, 3]$ and as an initial condition, the approximate vacuum is prescribed. The parameters are as follows:

$$\sigma_s(\mathbf{x}) = Q(\mathbf{x}) = 0$$

$$\sigma_a(\mathbf{x}) = \begin{cases} 50 & \text{if } \mathbf{x} \in [2, 3] \times [1, 3] \times [0, 2] \\ 0 & \text{else.} \end{cases}$$

Whenever we need a reference scheme, we use the splitting scheme based on (2.36). The obvious choice might be the unsplit scheme (2.32), as the new scheme is just a coordinate transformation of this scheme. However, the splitting scheme is easy to implement and avoids the time step restriction due to the physical parameters. For the new scheme, a similar splitting approach is not straightforward. Thus, using (2.32) as a reference would arguably give the new scheme an unfair advantage.

## 5.1. Convergence

Under some assumed bounds on the Hessian $\mathbf{H}$ and the local particle density $\rho$, the transformed scheme will always converge to the same solution as the splitting scheme (2.36) (see Sect. 3.4). However, in practice, taking numerical errors into account, these bounds may not always hold (in particular the lower bound on the norm of $\mathbf{H}$).

In our first experiment, we thus want to validate that the two schemes converge to the same solutions also in our numerical test cases. For that purpose, we compute numerical solutions with both schemes for varying tolerance and time step parameter, respectively, and calculate the errors with respect to a reference solution (new scheme with $\tau = 10^{-9}$).

**Remark 5.1.** It might be more intuitive to compute the reference solution using the standard finite volume scheme with a very small time step $\Delta t$. However, since we only want to show that both schemes converge to the same solution, it does not matter whether we use the standard scheme or the new transformed scheme as a reference, and computing the new scheme for a small tolerance $\tau$ is significantly faster.

As an error measure, we choose the $L^1$-error of the piecewise constant finite volume approximations $E^1(\mathbf{u}) = \|\mathbf{u} - \mathbf{u}_{\text{ref}}\|_{L^1(X)}$ at the final time $t_{\text{end}}$. We use a relatively coarse grid for all test cases to be able to compute the results for very small tolerance parameter $\tau$ or time step $\Delta t$ in reasonable time. However, we confirmed at least for large parameters ($\tau \in \{10^{-2}, 10^{-3}\}$, $\Delta t = \Delta t_{\max}$) that the results are similar for the grid sizes and final times used in Section 5.5 (see Tabs. S1 and S2 in the supplementary materials).

As can be seen in Figure 1 (for the plane source test) and Figures S1 and S2 in the supplementary materials (source-beam and point-source), both schemes nicely converge to the same solution. For the standard scheme, the error is basically independent of the model which is not true for the new scheme. This is probably due to the fact that for the new scheme, the error estimate during the time stepping is calculated in transformed ($\boldsymbol{\alpha}$-)variables while the final error is plotted in the original ($\mathbf{u}$-)variables. The $L^\infty$-errors behave similarly (data not shown).

## 5.2. Time stepping behavior

Now that we have confirmed also numerically that the new scheme indeed yields the same solutions as the standard scheme, we would like to investigate the properties of the new scheme. We will focus on the time stepping behavior first. We will only present the results for some exemplary models in the one-dimensional test cases here. Results for additional models and for the three-dimensional tests are similar and can be found in the supplementary materials (Figs. S3–S10).

Figure 2 shows the time steps chosen by the adaptive time stepping scheme (with a tolerance of $\tau = 10^{-3}$) in the one-dimensional test cases for some exemplary models. As expected, for all models, the new scheme takes

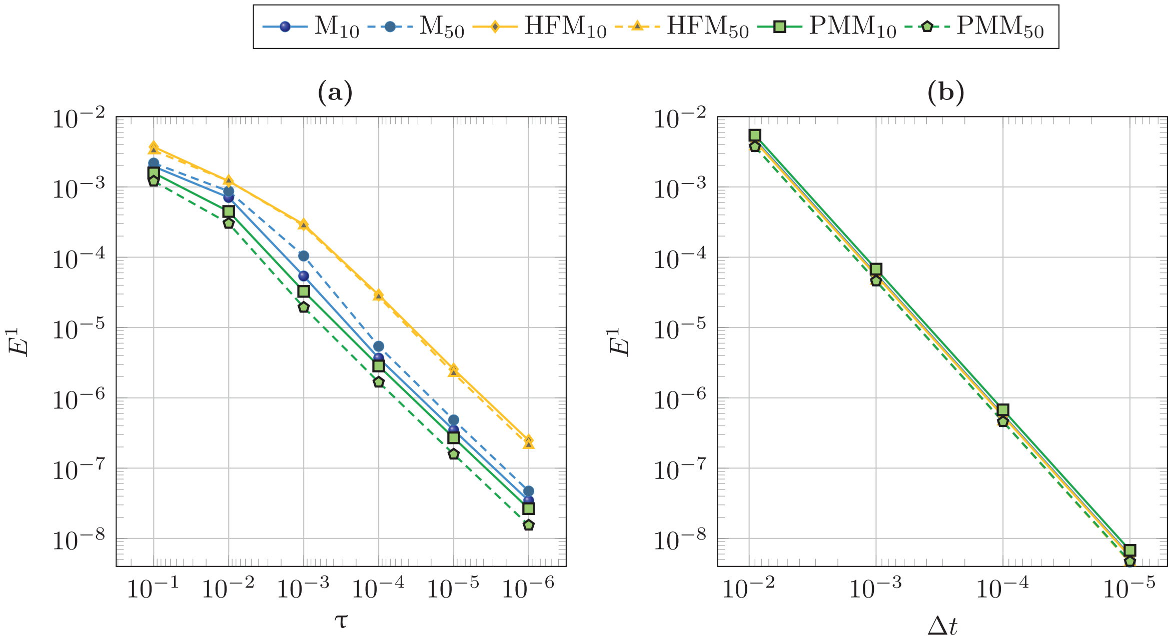

FIGURE 1. $L^1$-error against reference solution (new scheme with $\tau = 10^{-9}$) in the plane-source test ($n_x = 240, t_{\text{end}} = 0.5$). (a) New scheme for decreasing tolerance parameter τ. (b) Standard scheme for decreasing time step $\Delta t$.

very small time steps initially. This can be explained by the large time derivatives of the solution in $\boldsymbol{\alpha}$-variables at the beginning of the test (see Example 3.5).

In the plane-source test (Fig. 2a), the time steps are rapidly and almost monotonically increasing for all models and finally reach a time step that is even above the maximal realizability-preserving time step (2.37) used in the standard scheme (except for the PMM$_2$ model). In the source-beam test (Fig. 2b), the time steps are also increasing initially but are less stable afterwards. In particular for the M$_N$ models there are strong oscillations in the time step sizes.

To test the influence of the tolerance parameter τ on the time steps, we compute the test cases for the M$_{10}$ model again for varying τ. The M$_{10}$ model was chosen since it is more complex than the M$_2$ model, not as expensive to compute as the M$_{100}$ model and still shows the time step oscillations in the source-beam test. However, we also tested several other models and found that they all show a similar behavior with respect to the τ tolerance. As can be seen in Figure 3, for small tolerances ($\tau \leq 10^{-3}$), increasing τ basically just scales the time step curve by a constant factor which is due to the third-order time-stepping scheme (increasing τ by a factor of 10 results in an increase of the time step by a factor of approximately $\sqrt[3]{10}$). However, this scaling does not extend to large tolerance parameters ($\tau = 10^{-1}, 10^{-2}$). In particular, for the plane-source test, increasing the tolerance above $\tau = 10^{-3}$ does not result in larger time steps (see Fig. 3a). In addition, the time steps oscillate much more. This is due to the fact that the time step predicted from the standard error estimate for these tolerances is often too large, leading to `inf`s and `NaN`s during the computations and a subsequent reduction in the time step (compare Sect. 4.2.1).

Finally, we measured the times needed to compute a single time step of the new scheme or the standard scheme for several models (see Fig. 4). For the new scheme, the time needed to compute a time step is basically constant, except for time steps which have to be recomputed because the error estimate is above the tolerance. In contrast, time step computation times of the standard scheme are increasing over time. This is probably mostly

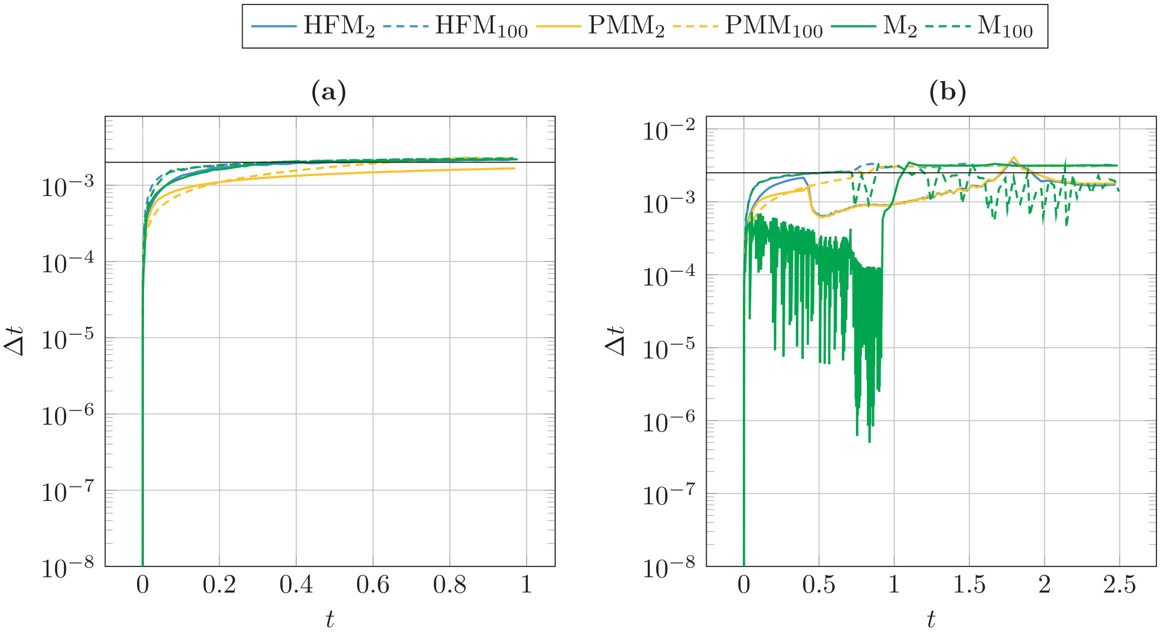

FIGURE 2. Time steps taken in the adaptive Runge–Kutta scheme. For clarity, instead of plotting each time point we plot the mean of 10 time steps each to avoid small oscillations. The solid horizontal line represents the upper bound (2.37) on the time step used in the standard splitting scheme (which approximately agrees with the bound (2.34) because $\frac{1}{\Delta x} \gg \sigma_t$ for these test cases). (a) Plane-source test, $n_x = 1200$, $t_{\text{end}} = 1$, $\tau = 0.001$. (b) Source-beam test, $n_x = 1200$, $t_{\text{end}} = 2.5$, $\tau = 0.001$.

due to the caching used in the implementation of the standard scheme (see Sect. 4.1.3) which is particularly effective during the first time steps where most grid cells still contain the initial approximate vacuum. The new scheme does not use any caching. As a consequence, the standard scheme is faster for the first few time steps but after a short time the new scheme's time steps are computed significantly faster.

As can also be seen from Figure 4, recomputations of time steps in the adaptive time stepping scheme (which show up in Fig. 4 as spikes in the computation times of the new scheme) occur rarely, except for the M$_N$ models in the source-beam test. This is in line with the erratic time stepping behavior that we observed for these models (compare Figs. 2b and 3b) and can be improved using regularization (see next section).

## 5.3. Regularization

### *5.3.1. Isotropic regularization of the Hessian*

In the previous section, we saw that the time steps for the M$_{10}$ model in the source-beam test (Fig. 3b) oscillate a lot between $t = 0.5$ and $t = 1$ and sometimes even get as small as $10^{-8}$. These oscillations are mostly caused by ill-conditioned Hessian matrices which arise from the interaction between the highly anisotropic particle beam from the left boundary and the particles from the source $Q$ in the absorbing but non-scattering region $[0, 1]$.

A possible workaround for this problem is the regularization (3.15) which adds a small isotropic particle density during computation of the Hessian matrix. As can be seen when comparing Figures 5a to 3b, this indeed improves the time step sizes and removes the very small time steps. As a consequence, the regularized

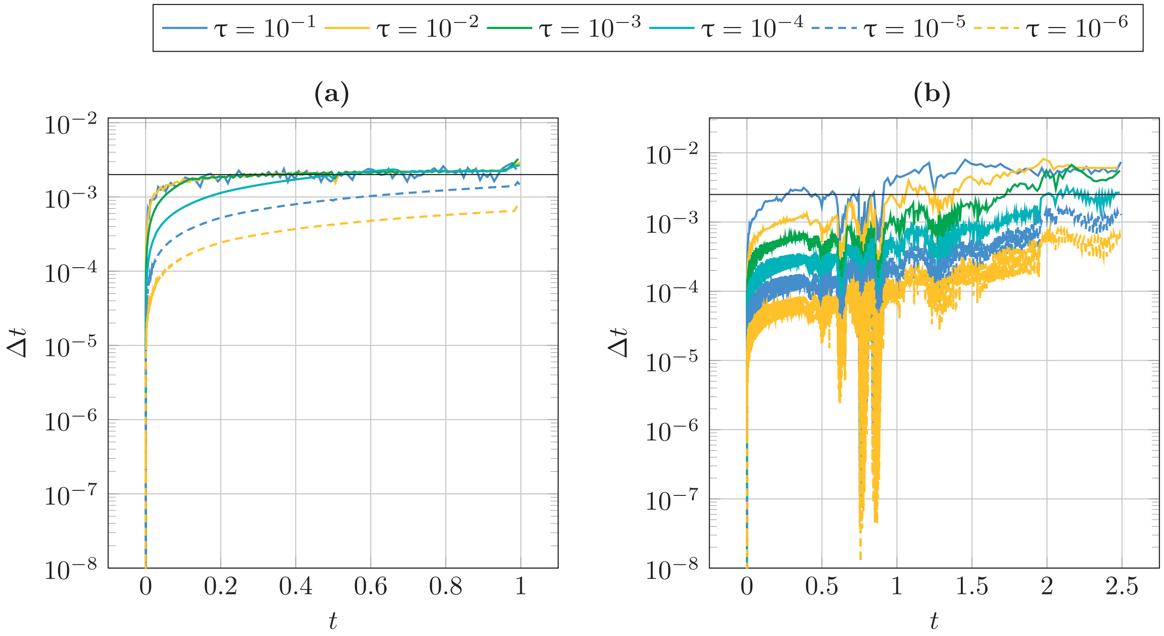

FIGURE 3. Time steps for the $M_{10}$ model in the plane-source ($n_{\mathbf{x}} = 1200$, $t_{\text{end}} = 1$) and the source-beam test ($n_{\mathbf{x}} = 600$, $t_{\text{end}} = 2.5$) for different tolerance parameters $\tau$. Again, the solid horizontal line represents the upper bound (2.37) and the time steps are plotted as the mean of 10 time steps each. (a) Plane-source. (b) Source-beam.

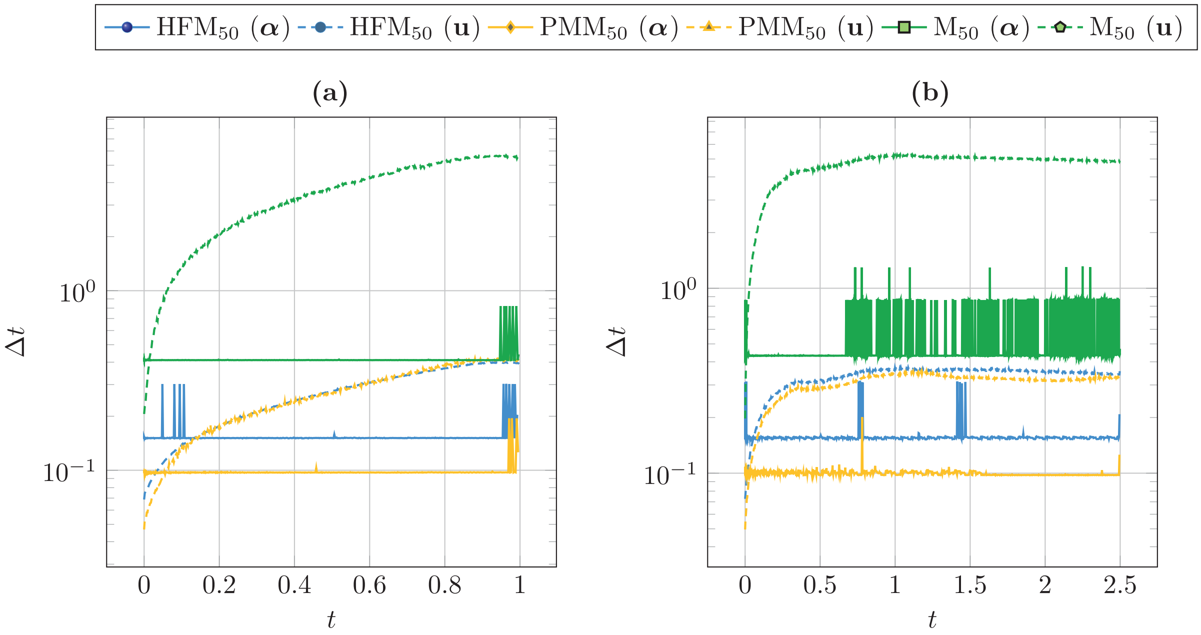

FIGURE 4. Wall times for computing a single time step of new and standard schemes (using $\boldsymbol{\alpha}$ and $\mathbf{u}$ variables, respectively) in one-dimensional tests. (a) Plane-source test, $n_x = 1200$, $t_{\text{end}} = 1$, $\tau = 0.001$. (b) Source-beam test, $n_x = 1200$, $t_{\text{end}} = 2.5$, $\tau = 0.001$.

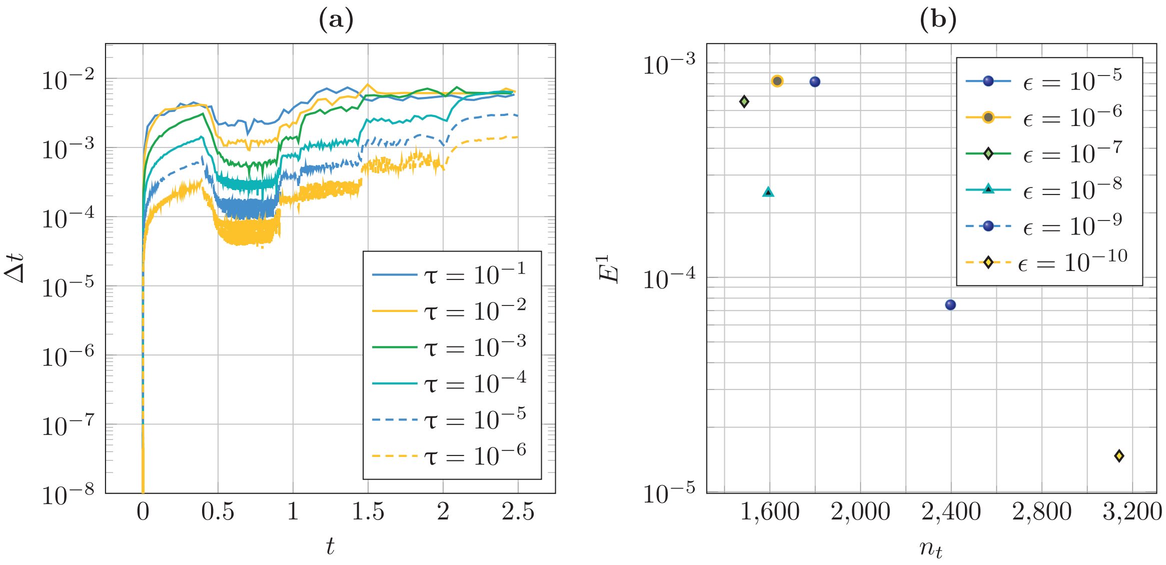

FIGURE 5. Effect of the regularization (3.15) on time steps and errors for the M$_{10}$ model in the source-beam test. (a) Time steps for different tolerance parameters $\tau$ using a regularization parameter of $\epsilon = 10^{-7}$. (b) Errors ($L^1$-error with respect to the non-regularized solution) and number of time steps $n_t$ for $\tau = 10^{-9}$ and varying regularization parameter $\epsilon$. The non-regularized version uses $n_t = 3918$ time steps.

scheme needs significantly less time steps (see Fig. 5b). For example, the regularized scheme with regularization parameter $\epsilon = 10^{-7}$ only needs $n_t = 1495$ time steps instead of 3918 for the non-regularized scheme.

The price to pay for the regularization is an additional error in the order of $10^{-3}$–$10^{-4}$ at the final time $t_{\text{end}}$ (see Fig. 5b for the $L^1$ error, the $L^\infty$ error is of the same order (data not shown)). This seems to be below the typical error introduced by the temporal discretization in the standard scheme (see Sect. 5.5.1 and Fig. S1b).

Increasing the regularization parameter $\epsilon$ above $10^{-7}$ increases the regularization error but does not further decrease the number of time steps.

### *5.3.2. Regularization for the hat function basis*

For the hat function (HFM$_n$) models, we can use a different regularization technique that might result in smaller regularization errors. Although, according to our testing, time step declines are much less common for the HFM$_n$ models than for the M$_N$ models, we observed such declines for the HFM$_6$ model in the shadow test around $t = 8$ (see Fig. 6a). In this test, (almost) all particles enter the domain with positive $x$-velocity. Since there is no scattering and a strongly absorbing region, the density of particles with negative $x$-velocity becomes very low in parts of the domain which also leads to large (in absolute values) negative entries in the $\hat{\boldsymbol{\alpha}}$ coefficient vectors. To improve the situation, we tested the regularization technique introduced in Section 3.5.2: Whenever the time step falls below $\Delta t_{\min} = 0.01$, we replace all entries in the $\hat{\boldsymbol{\alpha}}$ vector that are smaller than $\alpha_{\min} = -1000$ by $\alpha_{\min}$.

As can be seen in Figure 6b, this simple regularization technique removes the very small time steps. The overall number of time steps is reduced from 1152 to 547. The $L^1$-error at the final time $t_{\text{end}}$ between the results with and without regularization is only about $10^{-6}$ in this case.

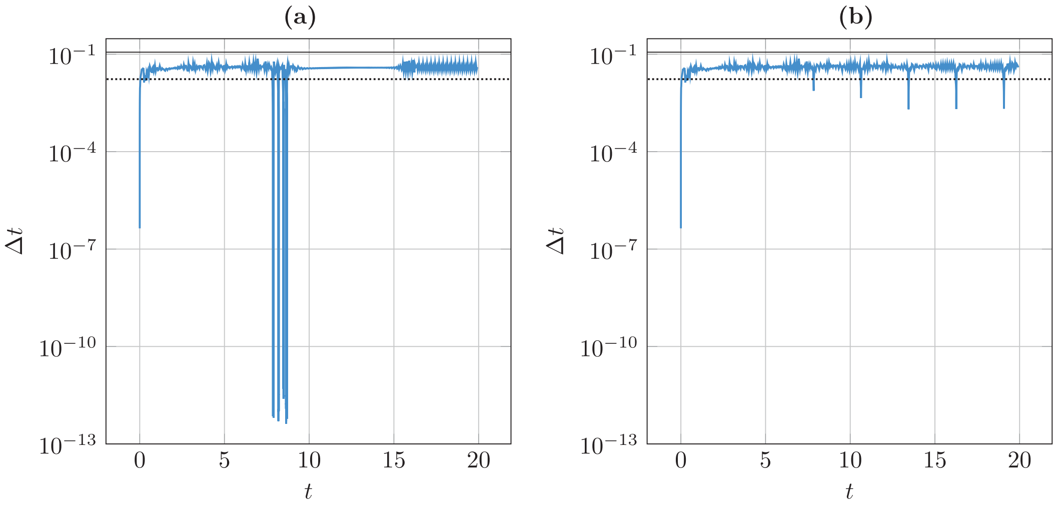

FIGURE 6. Time steps taken by the adaptive Runge–Kutta scheme in the $\mathrm{HFM}_6$ shadow test case ($n_{\mathbf{x}} = 60 \times 20 \times 15$, $t_{\mathrm{end}} = 20$) for a tolerance of $\tau = 0.01$. The solid and dotted horizontal line represent the time step restrictions (2.37) and (2.34), respectively. (a) No regularization. The time step sharply declines around $t = 8$. (b) When enforcing a lower bound of $-1000$ on the entries of the multipliers $\boldsymbol{\alpha}$, the time steps stay above $10^{-3}$.

### 5.4. Entropy stability

To test the entropy stability properties of the different schemes, we calculate the entropy $\hat{\mathcal{H}}$ (see Sect. 3.6) at each time step and compute the difference between the actual entropy $\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa+1})$ and the entropy given by the discrete entropy law. To that end, let $\hat{\mathcal{H}}_{\mathrm{est}}(\hat{\boldsymbol{\alpha}}^{\kappa})$ be the entropy estimated for the next time point $t_{\kappa+1}$ from $\hat{\boldsymbol{\alpha}}^{\kappa}$ using the entropy law (3.18). To get an compact error measure, we use the cumulated difference between actual entropy and estimated entropy, *i.e.*,

$$\Delta\hat{\mathcal{H}} := \sum_{\kappa=0}^{n_t-1} \left| \hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa+1}) - \hat{\mathcal{H}}_{\mathrm{est}}(\hat{\boldsymbol{\alpha}}^{\kappa}) \right| \Delta t_{\kappa}$$

We tested several representative moment models in the one-dimensional test cases using the new transformed scheme with either the non-modified Runge–Kutta scheme (RK) or the relaxed Runge–Kutta scheme (RRK). We restrict our investigation to one dimension here since we do not expect a qualitatively different behavior in three dimensions.

As can be seen in Table 1, with the non-relaxed Runge–Kutta scheme the difference $\Delta\hat{\mathcal{H}}$ between the entropy of the solutions and the discrete entropy law (3.18) is in the order of $10^{-4}$ to $10^{-7}$. The entropy $\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^{\kappa})$ is in the order of 100 to 1000, so the relative error is quite low even without relaxation. If relaxation is used, the error vanishes for all test cases (up to a remainder in the order of $10^{-12}$ which is the tolerance parameter for our root finding algorithm). However, the relaxation comes at a price, as can be seen from the computation times. With relaxation, the computations take at least twice as long (for the $\mathrm{M}_{50}$ model in the source-beam test), up to a factor of about 20 for the $\mathrm{HFM}_{50}$ model in the plane-source test. Note that we use a simple custom bisection root finding algorithm to compute the relaxation parameters and that we did not optimize the relaxed version for performance, so it should be possible to significantly reduce the performance impact of the relaxation. Still, given that the error is already quite low without relaxation, it seems advisable to simply use the non-modified Runge–Kutta scheme unless exact entropy stability is required in the application.

Table 1. Entropy stability results. The average absolute value of the entropy is computed as $|\hat{\mathcal{H}}|_{\mathrm{av}} = \frac{1}{t_{\mathrm{end}}} \sum_{\kappa=0}^{n_t-1} |\hat{\mathcal{H}}(\hat{\boldsymbol{\alpha}}^\kappa)|\Delta t_\kappa$. RK: Transformed scheme using Runge–Kutta method. RRK: Transformed scheme using relaxed Runge–Kutta method.

| | | | Wall time | | | $\Delta\hat{\mathcal{H}}$ | |
|---|---|---|---|---|---|---|---|
| Test case | Model | $\lvert\hat{\mathcal{H}}\rvert_{\mathrm{av}}$ | RK (s) | RRK (s) | Factor | RK | RRK |
| Plane-source | $\mathrm{HFM}_{10}$ | 877 | 22 | 170 | 7.7 | 3.9e-04 | 4.4e-13 |
| Plane-source | $\mathrm{HFM}_{50}$ | 878 | 81 | 1689 | 20.8 | 4.7e-04 | 4.5e-13 |
| Plane-source | $\mathrm{PMM}_{10}$ | 881 | 24 | 135 | 5.6 | 3.7e-05 | 4.7e-13 |
| Plane-source | $\mathrm{PMM}_{50}$ | 872 | 83 | 1181 | 14.2 | 5.9e-05 | 4.6e-13 |
| Plane-source | $\mathrm{M}_{10}$ | 865 | 45 | 189 | 4.2 | 2.5e-04 | 4.4e-13 |
| Plane-source | $\mathrm{M}_{50}$ | 876 | 253 | 821 | 3.3 | 3.2e-04 | 4.6e-13 |
| Source-beam | $\mathrm{HFM}_{10}$ | 411 | 41 | 442 | 10.8 | 1.3e-04 | 1.1e-12 |
| Source-beam | $\mathrm{HFM}_{50}$ | 500 | 140 | 2073 | 14.8 | 2.2e-05 | 1.1e-12 |
| Source-beam | $\mathrm{PMM}_{10}$ | 336 | 38 | 474 | 12.5 | 3.7e-04 | 2.5e-12 |
| Source-beam | $\mathrm{PMM}_{50}$ | 492 | 122 | 1413 | 11.6 | 5.2e-06 | 1.1e-11 |
| Source-beam | $\mathrm{M}_{10}$ | 507 | 955 | 3519 | 3.7 | 3.8e-07 | 1.4e-11 |
| Source-beam | $\mathrm{M}_{50}$ | 518 | 1827 | 3398 | 1.9 | 1.4e-05 | 1.3e-12 |

## 5.5. Performance

We now want to compare the performance of the new scheme to the standard splitting scheme. Since the standard scheme uses a fixed time step $\Delta t$ and the new scheme uses adaptive time steps controlled by the tolerance parameter $\tau$, we first have to decide on how to choose these parameters to have a fair comparison.

### *5.5.1. Choice of time stepping parameters*

For the standard scheme, we have an upper bound $\Delta t_{\max} := \frac{1-\varepsilon_\gamma}{\sqrt{d}}\Delta x$ (see Eq. (2.39)) on the time step due to realizability considerations. If we again consider the convergence results for the standard scheme (compare Figs. 1b, S1b, S2b and Tabs. S1, S2) we see that choosing $\Delta t = \Delta t_{\max}$ results in a time stepping error in the order of about $10^{-2}$ to $10^{-3}$. Note that the errors regarded here are solely due to the time stepping, *i.e.*, the solution both schemes are converging to in the convergence tests is the exact solution of the semidiscrete moment equation (2.29). Since we are usually interested in an approximation of the solution to the kinetic equation (2.1), we also have to take errors into account that arise due to the space discretization and due to the moment approximation. Choosing a time step smaller than $\Delta t_{\max}$ thus would only be reasonable if the time stepping error is of the same order or even larger than the errors due to the spatial discretization and the moment approximation. This seems to be the case, *e.g.*, for the checkerboard test where the moment approximation errors are relatively small and for the shadow test which has a large final time $t_{\mathrm{end}}$ such that time discretization errors accumulate over time [63]. However, for most of the regarded test cases, the errors introduced by the moment approximation (compare [63]) are much larger than the error of up to $10^{-2}$ we observed due to the time stepping. We thus always use $\Delta t = \Delta t_{\max}$ with $\varepsilon_\gamma = 0.1$ for the standard scheme.

By the same arguments, we could use a tolerance parameter of $\tau = 0.1$ for the new scheme which also yields time stepping errors in the order of $10^{-3}$ to $10^{-2}$ in our tests. However, we observed in Section 5.2 that, with with the current standard error estimate, increasing the tolerance above about $10^{-3}$ to $10^{-2}$ does not necessarily improve performance since the time steps do not increase accordingly. In some cases, we even observed increased computation times for larger tolerances as time steps had to be recomputed more frequently. We thus choose a tolerance of $\tau = 10^{-3}$ in slab geometry and $\tau = 10^{-2}$ in three dimensions.

Note that with this choice of parameters, the time stepping error is probably considerably lower for the new scheme. For the test cases where this error is insignificant, it might be possible to choose larger time steps for the new scheme with an improved error estimate that is specifically adapted to the moment equations. However, we did not find such an error estimate yet. On the other hand, for test cases (*e.g.*, checkerboard) where the time stepping error is relevant, a smaller time step would have to be used for the standard scheme which then would be significantly slower than the following results indicate. Alternatively, adaptive time stepping could also for the standard scheme. However, finding an adaptive time stepping scheme of the desired order that (provenly) preserves realizability may be difficult. The Bogacki–Shampine method used here is realizability-preserving (with the same times step restriction (2.34) as the forward Euler method) since the intermediate stages are just convex combinations of forward Euler steps and the zero vector. Similarly, the strong-stability-preserving embedded methods from the preprint [15] could be used. For the Dormand–Prince method [17], on the other hand, it is not clear under which conditions realizability is preserved (due to the negative coefficients in its Butcher tableau). In contrast, any adaptive time stepping scheme can be used for the new scheme.

For the standard scheme, we use the regularization techniques described in Section 4.1.2 to ensure that we are always able to solve the optimization problems. For the new scheme, we do not use any regularization.

### *5.5.2. Timings*

Computational times for the one-dimensional test cases can be found in Figure 7. As expected (see [63]), computational times are increasing linearly (HFM$_n$, PMM$_n$) or quadratically (M$_N$).

In the plane-source test (see Fig. 7a), the new scheme is several times faster than the standard scheme for all models except for the low-order HFM$_n$ and PMM$_n$ models.

For the source-beam test, we saw in Section 5.2 that the time steps significantly vary over time for some models. In particular, the M$_N$ models show strongly oscillating time steps. These oscillations seem to be much less pronounced for the higher-order models (see Supplementary Figs. S4 and S5). As a consequence, computational times for the new scheme are not increasing monotonically with the moment order (see Fig. 7b), *e.g.*, computing the M$_{20}$ model takes longer than computing the M$_{60}$ model. Thus, the standard scheme is faster for the low-order models and again significantly slower for the high-order models. Except for the HFM$_2$ and PMM$_2$ model, the HFM$_n$ and PMM$_n$ models do not show these oscillations and again reach time steps that are larger than $\Delta t_{\max}$ after some time. Consequently, overall computation times are faster with the new scheme.

For all models, the time steps are initially very small (see analysis in Sect. 5.1) but are rapidly increasing (see Fig. 2a). After some time, the time steps are even larger than the time steps taken by the standard scheme for most models. In addition, the time to compute a time step is (on average) much smaller for the new scheme (see Fig. 4a). This is especially true for the M$_N$ models which is why the speed-up for these models is significantly larger than for the HFM$_n$ and PMM$_n$ models where solving the optimization problem is already quite fast.

In the three-dimensional point-source test, the final time $t_{\text{end}} = 0.75$ is relatively small and none of the models reaches $\Delta t_{\max}$ during the test (compare Supplementary Fig. S6). Consequently, the computation times are only slightly faster for most models with the new scheme (see Fig. 8a). The higher-order PMM$_n$ models show considerably smaller time steps than the other models and thus overall computation times are even higher than with the standard scheme.

The checkerboard test case has strongly absorbing regions and thus is the first test where the time step restrictions (2.37) and (2.34) significantly differ. After some time, the time steps are mostly between these two bounds and even exceed the upper bound several times (compare Supplementary Fig. S6b). The higher order M$_N$ models show some oscillations in the beginning but much less than in the source-beam test and the time steps always stay relatively large. Thus, overall computation times are greatly improved and up to ten times as fast as with the standard splitting scheme (see Fig. 8b). In addition, as mentioned earlier, for this test case the increased accuracy of the new scheme might be important, as the error due to the timestepping with the standard scheme are of the same order as the error due to the spatial and moment approximation. Again, the speed-up is smaller for the HFM$_n$ models and non-existent for the PMM$_n$ models.

The shadow test case is highly challenging for the numerical solvers. In the absorbing domain, very small local particle densities occur which lead to numerical problems when inverting the Hessians (whose entries scale with the density). In addition, as only right-going particles are entering the domain, densities for particles with negative $x$-velocity decline much faster than positive $x$-velocity densities, resulting in very anisotropic

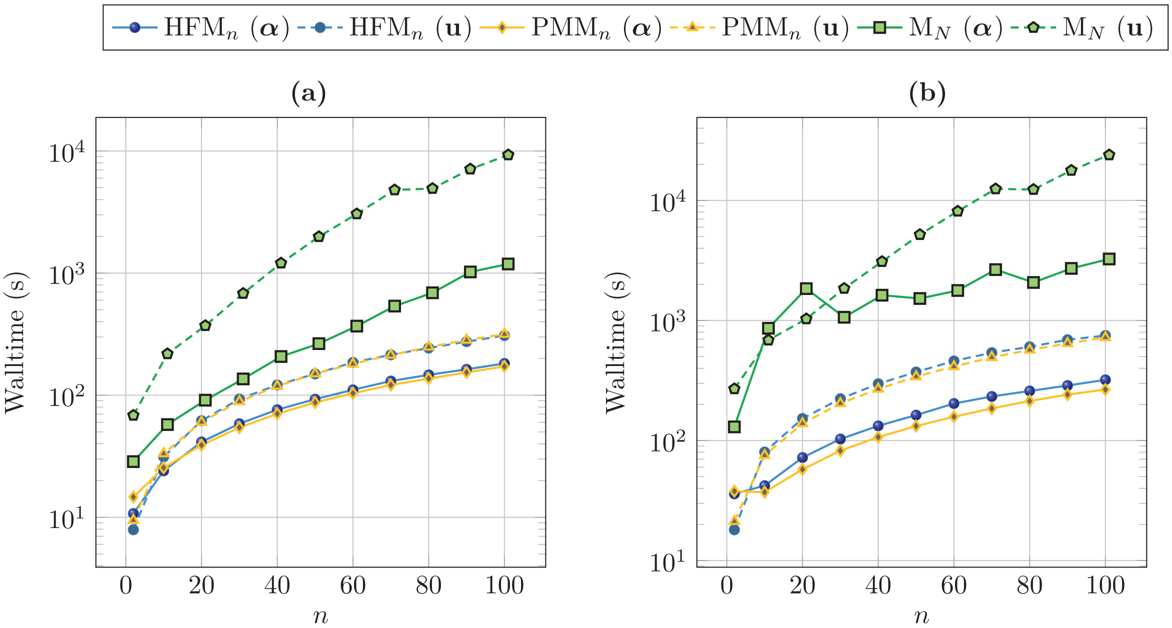

FIGURE 7. Computational times *versus* moment number $n$ for the two schemes in the one-dimensional tests ($n_x = 1200$ grid elements, $\tau = 0.001$, no parallelization). (a) Plane-source. (b) Source-beam.

distributions and ill-conditioned Hessians. For the standard scheme, we are dealing with these problems by enforcing a minimum density and using an isotropic regularization technique to replace ill-conditioned moment vectors (see Sect. 4.1.2). These techniques, in particular the isotropic regularization, introduce additional errors which in theory might completely alter the solution. In practice, regularization is usually mainly applied to moment vectors with very low densities and thus does not destroy accuracy. However, it should be noted that this is not guaranteed automatically and has to be verified for every new application of the scheme. In our case, regularization is massively used by the M$_N$ and PMM$_n$ models. The HFM$_n$ models do not use regularization.

For the new scheme, for this comparison, we did not use any regularization. Thus, it is particularly remarkable that, for the M$_N$ models, the new scheme is about as fast as the standard splitting scheme, although frequent recomputations can be observed (see Supplementary Fig. S3d) and the time steps are considerably smaller and highly varying (compare Supplementary Fig. S8).

For the PMM$_n$ and higher-order HFM$_n$ models, computational times are several times higher for the new scheme than for the standard splitting scheme (see Fig. 8c). For these models, the time steps are converging to a value well above the maximum time step of the unsplit scheme but also significantly below the time step of the splitting scheme (compare Supplementary Fig. S7). In addition, computing a time step for these models is already quite fast with the standard scheme and the speed-up of the new scheme is not large enough to compensate the smaller time step (compare Supplementary Fig. S3c). Note however that the PMM$_n$ models do not show significant oscillations or other problems though the standard scheme has to use regularization. To be competitive with respect to computation times in this test case, the new scheme probably also needs to use a splitting technique. Though there is no formal limit on the time step for the new scheme due to the strong absorption, we would expect the approximation error (and thus the time step) to be dominated by this term.

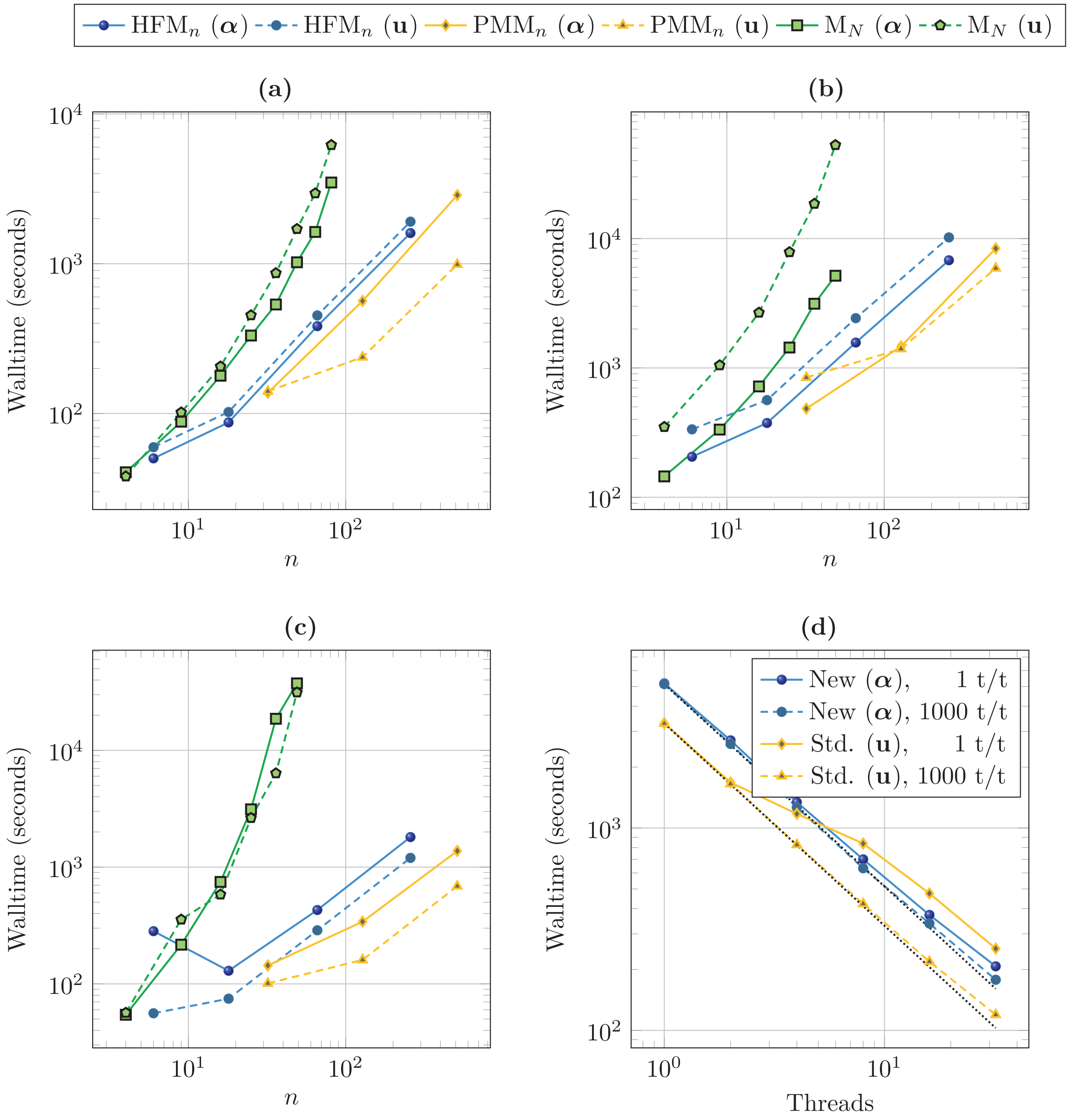

FIGURE 8. Computational times *versus* moment number $n$ for the two schemes in the three-dimensional tests ($\tau = 0.01$, 32 threads, 1000 tasks per thread) and parallel scaling. (a) Point-source problem ($n_{\mathbf{x}} = 50^3$, $t_{\text{end}} = 0.75$). The new scheme is slightly faster for the M$_N$ and HFM$_n$ models and slower for the PMM$_n$ models. (b) Checkerboard problem ($n_{\mathbf{x}} = 70^3$, $t_{\text{end}} = 3.2$). Here, the new scheme is several times faster for the M$_N$ models and slightly faster for the HFM$_n$ models. In the PMM$_n$ tests, the new model is on par or slightly faster in the low-order tests and slightly slower for PMM$_{512}$. (c) Shadow problem ($n_{\mathbf{x}} = 60 \times 20 \times 15$, $t_{\text{end}} = 20$). The new scheme is significantly slower for the HFM$_n$ and PMM$_n$ models and about as fast as the standard scheme for most M$_N$ models. (d) Computational times for 10 time steps in the point-source test ($n_{\mathbf{x}} = 100^3$, M$_3$ model) against number of threads. The dotted lines represent perfect scaling. t/t: tasks per thread.

### *5.5.3. Parallel scaling*

As already mentioned, one of the major drawbacks of the minimum-entropy-based moment models are their computational costs. As we have seen above, the new scheme often is several times faster than the scheme in standard variables. However, even with this speed-up computations without parallelization still take excessively long. In addition, there are cases where the new scheme is not faster or even slower than the standard scheme.

As the minimization problems or matrix inversions on different grid cells are independent, parallelization is easily possible for both schemes. However, for the standard scheme, load balancing may be a serious issue [34]. Usually, some minimization problems are harder to solve than others, resulting in different numbers of iterations in the Newton scheme. The new scheme does not have this problem, as it only needs the inversion of a relatively small positive definite matrix in each grid cell. For these matrix sizes, direct solvers usually perform at least as good as iterative methods and take an approximately constant time per inversion.

To investigate the scaling behaviour, for both schemes we computed ten time steps of the point-source test case with a varying number of threads. We use a work-stealing task-based parallelization (implemented using Intel `TBB` [32]). To see the impact of load balancing we perform all test both with 1 task per thread (no load balancing) and 1000 tasks per thread. The results are shown in Figure 8d. If load balancing is used, both schemes scale almost perfectly to 16 threads. When going to 32 threads the scaling is slightly worse which may be due to the used dual-socket system with 2 × 16 CPU cores.

Removing the load balancing has a large impact on the standard scheme while the new scheme is much less affected. We would expect that this difference is emphasized if even more threads (or processes) are used. The new scheme thus should be better suited to massively (MPI)-parallel computations.

### *5.5.4. Masslumping for the transformed scheme with hat function basis*

The basis functions used by the $\mathrm{HFM}_n$ models are basically the Lagrange $\mathbb{P}_1$ nodal basis functions used in the (continuous) finite element method, *i.e.*, each basis function evaluates to 1 on exactly one node of the triangulation and to 0 on all other nodes (see Sect. 2.4.2). As a consequence, the Hessian matrix (3.1) is tridiagonal (in one dimension) or sparse (three dimensions). Compared to the $\mathrm{M}_N$ models where the Hessian is dense, this significantly reduces the computational effort required for assembly and inversion. However, especially in three dimensions, assembling and inverting the Hessian matrix still account for the vast majority of the new scheme's computational time.

We can significantly speed up these computations by using a quadrature that only contains the nodes of the triangulation. With such a quadrature, the basis functions always evaluate to either zero or one and the Hessian matrix becomes diagonal. The downside is, of course, that an additional quadrature error is introduced as the nodal quadrature is only of first order. However, this additional error is of the same order as introduced by the linear finite element discretization. This approach is sometimes called masslumping as using such a quadrature diagonalizes the mass matrix in the finite element method ("all mass is lumped together on the diagonal").

**Remark 5.2.** Masslumping could also be used for the standard scheme and should lead to similar speed-ups (assuming that the masslumping does not negatively affect the number of iterations needed for the solution of the optimization problems). Since our focus in this work is on the new scheme, we did not yet test masslumping for the standard scheme.

For the one-dimensional tests, we use the two-point Gauss–Lobatto quadrature in each interval (containing only the end-points of interval) for the masslumped version. As quadrature points that are on the same vertex of the partition can be merged, we only have one quadrature point per vertex. The reference quadrature uses 24 quadrature points per interval. In addition, we only have to evaluate one component of the integrand per quadrature point (the one corresponding to the non-zero basis function) instead of two. Overall, this reduces the number of integrand evaluations by a factor of about 48. The results can be found in Figure 9. For both test cases, almost independently of the number of moments $n$, the computations are about 40 times as fast using masslumping (see Fig. 9a). This is in line with the reduction in the number of evaluations. The $L^1$ errors (compared to the non-masslumped result) are decreasing with second order (see Fig. 9b). The $L^\infty$ errors in

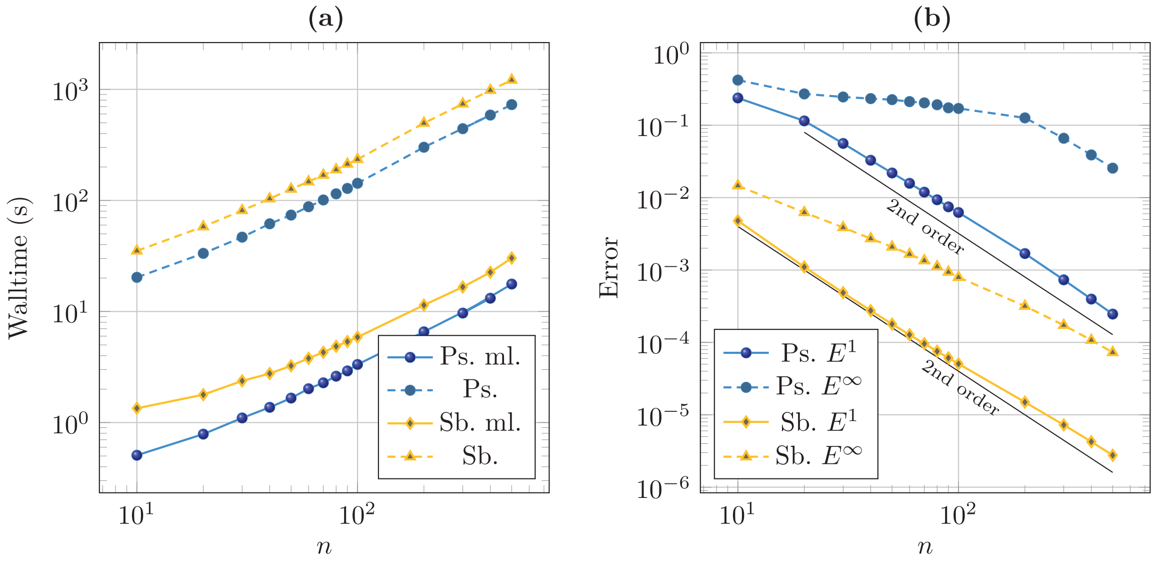

FIGURE 9. Effect of masslumping on computational times and accuracy of the HFM$_n$ models in the one-dimensional tests (1200 grid elements, $\tau = 0.001$, no parallelization). Ps: Plane-source, Sb: Source-beam, ml: masslumped. (a) Computational times *versus* moment number. (b) Errors introduced by masslumping (reference is the non-masslumped solution).

the source-beam test are decreasing with order about 1.3, while the $L^\infty$ errors in the plane-source tests are converging with very low order.

For the three-dimensional tests, we use the vertex quadrature on the reference triangle (with the vertexes $(0,0)$, $(1,0)$, $(0,1)$ as quadrature points and weight $\frac{1}{6}$ each) transferred to each spherical triangle. This results in one quadrature point per vertex of the triangulation. The triangulation consists of $2 \cdot 4^{r+1}$ triangles and $2 + 4^{r+1}$ vertices (where $r$ is the number of refinements of the initial octants, see Sect. 2.4.2), and the reference quadrature has 55 quadrature points per spherical triangle. The standard implementation thus uses about 110 times as many evaluations. The results can be found in Figure 10. For all test cases, the masslumped version is more than two orders of magnitude faster than the version using the reference quadrature. The maximum speed-up is 216 times (point-source, HFM$_{1026}$), which is considerably higher than the reduction in quadrature points. The additional speed-up is due to the more efficient implementation, as the masslumped version does not need to use sparse matrices and the associated indirect indexing. When looking at the profiler results, we see that the time for assembling and inverting the (diagonal) Hessian is negligible in the masslumped version. Overall, the time needed for the operator evaluation (which consists of calculating the kinetic fluxes and the source term and applying the inverse Hessian matrix) has been reduced to a point where the vector operations in the adaptive Runge–Kutta scheme now make up a major part of the computation time.

For all three-dimensional test cases, the errors compared to the non-masslumped version are quite large (see Fig. 10b). However, both the $L^1$ and the $L^\infty$ error converge with first order in $n$ for all test cases which corresponds to second-order convergence in the grid width of the (velocity space) triangulation, as each refinement halves the grid width but increases the number of vertices (approximately) by a factor of 4. The convergence rate thus is similar (for the checkerboard test) to or even higher (point-source, shadow) than the convergence of the (second-order discretization of the) moment approximation (compare [63]). Thus, for high-order models, the additional quadrature error introduced by masslumping might be acceptable given the massive speed-up. In any case, it might be preferable to replace a lower-order moment model with high-order quadrature by a higher-order masslumped model.

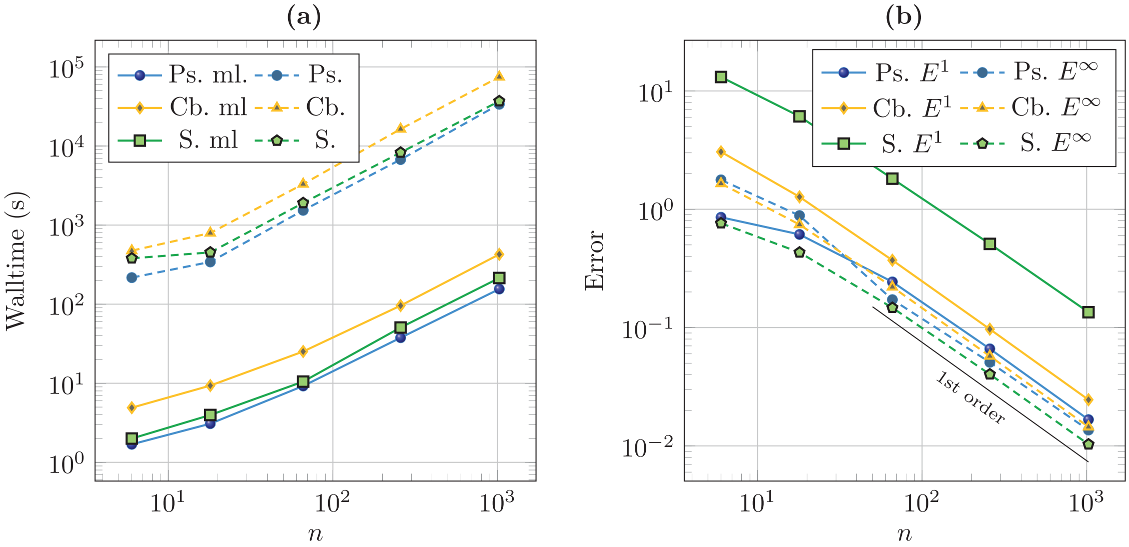

FIGURE 10. Effect of masslumping on computational times and accuracy of the HFM$_n$ models in the three-dimensional tests ($\tau = 0.01$, no parallelization). Ps: Point-source ($n_{\mathbf{x}} = 30^3$, $t_{\text{end}} = 0.75$), Cb: Checkerboard ($n_{\mathbf{x}} = 35^3$, $t_{\text{end}} = 3.2$), S: Shadow ($n_{\mathbf{x}} = 36 \times 12 \times 9$, $t_{\text{end}} = 20$), ml: masslumped. (a) Computational times *versus* moment number. (b) Errors introduced by masslumping (reference is the non-masslumped solution).

## 6. Conclusion and outlook

In this paper, we introduced a new numerical scheme for entropy-based moment equations that is based on a variable transformation of the semi-discretized equations and gets rid of the minimum-entropy optimization problems (except for the initial values). We have shown analytically and numerically that the new scheme converges to the correct solutions, and that it follow a discrete entropy law if a relaxed Runge–Kutta method is used for time stepping. In addition, we investigated the performance of the new scheme in several numerical benchmarks and showed that it is often several times faster than the untransformed scheme, at the same or even higher accuracy in time. In addition, for the hat function basis, we showed that a massive speed-up can be obtained by using a quadrature that contains only the vertices of the triangulation (at the cost of additional quadrature error), making very high-order models computable in reasonable time. Finally, we did some tests on parallel scaling of the schemes which suggest that the new scheme does not have the same load-balancing problems as the untransformed scheme.

To improve the scheme, better error estimates for the adaptive timesteppers should be investigated to get rid of the erratic time step behaviour observed in the source-beam and shadow tests. Here, larger errors could be allowed for multipliers that correspond to small densities and thus only have a minor effect on the solution in original variables. In addition, regularization techniques could be used to replace such multipliers if they limit the time step. These regularization techniques might also be needed to be able to solve problems where some Hessians are numerically singular. For applications involving strong scattering or absorption, splitting methods for the new scheme might be of interest to remove the time step restriction induced by the corresponding terms.

While we restricted ourselves to a first-order scheme, the same variable transformation can also be applied to higher-order kinetic schemes as regarded, *e.g.*, in [58, 61].

In future work, we will investigate further model reduction by POD-based reduced basis methods [31, 47] which should be much easier with the new scheme as it is well-defined on the whole $\mathbb{R}^n$ and not only on the realizable set (which is a convex cone in $\mathbb{R}^n$).

## Appendix A. Proof of Theorem 3.7

*Proof.* We want to show that the derivatives of $\boldsymbol{\alpha}_i^{\upharpoonleft}$ are bounded, *i.e.*, $\left\|\frac{\partial \boldsymbol{\alpha}_i^{\upharpoonleft}}{\partial \boldsymbol{\alpha}_j}\right\| \leq C$. Here, since all matrix norms are equivalent, $\|\cdot\|$ can be an arbitrary matrix norm. We will show that the rows of $\frac{\partial \boldsymbol{\alpha}_i^{\upharpoonleft}}{\partial \boldsymbol{\alpha}_j}$ are bounded in Euclidean norm which bounds the Frobenius matrix norm. By the definition of $\boldsymbol{\alpha}_i^{\upharpoonleft}$ (see (3.11)), we have

$$\begin{aligned}
\frac{\partial \boldsymbol{\alpha}_i^{\upharpoonleft}}{\partial \boldsymbol{\alpha}_j} &= \frac{\partial}{\partial \boldsymbol{\alpha}_j}\Big(\mathbf{H}(\boldsymbol{\alpha}_i)^{-1}\mathbf{u}_i^{\upharpoonleft}(\mathbf{u}(\boldsymbol{\alpha}_0), \dots, \mathbf{u}(\boldsymbol{\alpha}_{n_{\mathbf{x}}-1}))\Big) = \frac{\partial}{\partial \boldsymbol{\alpha}_j}\begin{pmatrix} \sum_l H_{0,l}^{-1} u_{i,l}^{\upharpoonleft} \\ \vdots \\ \sum_l H_{n-1,l}^{-1} u_{i,l}^{\upharpoonleft} \end{pmatrix} \\
&= \left(\sum_l \frac{\partial H_{k,l}^{-1}}{\partial \alpha_{jm}} u_{i,l}^{\upharpoonleft} + H_{k,l}^{-1}\frac{\partial u_{i,l}^{\upharpoonleft}}{\partial \alpha_{j,m}}\right)_{k,m=0,\dots,n-1} \\
&= \left(\frac{\partial \mathbf{H}^{-1}}{\partial \alpha_{j,0}}\mathbf{u}_i^{\upharpoonleft}, \dots, \frac{\partial \mathbf{H}^{-1}}{\partial \alpha_{j,n-1}}\mathbf{u}_i^{\upharpoonleft}\right) + \mathbf{H}^{-1}\left(\frac{\partial \mathbf{u}_i^{\upharpoonleft}}{\partial \alpha_{j,1}}, \dots, \frac{\partial \mathbf{u}_i^{\upharpoonleft}}{\partial \alpha_{j,n-1}}\right) \qquad \text{(A.1)}
\end{aligned}$$

where $H_{k,l}^{-1}$ is the $(k,l)$-th entry of $\mathbf{H}^{-1}$ and $u_{il}^{\upharpoonleft}$ and $\alpha_{jl}$ are the $l$-th component of $\mathbf{u}_i^{\upharpoonleft}$ and $\boldsymbol{\alpha}_j$, respectively.

For $i \neq j$, the first term in (A.1) vanishes. For $i = j$, since (see, *e.g.*, [24])

$$\frac{\partial \mathbf{H}^{-1}}{\partial \alpha} = -\mathbf{H}^{-1}\frac{\partial \mathbf{H}}{\partial \alpha}\mathbf{H}^{-1}, \qquad \text{(A.2)}$$

the first term in (A.1) depends on $\mathbf{H}^{-1}(\boldsymbol{\alpha}_i)$, $\mathbf{u}_i^{\upharpoonleft}$ and derivatives of $\mathbf{H}(\boldsymbol{\alpha}_i)$. The terms in $\mathbf{u}_i^{\upharpoonleft}$, except for the constant source term $\langle \mathbf{b}Q\rangle$, all scale with either $\rho(\mathbf{u}(\overline{\boldsymbol{\alpha}}_i))$ or $\rho(\mathbf{u}(\overline{\boldsymbol{\alpha}}_j))$, $j \in \mathcal{N}(i)$ (compare (2.33) and (2.14)). By (3.14b), $\mathbf{u}_i^{\upharpoonleft}$ is thus bounded. By (3.14a), after multiplication with $\mathbf{H}^{-1}$ the term is still bounded (as $\left\|\mathbf{H}^{-1}\right\| = \|\mathbf{H}\|^{-1}$). For the first term of (A.1), it remains to show that $\left\|\frac{\partial \mathbf{H}(\boldsymbol{\alpha}_i)}{\partial \alpha_{il}}\right\|$ is bounded for $l \in \{0, \dots, n-1\}$. We have

$$\begin{aligned}
\mathbf{w}^T\frac{\partial \mathbf{H}(\boldsymbol{\alpha}_i)}{\partial \alpha_{il}}\mathbf{w} &= \mathbf{w}^T\frac{\partial}{\partial \alpha_{il}}\langle \mathbf{b}\mathbf{b}^T\eta_*''(\boldsymbol{\alpha}_i \cdot \mathbf{b})\rangle\mathbf{w} = \Big\langle (\mathbf{b}\cdot\mathbf{w})^2 b_l \eta_*'''(\boldsymbol{\alpha}_i \cdot \mathbf{b})\Big\rangle \\
&\leq \left(\max_{\boldsymbol{\Omega}\in V}|b_l(\boldsymbol{\Omega})|\|\mathbf{b}(\boldsymbol{\Omega})\|^2\right)\|\mathbf{w}\|^2\langle \eta_*'''(\boldsymbol{\alpha}_i \cdot \mathbf{b})\rangle.
\end{aligned}$$

Thus, by (3.14d), $\left\|\frac{\partial \mathbf{H}(\boldsymbol{\alpha}_i)}{\partial \alpha_{il}}\right\|$ is bounded.

To show that the second summand in (A.1) is bounded, by (3.14a), we only have to show boundedness of $\frac{\partial \mathbf{u}_i^{\upharpoonleft}}{\partial \alpha_{jl}}$ for $l \in \{0, \dots, n-1\}$. We have (compare (2.33))

$$\frac{\partial \mathbf{u}_i^{\upharpoonleft}}{\partial \alpha_{jl}} = \frac{\partial}{\partial \alpha_{jl}}\left(\sum_{k\in\mathcal{N}(i)}\frac{|S_{ik}|}{|T_i|}\left(\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ik})\eta_*'(\boldsymbol{\alpha}_i^{\kappa}\cdot\mathbf{b})\mathbf{b}\rangle_+ + \langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ik})\eta_*'(\boldsymbol{\alpha}_k^{\kappa}\cdot\mathbf{b})\mathbf{b}\rangle_-\right) - \mathbf{s}(\mathbf{x}_i, \mathbf{u}_i^{\kappa})\right). \qquad \text{(A.3)}$$

Since $\mathbf{s}$ is linear in $\mathbf{u}$, and $\frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}\boldsymbol{\alpha}_j} = \mathbf{H}(\mathbf{u}_i)\delta_{ij}$ (compare (3.1)), the term in (A.3) involving $\mathbf{s}$ results in columns of the Hessian (multiplied by some constant matrices, see (2.14)) and thus is bounded by (3.14a). Computing the derivative of the flux terms in (A.3), we obtain terms like

$$\frac{\partial}{\partial \alpha_{jl}}\langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ik})\eta_*'(\boldsymbol{\alpha}_i^{\kappa}\cdot\mathbf{b})\mathbf{b}\rangle_+ = \langle(\boldsymbol{\Omega}\cdot\mathbf{n}_{ik})\eta_*''(\boldsymbol{\alpha}_i^{\kappa}\cdot\mathbf{b})b_l\delta_{il}\mathbf{b}\rangle_+$$

which are bounded by (3.14c). □

*Acknowledgements.* The authors thank Florian Schneider and Hendrik Ranocha for fruitful discussions.
The authors acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044–390685587, Mathematics Münster: Dynamics-Geometry-Structure.

## References

[1] G. Alldredge and F. Schneider, A realizability-preserving discontinuous Galerkin scheme for entropy-based moment closures for linear kinetic equations in one space dimension. *J. Comput. Phys.* **295** (2015) 665–684.

[2] G.W. Alldredge, C.D. Hauck and A.L. Tits, High-order entropy-based closures for linear transport in slab geometry II: A computational study of the optimization problem. *SIAM J. Sci. Comput.* **34** (2012) B361–B391.

[3] G.W. Alldredge, C.D. Hauck, D.P. O'Leary and A.L. Tits, Adaptive change of basis in entropy-based moment closures for linear kinetic equations. *J. Comput. Phys.* **258** (2014) 489–508.

[4] G.W. Alldredge, M. Frank and C.D. Hauck, A regularized entropy-based moment method for kinetic equations. *SIAM J. Appl. Math.* **79** (2019) 1627–1653.

[5] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen, LAPACK Users' Guide, 3rd edition. Society for Industrial and Applied Mathematics. Philadelphia, PA (1999).

[6] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger and O. Sander, A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE. *Computing* **82** (2008) 121–138.

[7] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger and O. Sander, A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework. *Computing* **82** (2008) 103–119.

[8] P. Bogacki and L.F. Shampine, A 3(2) pair of Runge–Kutta formulas. *Appl. Math. Lett.* **2** (1989) 321–325.

[9] L. Boltzmann, Weitere studien über das wärmegleichgewicht unter gasmolekülen. Sitzungsberichte der Akademie der Wissenschaften. *Mathematische-Naturwissenschaftliche Klasse* **66** (1872) 275–370.

[10] R. Borsche, A. Klar and F. Schneider, Kinetic and moment models for cell motion in fiber structures. In: Vol. 2 of Active Particles. *Model. Simul. Sci. Eng. Technol.* Birkhäuser/Springer, Cham (2019) 1–38.

[11] T.A. Brunner and J.P. Holloway, Two-dimensional time dependent riemann solvers for neutron transport. *J. Comput. Phys.* **210** (2005) 386–399.

[12] S.R. Buss and J.P. Fillmore, Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graphics* **20** (2001) 95–126.

[13] C. Cercignani, The Boltzmann Equation and its Applications. Vol. 67 of *Applied Mathematical Sciences*. Springer New York, New York, NY (1988).

[14] P. Chidyagwai, M. Frank, F. Schneider and B. Seibold, A comparative study of limiting strategies in discontinuous Galerkin schemes for the $M_1$ model of radiation transport. *J. Comput. Appl. Math.* **342** (2018) 399–418.

[15] S. Conde, I. Fekete and J.N. Shadid, Embedded error estimation and adaptive step-size control for optimal explicit strong stability preserving Runge–Kutta methods. Preprint arXiv:1806.08693 (2018).

[16] R.E. Curto and L.A. Fialkow, Recursiveness, positivity, and truncated moment problems. *Houston J. Math.* **17** (1991) 603–635.

[17] J.R. Dormand and P.J. Prince, A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6** (1980) 19–26.

[18] B. Dubroca and J.-L. Feugeas, Entropic moment closure hierarchy for the radiative transfer equation. *C. R. Acad. Sci. Paris Ser. I* **329** (1999) 915–920.

[19] B. Dubroca and A. Klar, Half-moment closure for radiative transfer equations. *J. Comput. Phys.* **180** (2002) 584–596.

[20] M. Frank, B. Dubroca and A. Klar, Partial moment entropy approximation to radiative heat transfer. *J. Comput. Phys.* **218** (2006) 1–18.

[21] K.O. Friedrichs and P.D. Lax, Systems of conservation equations with a convex extension. *Proc. Nat. Acad. Sci. USA* **68** (1971) 1686–1688.

[22] B.D. Ganapol, R.S. Baker, J.A. Dahl and R.E. Alcouffe, Homogeneous infinite media time-dependent analytical benchmarks. Tech. Rep. LA-UR-01-1854. Los Alamos National Laboratory (2001).

[23] C.K. Garrett and C.D. Hauck, A comparison of moment closures for linear kinetic transport equations: the line source benchmark. *Transp. Theory Stat. Phys.* **42** (2013) 203–235.

[24] M.B. Giles, Collected matrix derivative results for forward and reverse mode algorithmic differentiation. In: Advances in Automatic Differentiation, editrd by C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann and J. Utke. Springer Berlin Heidelberg, Berlin, Heidelberg (2008) 35–44.

[25] S. Gottlieb, On High Order Strong Stability Preserving Runge–Kutta and multi step time discretizations. *J. Sci. Comput.* **25** (2005) 105–128.

[26] G. Guennebaud, B. Jacob, et al., Eigen v3 (2010). https://eigen.tuxfamily.org.

[27] K.P. Hadeler, Reaction transport systems in biological modelling. In: Mathematics Inspired by Biology, edited by V. Capasso. *Lecture Notes in Mathematics*. Springer Berlin Heidelberg (1999).

[28] E. Hairer, G. Wanner and S.P. Nørsett, Solving Ordinary Differential Equations I: Nonstiff Problems. Vol. 8 of *Springer Series in Computational Mathematics*, 2nd revised edition. Springer-Verlag Berlin Heidelberg (1993).

[29] C.D. Hauck, High-order entropy-based closures for linear transport in slab geometry. *Commun. Math. Sci.* **9** (2010) 187–205.

[30] T. Hillen and K.J. Painter, Transport and anisotropic diffusion models for movement in oriented habitats. In: Dispersal, Individual Movement and Spatial Ecology, edited by M.A. Lewis, P.K. Maini and S.V. Petrovskii. Vol. 2071 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 177–222.

[31] C. Himpe, T. Leibner and S. Rave, Hierarchical approximate proper orthogonal decomposition. *SIAM J. Sci. Comput.* **40** (2018) A3267–A3292.

[32] Intel, Threading building blocks (2020). https://software.intel.com/en-us/tbb.

[33] M. Junk, Maximum entropy for reduced moment problems. *Math. Models Methods Appl. Sci.* **10** (2000) 1001–1025.

[34] C. Kristopher Garrett, C. Hauck and J. Hill, Optimization and large scale computation of an entropy-based moment closure. *J. Comput. Phys.* **302** (2015) 573–590.

[35] K. Lanckau, Cercignani, c., the Boltzmann equation and its applications. *ZAMM – J. Appl. Math. Mech./Z. Ang. Math. Mech.* **69** (1989) 423.

[36] T. Langer, A. Belyaev and H.-P. Seidel, Spherical barycentric coordinates. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing (2006) 81–88.

[37] T. Leibner and M. Ohlberger, Replication data for: a new entropy-variable-based discretization method for minimum entropy moment approximations of linear kinetic equations (2021). https://doi.org/10.7910/DVN/7XTREY.

[38] C.D. Levermore, Moment closure hierarchies for kinetic theories. *J. Stat. Phys.* **83** (1996) 1021–1065.

[39] E.E. Lewis and W.F. Miller, Jr., Computational Methods in Neutron Transport. John Wiley and Sons, New York (1984).

[40] P.A. Markowich, C.A. Ringhofer and C. Schmeiser, Semiconductor Equations. Springer-Verlag, Vienna (1990).

[41] D. Mihalas and B. Weibel-Mihalas, Foundations of Radiation Hydrodynamics. Dover (1999).

[42] R. Milk, F. Schindler and T. Leibner, dune-xt (2017). http://github.com/dune-community/dune-xt-super.

[43] R. Milk, F. Schindler and T. Leibner, Extending dune: the dune-xt modules. *Arch. Numer. Softw.* **5** (2017) 193–216.

[44] G. Minerbo, Ment: a maximum entropy algorithm for reconstructing a source from projection data. *Comput. Graphics Image Process.* **10** (1979) 48–68.

[45] G.N. Minerbo, Maximum entropy Eddington factors. *J. Quant. Spectrosc. Radiat. Trans.* **20** (1978) 541–545.

[46] E. Olbrant, C.D. Hauck and M. Frank, A realizability-preserving discontinuous Galerkin method for the M1 model of radiative transfer, *J. Comput. Phys.* **231** (2012) 5612–5639.

[47] A. Quarteroni, A. Manzoni and F. Negri, Reduced Basis Methods For Partial Differential Equations. Vol. 92 of *Unitext. An Introduction, La Matematica per il 3+2.* Springer, Cham (2016).

[48] H. Ranocha, M. Sayyari, L. Dalcin, M. Parsani and D.I. Ketcheson, Relaxation Runge–Kutta methods: fully discrete explicit entropy-stable schemes for the compressible Euler and Navier–Stokes equations. *SIAM J. Sci. Comput.* **42** (2020) A612–A638.

[49] J. Ritter, A. Klar and F. Schneider, Partial-moment minimum-entropy models for kinetic chemotaxis equations in one and two dimensions. *J. Comput. Appl. Math.* **306** (2016) 300–315.

[50] R.T. Rockafellar, Convex Analysis. Princeton University Press (1970).

[51] R.M. Rustamov, Barycentric coordinates on surfaces. *Eurographics Symp. Geom. Process.* **29** (2010) 1507–1516.

[52] R.P. Schaerer, P. Bansal and M. Torrilhon, Efficient algorithms and implementations of entropy-based moment closures for rarefied gases. *J. Comput. Phys.* **340** (2017) 138–159.

[53] C. Schär and P.K. Smolarkiewicz, A synchronous and iterative flux-correction formalism for coupled transport equations. *J. Comput. Phys.* **128** (1996) 101–120.

[54] F. Schindler, dune-gdt (2017). http://github.com/dune-community/dune-gdt.

[55] F. Schneider, First-order quarter- and mixed-moment realizability theory and Kershaw closures for a Fokker–Planck equation in two space dimensions: Code (2016). DOI: 10.5281/zenodo.48753.

[56] F. Schneider, Implicit-explicit, realizability-preserving first-order scheme for moment models with lipschitz-continuous source terms. Preprint arXiv:1611.01314 (2016).

[57] F. Schneider, Kershaw closures for linear transport equations in slab geometry II: high-order realizability-preserving discontinuous-Galerkin schemes. *J. Comput. Phys.* **322** (2016) 920–935.

[58] F. Schneider, Moment Models in Radiation Transport Equations. Verlag Dr. Hut (2016).

[59] F. Schneider and T. Leibner, First-order continuous- and discontinuous-galerkin moment models for a linear kinetic equation: model derivation and realizability theory. *J. Comput. Phys.* **416** (2020) 109547.

[60] F. Schneider, G.W. Alldredge, M. Frank and A. Klar, Higher order mixed-moment approximations for the Fokker–Planck equation in one space dimension. *SIAM J. Appl. Math.* **74** (2014) 1087–1114.

[61] F. Schneider, G.W. Alldredge and J. Kall, A realizability-preserving high-order kinetic scheme using weno reconstruction for entropy-based moment closures of linear kinetic equations in slab geometry. *Kinet. Relat. Models* **9** (2016) 193.

[62] F. Schneider, A. Roth and J. Kall, First-order quarter- and mixed-moment realizability theory and Kershaw closures for a Fokker–Planck equation in two space dimensions. *Kinet. Relat. Models* **10** (2017) 1127–1161.

[63] F. Schneider and T. Leibner, First-order continuous- and discontinuous-galerkin moment models for a linear kinetic equation: realizability-preserving splitting scheme and numerical analysis. Preprint arXiv:1904.03098 (2019).

[64] B. Seibold and M. Frank, StaRMAP – a second order staggered grid method for spherical harmonics moment equations of radiative transfer. *ACM Trans. Math. Softw.* **41** (2014) 1–28.

[65] X. Zhang and C.W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *J. Comput. Phys.* **229** (2010) 8918–8934.

**Subscribe to Open (S2O)**

A fair and sustainable open access model

This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

**Please help to maintain this journal in open access!**

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: https://www.edpsciences.org/en/maths-s2o-programme