

NUMERICAL SOLVER FOR THE BOLTZMANN EQUATION WITH SELF-ADAPTIVE COLLISION OPERATORS

ZHENNING CAI AND YANLI WANG

ABSTRACT. We use the Burnett spectral method to solve the Boltzmann equation whose collision term is modeled by separate treatments for the low-frequency part and high-frequency part of the solution. For the low-frequency part representing the sketch of the distribution function, the binary collision is applied, while for the high-frequency part representing the finer details, the BGK approximation is applied. The parameter controlling the ratio of the high-frequency part and the low-frequency part is selected adaptively on every grid cell at every time step. This self-adaptation is based on an error indicator describing the difference between the model collision term and the original binary collision term. The indicator is derived by controlling the quadratic terms in the modeling error with linear operators. Our numerical experiments show that such an error indicator is effective and computationally affordable.

Keywords: Boltzmann equation, Burnett spectral method, self-adaptation

1. INTRODUCTION

Due to the extensive applications of rarefied gas dynamics in a number of engineering fields, including the manufacturing of spacecrafts and micro-electro-mechanical systems, the numerical simulation of gas kinetic theory is under active research in recent years. In the kinetic theory, the fluid state is described using the distribution function $f(\mathbf{x}, \mathbf{v}, t)$, where t is the time, and \mathbf{x} and \mathbf{v} represent the spatial coordinates and the velocity of gas molecules, respectively. The distribution function represents the number density of gas molecules in the joint position-velocity space. In this paper, we consider the Boltzmann equation:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = Q[f, f],$$

where $Q[f, f]$ is the binary collision term defined by:

$$Q[f, g](\mathbf{v}) = \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbf{n} \perp \mathbf{g}} \int_0^\pi B(|\mathbf{g}|, \chi) [f(\mathbf{v}')g(\mathbf{v}'_*) + f(\mathbf{v}'_*)g(\mathbf{v}') - f(\mathbf{v})g(\mathbf{v}_*) - f(\mathbf{v}_*)g(\mathbf{v})] d\chi d\mathbf{n} d\mathbf{v}_*.$$

In the equation above, the relative velocity \mathbf{g} is defined by $\mathbf{g} = \mathbf{v} - \mathbf{v}_*$, and the post-collisional velocities \mathbf{v}' and \mathbf{v}'_* are given by

$$\begin{aligned} \mathbf{v}' &= \cos^2(\chi/2)\mathbf{v} + \sin^2(\chi/2)\mathbf{v}_* - |\mathbf{g}| \sin(\chi/2) \cos(\chi/2)\mathbf{n}, \\ \mathbf{v}'_* &= \cos^2(\chi/2)\mathbf{v}_* + \sin^2(\chi/2)\mathbf{v} + |\mathbf{g}| \sin(\chi/2) \cos(\chi/2)\mathbf{n}. \end{aligned}$$

Note that here \mathbf{n} is a unit vector in \mathbb{S}^2 , which implies that the integral with respect to \mathbf{n} is a one-dimensional integral over a circle perpendicular to \mathbf{g} . The non-negative function $B(\cdot, \cdot)$ is the collision kernel determined by the mutual force between gas molecules.

One of the numerical difficulties in the discretization of the Boltzmann equation lies in the high-dimensional integral form of $Q[f, f]$. To compute the collision term efficiently, the velocity variable in the distribution function is usually discretized by high-order schemes such as the spectral methods [37, 5] and discontinuous Galerkin methods [2], so that the number of degrees of freedom can be reduced. In the literature, the spectral methods mainly include the Fourier spectral method [5, 16, 20] based on the periodization of the velocity variable and the Hermite/Burnett spectral method based on the unbounded velocity domain [21, 29]. The Fourier spectral method provides a significant improvement in computational efficiency [15, 27], and the recent development of the Hermite/Burnett spectral method shows its advantage due to its connection with

Key words and phrases. Boltzmann equation, Burnett spectral method, self-adaptation.

Zhenning Cai's work was supported by the Academic Research Fund of the Ministry of Education of Singapore under Grant No. R-146-000-305-114. The work of Yanli Wang is partially supported by Science Challenge Project (No. TZ2016002) and the National Natural Science Foundation of China (Grant No. U1930402 and 12031013).

modelling in the gas kinetic theory. Specifically, the Hermite/Burnett spectral method can be linked to the moment method since the coefficients in the spectral expansion are actually the moments of the distribution functions. Such a property has been applied to derive the regularized 13-moment equations in [9], and inversely, some modelling techniques can therefore also be applied to the spectral methods. In [7], by taking the idea of the Shakhov operator [38], the authors divided all moments of the distribution function into two sets, with one set including low-order moments describing the sketch of the distribution function, and the other including high-order moments providing the details. For the set with low-order moments, the linearized collision operator is applied, while for the set with high-order moments, a simple decay towards the equilibrium is used as an approximation. This hybrid approach is later extended to quadratic collision operators in [40, 6]. One parameter in this hybrid approach is the critical order M_0 that defines the “low-order” and “high-order” moments. In this paper, we will focus on the selection of this parameter in the spatially inhomogeneous Boltzmann equation.

Since the parameter M_0 defines the modeling accuracy, it is expected that the choice of M_0 should depend on the “modeling error” given by some differences between the current collision model and the exact binary collision model when applied to the current distribution function. Once such an error indicator is obtained, we can change the value of M_0 dynamically during our simulation. However, the construction of such an error indicator is far from trivial due to the following reasons:

- (1) Unlike the *a posteriori* error estimation in the finite element methods, we do not have an equation to define the “residual” as an error indicator.
- (2) The collision operator is generally unbounded, so that even an *a priori* error estimation is non-trivial.
- (3) Another common technique by comparing the current model and a more accurate model with larger M_0 is not applicable here due to the rapid growth of the computational cost with respect to M_0 (usually M_0^8).

Because of these difficulties, we have to look for non-standard techniques to quantify the error. Since a rigorous and numerically affordable error bound is difficult to find, as an initial study, the goal of this paper is to establish an error indicator with low computational cost compared to the collision term. With this error indicator, we are able to choose this modeling parameter M_0 adaptively on each spatial grid cell at each time step with the purpose to reduce the computational time on the collision terms. Due to the high computational complexity with respect to M_0 , reducing M_0 can effectively save the computational cost.

This work contributes to the adaptive methods for the Boltzmann equation. In the literature, the self-adaptive methods have been applied to both spatial discretization and velocity discretization [30, 11, 3, 1], which can effectively reduce the degrees of freedom in the simulation. There have been also many works coupling the kinetic equations and fluid equations, so that the cheaper Navier-Stokes equations or Euler equations can be solved where the fluid is close to its local equilibrium [13, 12, 18], and many criteria have been proposed to predict the breakdown of fluid equations [34, 39, 19]. While the method in this work does not change the number of variables, we consider the modeling adaptivity, which changes the complexity of the collision model. We hope that this work can provide a new perspective for the simulation of Boltzmann equations, which might also be applicable in other related areas.

In the rest of this paper, we will first review the Burnett spectral method introduced in [24] (Section 2), and then in Section 3, we will detail the derivation of the error indicator, and the general structure of our numerical algorithm will also be presented. One- and two-dimensional numerical experiments showing the efficiency of the self-adaptive method will be given in Section 4, and the paper is concluded by a brief summary in Section 5.

2. BURNETT SPECTRAL METHOD FOR THE BOLTZMANN EQUATION

The Burnett spectral method is based on the following expansion of the distribution function:

$$(1) \quad f(\mathbf{x}, \mathbf{v}, t) = \sum_{l=0}^{+\infty} \sum_{m=-l}^l \sum_{n=0}^{+\infty} f_{lmn}(\mathbf{x}, t) \varphi_{lmn}(\mathbf{v}),$$

where $\varphi_{lmn}(\mathbf{v}) = \varphi_{lmn}^0(\mathbf{v} - \bar{\mathbf{u}})$, and

$$\varphi_{lmn}^0(\mathbf{v}) = \sqrt{\frac{2^{1-l}\pi^{3/2}n!}{\Gamma(n+l+3/2)}} L_n^{(l+1/2)}\left(\frac{|\mathbf{v}|^2}{2\bar{\theta}}\right) \left(\frac{|\mathbf{v}|}{\sqrt{\bar{\theta}}}\right)^l Y_l^m\left(\frac{\mathbf{v}}{|\mathbf{v}|}\right) \cdot \frac{1}{(2\pi\bar{\theta})^{3/2}} \exp\left(-\frac{|\mathbf{v}|^2}{2\bar{\theta}}\right),$$

$$l, n = 0, 1, \dots, \quad m = -l, \dots, l,$$

with $L_n^{(\alpha)}(\cdot)$ being the Laguerre polynomials and $Y_l^m(\cdot)$ being the spherical harmonics. The parameters $\bar{\mathbf{u}}$ and $\bar{\theta}$ are chosen to specify the center and the scaling of the basis functions. Note that φ_{lmn} is the product of a Gaussian and a polynomial of degree $l + 2n$. When we truncate the series (1) in our numerical method, we select a positive integer M as the upper bound of the polynomial, and preserve only the terms with $l + 2n \leq M$. Thus, the truncated series reads

$$(2) \quad f_M(\mathbf{x}, \mathbf{v}, t) = \sum_{l=0}^M \sum_{m=-l}^l \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} f_{lmn}(\mathbf{x}, t) \varphi_{lmn}(\mathbf{v}).$$

In this expansion, the basis functions φ_{lmn} satisfy the orthogonality

$$(3) \quad \langle \varphi_{lmn}(\mathbf{v}), \varphi_{l'm'n'}(\mathbf{v}) \rangle_\omega := \int_{\mathbb{R}^3} \varphi_{lmn}^\dagger(\mathbf{v}) \varphi_{l'm'n'}(\mathbf{v}) \omega(\mathbf{v}) \, d\mathbf{v} = \delta_{ll'} \delta_{mm'} \delta_{nn'},$$

where \dagger refers to the complex conjugate, and the weight function is

$$\omega(\mathbf{v}) = \left[\frac{1}{(2\pi\bar{\theta})^{3/2}} \exp\left(-\frac{|\mathbf{v} - \bar{\mathbf{u}}|^2}{2\bar{\theta}}\right) \right]^{-1}.$$

Note that here $\omega(\mathbf{v})$ is the reciprocal of a global Maxwellian, while in what follows, we use the term ‘‘local Maxwellian’’ to refer to the Maxwellian $\mathcal{M}(\mathbf{v}) = \exp(\alpha + \boldsymbol{\beta} \cdot \mathbf{v} + \gamma|\mathbf{v}|^2)$ associated with a distribution function $f(\mathbf{v})$ by

$$(4) \quad \int_{\mathbb{R}^3} \begin{pmatrix} 1 \\ \mathbf{v} \\ |\mathbf{v}|^2 \end{pmatrix} \mathcal{M}(\mathbf{v}) \, d\mathbf{v} = \int_{\mathbb{R}^3} \begin{pmatrix} 1 \\ \mathbf{v} \\ |\mathbf{v}|^2 \end{pmatrix} f(\mathbf{v}) \, d\mathbf{v}.$$

Let $\mathbf{f}(\mathbf{x}, t)$ be the vector including all the coefficients $f_{lmn}(\mathbf{x}, t)$ with $l + 2n \leq M$, and define $\boldsymbol{\varphi}$ as the vector including all the basis function $\varphi_{lmn}(\mathbf{v})$ also with $l + 2n \leq M$ arranged in the same order as \mathbf{f} . Then

$$f_M(\mathbf{x}, \mathbf{v}, t) = [\mathbf{f}(\mathbf{x}, t)]^T \boldsymbol{\varphi}(\mathbf{v}).$$

With the Petrov-Galerkin method [21, 24] based on the orthogonality (3), the semi-discrete Boltzmann equation has the form

$$(5) \quad \frac{\partial \mathbf{f}}{\partial t} + \sum_{k=1}^3 \mathbf{A}_k \frac{\partial \mathbf{f}}{\partial x_k} = \mathbf{Q} : (\mathbf{f} \otimes \mathbf{f}).$$

Here \mathbf{A}_k , $k = 1, 2, 3$ are sparse matrices coming from the discretization of the advection term, and \mathbf{Q} is a 3-tensor representing the discrete collision kernel. Since \mathbf{f} has $O(M^3)$ components, the tensor \mathbf{Q} has $O(M^9)$ elements, where only $O(M^8)$ elements are nonzero due to the rotational invariance of the collision operator [6]. Despite this sparsity of \mathbf{Q} , the computational cost grows quickly as M increases. To reduce the computational cost, in [6, 24], the authors chose $M_0 < M$ and split the discrete distribution function into two parts $f_M = f^{(1)} + f^{(2)}$, where

$$(6) \quad f^{(1)}(\mathbf{x}, \mathbf{v}, t) = \sum_{l=0}^{M_0} \sum_{m=-l}^l \sum_{n=0}^{\lfloor (M_0-l)/2 \rfloor} f_{lmn}(\mathbf{x}, t) \varphi_{lmn}(\mathbf{v}),$$

$$f^{(2)}(\mathbf{x}, \mathbf{v}, t) = \sum_{l=0}^M \sum_{m=-l}^l \sum_{n=\max(0, \lfloor (M-l)/2 \rfloor + 1)}^{\lfloor (M-l)/2 \rfloor} f_{lmn}(\mathbf{x}, t) \varphi_{lmn}(\mathbf{v}).$$

Due to the high efficiency of the spectral approximation, we expect that by choosing $M_0 < M$, the first part $f^{(1)}$ can capture the sketch of distribution function f , while $f^{(2)}$ provides more details of its profile. For

simplicity, we write \mathbf{f} as

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}^{(1)} \\ \mathbf{f}^{(2)} \end{pmatrix},$$

where $\mathbf{f}^{(1)}$ includes all the coefficients in the expansion of $f^{(1)}$ and $\mathbf{f}^{(2)}$ includes all the coefficients in the expansion of $f^{(2)}$. Similarly, we let \mathbf{M} denote the coefficients in the truncated expansion of the local Maxwellian defined by (4), and use $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$ to denote its subvectors. Then using the idea of the BGK and Shakhov collision operators, the original collision term $\mathbf{Q} : (\mathbf{f} \otimes \mathbf{f})$ can be approximated by

$$(7) \quad \begin{pmatrix} \mathbf{Q}_{M_0} : (\mathbf{f}^{(1)} \otimes \mathbf{f}^{(1)}) \\ \nu_{M_0} (\mathbf{M}^{(2)} - \mathbf{f}^{(2)}) \end{pmatrix}.$$

Here \mathbf{Q}_{M_0} is the discrete collision kernel for $M = M_0$. The approximation (7) applies the accurate binary collision operator to the sketch of the distribution function, while for the part representing the finer details, a simpler BGK-like expression is used instead. Such an idea can be found in [10] as a generalization of the classical BGK model. It was later realized for the linearized Boltzmann collision operator in [7], where the authors proved that for the linearized collision term, our BGK-like operator converges to the original operator in the resolvent sense as $M_0 \rightarrow +\infty$. The generalization to quadratic collision operators was first introduced in [40], where the choice of the parameter ν_{M_0} was chosen to be the spectral radius of the truncated linearized collision operator following the approach in [7]. Here we adopt the same choice of ν_{M_0} , and the details are given in the Appendix A.

To preserve Maxwellian in the homogeneous Boltzmann equation, in [24], the approximate collision term (7) is supplemented by adding a close-to-zero term so that the semi-discrete equation reads

$$(8) \quad \frac{\partial \mathbf{f}}{\partial t} + \sum_{k=1}^3 \mathbf{A}_k \frac{\partial \mathbf{f}}{\partial x_k} = \tilde{\mathbf{Q}}(M_0; \mathbf{f}) := \begin{pmatrix} \mathbf{Q}_{M_0} : (\mathbf{f}^{(1)} \otimes \mathbf{f}^{(1)} - \mathbf{M}^{(1)} \otimes \mathbf{M}^{(1)}) \\ \nu_{M_0} (\mathbf{M}^{(2)} - \mathbf{f}^{(2)}) \end{pmatrix}.$$

Here $\mathbf{M}^{(1)}$ plays the same role as $\mathbf{f}^{(1)}$ and provides the sketch of the local Maxwellian. Since any Maxwellian is a smooth function, we again expect that $\mathbf{M}^{(1)}$ can well capture the general structure of the Maxwellian with a moderate value of M_0 . Thus the corresponding collision term $\mathbf{Q}_{M_0} : (\mathbf{M}^{(1)} \otimes \mathbf{M}^{(1)})$ is likely to be close to zero. Such a discrete collision term ensures that it vanishes if $\mathbf{f}^{(1)} = \mathbf{M}^{(1)}$ and $\mathbf{f}^{(2)} = \mathbf{M}^{(2)}$. The idea of this approach comes from the steady-state preserving method introduced in [17], which uses an equivalent form

$$\mathbf{Q}_{M_0} : (\mathbf{f}^{(1)} \otimes \mathbf{f}^{(1)} - \mathbf{M}^{(1)} \otimes \mathbf{M}^{(1)}) = \mathbf{Q}_{M_0} : (\mathbf{g}^{(1)} \otimes \mathbf{g}^{(1)} + \mathbf{g}^{(1)} \otimes \mathbf{M}^{(1)} + \mathbf{M}^{(1)} \otimes \mathbf{g}^{(1)}),$$

where $\mathbf{g}^{(1)} = \mathbf{f}^{(1)} - \mathbf{M}^{(1)}$ denotes the non-equilibrium part of the distribution function. Such splitting also borrows ideas from the method of micro-macro decomposition to develop asymptotic preserving schemes [4].

The idea of hybridizing expensive and cheap models has been tested in a number of previous works [7, 8, 40, 25, 6, 24]. However, the choice of M_0 remains to be problem-dependent, and currently its determination can only be based on trial-and-error approaches. In this work, we would like to determine M_0 based on an error estimate, and different M_0 will be used on different spatial grids.

3. ERROR INDICATOR FOR THE ADAPTIVE COLLISION OPERATOR

According to the discussion in the previous section, given \mathbf{f} defined on any spatial grid cell, our purpose is to choose appropriate M_0 such that the right-hand side of (8) is a good approximation of the collision term. Since the collision operator is defined locally in both time and space, we expect that the choice M_0 on any cell depends only on the distribution function defined thereon. Hence, we will omit the arguments \mathbf{x} and t in the following discussion. Also, we assume that the distribution function f has been normalized such that its integral equals 1. The purpose of this normalization is to provide the bounds for the relative error. To begin with, we will introduce some notations and assumptions for the sake of convenience.

3.1. Notations and hypotheses. Let \mathcal{T}_M be truncation operator that cut off the series defined in (1) by discarding all the terms with polynomials of degree greater than M . Therefore

$$\mathcal{T}_M f = f_M$$

with f_M given in (2). Then in the original spectral method (5), the right-hand side represents the coefficients in the expansion of $\mathcal{T}_M Q[f_M, f_M]$. In other words, we have

$$(9) \quad [\mathbf{Q} : (\mathbf{f} \otimes \mathbf{f})]^T \boldsymbol{\varphi} = \mathcal{T}_M Q[f_M, f_M].$$

Similarly, for the right-hand side of (8), we have

$$(10) \quad \begin{pmatrix} \mathbf{Q}_{M_0} : (\mathbf{f}^{(1)} \otimes \mathbf{f}^{(1)} - \mathbf{M}^{(1)} \otimes \mathbf{M}^{(1)}) \\ \nu_{M_0}(\mathbf{M}^{(2)} - \mathbf{f}^{(2)}) \end{pmatrix}^T \boldsymbol{\varphi} = \mathcal{T}_{M_0} \left(Q[f^{(1)}, f^{(1)}] - Q[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}] \right) + \nu_{M_0}(\mathcal{M}^{(2)} - f^{(2)}),$$

where we have used \mathcal{M} to denote the local Maxwellian associated with the distribution function f , and we will use the notations \mathcal{M}_M , $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$ defined similarly to (2) and (6). Likewise, we define the operator

$$Q_M = \mathcal{T}_M Q,$$

so that the right-hand side of (9) can be written as $Q_M[f_M, f_M]$.

To choose M_0 , we are interested in the estimation of the difference between the two right-hand sides in (9) and (10). For this aim, we make the following assumptions:

- The truncated series f_M provides a sufficiently good approximation of the distribution function f , so that we can assume

$$f_M(\mathbf{v}) \gtrsim 0, \quad \forall \mathbf{v} \in \mathbb{R}^3,$$

where “ \gtrsim ” means the inequality holds approximately.

- The truncated series \mathcal{M}_M provides a sufficiently good approximation of the local Maxwellian \mathcal{M} , so that we can assume $Q[\mathcal{M}_M, \mathcal{M}_M] \approx 0$.
- The operator Q_M provides a sufficiently good approximation of the collision operator Q , so that Q_M and Q are interchangeable in the derivation below.

In general, these assumptions mean that M is sufficiently large so that truncated functions and operators can almost preserve the properties of the original functions and operators. The purpose of these conditions is to focus mainly on the modeling error to be described below, and temporarily ignore the error introduced by the spectral method itself. Alternatively, one can regard M as infinity in our following derivations so that the conditions above hold naturally. After the error indicator is derived, to make the computation feasible, the infinities that appear in its expression are replaced by a finite M to approximate our error indicator.

Now we use ΔQ to denote the modeling error, i.e. the difference between the right-hand sides of (9) and (10):

$$\Delta Q := Q_M[f_M, f_M] - \left(Q_{M_0}[f^{(1)}, f^{(1)}] - Q_{M_0}[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}] + \nu_{M_0}(\mathcal{M}^{(2)} - f^{(2)}) \right).$$

Our aim is to find a heuristic error indicator characterizing the size of the quantity above. This error indicator must be relatively cheap to compute given the expansion of f_M . To this end, we split ΔQ into three terms, written in the three lines below:

$$(11) \quad \begin{aligned} \Delta Q &= (Q_M[f^{(1)}, f^{(1)}] - Q_M[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]) - (Q_{M_0}[f^{(1)}, f^{(1)}] - Q_{M_0}[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]) \\ &\quad + Q_M[f_M, f_M] - (Q_M[f^{(1)}, f^{(1)}] - Q_M[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]) \\ &\quad - \nu_{M_0}(\mathcal{M}^{(2)} - f^{(2)}). \end{aligned}$$

The first line in this equation is the truncation error of the function $Q_M[f^{(1)}, f^{(1)}] - Q_M[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]$. The estimation of the truncation error usually requires the information of the function, which is expensive to retrieve. As a workaround, we choose to ignore this term in our error indicator. In fact, this term may be relatively small due to the following two reasons:

- (1) Both $f^{(1)}$ and $\mathcal{M}^{(1)}$ are early truncations of the series (only include polynomials of degree up to M_0), which usually appear to be very smooth. Therefore both $Q_M[f^{(1)}, f^{(1)}]$ and $Q_M[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]$ are sufficiently smooth functions which can be well approximated by an early truncation of their expansions.
- (2) According to the observations in [7], the dependence of higher moments on the lower moments is relatively weak, meaning that $f^{(1)}$ does not produce large numbers in the higher coefficients in the expansion of $Q_M[f^{(1)}, f^{(1)}]$, which is similar for $\mathcal{M}^{(1)}$.

These statements are yet to be verified rigorously. We would like to leave it to future work. Below we will mainly focus on the quantification of the second line in (11).

3.2. Building indicator. According to our working hypotheses, we rewrite the second line of (11) by replacing Q_M with Q and subtracting an approximately zero term $Q[\mathcal{M}_M, \mathcal{M}_M]$. Thereby we can rewrite this term as

$$(12) \quad \delta Q := (Q[f_M, f_M] - Q[\mathcal{M}_M, \mathcal{M}_M]) - (Q[f^{(1)}, f^{(1)}] - Q[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]).$$

Using $f_M = f^{(1)} + f^{(2)}$ and the fact that $Q[\cdot, \cdot]$ is quadratic and symmetric, we have

$$(13) \quad \begin{aligned} \delta Q &= (Q[\mathcal{M}_M + (f_M - \mathcal{M}_M), \mathcal{M}_M + (f_M - \mathcal{M}_M)] - Q[\mathcal{M}_M, \mathcal{M}_M]) \\ &\quad - (Q[\mathcal{M}^{(1)} + (f^{(1)} - \mathcal{M}^{(1)}), \mathcal{M}^{(1)} + (f^{(1)} - \mathcal{M}^{(1)})] - Q[\mathcal{M}^{(1)}, \mathcal{M}^{(1)}]) \\ &= 2 \underbrace{(Q[f_M - \mathcal{M}_M, \mathcal{M}_M] - Q[f^{(1)} - \mathcal{M}^{(1)}, \mathcal{M}^{(1)}])}_{\delta Q_a} + \underbrace{(Q[f_M - \mathcal{M}_M, f_M - \mathcal{M}_M] - Q[f^{(1)} - \mathcal{M}^{(1)}, f^{(1)} - \mathcal{M}^{(1)}])}_{\delta Q_b}, \end{aligned}$$

where δQ_a and δQ_b can be further simplified as

$$(14) \quad \begin{aligned} \delta Q_a &= 2Q[(f^{(1)} - \mathcal{M}^{(1)}) + (f^{(2)} - \mathcal{M}^{(2)}), \mathcal{M}^{(1)} + \mathcal{M}^{(2)}] - 2Q[f^{(1)} - \mathcal{M}^{(1)}, \mathcal{M}^{(1)}] \\ &= 2Q[f^{(2)} - \mathcal{M}^{(2)}, \mathcal{M}_M] + 2Q[f^{(1)} - \mathcal{M}^{(1)}, \mathcal{M}^{(2)}], \\ \delta Q_b &= Q[f_M - \mathcal{M}_M, f_M - \mathcal{M}_M] - Q[(f_M - \mathcal{M}_M) - (f^{(2)} - \mathcal{M}^{(2)}), (f_M - \mathcal{M}_M) - (f^{(2)} - \mathcal{M}^{(2)})] \\ &= Q[f^{(2)} - \mathcal{M}^{(2)}, f^{(2)} - \mathcal{M}^{(2)}] + 2Q[f_M - \mathcal{M}_M, f^{(2)} - \mathcal{M}^{(2)}]. \end{aligned}$$

Inserting (14) into (13) yields

$$(15) \quad \delta Q = 2 \underbrace{Q[f_M, f^{(2)} - \mathcal{M}^{(2)}]}_{\delta Q_1} + 2 \underbrace{Q[f^{(1)} - \mathcal{M}^{(1)}, \mathcal{M}^{(2)}]}_{\delta Q_2} + \underbrace{Q[f^{(2)} - \mathcal{M}^{(2)}, f^{(2)} - \mathcal{M}^{(2)}]}_{\delta Q_3}.$$

This expression implies that δQ is small in either of the following two scenarios:

- (1) The functions $f^{(1)}$ and $\mathcal{M}^{(1)}$ can accurately describe f_M and \mathcal{M}_M , respectively, which means $f^{(2)}$ and $\mathcal{M}^{(2)}$ are small.
- (2) The function f_M is close to the equilibrium \mathcal{M}_M , which means $f^{(1)} - \mathcal{M}^{(1)}$ and $f^{(2)} - \mathcal{M}^{(2)}$ are both small.

In either case, the term δQ_3 appears to be a quadratic term, and is expected to be smaller than the previous two terms. Hence, we ignore this term in our error indicator and mainly discuss the estimation of δQ_1 and δQ_2 . The analysis above indicates that δQ_1 and δQ_2 present two different sources of the error: δQ_1 mainly captures the error due to the BGK approximation of the high-frequency part, and δQ_2 mainly captures the error due to the missing interaction between the low-frequency part and the high-frequency part.

Before proceeding, we would like to first discuss how much computational cost we can afford to estimate (15). Recall that the time complexity for evaluating all the coefficients in the expansion of $Q_M[f_M, f_M]$ is $O(M^8)$. Therefore, the computational cost of the indicator should be essentially smaller than $O(M^8)$ to achieve savings. Besides, the computational cost for the right-hand side of (8) is $O(M_0^8 + M^3)$ according to [24], and our computational cost for the indicators should not be significantly larger than this. Thus, it is unrealistic to compute δQ_1 directly since its computation requires already $O(M^8)$ operations. Meanwhile, we would also like to avoid direct computation of δQ_2 since it requires $O(M^6 M_0^2)$ operations, which would take the most time of the simulation if computed. As a result, we need to estimate these terms using a computationally cheaper expression. The most naive approach is to consider the following type of estimation:

$$\|Q[f, g]\| \leq C \|f\| \cdot \|g\|.$$

Unfortunately, the collision operator Q is unbounded for most collision kernels. Note that the gain term may be bounded when the assumption of Grad's angular cut-off holds (see e.g. [36]), while our approach to be proposed in the rest part of this section does not rely on this assumption.

The basic idea is to bound $Q[f, g]$ by splitting it into the gain and loss operators:

$$\begin{aligned} Q^+[f, g] &= \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbf{n} \perp \mathbf{g}} \int_0^\pi B(|\mathbf{g}|, \chi) [f(\mathbf{v}')g(\mathbf{v}') + f(\mathbf{v}'_*)g(\mathbf{v}'_*)] d\chi d\mathbf{n} d\mathbf{v}_*, \\ Q^-[f, g] &= \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbf{n} \perp \mathbf{g}} \int_0^\pi B(|\mathbf{g}|, \chi) [f(\mathbf{v})g(\mathbf{v}_*) + f(\mathbf{v}_*)g(\mathbf{v})] d\chi d\mathbf{n} d\mathbf{v}_*. \end{aligned}$$

Then for any functions f and g satisfying $f \geq 0$, if we can find another distribution function h satisfying $|g(\mathbf{v})| \leq h(\mathbf{v})$ for every $\mathbf{v} \in \mathbb{R}^3$, then it holds that

$$(16) \quad |Q[f, g]| = |Q^+[f, g] - Q^-[f, g]| \leq Q^+[f, |g|] + Q^-[f, |g|] \leq Q^+[f, h] + Q^-[f, h],$$

where we have used the positivity of the collision kernel $B(|\mathbf{g}|, \chi)$ to get the last inequality. Thus, the right-hand side of (16) can be used as an upper bound of $|Q[f, g]|$. In general, the computational cost of this upper bound is as high as a full collision operator. Therefore, the function h must be chosen carefully so that $Q^+[f, h] + Q^-[f, h]$ can be efficiently computed. For simplicity, we define

$$Q^{\text{abs}}[f, h] = Q^+[f, h] + Q^-[f, h],$$

which is different from $Q[f, h]$ since $Q[f, h]$ is the difference of these two terms. Based on this idea, we need to answer the following two questions:

- (1) What should the general form of h be such that the computation of $Q^{\text{abs}}[f, h]$ is efficient?
- (2) Given f_M , how to find the function h as the bound of f_M ?

These two questions will be addressed in the following two subsections.

3.2.1. *Space of the bounding function.* Our choice of h is established on the rotational invariance of the collision operator:

Lemma 1. *For any orthogonal matrix \mathbf{R} , let*

$$f^{\mathbf{R}}(\mathbf{v}) = f(\mathbf{R}\mathbf{v}), \quad g^{\mathbf{R}}(\mathbf{v}) = g(\mathbf{R}\mathbf{v}).$$

Then

$$Q[f, g](\mathbf{R}\mathbf{v}) = Q[f^{\mathbf{R}}, g^{\mathbf{R}}](\mathbf{v}), \quad Q^{\text{abs}}[f, g](\mathbf{R}\mathbf{v}) = Q^{\text{abs}}[f^{\mathbf{R}}, g^{\mathbf{R}}](\mathbf{v}).$$

Here the rotational invariance of the collision operator is a classical result and can be found in [14, p. 45]. The rotational invariance of Q^{abs} can be derived from the rotational invariance of both Q^+ and Q^- . A natural consequence of this result is

Corollary 2. *Let $h(\mathbf{v})$ be a function depending only on $|\mathbf{v}|$. Then the linear operator $\mathcal{L}_h[\cdot] := Q^{\text{abs}}[\cdot, h]$ is a rotational invariant operator, i.e., for any orthogonal matrix \mathbf{R} , we have*

$$\mathcal{L}_h[f](\mathbf{R}\mathbf{v}) = \mathcal{L}_h[f^{\mathbf{R}}](\mathbf{v}),$$

for $f^{\mathbf{R}}(\mathbf{v}) = f(\mathbf{R}\mathbf{v})$.

For linear rotationally invariant operators, we have the following result:

Theorem 3. *Suppose $\mathcal{L}[\cdot]$ is a linear rotationally invariant operator. For any non-negative integers l and n , there exists an isotropic function $c_{ln}(|\mathbf{v}|)$ such that*

$$\mathcal{L}[\varphi_{lmn}](\mathbf{v}) = c_{ln}(|\mathbf{v}|)\varphi_{lmn}(\mathbf{v}), \quad \forall m = -l, \dots, l.$$

This result is already given in [7, Theorem 1], where the statement is written for the linearized collision operator but the proof only requires the rotational invariance of \mathcal{L} . This theorem indicates that the series form of $\mathcal{L}_h[\varphi_{lmn}]$ should be

$$(17) \quad \mathcal{L}_h[\varphi_{lmn}] = \sum_{n_1=0}^{+\infty} a_{lnn_1}^{(h)} \varphi_{lmn_1}.$$

Consequently, if we choose $h(\mathbf{v})$ to be an isotropic function that depends only on $|\mathbf{v}|$, we have

$$Q^{\text{abs}}[f_M, h] = \mathcal{L}_h[f_M] = \sum_{l=0}^M \sum_{m=-l}^l \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} f_{lmn} \mathcal{L}_h[\varphi_{lmn}] = \sum_{l=0}^M \sum_{m=-l}^l \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} \sum_{n_1=0}^{+\infty} f_{lmn} a_{lnn_1}^{(h)} \varphi_{lmn_1}.$$

Numerically, we truncate the series above by replacing $+\infty$ with $\lfloor (M-l)/2 \rfloor$. Thus, when all the coefficients $a_{lnn}^{(h)}$ are given, the computational cost for the expansion of $Q^{\text{abs}}[f_M, h]$ is $O(M^4)$. Technically, this can be done by carrying out the matrix-vector multiplication

$$(18) \quad Q_{lmn_1}^{\text{abs}} = \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} a_{lnn_1}^{(h)} f_{lmn}, \quad l = 0, 1, \dots, M, \quad m = -l, \dots, l, \quad n_1 = 0, 1, \dots, \lfloor (M-l)/2 \rfloor,$$

as allows efficient libraries of linear algebra to be used.

To find the coefficients $a_{l n n_1}^{(h)}$, we need the expansion of h in terms of the basis functions $\varphi_{l m n}$. Since we choose h to be isotropic, the expansion holds the form

$$h(\mathbf{v}) = \sum_{n'=0}^{+\infty} h_{n'} \varphi_{00n'}(\mathbf{v}).$$

For each n' , the function $\varphi_{00n'}$ is also isotropic, implying that $\mathcal{L}_{\varphi_{00n'}}[\cdot] = Q^{\text{abs}}[\varphi_{00n'}, \cdot]$ is rotationally invariant. Thus we can assume

$$\mathcal{L}_{\varphi_{00n'}}[\varphi_{l m n}] = \sum_{n_1=0}^{+\infty} a_{l n n_1}^{n'} \varphi_{l m n_1},$$

so that

$$\mathcal{L}_h[\varphi_{l m n}] = \sum_{n'=0}^{+\infty} h_{n'} \mathcal{L}_{\varphi_{00n'}}[\varphi_{l m n}] = \sum_{n_1=0}^{+\infty} \sum_{n'=0}^{+\infty} a_{l n n_1}^{n'} h_{n'} \varphi_{l m n_1}.$$

By comparing this equation with (17), one can find that

$$(19) \quad a_{l n n_1}^{(h)} = \sum_{n'=0}^{+\infty} a_{l n n_1}^{n'} h_{n'}, \quad l = 0, 1, \dots, M, \quad n, n_1 = 0, 1, \dots, \lfloor (M-l)/2 \rfloor.$$

In practice, we again truncate the above series by replacing $+\infty$ with some $N_0 = O(M)$. Then the equation (19) shows that the computation of all required coefficients again needs $O(M^4)$ operations.

By now, we have concluded that choosing $h(\mathbf{v})$ to be an isotropic function can reduce the computational cost of $Q^{\text{abs}}[f, h]$ to $O(M^4)$ (including (19) and (18)). To complete the computation, we still need to obtain $a_{l n n_1}^{n'}$ for a given collision operator. These coefficients can be precomputed before the simulation, which will be discussed in detail in Section 3.2.3. Now we will first discuss the construction of h such that $|g| \leq h$ for some given function g in order that the estimation (16) holds.

3.2.2. The approximation of the bounding function and the error indicator. In our implementation, instead of looking for h that bounds $|g|$ pointwisely, we choose to find an approximate upper bound with the form

$$(20) \quad h(\mathbf{v}) = \sum_{n'=0}^{N_0} h_{n'} \varphi_{00n'}(\mathbf{v}), \quad N_0 = \left\lceil \frac{M}{2} \right\rceil.$$

The general idea to find h is to bound the radial part and the angular part separately. To this end, we write the basis functions as

$$(21) \quad \varphi_{l m n}(\mathbf{v}) = \varphi_{ln}^1(\mathbf{v} - \bar{\mathbf{u}}) Y_l^m \left(\frac{\mathbf{v} - \bar{\mathbf{u}}}{|\mathbf{v} - \bar{\mathbf{u}}|} \right),$$

where

$$(22) \quad \varphi_{ln}^1(\mathbf{v}) = \sqrt{\frac{2^{1-l} \pi^{3/2} n!}{\Gamma(n+l+3/2)}} L_n^{(l+1/2)} \left(\frac{|\mathbf{v}|^2}{2\bar{\theta}} \right) \left(\frac{|\mathbf{v}|}{\sqrt{\bar{\theta}}} \right)^l \cdot \frac{1}{(2\pi\bar{\theta})^{3/2}} \exp\left(-\frac{|\mathbf{v}|^2}{2\bar{\theta}}\right)$$

represents the radial part of the basis function. Without loss of generality, we set $\bar{\mathbf{u}} = 0$ and $\bar{\theta} = 1$ in the analysis below. Since $\varphi_{ln}^1(\mathbf{v})$ depends only on $|\mathbf{v}|$, here we approximate its absolute value by a linear combination of φ_{00n} :

$$(23) \quad |\varphi_{ln}^1(\mathbf{v})| \approx \sum_{n'=0}^{N_0} s_{ln}^{n'} \varphi_{00n'}(\mathbf{v}),$$

and we choose to find the approximation by orthogonal projection:

$$(24) \quad s_{ln}^{n'} = \int_{\mathbb{R}^3} |\varphi_{ln}^1(\mathbf{v})| \varphi_{00n'}(\mathbf{v}) \omega(\mathbf{v}) d\mathbf{v}.$$

Since φ_{00n} contains a polynomial with even order $2n$, by choosing $N_0 = \lceil \frac{M}{2} \rceil$, we can guarantee that the degree of the polynomial in the right-hand side of (23) is no less than that in the radial function φ_{ln}^1 . Thus, it holds for any distribution function g that

$$(25) \quad \begin{aligned} |g(\mathbf{v})| &= \left| \sum_{l=0}^M \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} \sum_{m=-l}^l g_{lmn} Y_l^m \left(\frac{\mathbf{v}}{|\mathbf{v}|} \right) \varphi_{ln}^1(\mathbf{v}) \right| \leq \sum_{l=0}^M \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} \left| \sum_{m=-l}^l g_{lmn} Y_l^m \left(\frac{\mathbf{v}}{|\mathbf{v}|} \right) \right| |\varphi_{ln}^1(\mathbf{v})| \\ &\leq \sum_{l=0}^M \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} g_{ln} |\varphi_{ln}^1(\mathbf{v})| \approx \sum_{l=0}^M \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} g_{ln} \sum_{n'=0}^{N_0} s_{ln}^{n'} \varphi_{00n'}(\mathbf{v}), \end{aligned}$$

where

$$(26) \quad g_{ln} = \max_{|\mathbf{n}|=1} \left| \sum_{m=-l}^l g_{lmn} Y_l^m(\mathbf{n}) \right|.$$

Equation (25) shows that we can choose

$$(27) \quad h_{n'} = \sum_{l=0}^M \sum_{n=0}^{\lfloor (M-l)/2 \rfloor} g_{ln} s_{ln}^{n'}, \quad n' = 0, 1, \dots, N_0$$

such that $h(\mathbf{v})$ is an approximate upper bound of $g(\mathbf{v})$.

In the calculation of $h_{n'}$, the coefficients $s_{ln}^{n'}$ can be precomputed by numerical integration before the simulation. Thus once g_{ln} are obtained, the computational cost is $O(M^3)$. As for g_{ln} , we again choose to approximate them instead of computing them exactly. We pick a finite set of points $\Omega \in \mathbb{S}^2$ and approximate g_{ln} by

$$(28) \quad g_{ln} \approx \max_{\mathbf{n} \in \Omega} \left| \sum_{m=-l}^l g_{lmn} Y_l^m(\mathbf{n}) \right|.$$

In our implementation, the fifty-point Lebedev-Gauss integral points [32] are chosen to form the set Ω . Thus the computational cost to find all g_{ln} is also $O(M^3)$.

By such means, we can find $h(\mathbf{v})$ with the form (20) such that $|f^{(2)}(\mathbf{v}) - \mathcal{M}^{(2)}(\mathbf{v})| \lesssim h(\mathbf{v})$, meaning that $h(\mathbf{v})$ is an approximate bound of $f^{(2)} - \mathcal{M}^{(2)}$. Thus δQ_1 defined in (15) can be bounded by

$$(29) \quad |\delta Q_1| \leq Q^{\text{abs}}[f_M, |f^{(2)} - \mathcal{M}^{(2)}|] \lesssim Q^{\text{abs}}[f_M, h],$$

which can be computed via (18) and (19) with computational cost $O(M^4)$. To bound δQ_2 , we adopt the similar approach: by constructing $h^{(1)}(\mathbf{v})$ and $h^{(2)}(\mathbf{v})$ such that $|f^{(1)} - \mathcal{M}^{(1)}| \lesssim h^{(1)}(\mathbf{v})$, $|\mathcal{M}^{(2)}| \lesssim h^{(2)}(\mathbf{v})$ and

$$(30) \quad h^{(1)}(\mathbf{v}) = \sum_{n'=0}^{N_0} h_{n'}^{(1)} \varphi_{00n'}(\mathbf{v}), \quad h^{(2)}(\mathbf{v}) = \sum_{n'=0}^{N_0} h_{n'}^{(2)} \varphi_{00n'}(\mathbf{v}),$$

we have

$$(31) \quad \begin{aligned} |\delta Q_2| &\leq Q^{\text{abs}}[|f^{(1)} - \mathcal{M}^{(1)}|, |\mathcal{M}^{(2)}|] \lesssim Q^{\text{abs}}[h^{(1)}, h^{(2)}] \\ &= \sum_{n'=0}^{N_0} \sum_{n=0}^{N_0} h_n^{(1)} h_{n'}^{(2)} \mathcal{L}_{\varphi_{00n'}}[\varphi_{00n}] = \sum_{n'=0}^{N_0} \sum_{n=0}^{N_0} h_n^{(1)} h_{n'}^{(2)} \sum_{n_1=0}^{N_0} a_{0nn_1}^{n'} \varphi_{00n_1} \\ &= \sum_{n_1=0}^{N_0} \left(\sum_{n'=0}^{N_0} \sum_{n=0}^{N_0} h_n^{(1)} h_{n'}^{(2)} a_{0nn_1}^{n'} \right) \varphi_{00n_1}, \end{aligned}$$

where the coefficients

$$(32) \quad \tilde{Q}_{n_1}^{\text{abs}} = \sum_{n'=0}^{N_0} \sum_{n=0}^{N_0} h_n^{(1)} h_{n'}^{(2)} a_{0nn_1}^{n'}, \quad n = 0, 1, \dots, N_0$$

can be computed with time complexity $O(M^3)$. Finally, we choose our error indicator to be the sum of the bounds of both δQ_1 and δQ_2 :

$$(33) \quad \text{Indicator} = \sqrt{\sum_{l=0}^M \sum_{m=-l}^l \sum_{n_1=0}^{N_0} |Q_{lmn_1}^{\text{abs}}|^2} + \sqrt{\sum_{n_1=0}^{N_0} |\tilde{Q}_{n_1}^{\text{abs}}|^2}.$$

Here we have used the weighted L^2 -norm with weight function $\omega(\mathbf{v})$ so that the norm is simply the sum of squares of the coefficients. This error indicator can be considered as an *a posteriori* estimation of the truncation error, since it depends on the numerical solution but only estimates the error of the collision term instead of the solution itself.

In the indicator above, we did not consider the third line of (11) since the contribution of this term is similar to h times a constant (since $h(\mathbf{v})$ is the approximate bound of $|f^{(2)}(\mathbf{v}) - \mathcal{M}^{(2)}(\mathbf{v})|$). Such contribution has been covered by the term $Q^{\text{abs}}[f_M, h]$ and it is less meaningful to duplicate it in the error indicator.

3.2.3. *Computation of the coefficients $a_{l n n_1}^{n'}$.* By the orthogonality of the basis functions (3), we have

$$(34) \quad a_{l n n_1}^{n'} = \int_{\mathbb{R}^3} p_{l m n}^\dagger(\mathbf{v}) Q^{\text{abs}}[\varphi_{l m n_1}, \varphi_{0 0 n'}](\mathbf{v}) d\mathbf{v}$$

for any $m = -l, \dots, l$. For simplicity, here we have used

$$p_{l m n}(\mathbf{v}) = \varphi_{l m n}(\mathbf{v}) \omega(\mathbf{v}),$$

and $p_{l m n}$ is a polynomial of degree $l + 2n$. Below we will also use $p_{l m n}^0$ to denote the polynomial $p_{l m n}$ with $\bar{\mathbf{u}}$ set to be zero. In our calculation, we choose $m = 0$ so that all the functions are real and we can remove “ \dagger ” in (34). A well-known property of the collision integral is

$$(35) \quad \begin{aligned} & \int_{\mathbb{R}^3} p_{l 0 n}(\mathbf{v}) Q^{\text{abs}}[\varphi_{l 0 n_1}, \varphi_{0 0 n'}](\mathbf{v}) d\mathbf{v} \\ &= \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \int_{\mathbf{n} \perp \mathbf{g}} \int_0^\pi [p_{l 0 n}(\mathbf{v}') + p_{l 0 n}(\mathbf{v}'_*) + p_{l 0 n}(\mathbf{v}) + p_{l 0 n}(\mathbf{v}_*)] B(|\mathbf{g}|, \chi) \varphi_{l 0 n_1}(\mathbf{v}) \varphi_{0 0 n'}(\mathbf{v}_*) d\chi d\mathbf{n} d\mathbf{v}_* d\mathbf{v}. \end{aligned}$$

A classical approach [22, 31] to computing this integral is to define

$$\mathbf{g}' = \mathbf{v}' - \mathbf{v}'_*, \quad \mathbf{h} = \frac{\mathbf{v} + \mathbf{v}_*}{2},$$

which yields

$$|\mathbf{g}| = |\mathbf{g}'|, \quad \mathbf{v} = \mathbf{h} + \frac{1}{2}\mathbf{g}, \quad \mathbf{v}_* = \mathbf{h} - \frac{1}{2}\mathbf{g}, \quad \mathbf{v}' = \mathbf{h} + \frac{1}{2}\mathbf{g}', \quad \mathbf{v}'_* = \mathbf{h} - \frac{1}{2}\mathbf{g}', \quad d\mathbf{v} d\mathbf{v}_* = d\mathbf{g} d\mathbf{h}.$$

The purpose of these changes of variables is to convert the integral with respect to \mathbf{v}_* and \mathbf{v} to the integral with respect to \mathbf{g} and \mathbf{h} . This requires us to express the polynomials in (35) by linear combinations of the basis polynomials of \mathbf{g} and \mathbf{h} . This requires the following result:

Theorem 4. *For any non-negative integers l , n and n' , it holds that*

$$(36) \quad \begin{aligned} & p_{l 0 n} \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) p_{0 0 n'} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) \\ &= \sum_{\substack{l_1, l_2, n_1, n_2 \geq 0 \\ l_1 + l_2 + 2(n_1 + n_2) = l + 2(n + n')}} \sum_{\substack{m_1 = -l_1, \dots, l_1 \\ m_2 = -l_2, \dots, l_2 \\ m_1 + m_2 = 0}} A_{l n n'}^{l_1 l_2 m_2 n_2} p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right). \end{aligned}$$

The coefficients $A_{lnn'}^{l_1 l_2 m_2 n_2}$ are constants satisfying the recurrence relation:

$$(37) \quad A_{ln,n'+1}^{l_1 l_2 m_2 n_2} = \frac{1}{\sqrt{(n'+1)(n'+3/2)}} \left[\sum_{k=1}^2 \frac{\sqrt{n_k(n_k+l_k+1/2)}}{2} A_{lnn'}^{l_1 l_2 m_2, n_2-\delta_{2k}} \right. \\ \left. + \sum_{\mu=-1}^1 (-1)^\mu \left(\sqrt{(l_1+n_1+1/2)(l_2+n_2+1/2)} \gamma_{l_1, m_2+\mu}^{-\mu} \gamma_{l_2, m_2+\mu}^{-\mu} A_{lnn'}^{l_1-1, l_2-1, m_2+\mu, n_2} \right. \right. \\ \left. \left. - (-1)^\mu \sqrt{n_1(l_2+n_2+1/2)} \gamma_{l_1-1, m_2+\mu}^{-\mu} \gamma_{l_2, m_2+\mu}^{-\mu} A_{lnn'}^{l_1+1, l_2-1, m_2+\mu, n_2} \right. \right. \\ \left. \left. - (-1)^\mu \sqrt{n_2(l_1+n_1+1/2)} \gamma_{l_1, m_2+\mu}^{-\mu} \gamma_{l_2-1, m_2+\mu}^{-\mu} A_{lnn'}^{l_1-1, l_2+1, m_2+\mu, n_2-1} \right. \right. \\ \left. \left. + \sqrt{n_1 n_2} \gamma_{l_1-1, m_2+\mu}^{-\mu} \gamma_{l_2-1, m_2+\mu}^{-\mu} A_{lnn'}^{l_1+1, l_2+1, m_2+\mu, n_2-1} \right) \right],$$

where $n_1 = (l - l_1 - l_2)/2 + (n + n' - n_2)$, and γ_{lm}^μ is defined by

$$(38) \quad \gamma_{lm}^\mu = \sqrt{\frac{[l + (2\delta_{1,\mu} - 1)m + \delta_{1,\mu}][l - (2\delta_{-1,\mu} - 1)m + \delta_{-1,\mu}]}{2^{|\mu|}(2l-1)(2l+1)}},$$

and $A_{lnn'}^{l_1 m_1 n_1, l_2 m_2 n_2}$ is regarded as zero if any of the following conditions are violated:

- (1) l_1, l_2, n_2 are positive integers; (2) $l - l_1 - l_2 + 2(n + n' - n_2) \geq 0$; (3) $|m_2| \leq \min(l_1, l_2)$.

The proof of this theorem is similar to the proof of [7, Proposition 3], and the details can be found in Appendix B. The recurrence relation (37) helps compute the coefficients in the expansion (36). The initial condition corresponds to the case $n' = 0$, for which $p_{00n'}(\cdot) \equiv 1$ and the corresponding coefficients $A_{lnn'}^{l_1 l_2 m_2 n_2}$ have been derived in [7, Proposition 3].

By (36), we have

$$\begin{aligned} & p_{l0n}(\mathbf{v}') + p_{l0n}(\mathbf{v}'_*) + p_{l0n}(\mathbf{v}) + p_{l0n}(\mathbf{v}_*) \\ = & \sum_{\substack{l_1, l_2, n_1, n_2 \geq 0 \\ l_1 + l_2 + 2(n_1 + n_2) = l + 2n}} \sum_{\substack{m_1 = -l_1, \dots, l_1 \\ m_2 = -l_2, \dots, l_2 \\ m_1 + m_2 = 0}} A_{ln0}^{l_1 l_2 m_2 n_2} p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) \times \\ & \left[p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}'}{\sqrt{2}} \right) + p_{l_2 m_2 n_2}^0 \left(-\frac{\mathbf{g}'}{\sqrt{2}} \right) + p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) + p_{l_2 m_2 n_2}^0 \left(-\frac{\mathbf{g}}{\sqrt{2}} \right) \right] \\ = & \sum_{\substack{l_1, l_2, n_1, n_2 \geq 0 \\ l_1 + l_2 + 2(n_1 + n_2) = l + 2n}} \sum_{\substack{m_1 = -l_1, \dots, l_1 \\ m_2 = -l_2, \dots, l_2 \\ m_1 + m_2 = 0}} [1 + (-1)^{l_2}] A_{ln0}^{l_1 l_2 m_2 n_2} p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) \left[p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}'}{\sqrt{2}} \right) + p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) \right], \end{aligned}$$

where we have used the property that p_{lmn}^0 is odd/even if l is odd/even. By the equality [26, Eq. (7.1)]

$$\int_{\mathbf{n} \perp \mathbf{g}} p_{lmn}^0 \left(\frac{\mathbf{g}'}{\sqrt{2}} \right) d\mathbf{n} = 2\pi p_{lmn}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) P_l(\cos \chi),$$

we conclude that

$$(39) \quad \int_{\mathbf{n} \perp \mathbf{g}} [p_{l0n}(\mathbf{v}') + p_{l0n}(\mathbf{v}'_*) + p_{l0n}(\mathbf{v}) + p_{l0n}(\mathbf{v}_*)] d\mathbf{n} \\ = \sum_{\substack{l_1, l_2, n_1, n_2 \geq 0 \\ l_1 + l_2 + 2(n_1 + n_2) = l + 2n}} \sum_{\substack{m_1 = -l_1, \dots, l_1 \\ m_2 = -l_2, \dots, l_2 \\ m_1 + m_2 = 0}} 2\pi [1 + (-1)^{l_2}] [P_{l_2}(\cos \chi) + 1] A_{ln0}^{l_1 l_2 m_2 n_2} p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right).$$

Next, we perform the following operations:

- (1) Insert (39) into (35);
- (2) Use (36) to expand $\varphi_{l0n_1}(\mathbf{v})\varphi_{00n'}(\mathbf{v}_*)$ in (35);
- (3) Use the orthogonality (3) to integrate with respect to \mathbf{h} .

- (4) Represent \mathbf{g} using spherical coordinates as $g\boldsymbol{\omega}$ and use the orthogonality of spherical harmonics to integrate with respect to $\boldsymbol{\omega}$.

We omit the details of these steps since they are standard procedures to compute the moments of the collision operators. Afterward, we obtain

(40)

$$a_{lnn_1}^{n'} = \sum_{\substack{l_1, l_2 \geq 0 \\ l - (l_1 + l_2) \text{ is even} \\ l_1 + l_2 \leq l + 2 \min(n, n_1 + n')}}^{\min(l_1, l_2)} \sum_{m_2 = -\min(l_1, l_2)}^{\min(l_1, l_2)} \sum_{\substack{n_2, n_2' \geq 0 \\ n - n_2 = n_1 + n' - n_2' \\ n_2 \leq (l - l_1 - l_2)/2 + n}} \sqrt{\frac{n_2! n_2'!}{\Gamma(l_2 + n_2 + 3/2) \Gamma(l_2 + n_2' + 3/2)}} A_{ln_2 0}^{l_1 l_2 m_2 n_2} A_{ln_1 n'}^{l_1 l_2 m_2 n_2'} \\ \times \pi [1 + (-1)^{l_2}] \int_0^{+\infty} \int_0^\pi B(\sqrt{4\theta}r, \chi) [P_{l_2}(\cos \chi) + 1] L_{n_2}^{(l_2+1/2)}(r) L_{n_2'}^{(l_2+1/2)}(r) r^{l_2+1/2} \exp(-r) d\chi dr,$$

where r comes from the change of variables $r = g^2/(4\bar{\theta})$.

The integrals with respect to χ and r depend on the collision kernel. For the variable-hard-sphere (VHS) model, the collision kernel is

$$(41) \quad B(g, \chi) = g^\nu \sin \chi \quad \text{for } \nu \in [0, 1].$$

In this case, the underlined integral in the equation above can be computed explicitly as

(42)

$$2^{\nu+1} (\delta_{0, l_2} + 1) \bar{\theta}^{\nu/2} \int_0^{+\infty} L_{n_2}^{(l_2+1/2)}(r) L_{n_2'}^{(l_2+1/2)}(r) r^{l_2+(\nu+1)/2} \exp(-r) dr \\ = (-1)^{n_2+n_2'} 2^{\nu+1} (\delta_{0, l_2} + 1) \bar{\theta}^{\nu/2} \Gamma\left(l_2 + 1 + \frac{\nu+1}{2}\right) \sum_{k=0}^{\min(n_2, n_2')} \binom{\nu/2}{n_2 - k} \binom{\nu/2}{n_2' - k} \binom{k + l_2 + (\nu+1)/2}{k}.$$

Note that the coefficients $a_{lnn_1}^{n'}$ can all be precomputed before the simulation. In our implementation, we ignore the coefficient $\bar{\theta}^{\nu/2}$ since it only introduces a universal constant to the indicator.

3.3. Adaptive strategy. With the error indicator defined by (33), we compute this quantity for the distribution function on each spatial grid cell after every time step. For distribution functions with a large indicator, we increase the value of M_0 at the next time step, and vice versa. In our implementation, in order that the numerical solution does not oscillate due to the self-adaptation, we would like to maintain the stability of the collision model by avoiding the drastic change of the value of M_0 . To this end, we adopt the following two strategies:

- Instead of a single threshold for the indicator like in most adaptive methods, we introduce two thresholds ϵ_1 and ϵ_2 . If the error indicator of a certain distribution function lies between (ϵ_1, ϵ_2) , we keep the value of M_0 unchanged.
- The value of M_0 changes only by 1 at each time step. More precisely, if the error indicator exceeds ϵ_2 , we increase M_0 by 1; if the error indicator falls below ϵ_1 , we reduce M_0 by 1.

In general, if the range of the interval (ϵ_1, ϵ_2) is wider, M_0 is more stable and the algorithm is less adaptive. A larger lower bound ϵ_1 makes it easier for M_0 to drop; a larger upper bound ϵ_2 makes it harder for M_0 to increase. In many applications, the bounds do not need to be too tight since the first few moments (e.g., density, velocity, and temperature) are usually not very sensitive about small changes of the collision models. This is why some simpler collision models such as the ES-BGK and the Shakhov models can still provide decent numerical results for these macroscopic variables.

To select the proper values of ϵ_1 and ϵ_2 , we adopt the following strategy:

- (1) Do a test run with a small M_0 (e.g. $M_0 = 3$) being fixed without self-adaptation. Calculate the indicators for all time steps on all grid cells, and find its maximum value ϵ_{\max} .
- (2) Use ϵ_{\max} as a reference value and choose the initial guesses of ϵ_1 and ϵ_2 . They should be less than but not too far away from ϵ_{\max} , e.g. $\epsilon_1 = \epsilon_{\max}/4$ and $\epsilon_2 = \epsilon_{\max}/2$.
- (3) Do a test run for the self-adaptive algorithm with the chosen ϵ_1 and ϵ_2 , and then reduce ϵ_1 (e.g. set ϵ_1 to be $\epsilon_1/2$).
- (4) Do another test run for the current ϵ_1 and ϵ_2 .

- (5) Compare the results of the two most recent runs. Go to the next step if the two results are sufficiently close to each other; otherwise, reduce ϵ_1 again and return to the previous step.
- (6) Keep ϵ_1 fixed and reduce the value of ϵ_2 (e.g. set ϵ_2 to be $\epsilon_2/2$).
- (7) Compare the results of the two most recent runs. Stop if the two results are sufficiently close to each other; otherwise, reduce the value of ϵ_2 and check if $\epsilon_1 < \epsilon_2$. If so, return to Step 6; otherwise, reduce ϵ_1 and return to Step 3.

Here, the purpose of the first step is to provide a general range of the acceptable error indicator. Then, based on the initial guess of ϵ_1 and ϵ_2 in the second step, we first determine the lower bound by reducing ϵ_1 until the solution looks stable, and then apply the same approach to ϵ_2 to find a suitable upper bound. All test runs can be carried out on a coarse grid to save computational time. An example will be given in Section 4.2.1.

3.4. Outline of the algorithm. As a summary, below we list out the general steps of our algorithm:

- (1) Precompute the coefficients $s_{l_n}^{n'}$ according to (24).
- (2) Precompute the coefficients $A_{l_n n'}^{l_1 l_2 m_2 n_2}$ according to (37).
- (3) Precompute the coefficients $a_{l_n n_1}^{n'}$ according to (40) with the underlined term replaced by (42).
- (4) Solve the Boltzmann equation by one time step. Terminate if the final time is reached.
- (5) For each distribution function, use (27) and (28) to find the bounding functions h , $h^{(1)}$ and $h^{(2)}$, which bound $f^{(2)} - \mathcal{M}^{(2)}$, $f^{(1)} - \mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$, respectively.
- (6) Use (18)(19)(32)(33) to compute the error indicator for each collision term.
- (7) Perform self-adaptation on each grid cell: if the error indicator is greater than the threshold ϵ_1 , we increase M_0 by 1; if the error indicator is less than the threshold ϵ_2 , we decrease M_0 by 1 if $M_0 > 3$.
- (8) Return to Step 4.

In the algorithm, we have required that M_0 is no less than 3, which is the smallest M_0 that includes the heat flux in the quadratic part of the collision operator. This ensures that the Navier-Stokes limit can always be correctly captured. The computation of the indicator lies in Steps 5 and 6. In these two steps, the computation of (18) and (19) has complexity $O(M^4)$, and the computation (28) requires $O(|\Omega|M^3)$ operations, where $|\Omega|$ denotes the number of quadrature points on the sphere. These parts take up most of the computational time in Steps 5 and 6. Note that the computational cost of the indicator depends on M instead of M_0 , since the computation of $Q^{\text{abs}}(\cdot, \cdot)$ involves the complete distribution function instead of only the low-frequency part, so that the error due to the inaccuracy of the high-frequency part can be captured. Nevertheless, as we will see in the next section, such a cost is quite small compared to the evaluation of the collision operator for a large M_0 .

4. NUMERICAL SCHEME AND EXPERIMENTS

We are now ready to integrate the adaptation technique into the Boltzmann solver and carry out numerical experiments. In what follows, we will first brief our numerical algorithm to solve the system (8), and then present several numerical experiments to demonstrate the effectiveness of the proposed indicator.

4.1. Numerical scheme. For convenience, we will only provide the numerical algorithm for the spatially one-dimensional case (the velocity space is still three-dimensional), where we assume that

$$(43) \quad \frac{\partial \mathbf{f}}{\partial x_2} = \frac{\partial \mathbf{f}}{\partial x_3} = 0.$$

The algorithm can be naturally generalized to the multi-dimensional case with uniform grids. Suppose the spatial domain $\Omega \subset \mathbb{R}$ is discretized by a uniform grid with cell size Δx . Using \mathbf{f}_j^n to approximate the average of \mathbf{f} over the j th grid cell $[x_{j-1/2}, x_{j+1/2}]$ at time t^n , we can solve the system (8) by the following finite volume method with time step size Δt :

$$(44) \quad \mathbf{f}_j^* = \mathbf{f}_j^n - \frac{\Delta t}{\Delta x} [\mathbf{F}_{j+1/2}^n - \mathbf{F}_{j-1/2}^n], \quad \mathbf{f}_j^{n+1} = \mathbf{f}_j^* + \Delta t \tilde{\mathbf{Q}}(M_0; \mathbf{f}_j^n),$$

where $\tilde{\mathbf{Q}}(M_0; \mathbf{f}_j^n)$ is the modified collision operator defined in (8). The numerical fluxes $\mathbf{F}_{j\pm 1/2}^n$ are chosen according to the HLL scheme [23]:

$$(45) \quad \mathbf{F}_{j+1/2}^n = \frac{\lambda^R \mathbf{A}_1 \mathbf{f}_j^n - \lambda^L \mathbf{A}_1 \mathbf{f}_{j+1}^n + \lambda^R \lambda^L (\mathbf{f}_{j+1}^n - \mathbf{f}_j^n)}{\lambda^R - \lambda^L},$$

where λ^R and λ^L are the minimum and maximum eigenvalues of \mathbf{A}_1 , respectively. Precisely, we have

$$(46) \quad \lambda^L = \bar{u}_1 - C_{M+1} \sqrt{\bar{\theta}}, \quad \lambda^R = \bar{u}_1 + C_{M+1} \sqrt{\bar{\theta}},$$

and C_{M+1} is the largest zero of the Hermite polynomial of degree $M+1$. Here the parameters are always chosen such that $\lambda^L < 0$ and $\lambda^R > 0$ to avoid advection only in one direction. Besides, the time step size is determined by the CFL condition

$$(47) \quad \Delta t \frac{|\bar{u}_1| + C_{M+1} \sqrt{\bar{\theta}}}{\Delta x} = \text{CFL} < 1.$$

In our actual implementation, we have upgraded this scheme to the second order by linear reconstruction with minmod limiter, Heun's time integrator, and the Strang splitting. Such strategies are standard techniques and can be found in many textbooks (e.g. [33]).

4.2. One-dimensional numerical examples. In this section, we present two numerical examples, both of which use the variable hard sphere model with $\nu = 5/9$ (see (41)). In our simulation, in order to prevent the computational cost from getting out of control, we set a cap for the value of M_0 to be 15, and we use the non-adaptive results with $M_0 = 15$ being fixed as our reference solution. All the numerical tests in this section are carried out on a desktop with CPU model Intel[®] Core[™] i7-7600U.

4.2.1. *Colliding flow.* We consider the colliding flow with the initial condition

$$(48) \quad f(x, \mathbf{v}, 0) = \frac{\rho(x)}{(2\pi\theta(x))^{3/2}} \exp\left(-\frac{|\mathbf{v} - \mathbf{u}(x)|^2}{2\theta(x)}\right)$$

with

$$(49) \quad \rho(x) = 1, \quad \mathbf{u}(x) = \begin{cases} (1, 0, 0)^T, & \text{if } x < 0, \\ (-1, 0, 0)^T, & \text{if } x > 0, \end{cases} \quad \theta(x) = 1/3.$$

We scale the collision term such that the Knudsen number equals 0.5. The initial condition consists of two equilibrium flows with the same temperature moving in opposite directions, and it is expected that the collision of the two Maxwellians will create some non-equilibrium effects, which require a relatively large M_0 to accurately capture the flow states.

The computational domain is set as $[-20, 20]$. In (2), we choose M to be 30, and $\bar{\mathbf{u}}$ and $\bar{\theta}$ are set to be $\mathbf{0}$ and 1, respectively. To determine the values of the thresholds ϵ_1 and ϵ_2 , we follow the strategy in Section 3.3 and carry out a test run for M_0 fixed to be 3 up to $t = 15$. The cell size is chosen to be $\Delta x = 0.4$ in all the test runs below. The numerical solutions of this test run at different times are given in Figure 1, which plots the density ρ and heat flux \mathbf{q} of the gas, which are defined by

$$(50) \quad \rho = \int_{\mathbb{R}^3} f(\mathbf{v}) \, d\mathbf{v}, \quad \mathbf{q} = \frac{1}{2} \int_{\mathbb{R}^3} |\mathbf{v} - \mathbf{u}|^2 (\mathbf{v} - \mathbf{u}) f(\mathbf{v}) \, d\mathbf{v},$$

where \mathbf{u} is the average velocity of gas molecules:

$$(51) \quad \mathbf{u} = \frac{1}{\rho} \int_{\mathbb{R}^3} \mathbf{v} f(\mathbf{v}) \, d\mathbf{v}.$$

For this one-dimensional flow, only the first component of \mathbf{q} is plotted. Due to the insufficient resolution of the solution and the small value of M_0 , the peak values of both density and heat flux are not well captured, but the general behavior of the flow is still qualitatively correct: the collision of the flow generates two shock waves moving in opposite directions, and the heat flux is nonzero inside these shock waves. During the test run, we record the maximum value of the indicator, which turns out to be

$$\epsilon_{\max} = 15.6.$$

This leads us to the following initial guess of ϵ_1 and ϵ_2 :

$$(\epsilon_1, \epsilon_2) = (4, 8) \approx (\epsilon_{\max}/4, \epsilon_{\max}/2).$$

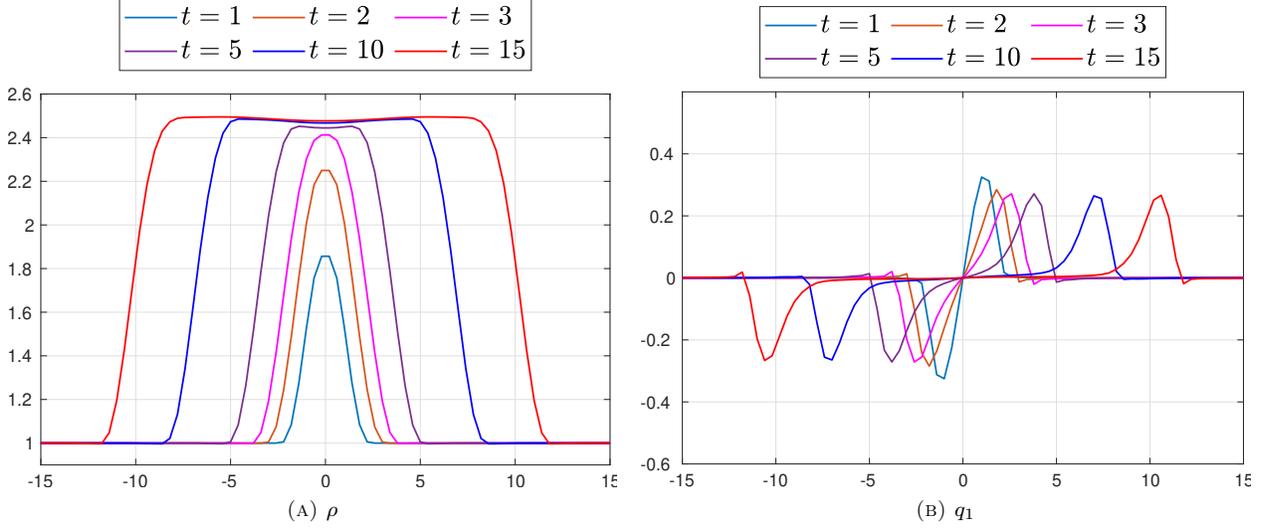


FIGURE 1. Plots of density and heat flux for the colliding flow with $M_0 = 3$ (fixed) at different times.

We now carry out a test with adaptive collision operators using this pair of parameters, and then reduce the value of ϵ_1 to 2 and run another test. The comparison of these two tests is given in Figure 2, where we only present the non-equilibrium variable q_1 that shows a more significant difference than the equilibrium variables. It can be seen from Figure 2b that the smaller value of ϵ_1 leads to slightly greater M_0 inside the shock wave. Since the graph of q_1 in Figure 2a still shows quite some difference between the two solutions, we further reduce ϵ_1 by a half and carry out another test run. The comparison of ϵ_1 to be 1. Figure 3. We are now satisfied with the small difference and will fix the value of ϵ_1 to be 1.

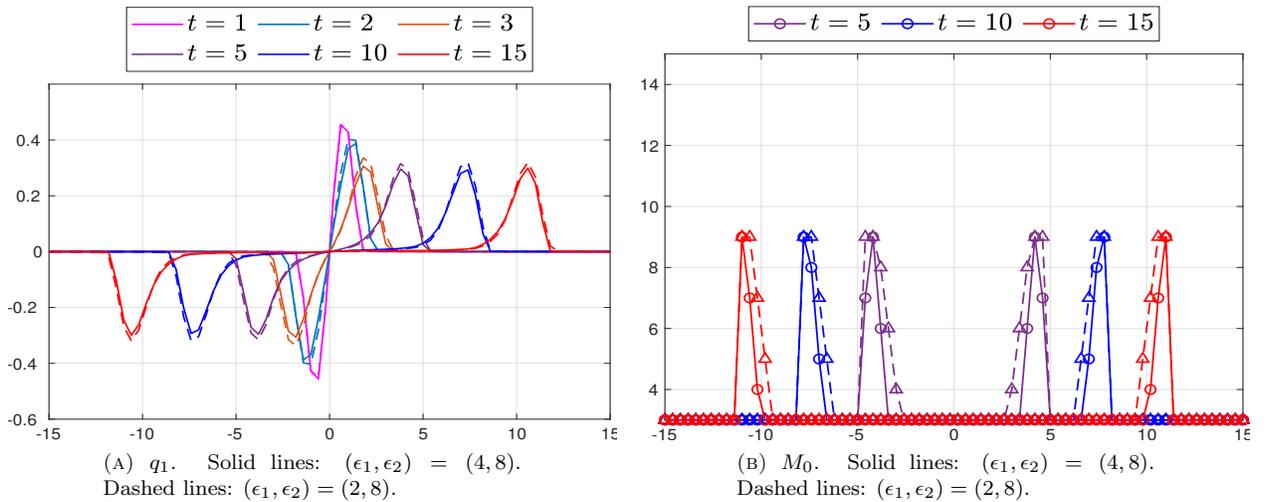


FIGURE 2. Profiles of heat flux and distributions of M_0 for the colliding flow with two different pairs of indicator thresholds $(\epsilon_1, \epsilon_2) = (4, 8)$ and $(\epsilon_1, \epsilon_2) = (2, 8)$.

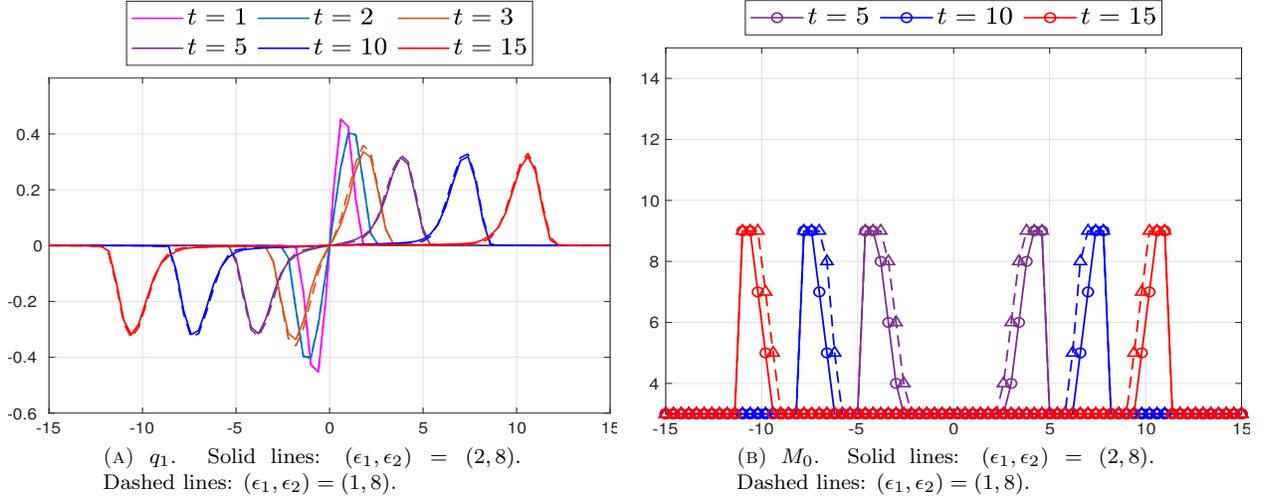


FIGURE 3. Profiles of heat flux and distributions of M_0 for the colliding flow with two different pairs of indicator thresholds $(\epsilon_1, \epsilon_2) = (2, 8)$ and $(\epsilon_1, \epsilon_2) = (1, 8)$.

The selection of ϵ_2 is done in a similar way. We reduce ϵ_2 from 8 to 4 and compare the results with the parameters $(\epsilon_1, \epsilon_2) = (1, 8)$ and $(\epsilon_1, \epsilon_2) = (1, 4)$. As shown in Figure 4, in a few grid cells inside the shock wave, the values of M_0 have been significantly increased, while the solution of the heat flux does not change too much. We therefore fix the value of ϵ_2 to be 4. The computational times for these test runs are given in Table 1, which look affordable due to the coarse grid size.

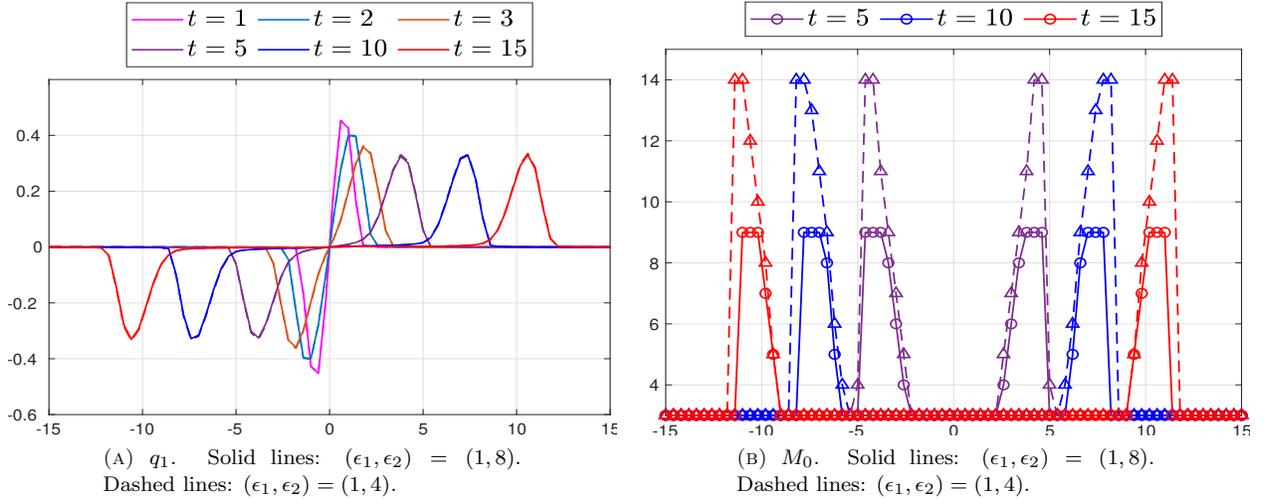


FIGURE 4. Profiles of heat flux and distributions of M_0 for the colliding flow with two different pairs of indicator thresholds $(\epsilon_1, \epsilon_2) = (1, 8)$ and $(\epsilon_1, \epsilon_2) = (1, 4)$.

Next, we refine the grid and set the cell size to be $\Delta x = 0.1$. With (ϵ_1, ϵ_2) chosen to be $(1, 4)$, we rerun the simulation up to $t = 15$. The numerical solutions at different times are given in Figure 5, including three equilibrium quantities (density ρ , velocity \mathbf{u} and temperature θ) and one non-equilibrium moment (heat flux \mathbf{q}). The temperature θ is related to the distribution function by

$$(52) \quad \theta = \frac{1}{3\rho} \int_{\mathbb{R}^3} |\mathbf{v} - \mathbf{u}|^2 f(\mathbf{v}) d\mathbf{v}.$$

TABLE 1. CPU times of the test run for the collision flow. The parameters $(0, +\infty)$ refers to the non-adaptive run with M_0 fixed to be 3.

(ϵ_1, ϵ_2)	$(0, +\infty)$	$(4, 8)$	$(2, 8)$	$(1, 8)$	$(1, 4)$
Total CPU time (s)	240.62	479.27	510.55	525.35	724.82

For the velocity and heat flux, only their first components are plotted due to the one-dimensional nature of the flow. Similar to our test runs, the flow structure emerges from the middle of the domain due to the interaction of the two Maxwellians, producing higher density and temperature. Then two shock waves are formed and move in opposite directions. After the two shock waves are separated, the center of the domain returns to local equilibrium state. We would like to emphasize that Figure 5 includes two sets of solutions (the reference solution and the self-adaptive solution) which almost coincide.

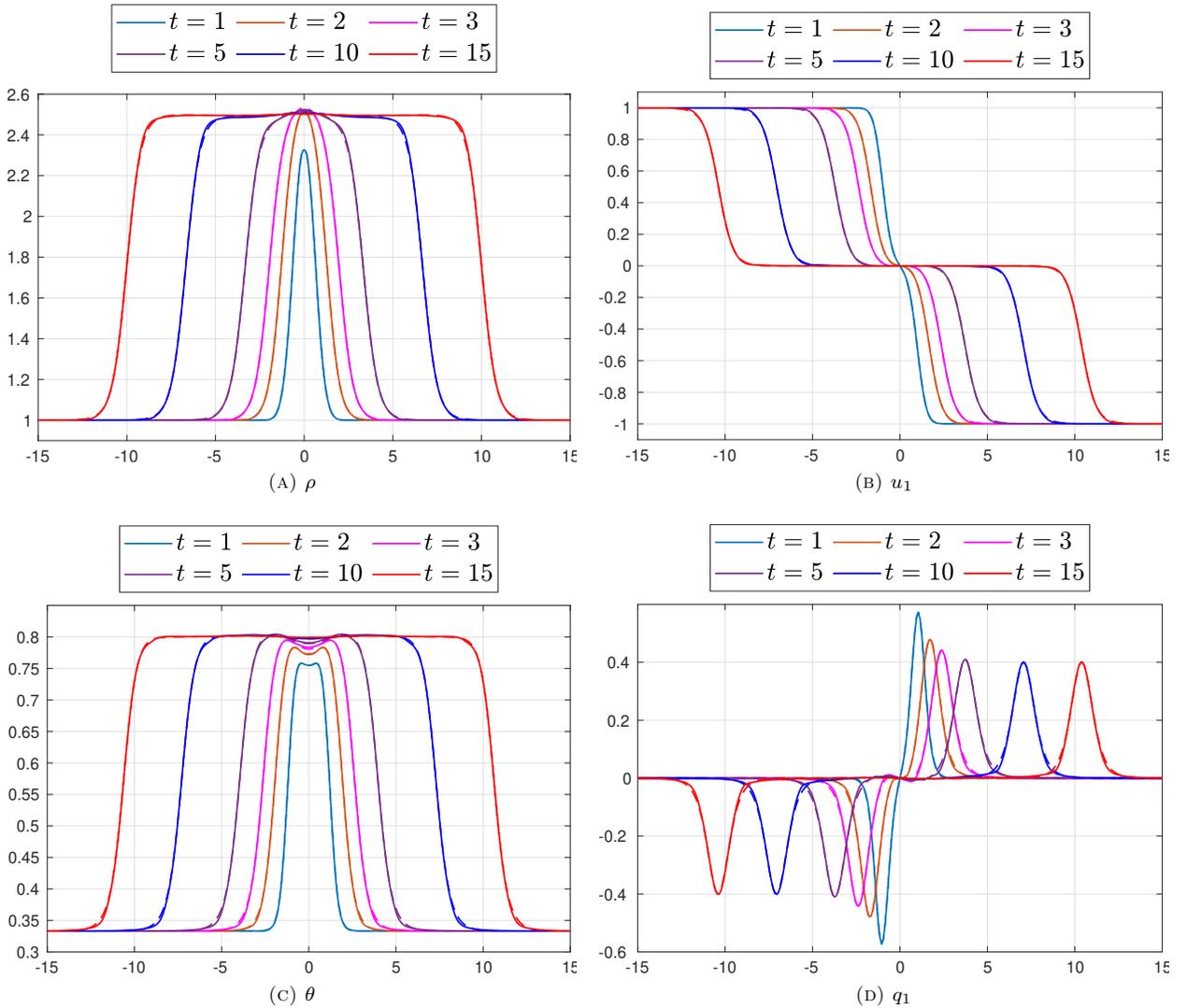


FIGURE 5. Solution of the colliding flow at different times. The solid lines are the numerical solution of the adaptive algorithm and the dashed lines are the reference solution.

The evolution of the distribution of M_0 is provided in Figure 6. Initially, we set $M_0 = 15$ on all grid cells. In a few time steps, this drops to 3 almost everywhere except the center of the domain. Afterward,

the evolution of M_0 well agrees with the evolution of the non-equilibrium. During the simulation, most part of the domain is in the local equilibrium state, where M_0 stays at its lowest value 3, requiring much less computational cost. Consequently, as shown in Table 2, the total CPU time is significantly reduced compared with the simulation using a uniform M_0 . Moreover, the evaluation of the error indicator only takes a relatively small portion of the total computational time, which agrees with the goal we set in Section 1. It is also worth mentioning that the total computational time is 3993.29s, which is even longer than the sum of all our test runs.

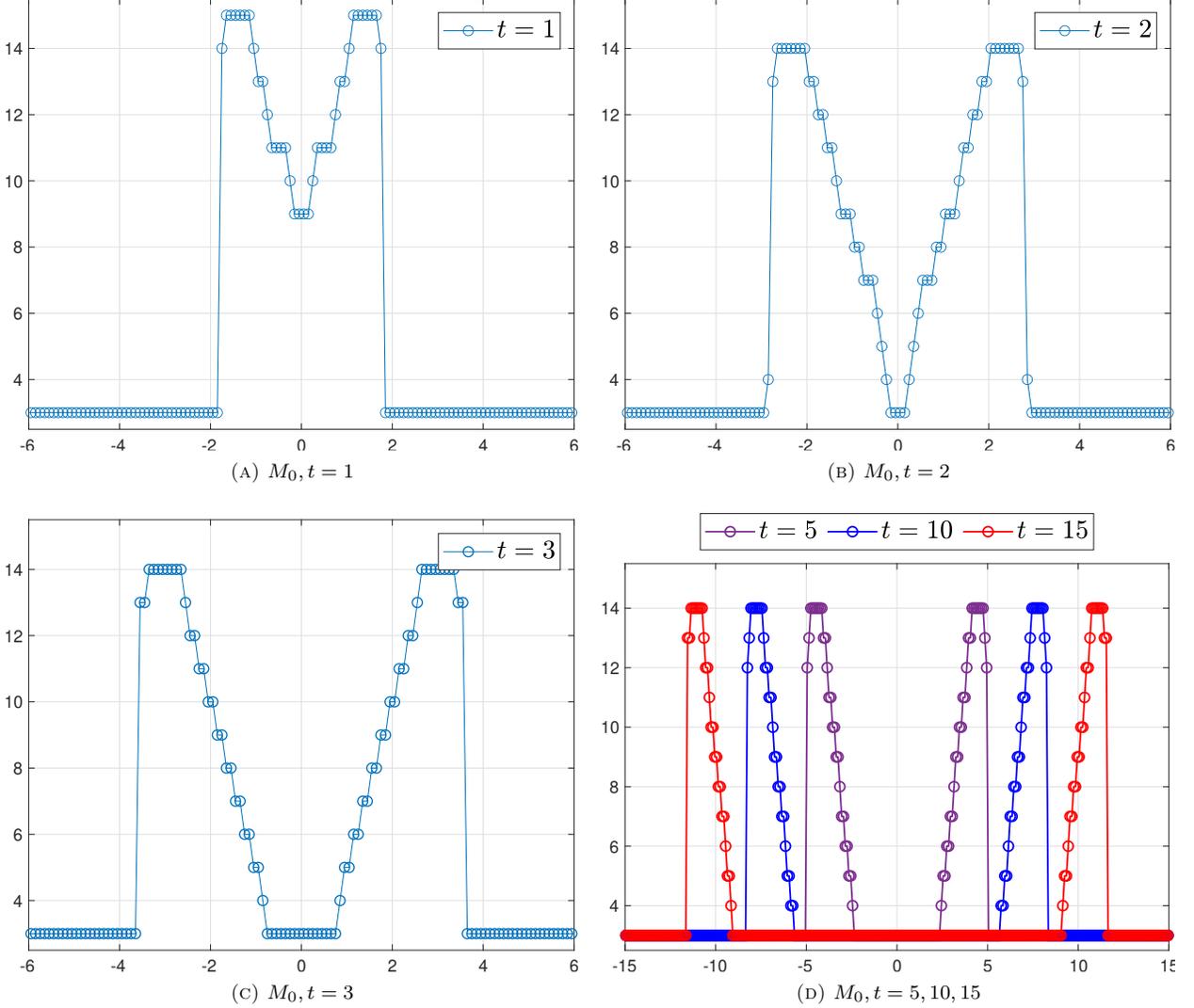


FIGURE 6. Distribution of M_0 for the colliding flow at different times.

TABLE 2. Statistical data for the colliding flow. T_{ref} and T_{adp} refer to the average CPU time per time step for the reference solution and the self-adaptive solution, respectively, and T_{ind} refers to the average CPU time per time step for the computation of the error indicator.

T_{ref}	T_{adp}	T_{ind}	$1 - T_{\text{adp}}/T_{\text{ref}}$	$T_{\text{ind}}/T_{\text{adp}}$
16.96s	1.06s	0.108s	93.7%	10.1%

4.2.2. *Planar Couette flow.* The planar Couette flow is a commonly used benchmark problem for the one-dimensional Boltzmann equation. We assume that the gas between two infinite parallel plates has an initial temperature $\theta = 1$, and the two plates move in the opposite directions with velocities parallel to the plates. The speeds of both plates are 0.5, and the distance between the two plates is $L = 1$. Both plates are assumed to be completely diffusive, meaning that for any particle hitting the wall, the reflected velocity is completely independent of the incident velocity. Instead, the distribution of the reflected velocity follows the Maxwellian with the wall velocity being the center and wall temperature being the variance. The implementation of the boundary condition in the Burnett spectral method has been detailed in [24, Section 3.4]. The initial state of the fluid is set to be a uniform Maxwellian with density $\rho = 1$, velocity $\mathbf{u} = \mathbf{0}$ and temperature $\theta = 1$. Driven by the motion of the plates, the flow will reach a steady state as time approaches infinity. Here we choose the Knudsen number to be 0.5, for which a strong non-equilibrium can be expected, especially on the boundary of the domain where the distribution function is discontinuous.

Numerically, we set $\bar{\mathbf{u}} = \mathbf{0}$, $\bar{\theta} = 1$ and $M = 30$ in (2). A uniform grid with 200 cells is used for spatial discretization, and the thresholds of the error indicator are set to be $\epsilon_1 = 1$ and $\epsilon_2 = 8$. Figure 7 shows the numerical solution of the four moments defined in (50)(51) and (52), and Figure 8 shows the evolution of the parameter M_0 . During the evolution to the steady state, some small differences between the self-adaptive solution and the reference solution can be observed. The discrepancy of the velocity profiles appears to be the most significant due to its small magnitude. In this example, large M_0 only appears near the boundary of the domain for small t , since the central part of the domain is still mostly in the initial equilibrium state. As the boundary effect propagates inward, the value of M_0 gradually increases. Interestingly, the distribution of M_0 reaches the “steady state” earlier than the fluid does. As shown in Figures 8c and 8d, at $t = 1.0$, while the fluid structure is still evolving, M_0 does not change with time any more. Compared to the example in Section 4.2.1, the non-equilibrium spreads more widely in this case, resulting in less reduction of the computational cost (see Table 3). Nevertheless, the CPU time per time step is still reduced to nearly one-sixth. Here, the computation of the indicator takes a smaller portion since longer time is spent on the evaluation of the collision term.

TABLE 3. Statistical data for the planar Couette flow. T_{ref} and T_{adp} refer to the CPU time per time step for the reference solution and the self-adaptive solution, respectively, and T_{ind} refers to the CPU time per time step for the computation of the error indicator.

T_{ref}	T_{adp}	T_{ind}	$1 - T_{\text{adp}}/T_{\text{ref}}$	$T_{\text{ind}}/T_{\text{adp}}$
8.54s	1.51s	0.057s	82.3%	3.78%

4.3. **Two-dimensional examples.** In our two-dimensional examples, we also consider the variable hard sphere model with $\nu = 5/9$, and M_0 is still capped at 15. Two examples with and without boundary conditions will be considered in the following two subsections.

4.3.1. *Fluid diffusion.* Our first two-dimensional example considers the initial data

$$f(x_1, x_2, \mathbf{v}, 0) = \frac{\rho(x_1, x_2)}{(2\pi\theta(x_1, x_2))^{3/2}} \exp\left(-\frac{|\mathbf{v} - \mathbf{u}(x_1, x_2)|^2}{2\theta(x_1, x_2)}\right),$$

where $\mathbf{u}(x_1, x_2) = \mathbf{0}$ and $\theta(x_1, x_2) = 1$ for all x_1 and x_2 , while $\rho(x)$ is set to be

$$\rho(x_1, x_2) = \begin{cases} 10, & \text{if } |x_1| \leq 0.05 \text{ and } |x_2| \leq 0.05, \\ 1, & \text{otherwise.} \end{cases}$$

We set the computational domain to be $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$ and apply the Neumann boundary condition to simulate the flow in the unbounded domain. In this example, there is a high density region in the center of the domain, and we are interested in the dynamics of its diffusion into the background fluid. Here we choose the Knudsen number to be $Kn = 0.05$. The parameters in (2) are set to be $M = 30$, $\bar{\mathbf{u}} = \mathbf{0}$ and $\bar{\theta} = 1$. A uniform grid of size 200×200 is utilized here to discretize the physical space. The thresholds of the error indicator are chosen as $(\epsilon_1, \epsilon_2) = (2.5, 8.5)$.

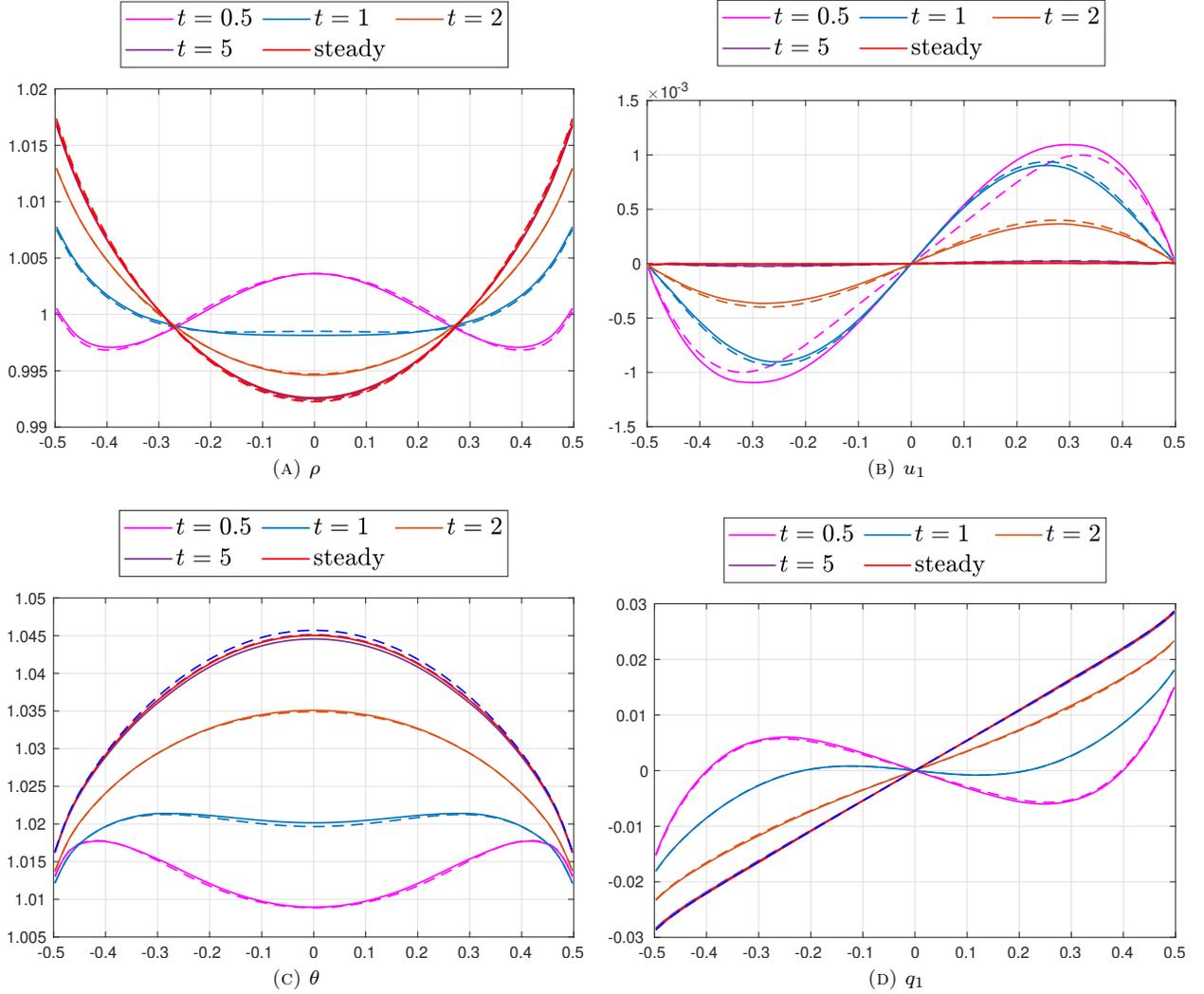


FIGURE 7. Solution of the Couette flow at different times. The solid lines are the numerical solution of the adaptive algorithm and the dashed lines are the reference solution.

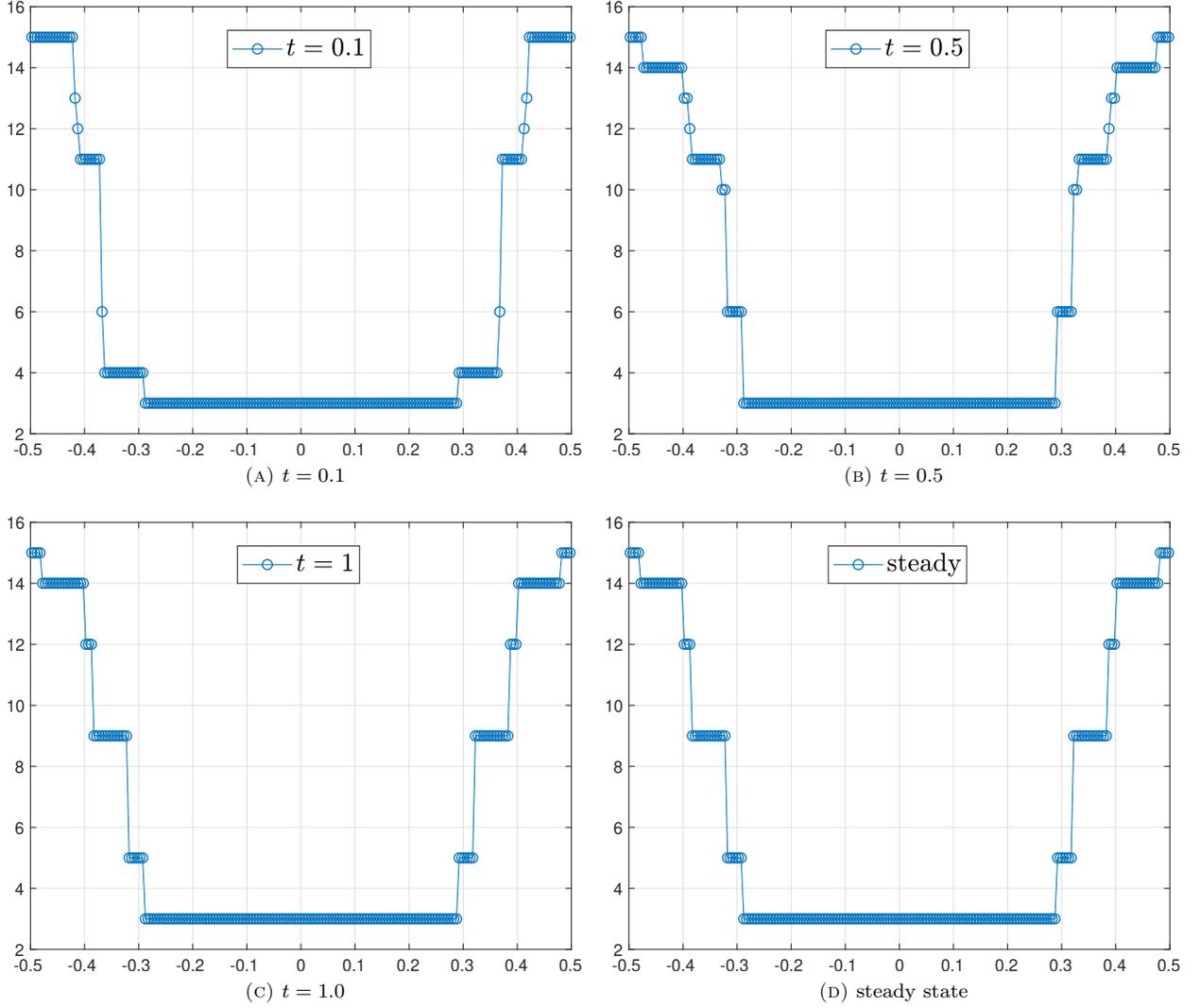
We plot the evolution of the fluid states in Figure 9, with reference solutions computed by using $M_0 = 15$ everywhere. Besides the equilibrium variables density ρ and temperature θ , we have also plotted the shear stress σ_{12} , which is related to the distribution function by

$$(53) \quad \sigma_{12} = \int_{\mathbb{R}^3} (v_1 - u_1)(v_2 - u_2) f(\mathbf{v}) d\mathbf{v}.$$

It can be seen that the density in the center of the domain gradually decreases, and as the mass flows out, the temperature also starts to decrease so that the total energy can be conserved. Due to the symmetry of the initial data, the value of the non-equilibrium variable σ_{12} equals zero on both x - and y -axes. As the fluid evolves with time, the non-equilibrium effect spreads out, while the peak values of σ_{12} start to decrease. With our adaptive method, these phenomena can be accurately captured. To get a clearer view of the difference between the adaptive solutions and the reference solutions, we define

$$(54) \quad \mathcal{E}_\rho = \rho^{\text{adp}} - \rho^{\text{ref}}, \quad \mathcal{E}_\theta = \theta^{\text{adp}} - \theta^{\text{ref}}, \quad \mathcal{E}_{\sigma_{12}} = \sigma_{12}^{\text{adp}} - \sigma_{12}^{\text{ref}},$$

where the superscripts “adp” and “ref” denote the adaptive solution and the reference solution, respectively. These quantities are plotted in Figure 10, and the corresponding relative L^2 differences are given in Table 4. One can observe that the difference between the two solutions increases with time due to the accumulation


 FIGURE 8. Distribution of M_0 for the Couette flow at different times.

of the error. This also implies that the thresholds ϵ_1 and ϵ_2 , whose values stay the same throughout the simulation, do not directly correspond to the error of the solution. Our error indicator only estimates the local truncation error, which may accumulate in time-dependent problems. In such circumstances, to ensure the numerical accuracy for longer simulations, one may need to choose smaller values of ϵ_1 and ϵ_2 . Such an effect is automatically incorporated into the procedure of parameter selection introduced in Section 3.3 if the test runs are also preformed until the desired final time.

 TABLE 4. Relative L_2 difference between the self-adaptive solution and the reference solution.

	$t = 0.04$	$t = 0.08$	$t = 0.12$	$t = 0.15$
$\ \mathcal{E}_\rho\ _{L^2}/\ \rho^{\text{ref}}\ _{L^2}$	0.01%	0.03%	0.11%	0.24%
$\ \mathcal{E}_\theta\ _{L^2}/\ \theta^{\text{ref}}\ _{L^2}$	0.01%	0.04%	0.12%	0.20%
$\ \mathcal{E}_{\sigma_{12}}\ _{L^2}/\ \sigma_{12}^{\text{ref}}\ _{L^2}$	0.30%	0.84%	1.29%	2.01%

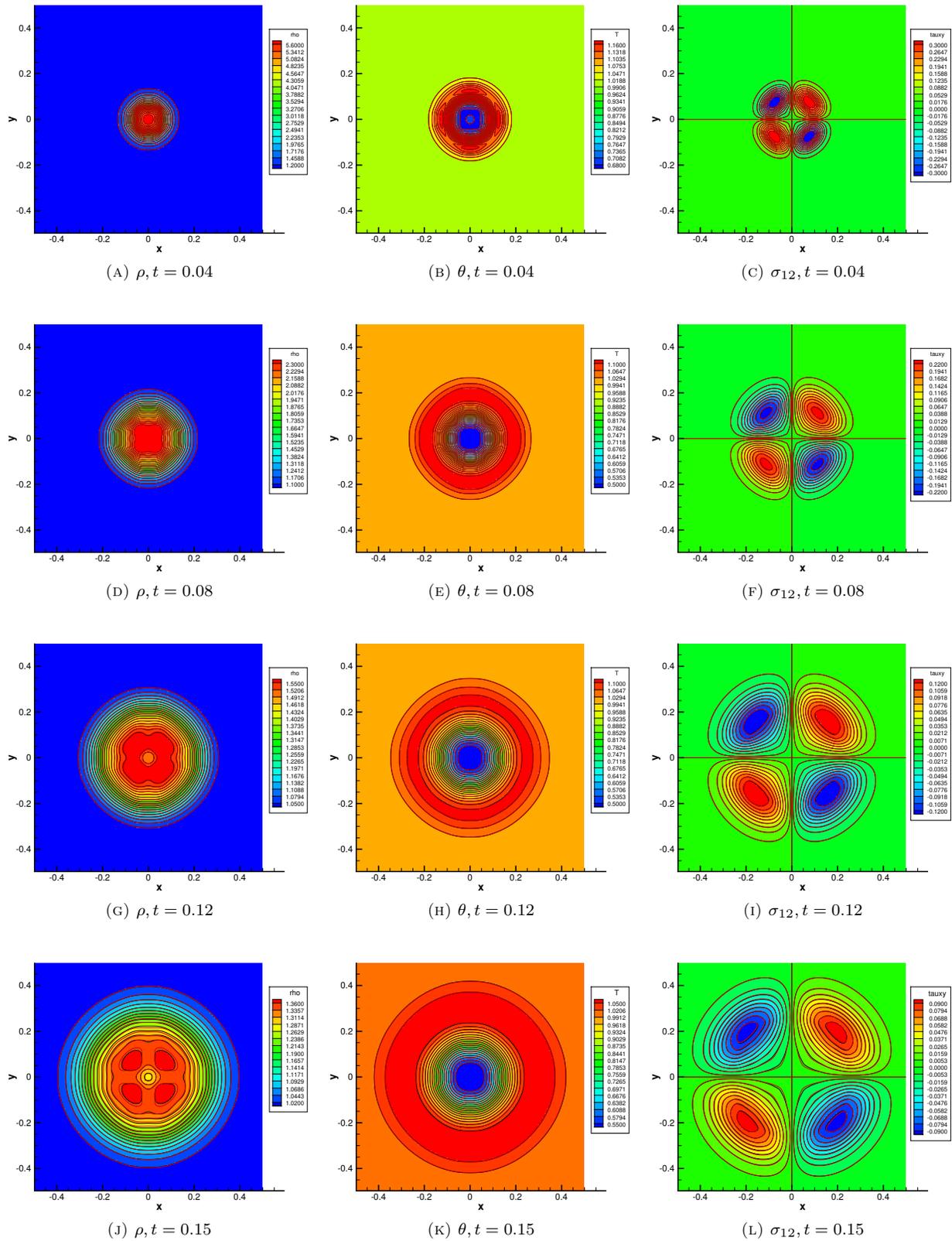


FIGURE 9. Solution of the fluid diffusion problem at different times. The red contours are the numerical solutions of the adaptive method and the black contours are the reference solutions.

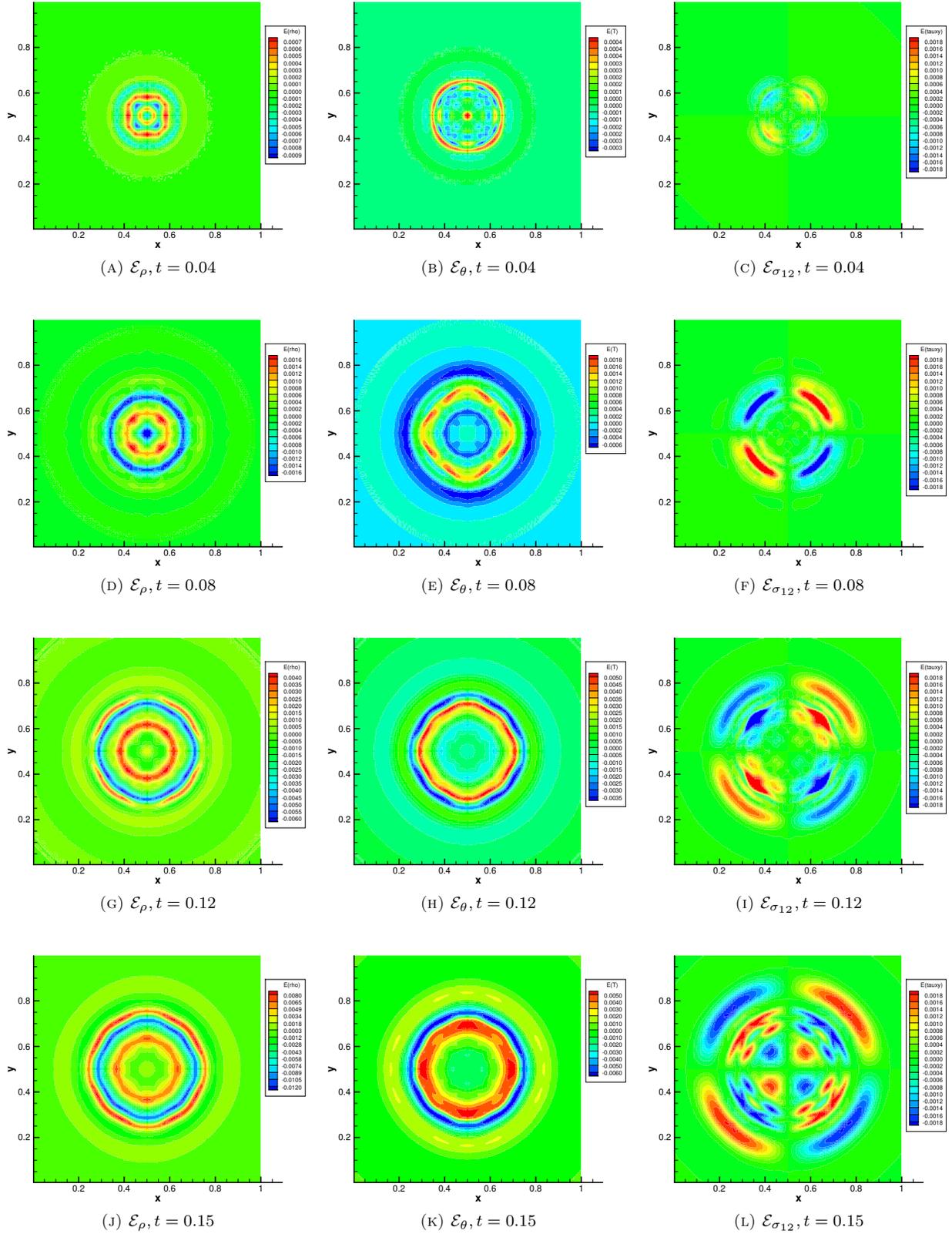


FIGURE 10. Error (54) of the fluid diffusion problem at different times.

The evolution of the distribution of M_0 is given in Figure 11. Initially, the mixing of two fluid regions creates some non-equilibrium. However, due to the high density in the central part of the domain, fast collisions of particles keep the fluid near its local equilibrium, so that M_0 is generally not too large at $t = 0.04$. As t increases, both the density and the temperature in the central area decrease, and correspondingly, M_0 needs to be increased to capture the non-equilibrium effects. From $t = 0.12$ to $t = 0.15$, although the fluid has spread more widely, the outside layers are almost in the equilibrium states, and therefore the distribution of M_0 does not change significantly. Compared with the reference solution, the average CPU time per time step is reduced from 1473 seconds to 123.7 seconds using our method.

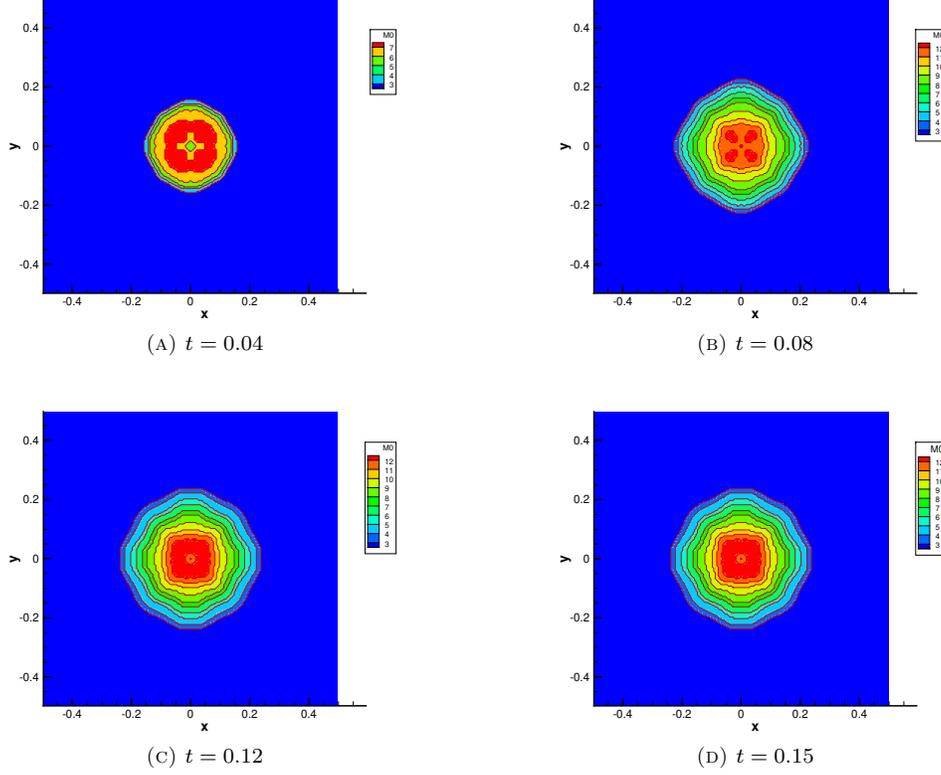


FIGURE 11. Distribution of M_0 for fluid diffusion at different times.

4.3.2. Lid-driven cavity flow. Our second example assumes that the gas is in a square cavity $\Omega = [0, 1] \times [0, 1]$, and we scale the collision term such that the Knudsen number is 0.1. The top lid of the cavity moves horizontally at a constant speed $v = 0.0208$, and all the four sides of Ω are assumed to be fully diffusive. Initially, the fluid is in the equilibrium state with density $\rho = 1$, velocity $\mathbf{u} = \mathbf{0}$ and temperature $\theta = 1$. The friction between the lid and the gas causes the rotation of the fluid, and a steady state will be developed after a sufficiently long time. Such an example has been widely studied in the literature [28, 35]. Here we discretize the domain Ω with a uniform grid of size 100×100 . The parameters in (2) are set to be $M = 40$, $\bar{\mathbf{u}} = \mathbf{0}$ and $\bar{\theta} = 1$. The reference solution will again be provided by the uniform M_0 equal to 15.

To study the effect of adaptive parameters, we consider two groups of thresholds: $(\epsilon_1^1, \epsilon_2^1) = (0.05, 0.20)$ and $(\epsilon_1^2, \epsilon_2^2) = (0.025, 0.08)$. Clearly, the second set of parameters is tighter and will lead to larger M_0 in the simulations. The evolution of the fluid states is plotted in Figure 12, which includes the density ρ , the temperature θ , and shear stress σ_{12} at time $t = 0.5, 1, 5$ and the steady state. One can observe the singular flow structure in the top two corners of the cavity, where the distributions are distorted due to the inconsistent boundary velocities. In Figure 12, three sets of solutions generally agree with each other. Some differences can be observed in the second column representing the temperature contours. Despite this, the relative difference in temperature between our results and the reference solution is well below 0.05%. In the

first and third columns, all the three sets of contour lines almost coincide with each other, indicating the effectiveness of our error indicator.

The distribution of M_0 is given in Figure 13. Since the flow is driven by the movement of the top lid, non-equilibrium emerges from the upper part of the domain, and then expands downward as t increases. For the first set of parameters (left column), the value of M_0 reaches the cap 15 only near the boundary of the domain, where the distribution function is discontinuous, while in the right column, more than a half of the grid cells are covered by the collision term with $M_0 = 15$, which is consistent with our prediction.

4.4. Discussion on the choice of parameters. In the previous numerical examples, one can observe that the choice of the thresholds ϵ_1 and ϵ_2 appears to be quite problem-dependent. This is mainly due to the different requirements of the numerical accuracy in different problems. Generally speaking, for flows with larger fluctuations such as Section 4.3.1, we tend to choose a larger pair of parameters since the small error is less noticeable, while for the lid-driven cavity flow in Section 4.3.2, the parameters are chosen smaller since the contour lines are more sensitive to the numerical error. In practice, when the flow structure is complicated, the flows in different areas may have different features, which may require different thresholds to obtain proper relative errors. To achieve this, a straightforward method is to add the spatial variable x to both thresholds. Then the method in Section 3.3 can still be applied to determine $\epsilon_1(x)$ and $\epsilon_2(x)$. Further study of this approach will be considered in our future work.

5. SUMMARY AND OUTLOOK

This paper contributes to the efficient simulation of the Boltzmann equation with the quadratic collision operator. Instead of a full discretization of the binary collision term, we choose to replace part of it with the BGK simulation, and the choice of ‘‘BGK part’’ changes with the distribution function. To make proper choices adaptively, we construct our error indicator based on a novel idea that uses a cheaper linear operator to control some quadratic parts of the error term, so that even in the case where the full binary collision operator has to be used widely, our adaptive method does not slow down the computation. Our numerical simulation shows the affordability and reliability of our indicators.

The error indicator introduced in this paper is specially designed for the Burnett spectral method, while we expect that the same idea can be applied to other approaches such as the Fourier spectral method, which has lower time complexity. As the Fourier spectral method is also much cheaper for certain particular models [16], we are exploring the possibilities of such extensions.

APPENDIX A. CHOICE OF THE PARAMETER ν_{M_0}

The parameter ν_{M_0} in the approximate collision term (7) is chosen following [7, 40, 6]. It can be obtained by the following steps:

- Set $\bar{\mathbf{u}}$ and $\bar{\theta}$ to be the velocity \mathbf{u} and temperature θ , respectively, so that

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}^{(1)} \\ \mathbf{M}^{(2)} \end{pmatrix} = (\rho, 0, \dots, 0)^T.$$

- Define the linearized collision operator

$$\mathbf{L}(\mathbf{f}^{(1)}) = \mathbf{Q}_{M_0} : (\mathbf{f}^{(1)} \otimes \mathbf{M}^{(1)}).$$

Since $\mathbf{M}^{(1)}$ denotes an isotropic distribution function, the operator can be expressed by

$$g_{lmn} = \sum_{n'=0}^{\lfloor (M_0-l)/2 \rfloor} a_{l n n'}^0 f_{l m n'}, \quad l = 0, 1, \dots, M_0, \quad m = -l, \dots, l, \quad n = 0, \dots, \lfloor (M_0 - l)/2 \rfloor$$

where $f_{l m n'}$ are the components of $\mathbf{f}^{(1)}$ and $g_{l m n}$ are the components of $\mathbf{L}(\mathbf{f}^{(1)})$.

- Set ν_{M_0} to be the spectral radius of \mathbf{L} , which can be computed via

$$\nu_{M_0} = \max_{l=0,1,\dots,M_0} \max\{|\lambda| : \lambda \text{ is the eigenvalue of the matrix } \mathbf{A}_l = (a_{l n n'}^0)\}.$$

The coefficients $a_{l n n'}^0$ are given in (34), and the matrix eigenvalues are numerically computed.

When $M_0 = 2$, the matrix \mathbf{A}_0 is a 2×2 matrix, and \mathbf{A}_1 and \mathbf{A}_2 are scalars. Due to the conservation of mass, momentum and energy, we have $\mathbf{A}_0 = 0$, $\mathbf{A}_1 = 0$. Thus, the absolute value of the only coefficient a_{200}^0 in \mathbf{A}_2 provides the value of ν_{M_0} . This coefficient indicates the decay rate of the stress tensor, which is often used as the collision frequency in the BGK model.

APPENDIX B. PROOF OF THEOREM 4

Lemma 5. *Given non-negative indices $l, n, n', l_1, n_1, l_2, n_2$ and integer indices $m_1 \in [-l_1, l_1]$ and $m_2 \in [-l_2, l_2]$, the integral*

$$\int_{\mathbb{R}^3} \int_{\mathbb{R}^3} p_{l_1 m_1 n_1}^\dagger(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^{0\dagger} \left(\frac{\mathbf{g}}{\sqrt{2}} \right) p_{l_0 n} \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) p_{00 n'} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) \omega \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) \omega \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) d\mathbf{g} d\mathbf{h}$$

is nonzero only if $l_1 + l_2 + 2(n_1 + n_2) = l + 2(n + n')$ and $m_1 + m_2 = 0$.

This conclusion can be found in [31, eqs. (112)(114)].

Proof of Theorem 4. Due to the orthogonality of the polynomials p_{lmn} , we know that

$$(55) \quad A_{lnn'}^{l_1 l_2 m_2 n_2} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} p_{l_1 m_1 n_1}^\dagger(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^{0\dagger} \left(\frac{\mathbf{g}}{\sqrt{2}} \right) p_{l_0 n} \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) p_{00 n'} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) \times \omega \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) \omega \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) d\mathbf{g} d\mathbf{h},$$

where $m_1 = -m_2$ and $n_1 = l - l_1 - l_2 + 2(n + n' - n_2)$. For simplicity, we assume that $\bar{\mathbf{u}} = 0$, and in the case of nonzero $\bar{\mathbf{u}}$, the result can be obtained by translation. To derive the recurrence relation of $A_{lnn'}^{l_1 l_2 m_2 n_2}$, we use the recurrence relation of Laguerre polynomials to get

$$p_{00, n'+1} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) = -\frac{1}{2\sqrt{(n+1)(n+3/2)}} \bar{\theta}^{-1} \left\| \mathbf{h} - \frac{1}{2}\mathbf{g} \right\|^2 p_{00 n'} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) + (\text{Lower degree polynomials}),$$

where ‘‘lower degree polynomials’’ refers to the polynomials of \mathbf{g} and \mathbf{h} of degree less than $2(n' + 1)$. Due to the orthogonality of p_{lmn} , these terms will vanish when calculating the integral (55) with n' replaced by $n' + 1$:

$$(56) \quad A_{ln, n'+1}^{l_1 l_2 m_2 n_2} = -\frac{1}{2\sqrt{(n+1)(n+3/2)}} \bar{\theta}^{-1} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \left(h^2 + \frac{1}{4}g^2 - \mathbf{h} \cdot \mathbf{g} \right) p_{l_1 m_1 n_1}^\dagger(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^{0\dagger} \left(\frac{\mathbf{g}}{\sqrt{2}} \right) \times p_{l_0 n} \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) p_{00 n'} \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) \omega \left(\mathbf{h} + \frac{1}{2}\mathbf{g} \right) \omega \left(\mathbf{h} - \frac{1}{2}\mathbf{g} \right) d\mathbf{g} d\mathbf{h}.$$

Using

$$(57) \quad \bar{\theta}^{-1} \left(h^2 + \frac{1}{4}g^2 \right) p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) = -\sum_{k=1}^2 \sqrt{n_k(n_k + l_k + 1/2)} p_{l_1, m_1, n_1 - \delta_{1k}}(\sqrt{2}\mathbf{h}) p_{l_2, m_2, n_2 - \delta_{2k}}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) + (\text{Higher degree polynomials})$$

and

$$(58) \quad \frac{1}{2} \bar{\theta}^{-1} (\mathbf{h} \cdot \mathbf{g}) p_{l_1 m_1 n_1}(\sqrt{2}\mathbf{h}) p_{l_2 m_2 n_2}^0 \left(\frac{\mathbf{g}}{\sqrt{2}} \right) = \sum_{\mu=-1}^1 (-1)^\mu \left[\sqrt{l_1 + n_1 + 1/2} \gamma_{l_1, m_1 - \mu}^\mu p_{l_1 - 1, m_1 - \mu, n_1}(\sqrt{2}\mathbf{h}) - (-1)^\mu \sqrt{n_1} \gamma_{-l_1 - 1, m_1 - \mu}^\mu p_{l_1 + 1, m_1 - \mu, n_1 - 1}(\sqrt{2}\mathbf{h}) \right] \times \left[\sqrt{l_2 + n_2 + 1/2} \gamma_{l_2, m_2 + \mu}^{-\mu} p_{l_2 - 1, m_2 + \mu, n_2}(\sqrt{2}\mathbf{h}) - (-1)^\mu \sqrt{n_2} \gamma_{-l_2 - 1, m_2 + \mu}^{-\mu} p_{l_2 + 1, m_2 + \mu, n_2 - 1}(\sqrt{2}\mathbf{h}) \right] + (\text{Higher degree polynomials}).$$

We refer the readers to [7, Appendix B] for the derivation of these equations. In the above two equations, ‘‘higher degree polynomials’’ refer to the orthogonal polynomials of \mathbf{g} and \mathbf{h} whose degrees are higher than

$l + 2(n + n') - 1$, and such terms will vanish after substituting (57) and (58) into (56). The recurrence relation (37) can be obtained by such substitution and using the properties

$$m_1 + m_2 = 0, \quad \gamma_{l,m}^\mu = \gamma_{l,-m}^{-\mu}. \quad \square$$

REFERENCES

1. M.R.A. Abdelmalik and E.H. van Brummelen, *Error estimation and adaptive moment hierarchies for goal-oriented approximations of the Boltzmann equation*, Comput. Methods Appl. Mech. Engrg. **325** (2017), 219–239.
2. A. Alekseenko and E. Josyula, *Deterministic solution of the spatially homogeneous Boltzmann equation using discontinuous Galerkin discretizations in the velocity space*, J. Comput. Phys. **272** (2014), 170–188.
3. C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens, *Locally refined discrete velocity grids for stationary rarefied flow simulations*, J. Comput. Phys. **257** (2014), 572–593.
4. M. Bennoune, M. Lemou, and L. Mieussens, *Uniformly stable numerical schemes for the Boltzmann equation preserving the compressible Navier-Stokes asymptotics*, J. Comput. Phys. **227** (2008), 3781–3803.
5. A. V. Bobylev and S. Rjasanow, *Difference scheme for the Boltzmann equation based on the fast Fourier transform*, Eur. J. Mech. B Fluids **16** (1997), 293–306.
6. Z. Cai, Y. Fan, and Y. Wang, *Burnett spectral method for the spatially homogeneous Boltzmann equation*, Comput. Fluids **200** (2020), 104456.
7. Z. Cai and M. Torrilhon, *Approximation of the linearized Boltzmann collision operator for hard-sphere and inverse-power-law models*, J. Comput. Phys. **295** (2015), 617–643.
8. ———, *Numerical simulation of microflows using moment method with linearized collision operator*, J. Sci. Comput. **74** (2018), 336–374.
9. Z. Cai and Y. Wang, *Regularized 13-moment equations for inverse power law models*, J. Fluid Mech. **894** (2020), A12.
10. C. Cercignani, *Slow rarefied flows: Theory and application to micro-electro-mechanical systems*, Progress in Mathematical Physics, vol. 41, Birkhäuser, 2006.
11. S. Chen, K. Xu, C. Lee, and Q. Cai, *A unified gas kinetic scheme with moving mesh and velocity space adaptation*, J. Comput. Phys. **231** (2012), no. 20, 6643–6664.
12. P. Degond and G. Dimarco, *Fluid simulations with localized boltzmann upscaling by direct simulation Monte-Carlo*, J. Comput. Phys. **231** (2012), 2414–2437.
13. P. Degond, S. Jin, and L. Mieussens, *A smooth transition model between kinetic and hydrodynamic equations*, J. Comput. Phys. **209** (2005), no. 2, 665–694.
14. P. Degond, L. Pareschi, and G. Russo, *Modeling and computational methods for kinetic equations*, Modeling and Simulation in Science, Engineering and Technology, Birkhäuser Basel, 2004.
15. G. Dimarco, R. Loubère, J. Narski, and T. Rey, *An efficient numerical method for solving the Boltzmann equation in multidimensions*, J. Comput. Phys. **353** (2018), 46–81.
16. F. Filbet, C. Mouhot, and L. Pareschi, *Solving the Boltzmann equation in $N \log_2 N$* , SIAM J. Sci. Compute. **28** (2006), no. 3, 1029–1053.
17. F. Filbet, L. Pareschi, and T. Rey, *On steady-state preserving spectral methods for homogeneous Boltzmann equations*, C. R. Acad. Sci. Paris, Ser. I **353** (2015), 309–314.
18. F. Filbet and T. Rey, *A hierarchy of hybrid numerical methods for multiscale kinetic equations*, SIAM J. Sci. Comput. **37** (2015), no. 3, A1218–A1247.
19. F. Filbet and T. Xiong, *A hybrid discontinuous Galerkin scheme for multi-scale kinetic equations*, J. Comput. Phys. **372** (2018), 841–863.
20. I. M. Gamba, J. R. Haack, C. D. Hauck, and J. Hu, *A fast spectral method for the Boltzmann collision operator with general collision kernels*, SIAM J. Sci. Comput. **39** (2017), no. 14, B658–B674.
21. I. M. Gamba and S. Rjasanow, *Galerkin-Petrov approach for the Boltzmann equation*, J. Comput. Phys. **366** (2018), 341–365.
22. H. Grad, *On the kinetic theory of rarefied gases*, Comm. Pure Appl. Math. **2** (1949), no. 1, 331–407.
23. A. Harten, P. D. Lax, and B. Van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Review **25** (1983), no. 1, 35–61.
24. Z. Hu and Z. Cai, *Burnett spectral method for high-speed rarefied gas flows*, SIAM J. Sci. Comput. **42** (2020), no. 5, B1193–B1226.
25. Z. Hu, Z. Cai, and Y. Wang, *Numerical simulation of microflows Hermite spectral methods*, SIAM J. Sci. Comput. **42** (2020), B105–B134.
26. E. Ikenberry and C. Truesdell, *On the pressures and the flux of energy in a gas according to Maxwell’s kinetic theory, I*, J. Rat. Mech. Anal. **5** (1956), no. 1, 1–54.
27. S. Jaiswal, A. A. Alexeenko, and J.W. Hu, *A discontinuous Galerkin fast spectral method for the full Boltzmann equation with general collision kernels*, J. Comput. Phys. **378** (2019), 178–208.
28. B. John, X. J. Gu, and D. R. Emerson, *Investigation of heat and mass transfer in a lid-driven cavity under nonequilibrium flow conditions*, Numer. Heat Tr. B-fund. **58** (2010), 287–303.
29. G. Kitzler and J. Schöberl, *A polynomial spectral method for the spatially homogeneous boltzmann equation*, SIAM J. Sci. Comput. **41** (2019), no. 1, B27–B49.
30. V.I. Kolobov, R.R. Arslanbekov, V.V. Aristov, A.A. Frolova, and S.A. Zabelok, *Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement*, J. Comput. Phys. **223** (2007), no. 2, 589–608.

31. K. Kumar, *Polynomial expansions in kinetic theory of gases*, Ann. Phys. **37** (1966), no. 1, 113–141.
32. V.I. Lebedev, *Values of the nodes and weights of ninth to seventeenth order gauss-markov quadrature formulae invariant under the octahedron group with inversion*, USSR Comput. Math. Math. Phys. **15** (1975), no. 1, 44–51.
33. R. J. Leveque, *Finite volume methods for hyperbolic problems*, Cambridge, 2002.
34. C. D. Levermore, W. J. Morokoff, and B. T. Nadiga, *Moment realizability and the validity of the Navier–Stokes equations for rarefied gas dynamics*, Phys. Fluids **10** (1998), no. 12, 3214–3226.
35. C. Liu, K. Xu, Q. Sun, and Q. Cai, *A unified gas-kinetic scheme for continuum and rarefied flows IV: Full Boltzmann and model equations*, J. Comput. Phys. **314** (2016), 305–340.
36. C. Mouhot and C. Villani, *Regularity theory for the spatially homogeneous Boltzmann equation with cut-off*, Arch. Rat. Mech. Anal. **173** (2004), 169–212.
37. L. Pareschi and B. Perthame, *A Fourier spectral method for homogeneous Boltzmann equations*, Transport Theory Statist. Phys. **25** (1996), 369–382.
38. E. M. Shakhov, *Generalization of the Krook kinetic relaxation equation*, Fluid Dyn. **3** (1968), no. 5, 95–96.
39. W.-L. Wang and I. D. Boyd, *Predicting continuum breakdown in hypersonic viscous flows*, Phys. Fluids **15** (2003), no. 1, 91–100.
40. Y. Wang and Z. Cai, *Approximation of the Boltzmann collision operator based on hermite spectral method*, J. Comput. Phys. **397** (2019), 108815.

(Zhenning Cai) DEPARTMENT OF MATHEMATICS, NATIONAL UNIVERSITY OF SINGAPORE, LEVEL 4, BLOCK S17, 10 LOWER KENT RIDGE ROAD, SINGAPORE 119076

Email address: matcz@nus.edu.sg

(Yanli Wang) BEIJING COMPUTATIONAL SCIENCE RESEARCH CENTER, BEIJING, CHINA 100193

Email address: ylwang@csrc.ac.cn

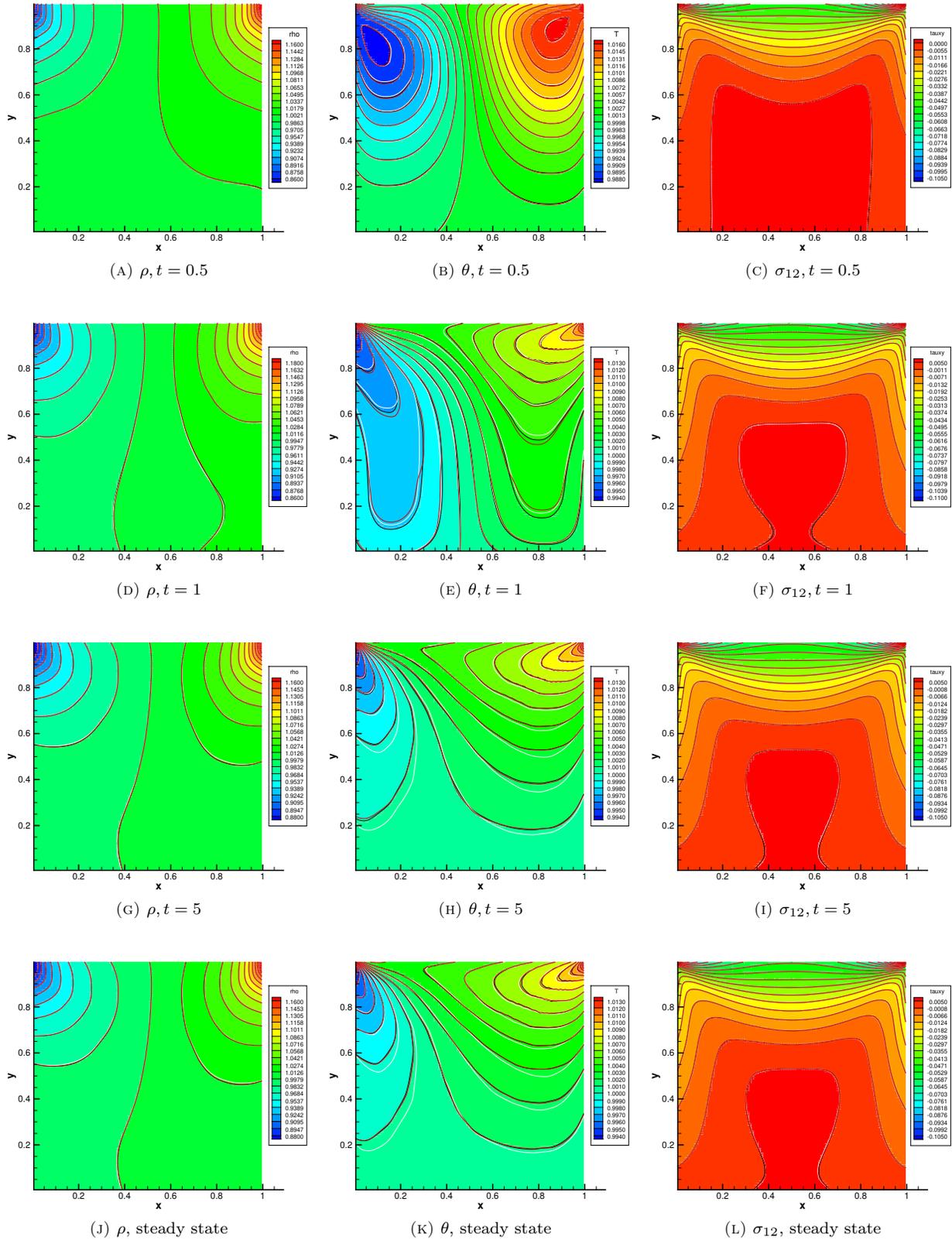


FIGURE 12. Solution of the lid-driven cavity flow at different times. The white contours and the red contours are the numerical solutions with threshold parameters $(\epsilon_1^1, \epsilon_2^1)$ and $(\epsilon_1^2, \epsilon_2^2)$, respectively. The black contours are the reference solution.

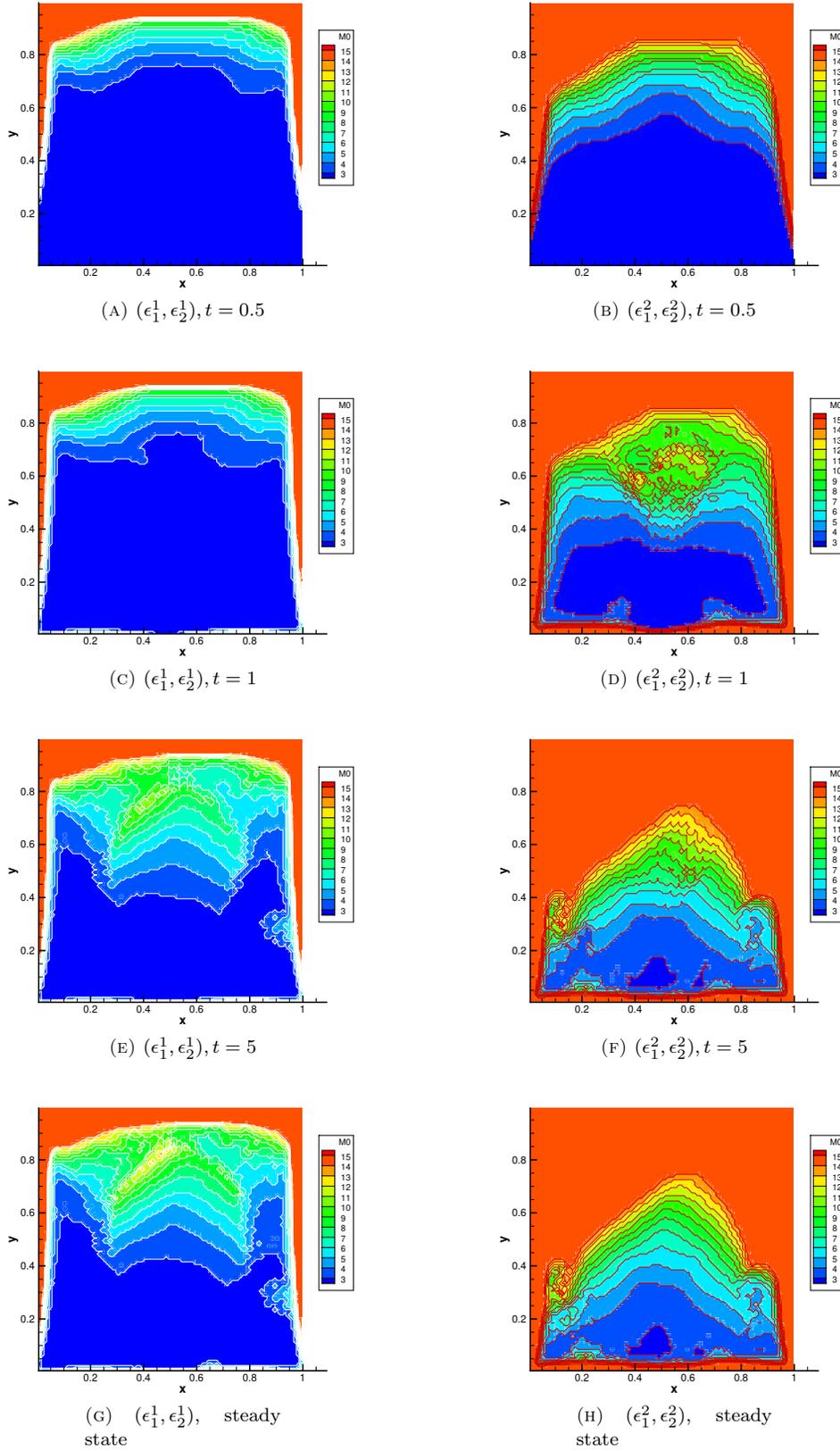


FIGURE 13. The distribution of M_0 for the lid-driven cavity flow at different times.