



An Adaptive Dynamical Low Rank Method for the Nonlinear Boltzmann Equation

Jingwei Hu¹ · Yubo Wang²

Received: 4 December 2021 / Revised: 22 March 2022 / Accepted: 24 June 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Efficient and accurate numerical approximation of the full Boltzmann equation has been a longstanding challenging problem in kinetic theory. This is mainly due to the high dimensionality of the problem and the complicated collision operator. In this work, we propose a highly efficient adaptive low rank method for the Boltzmann equation, concerning in particular the steady state computation. This method employs the fast Fourier spectral method (for the collision operator) and the dynamical low rank method to obtain computational efficiency. An adaptive strategy is introduced to incorporate the boundary information and control the computational rank in an appropriate way. Using a series of benchmark tests in 1D and 2D, we demonstrate the efficiency and accuracy of the proposed method in comparison to the full tensor grid approach.

Keywords Dynamical low rank method · Boltzmann equation · Steady state solution · Adaptive method · Fast Fourier spectral method · Normal shock wave

1 Introduction

Kinetic theory describes the non-equilibrium dynamics of gases or systems comprised of a large number of particles. It provides rich information at the mesoscopic level when the well-known fluid mechanical laws of Navier-Stokes and Fourier become inadequate. Various applications of kinetic theory can be found in fields such as rarefied gas dynamics [4], plasma physics [2], semiconductor modeling [24] and biological and social sciences [27].

In this work, we are interested in the efficient numerical approximation of the nonlinear Boltzmann equation [3, 34], which is the central model in kinetic theory and reads as

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = \mathcal{Q}(f, f), \quad t > 0, \quad \mathbf{x} \in \Omega_{\mathbf{x}} \subset \mathbb{R}^d, \quad \mathbf{v} \in \mathbb{R}^d, \quad (1.1)$$

✉ Jingwei Hu
hujw@uw.edu

Yubo Wang
wang3158@purdue.edu

¹ Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

² Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA

where $f = f(\mathbf{x}, \mathbf{v}, t)$ is the phase space distribution function of time t , position $\mathbf{x}$, and particle velocity $\mathbf{v}$; $\mathcal{Q}$ is the Boltzmann collision operator, which is a quadratic integral operator modeling the binary interactions between particles. It is convenient to introduce the bilinear form of $\mathcal{Q}$:

$$\mathcal{Q}(g, f)(\mathbf{v}) = \int_{\mathbb{R}^d} \int_{S^{d-1}} B(|\mathbf{v} - \mathbf{v}_*|, \cos \chi) [g(\mathbf{v}'_*)f(\mathbf{v}') - g(\mathbf{v}_*)f(\mathbf{v})] d\sigma d\mathbf{v}_*, \quad (1.2)$$

where the post-collisional velocities $(\mathbf{v}', \mathbf{v}'_*)$ are defined in terms of pre-collisional velocities $(\mathbf{v}, \mathbf{v}_*)$ through the conservation of momentum and energy during the collision:

$$\mathbf{v}' = \frac{\mathbf{v} + \mathbf{v}_*}{2} + \frac{|\mathbf{v} - \mathbf{v}_*|}{2} \sigma, \quad \mathbf{v}'_* = \frac{\mathbf{v} + \mathbf{v}_*}{2} - \frac{|\mathbf{v} - \mathbf{v}_*|}{2} \sigma, \quad (1.3)$$

with σ being a vector over the unit sphere S^{d-1} . The collision kernel B is a non-negative function depending on $|\mathbf{v} - \mathbf{v}_*|$ and cosine of the deviation angle χ , i.e., the angle between $\mathbf{v} - \mathbf{v}_*$ and $\mathbf{v}' - \mathbf{v}'_*$. It should be noted that collisions happen only in the velocity space, thus time and spatial dependence is omitted in (1.2). This property has important consequence in design of efficient numerical methods as we shall see later. With the distribution function f , one can retrieve the macroscopic quantities via its moments:

$$\int_{\mathbb{R}^d} f(\mathbf{x}, \mathbf{v}, t) \begin{bmatrix} 1 \\ \mathbf{v} \\ \frac{1}{2}|\mathbf{v}|^2 \end{bmatrix} d\mathbf{v} = \begin{bmatrix} \rho(\mathbf{x}, t) \\ \rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) \\ \frac{1}{2}\rho(\mathbf{x}, t)|\mathbf{u}(\mathbf{x}, t)|^2 + \frac{d}{2}\rho(\mathbf{x}, t)RT(\mathbf{x}, t) \end{bmatrix}, \quad (1.4)$$

where $\rho(\mathbf{x}, t)$, $\mathbf{u}(\mathbf{x}, t)$, and $T(\mathbf{x}, t)$ are the density, bulk velocity, and temperature, R is the Boltzmann constant.

Despite of the long history and wide application of the Boltzmann equation, numerically solving the Boltzmann equation still faces great challenges nowadays. This is mainly due to the high dimensionality of the equation and the complicated collision operator. The prevailing method is the direct simulation Monte Carlo (DSMC) method [1, 28] because it can avoid the curse of dimensionality. DSMC method models binary collisions stochastically but could suffer from slow convergence in certain cases such as low speed or near continuum flows. On the other hand, the deterministic method based on discretization of the equation on representative grids has undergone significant development over the past decade. This is partly due to the rapid growth of the computing power as well as the algorithmic advance in approximation of the Boltzmann collision operator. Regarding the latter, the Fourier spectral method [30, 31] stands out for its high accuracy and possibility of being further accelerated by the fast Fourier transform (FFT). The readers can refer to [8, 17] for a review of such methods. Relevant to the current work, we mention the fast algorithm proposed in [25] which can efficiently evaluate the collision operator for certain collision kernel in $\mathcal{O}(MN_{\mathbf{v}}^d \log N_{\mathbf{v}})$ complexity, where $N_{\mathbf{v}}$ is the number of points in each velocity dimension and M is the number of points over S^{d-1} . Even equipped with the fast solver for the collision operator, solving the Boltzmann equation deterministically can still be very expensive. In the full tensor grid approach, the overall complexity (per time step) would be $\mathcal{O}(N_{\mathbf{x}}^d M N_{\mathbf{v}}^d \log N_{\mathbf{v}})$, where $N_{\mathbf{x}}$ is the number of discretization points used in each spatial dimension. This motivates us to seek more efficient method to overcome the intrinsic high dimensionality of the problem.

Recently, a class of dynamical low rank method has been applied to solving kinetic equations including the Vlasov equation [12, 13], Boltzmann–BGK equation [9, 11] and radiation transfer equation [10, 32]. The basic idea is to find a low-rank approximation of the unknown function f by projecting the equation onto the tangent space of the low-rank solution manifold. Upon a further operator splitting, the original $N_{\mathbf{x}}^d N_{\mathbf{v}}^d$ dimensional problem can be

reduced to a few $N_{\mathbf{x}}^d$ or $N_{\mathbf{v}}^d$ dimensional problems. We mention that this formulation can only be done easily if the original equation has a tensor structure ($\mathbf{x}$ and $\mathbf{v}$ are separated in some sense). For some collision operators, e.g., the BGK operator which is highly nonlinear due to the f dependent Maxwellian function, extra effort is needed to make the method efficient [11].

In this work, we apply the dynamical low rank method to the Boltzmann equation (1.1) and investigate the performance of the method in a series of benchmark tests that concern the steady state solutions. Our contribution can be summarized as follows: 1) The dynamical low rank method is for the first time applied to the nonlinear Boltzmann equation (1.1). Even though the collision operator $\mathcal{Q}(f, f)$ is quite complicated, it is local in $\mathbf{x}$ hence is highly suited in the low rank framework. Moreover, the previously developed fast Fourier spectral method can be applied straightforwardly to accelerate the overall method. 2) We propose an adaptive strategy to add and remove basis along the time evolution. In particular, the stage of adding the basis is strongly motivated by our underlying problem. Most benchmark tests for the Boltzmann equation involve steady state solutions (e.g., normal shock, Couette flow, thermally driven cavity flow, etc. [18]) for which the boundary condition is highly non-trivial and plays an important role. We show that to accurately simulate this type of problems, the boundary information needs to be added to the solution on the fly. As a consequence, dropping the basis becomes mandatory, otherwise the numerical rank will increase constantly. This is in contrast to most of the previous dynamical low rank methods on kinetic equations, where a fixed rank can often be used throughout the simulation. 3) Using asymptotic analysis and heuristic arguments, we identify a class of problems – normal shock problem – whose steady state solutions are indeed low rank in some regimes, and further confirm it in numerical experiments. This provides some theoretical guarantee for the proposed low rank method to be an efficient approach for solving the nonlinear Boltzmann equation. Finally we mention some relevant work on rank adaptivity [5, 6, 16, 21, 33]. Compared to these work, our method investigates in particular the boundary effect and how to incorporate that information to enrich the solution expansion basis. Furthermore, we provide a way to choose a dynamically changed tolerance (rather than a prefixed tolerance) to drop the basis.

The rest of this paper is organized as follows. In Sect. 2, we describe the dynamical low rank method for the Boltzmann equation, including the time, velocity and physical space discretization as well as the treatment of the boundary condition. In Sect. 3, we introduce an adaptive strategy to add and drop basis in the dynamical low rank method during the time evolution. In Sect. 4, we analyze the normal shock problem and demonstrate the low rank property of the solution in both the weak and strong shock wave regimes. Section 5 presents numerical examples in 1D and 2D using the proposed adaptive dynamical low rank method. Several benchmark tests for the nonlinear Boltzmann equation are considered: normal shock, Fourier flow, lid driven cavity flow, and thermally driven cavity flow. The paper is concluded in Sect. 6.

2 The Dynamical Low Rank Method for the Boltzmann Equation

In this section, we introduce the dynamical low rank method for the Boltzmann equation (1.1). We first present the formulation in the continuous setup, where we highlight the special structure of the collision operator in obtaining an efficient low rank approximation. We then describe the discretization in the velocity space and physical space, and treatment of the

typical boundary conditions of the Boltzmann equation. Finally, we add the time discretization to obtain a fully discrete low rank scheme.

The starting point of the method is to constrain the distribution function $f(\mathbf{x}, \mathbf{v}, t)$ to a low rank manifold $\mathbb{M}$ such that

$$f(\mathbf{x}, \mathbf{v}, t) = \sum_{i,j=1}^r X_i(\mathbf{x}, t) S_{ij}(t) V_j(\mathbf{v}, t), \quad (2.1)$$

where r is the representation rank and the basis functions $\{X_i\}_{1 \leq i \leq r} \subset L^2(\Omega_{\mathbf{x}})$ and $\{V_j\}_{1 \leq j \leq r} \subset L^2(\Omega_{\mathbf{v}})$ are orthonormal:

$$\langle X_i, X_j \rangle_{\mathbf{x}} = \delta_{ij}, \quad \langle V_i, V_j \rangle_{\mathbf{v}} = \delta_{ij}, \quad 1 \leq i, j \leq r, \quad (2.2)$$

with $\langle \cdot, \cdot \rangle_{\mathbf{x}}$ and $\langle \cdot, \cdot \rangle_{\mathbf{v}}$ being the inner products on $L^2(\Omega_{\mathbf{x}})$ and $L^2(\Omega_{\mathbf{v}})$, respectively. Note here we consider a finite velocity domain $\Omega_{\mathbf{v}}$ rather than the whole space $\mathbb{R}^d$ to avoid the complication in the infinite domain. This is a reasonable assumption because the majority of the numerical methods for kinetic equations need to first truncate the velocity domain and then perform the discretization. It can often be done without much loss of accuracy since f decays sufficiently fast as $\mathbf{v}$ goes to infinity.

We rewrite Eq. (1.1) as

$$\partial_t f = -\mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathcal{Q}(f, f) := \text{RHS}. \quad (2.3)$$

To ensure uniqueness of the dynamical factors X_i , S_{ij} , and V_j through Eq. (2.3), we impose the following gauge conditions by constraining the derivatives in the null space (for details, see [19]):

$$\langle \partial_t X_i, X_j \rangle_{\mathbf{x}} = 0, \quad \langle \partial_t V_i, V_j \rangle_{\mathbf{v}} = 0, \quad 1 \leq i, j \leq r. \quad (2.4)$$

We now project the right hand side of (2.3) onto the tangent space of $\mathbb{M}$:

$$\partial_t f = P_f(\text{RHS}), \quad (2.5)$$

where the orthogonal projector P_f can be written as

$$P_f(\text{RHS}) = \sum_{j=1}^r \langle V_j, \text{RHS} \rangle_{\mathbf{v}} V_j - \sum_{i,j=1}^r X_i \langle X_i V_j, \text{RHS} \rangle_{\mathbf{x}, \mathbf{v}} V_j + \sum_{i=1}^r X_i \langle X_i, \text{RHS} \rangle_{\mathbf{x}}. \quad (2.6)$$

To avoid the possible ill-conditioning of the matrix $S = (S_{ij})_{1 \leq i, j \leq r}$, one can perform a simple operator splitting [22] to decompose (2.6) into solving three subflows consecutively:

$$\partial_t f = \sum_{j=1}^r \langle V_j, \text{RHS} \rangle_{\mathbf{v}} V_j, \quad (2.7)$$

$$\partial_t f = - \sum_{i,j=1}^r X_i \langle X_i V_j, \text{RHS} \rangle_{\mathbf{x}, \mathbf{v}} V_j, \quad (2.8)$$

$$\partial_t f = \sum_{i=1}^r X_i \langle X_i, \text{RHS} \rangle_{\mathbf{x}}, \quad (2.9)$$

with initial condition:

$$f(\mathbf{x}, \mathbf{v}, 0) = \sum_{i,j=1}^r X_i(\mathbf{x}, 0) S_{ij}(0) V_j(\mathbf{v}, 0). \quad (2.10)$$

Using the orthogonality condition (2.2), we can further simplify each subflow and proceed in the following three substeps:

- *K*-step: Define $K_j(\mathbf{x}, t) = \sum_{i=1}^r X_i(\mathbf{x}, t) S_{ij}(t)$ then $f(\mathbf{x}, \mathbf{v}, t) = \sum_{j=1}^r K_j(\mathbf{x}, t) V_j(\mathbf{v}, t)$. We can rewrite Eq. (2.7) as

$$\partial_t \left(\sum_{j=1}^r K_j V_j \right) = \sum_{j=1}^r (\partial_t K_j V_j + K_j \partial_t V_j) = \sum_{j=1}^r \langle V_j, \text{RHS} \rangle_{\mathbf{v}} V_j. \quad (2.11)$$

It is easy to see that $\partial_t K_j = \langle V_j, \text{RHS} \rangle_{\mathbf{v}}$ together with $\partial_t V_j = 0$ solve (2.11). Since the solution to the subflow is unique, we thus know $\{V_j\}_{1 \leq j \leq r}$ remains unchanged during this substep. Furthermore, we have

$$\begin{aligned} \partial_t K_j &= \langle V_j, \text{RHS} \rangle_{\mathbf{v}} \\ &= - \sum_{l=1}^r \langle \mathbf{v} V_j V_l \rangle_{\mathbf{v}} \cdot \nabla_{\mathbf{x}} K_l + \sum_{m,n=1}^r \langle V_j \mathcal{Q}(V_m, V_n) \rangle_{\mathbf{v}} K_m K_n, \quad j = 1, \dots, r, \end{aligned} \quad (2.12)$$

where the simplification of the last term relies crucially on the bilinearity of the collision operator (1.2) as well as the fact that collisions act locally in the physical space.

- *S*-step: We can argue similarly to obtain that the subflow (2.8) is equivalent to

$$\begin{aligned} \partial_t S_{ij} &= - \langle X_i V_j, \text{RHS} \rangle_{\mathbf{x}, \mathbf{v}} \\ &= \sum_{k,l=1}^r \langle \mathbf{v} V_j V_l \rangle_{\mathbf{v}} \cdot \langle X_i \nabla_{\mathbf{x}} X_k \rangle_{\mathbf{x}} S_{kl} \\ &\quad - \sum_{k,l,m,n=1}^r \langle X_i X_k X_l \rangle_{\mathbf{x}} \langle V_j \mathcal{Q}(V_m, V_n) \rangle_{\mathbf{v}} S_{km} S_{ln}, \quad i, j = 1, \dots, r. \end{aligned} \quad (2.13)$$

During this substep, both $\{V_j\}_{1 \leq j \leq r}$ and $\{X_i\}_{1 \leq i \leq r}$ remain unchanged.

- *L*-step: Define $L_i(\mathbf{v}, t) = \sum_{j=1}^r S_{ij}(t) V_j(\mathbf{v}, t)$ then $f(\mathbf{x}, \mathbf{v}, t) = \sum_{i=1}^r X_i(\mathbf{x}, t) L_i(\mathbf{v}, t)$. By similar arguments, the subflow (2.9) is equivalent to

$$\begin{aligned} \partial_t L_i &= \langle X_i, \text{RHS} \rangle_{\mathbf{x}}, \\ &= - \sum_{l=1}^r \mathbf{v} \cdot \langle X_i \nabla_{\mathbf{x}} X_l \rangle_{\mathbf{x}} L_l + \sum_{m,n=1}^r \mathcal{Q}(L_m, L_n) \langle X_i X_m X_n \rangle_{\mathbf{x}}, \quad i = 1, \dots, r. \end{aligned} \quad (2.14)$$

During this substep, $\{X_i\}_{1 \leq i \leq r}$ remains unchanged.

Therefore, we have obtained a set of low rank Eqs. (2.12)–(2.14) in the continuous setting. The task remains is to apply the proper discretization to these equations in the velocity space, physical space, and time, which we will detail in the following subsections.

2.1 Velocity Space Discretization

Examining the Eqs. (2.12)–(2.14), we can see that all terms pertaining to the collision operator have the form of $\mathcal{Q}(h_1, h_2)$, where h_1 and h_2 are some functions of $\mathbf{v}$ (e.g., $\mathcal{Q}(V_m, V_n)$ in (2.12) and (2.13), and $\mathcal{Q}(L_m, L_n)$ in (2.14)). Luckily this isn't much change from the original collision operator in (1.1) and we can apply the well-developed fast Fourier spectral methods. These methods can be implemented as a discrete velocity method: one chooses an appropriate velocity domain $[-L_v, L_v]^d$ and uniform grid points $\{\mathbf{v}_q\}$; the collision solver takes discrete values $\{h_1(\mathbf{v}_q)\}$ and $\{h_2(\mathbf{v}_q)\}$ and outputs $\{\mathcal{Q}(h_1, h_2)(\mathbf{v}_q)\}$ on the same set of grid points.

Specifically, for 2D Maxwell molecules ($d = 2$ and $B = \text{const}$) and 3D hard spheres ($d = 3$ and $B = \text{const}|\mathbf{v} - \mathbf{v}_*|$), we can use the algorithm proposed in [26] to evaluate each term $\mathcal{Q}(h_1, h_2)$ with complexity $\mathcal{O}(MN_v^d \log N_v)$, where N_v is the number of points in each dimension of the velocity space and $M \ll N_v^{d-1}$ is the number of points used on the sphere S^{d-1} ; for general collision kernel $B = B(|\mathbf{v} - \mathbf{v}_*|, \cos \chi)$, we can use the algorithm proposed in [14] to evaluate each term $\mathcal{Q}(h_1, h_2)$ with complexity $\mathcal{O}(MN_v^{d+1} \log N_v)$. For more details, the readers can refer to [14, 26]. Note that the above complexity is only for one time evaluation of the collision operator. The overall complexity of the low rank method (which depends on the rank r) is given in Sect. 2.4.

2.2 Physical Space Discretization

There are various ways to discretize the Eqs. (2.12)–(2.14) in the physical space, for example, one can apply the Fourier spectral method [12] or the high resolution finite difference scheme [11] directly to these equations. Generally speaking, the conventional scheme used for the original equation needs to be tailored when solving the equations resulted from the low rank projection. The issue also becomes a bit tricky when the boundary condition is not periodic.

Here we adopt a “first discretize, then project” strategy, which is simpler because it follows directly from the scheme for the original equation. We mention that this idea is similar to the so-called kinetic flux vector splitting (KFVS) scheme [7], a well-known method for solving the compressible Euler equations derived from the kinetic equation. For simplicity, we focus on the first order upwind scheme in this work. To extend it to high order, similar strategy for the KFVS scheme [23] can be considered.

We use the one-dimensional case ($d = 1$) to illustrate the idea. Extension to high dimension with rectangular grid is straightforward as implemented in our numerical examples. Assume $\Omega_x = [-L_x, L_x]$ with uniform grid points chosen as $x_p = -L_x + (p - \frac{1}{2})\Delta x$, $p = 1, \dots, N_x$, $\Delta x = \frac{2L_x}{N_x}$. Since the transport term in the Boltzmann equation (1.1) is linear, it is very easy to apply the upwind scheme:

$$\begin{aligned} \partial_t f(x, v, t) &= -\frac{v + |v|}{2} \frac{f(x, v, t) - f(x - \Delta x, v, t)}{\Delta x} \\ &\quad - \frac{v - |v|}{2} \frac{f(x + \Delta x, v, t) - f(x, v, t)}{\Delta x} + \mathcal{Q}(f(x, v, t), f(x, v, t)) \\ &:= -v^+ D_+ f(x, v, t) - v^- D_- f(x, v, t) + \mathcal{Q}(f(x, v, t), f(x, v, t)), \end{aligned} \quad (2.15)$$

where $v^\pm = \frac{v \pm |v|}{2}$, and $D_\pm$ are first order upwind operators.

For Eq. (2.15), we can apply the same projection process as we did previously to Eq. (2.3) to obtain (i.e., the analogs of (2.12)–(2.14)):

- *K*-step:

$$\begin{aligned}\partial_t K_j(x, t) = & - \sum_{l=1}^r \langle v^+ V_j(v, t) V_l(v, t) \rangle_v D_+ K_l(x, t) \\ & - \sum_{l=1}^r \langle v^- V_j(v, t) V_l(v, t) \rangle_v D_- K_l(x, t) \\ & + \sum_{m,n=1}^r \langle V_j(v, t) \mathcal{Q}(V_m(v, t), V_n(v, t)) \rangle_v K_m(x, t) K_n(x, t).\end{aligned}\quad (2.16)$$

- *S*-step:

$$\begin{aligned}\partial_t S_{ij}(t) = & \sum_{k,l=1}^r \langle v^+ V_j(v, t) V_l(v, t) \rangle_v \langle X_i(x, t) D_+ X_k(x, t) \rangle_x S_{kl} \\ & + \sum_{k,l=1}^r \langle v^- V_j(v, t) V_l(v, t) \rangle_v \langle X_i(x, t) D_- X_k(x, t) \rangle_x S_{kl} \\ & - \sum_{k,l,m,n=1}^r \langle X_i(x, t) X_k(x, t) X_l(x, t) \rangle_x \langle V_j(v, t) \\ & \mathcal{Q}(V_m(v, t), V_n(v, t)) \rangle_v S_{km} S_{ln}.\end{aligned}\quad (2.17)$$

- *L*-step:

$$\begin{aligned}\partial_t L_i(v, t) = & - \sum_{l=1}^r v^+ \langle X_i(x, t) D_+ X_l(x, t) \rangle_x L_l(v, t) \\ & - \sum_{l=1}^r v^- \langle X_i(x, t) D_- X_l(x, t) \rangle_x L_l(v, t) \\ & + \sum_{m,n=1}^r \mathcal{Q}(L_m(v, t), L_n(v, t)) \langle X_i(x, t) X_m(x, t) X_n(x, t) \rangle_x.\end{aligned}\quad (2.18)$$

Remark 2.1 During the review process, we were made aware of the work [20], which analyzes the stability of the dynamical low rank method applied to a linear hyperbolic system using the “first discretize, then project” strategy. It is shown that the method can lead to instabilities (which however only become visible for certain ranks). In all of our numerical experiments, we didn’t observe any instabilities. A detailed analysis for the kinetic problem is yet to be performed.

2.3 Treatment of the Boundary Condition

In the low rank framework, boundary condition for $f(\mathbf{x}, \mathbf{v}, t)$ needs to be transformed to the boundary condition of $\{K_j\}_{1 \leq j \leq r}$. In fact, this transformation has a non-trivial impact on the fully discrete scheme which we shall describe in the next subsection.

For a boundary point $\mathbf{x} \in \partial\Omega_{\mathbf{x}}$ with outward pointing normal $\mathbf{n}(\mathbf{x})$ and boundary velocity $\mathbf{u}_w(\mathbf{x}, t)$, general boundary conditions for Boltzmann equation (1.1) are defined through the inflow direction:

$$f(\mathbf{x}, \mathbf{v}, t) = f_{bdy}(\mathbf{x}, \mathbf{v}, t), \quad (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0, \quad (2.19)$$

where f_{bdy} is a prescribed function. The other half of $f(\mathbf{x}, \mathbf{v}, t)$ is given from interior of the domain (outflow). We thus define

$$f^b(\mathbf{x}, \mathbf{v}, t) = \begin{cases} f_{bdy}(\mathbf{x}, \mathbf{v}, t), & (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0, \\ f(\mathbf{x}, \mathbf{v}, t), & (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) \geq 0. \end{cases} \quad (2.20)$$

Accordingly, we can project the full boundary $f^b(\mathbf{x}, \mathbf{v}, t)$ to the space spanned by $\{V_j\}_{1 \leq j \leq r}$ to obtain boundary values for $\{K_j\}_{1 \leq j \leq r}$:

$$\begin{aligned} K_j(\mathbf{x}, t) &= \langle f^b(\mathbf{x}, \mathbf{v}, t), V_j(\mathbf{v}, t) \rangle_{\mathbf{v}} \\ &= \langle f_{bdy}(\mathbf{x}, \mathbf{v}, t) \mathbb{1}_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0}, V_j(\mathbf{v}, t) \rangle_{\mathbf{v}} \\ &\quad + \langle f(\mathbf{x}, \mathbf{v}, t) \mathbb{1}_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) \geq 0}, V_j(\mathbf{v}, t) \rangle_{\mathbf{v}} \\ &= \langle f_{bdy}(\mathbf{x}, \mathbf{v}, t) \mathbb{1}_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0}, V_j(\mathbf{v}, t) \rangle_{\mathbf{v}} \\ &\quad + \sum_{l=1}^r K_l(\mathbf{x}, t) \langle \mathbb{1}_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) \geq 0} V_l(\mathbf{v}, t) V_j(\mathbf{v}, t) \rangle_{\mathbf{v}}, \end{aligned} \quad (2.21)$$

where the K_l term appearing on the right hand side of (2.21) can be approximated using values inside the domain (extrapolation) since the term results from the outflow.

Two typical boundary conditions used when solving the Boltzmann equation (1.1) are the following inflow boundary and Maxwell diffusive boundary. For inflow boundary, we take $\mathbf{u}_w(\mathbf{x}, t) = \mathbf{0}$ and

$$f_{bdy}(\mathbf{x}, \mathbf{v}, t) = \frac{\rho_{in}(\mathbf{x}, t)}{(2\pi T_{in}(\mathbf{x}, t))^{d/2}} \exp\left(-\frac{|\mathbf{v} - \mathbf{u}_{in}(\mathbf{x}, t)|^2}{2RT_{in}(\mathbf{x}, t)}\right), \quad \mathbf{v} \cdot \mathbf{n}(\mathbf{x}) < 0, \quad (2.22)$$

where ρ_{in} , $\mathbf{u}_{in}$ and T_{in} are the density, bulk velocity and temperature of the prescribed inflow. For the Maxwell diffusive boundary, we take

$$f_{bdy}(\mathbf{x}, \mathbf{v}, t) = \rho_w(\mathbf{x}, t) \exp\left(-\frac{|\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)|^2}{2RT_w(\mathbf{x}, t)}\right), \quad (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0, \quad (2.23)$$

where T_w is the wall temperature, and ρ_w is determined by conservation of mass through the wall:

$$\rho_w(\mathbf{x}, t) = -\frac{\int_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) \geq 0} (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) f(\mathbf{x}, \mathbf{v}, t) \, d\mathbf{v}}{\int_{(\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) < 0} (\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)) \cdot \mathbf{n}(\mathbf{x}) \exp\left(-\frac{|\mathbf{v} - \mathbf{u}_w(\mathbf{x}, t)|^2}{2RT_w(\mathbf{x}, t)}\right) \, d\mathbf{v}}. \quad (2.24)$$

2.4 Time Discretization and the Fully Discrete Scheme

We now add the time discretization to (2.12)–(2.14) to obtain a fully discrete scheme. Since most of the examples we are interested in this paper concern the stationary Boltzmann equation, the first order time discretization suffices. For high order method in time, the readers can refer to [10] and references therein.

Given the initial condition $f(\mathbf{x}, \mathbf{v}, 0) = f^0(\mathbf{x}, \mathbf{v})$, we first perform the singular value decomposition $f^0(\mathbf{x}, \mathbf{v}) = \sum_{i,j=1}^r X_i^0(\mathbf{x}) S_{ij}^0 V_j^0(\mathbf{v})$ to obtain (X_i^0, S_{ij}^0, V_j^0) , where a fixed, reasonable rank r is chosen and used in the following computation.

Suppose at time step t^n , (X_i^n, S_{ij}^n, V_j^n) are available. With the fully discretized scheme, each element $X_i \in \mathbb{R}^{N_x^d}$, $S_{ij} \in \mathbb{R}$ and $V_j \in \mathbb{R}^{N_v^d}$. The time step Δt is chosen as $\frac{\Delta x}{2L_v}$. In order to obtain $(X_i^{n+1}, S_{ij}^{n+1}, V_j^{n+1})$ at $t^{n+1} = t^n + \Delta t$, we proceed as follows:

1. *K*-step.

- (a) Construct $K_j^n = \sum_{i=1}^r X_i^n S_{ij}^n$.
- (b) Perform the forward Euler step in (2.12) to obtain K_j^{n+1} :

$$\begin{aligned} K_j^{n+1} = & K_j^n - \Delta t \sum_{l=1}^r \left\langle \mathbf{v} V_j^n V_l^n \right\rangle_{\mathbf{v}} \cdot \nabla_{\mathbf{x}} K_l^n \\ & + \Delta t \sum_{m,n=1}^r \left\langle V_j^n \mathcal{Q}(V_m^n, V_n^n) \right\rangle_{\mathbf{v}} K_m^n K_n^n, \quad j = 1, \dots, r. \end{aligned} \quad (2.25)$$

- (c) Compute the QR decomposition of $K_j^{n+1} = \sum_{i=1}^r X_i^{n+1} S_{ij}^{(1)}$ to obtain updated X_i^{n+1} and $S_{ij}^{(1)}$.

The overall arithmetic complexity of this step is $\mathcal{O}(r^3 N_v^d + r^3 N_x^d + r^2 M N_v^d \log N_v)$ (suppose the algorithm in [26] is used for evaluating the collision operator).

2. *S*-step.

- (a) Perform the forward Euler step in (2.13) to obtain $S_{ij}^{(2)}$:

$$\begin{aligned} S_{ij}^{(2)} = & S_{ij}^{(1)} + \Delta t \sum_{l=1}^r \left\langle \mathbf{v} V_j^n V_l^n \right\rangle_{\mathbf{v}} \cdot \left\langle X_i^{n+1} \nabla_{\mathbf{x}} K_l^{n+1} \right\rangle_{\mathbf{x}} \\ & - \Delta t \sum_{m,n=1}^r \left\langle V_j^n \mathcal{Q}(V_m^n, V_n^n) \right\rangle_{\mathbf{v}} \sum_{l=1}^r \left(\sum_{k=1}^r \left(\left\langle X_i^{n+1} X_k^{n+1} X_l^{n+1} \right\rangle_{\mathbf{x}} S_{km}^{(1)} \right) S_{ln}^{(1)} \right), \\ & i, j = 1, \dots, r. \end{aligned} \quad (2.26)$$

Since some of the quantities have been computed in the *K*-step, they can be reused in this step, for example, the term $\mathcal{Q}(V_m^n, V_n^n)$. Note that we changed the second term on the right hand side such that it uses $\nabla_{\mathbf{x}} K_j^{n+1}$ rather than $\nabla_{\mathbf{x}} X_j^{n+1}$. This is crucial because we have only available boundary condition expressed in terms of K_j^{n+1} as seen in Sect. 2.3.

The overall arithmetic complexity of this step is $\mathcal{O}(r^3 N_x^d + r^4)$.

3. *L*-step.

- (a) Construct $L_i^n = \sum_{j=1}^r S_{ij}^{(2)} V_j^n$ and $\tilde{K}_j^{n+1} = \sum_{i=1}^r X_i^{n+1} S_{ij}^{(2)}$.

(b) Perform the forward Euler step in (2.14) to obtain L_i^{n+1} :

$$\begin{aligned} L_i^{n+1} = & L_i^n - \Delta t \sum_{l=1}^r \mathbf{v} \cdot \langle X_i^{n+1} \nabla_{\mathbf{x}} \tilde{K}_l^{n+1} \rangle_{\mathbf{x}} V_l^n \\ & + \Delta t \sum_{p,q=1}^r \mathcal{Q}(V_p^n, V_q^n) \sum_{n=1}^r \\ & \times \left(\sum_{m=1}^r \left(\langle X_i^{n+1} X_m^{n+1} X_n^{n+1} \rangle_{\mathbf{x}} S_{mp}^{(2)} \right) S_{nq}^{(2)} \right), \quad i = 1, \dots, r. \end{aligned} \quad (2.27)$$

The term involving the collision operator is rearranged so that the previously computed values can be reused. For the same reason as in the S -step, $\nabla_{\mathbf{x}} \tilde{K}_j^{n+1}$ is introduced to avoid using $\nabla_{\mathbf{x}} X_l^{n+1}$.

(c) Compute the QR decomposition of $L_i^{n+1} = \sum_{j=1}^r S_{ij}^{n+1} V_j^{n+1}$ to obtain updated V_j^{n+1} and S_{ij}^{n+1} .

The overall arithmetic complexity of this step is $\mathcal{O}(r^2 N_{\mathbf{x}}^d + r^3 N_{\mathbf{v}}^d + r^4)$.

To simplify the notation, we treat $\mathbf{x}, \mathbf{v}$ as the continuous variables in the above presentation. The discretization in $\mathbf{x}$ and $\mathbf{v}$ can be added straightforwardly following the discussion in Sects. 2.1 and 2.2. The inner products $\langle \cdot \rangle_{\mathbf{v}}, \langle \cdot \rangle_{\mathbf{x}}$ are evaluated using the midpoint rule at the discrete velocity and spatial grid points.

If r is small, the computational complexity of the above algorithm will be dominated by the evaluation of the collision operator $\mathcal{O}(r^2 M N_{\mathbf{v}}^d \log N_{\mathbf{v}})$, which can be much more efficient than the full tensor method whose complexity is $\mathcal{O}(N_{\mathbf{x}}^d M N_{\mathbf{v}}^d \log N_{\mathbf{v}})$.

3 An Adaptive Dynamical Low Rank Method

The dynamical low rank method introduced in the last section uses a fixed rank r throughout the entire time evolution. This turns out to be a bad strategy when solving the stationary Boltzmann equation subject to inflow or Maxwell diffusive boundary conditions. The reason is two-fold: 1) The boundary keeps sending new information to the interior of the domain so that the basis X_i, S_{ij}, V_j initialized according to the initial condition is not sufficient to capture the solution at later time. Thus new basis needs to be injected to the solution over time. 2) For many benchmark tests of the Boltzmann equation, the steady state solutions are often low rank (see Sect. 4 for a partial justification). Therefore, keeping adding basis without dropping anything would unnecessarily increase the computational cost. In this section, we provide an adaptive strategy to add and delete basis during the time evolution of a dynamical low rank method.

3.1 Adding Basis from the Boundary

Assume that at the boundary $\mathbf{x} \in \partial\Omega_{\mathbf{x}}$, $f(\mathbf{x}, \mathbf{v}, t)$ is given by

$$f(\mathbf{x}, \mathbf{v}, t) = f^b(\mathbf{x}, \mathbf{v}, t), \quad (3.1)$$

where $f^b(\mathbf{x}, \mathbf{v}, t)$ is defined in (2.20).

Since the function $f^b(\mathbf{x}, \mathbf{v}, t)$ does not necessarily belong to the space spanned by $\{V_j\}_{1 \leq j \leq r}$, using a fixed set of basis will result in information loss.

We can fix this problem by explicitly adding boundary conditions as basis into $\{V_j\}_{1 \leq j \leq r}$. For example, at the beginning of time step t^n , suppose in the fully discrete scheme there are N_{bx} spatial points on the boundary $\partial\Omega_x$, N_v^d velocity points over the velocity space Ω_v and N_x^d spatial points over the physical space Ω_x . We can represent the function $f^b(\mathbf{x}, \mathbf{v}, t^n)$ using a matrix $F_b \in \mathbb{R}^{N_{bx} \times N_v^d}$ (typically $N_{bx} \ll N_v^d$). We then proceed as follows:

1. Compute the SVD of F_b to obtain $F_b = U_b \Sigma_b Q_b^T$ where $U_b \in \mathbb{R}^{N_{bx} \times N_{bx}}$, $Q_b \in \mathbb{R}^{N_v^d \times N_v^d}$ are orthonormal and $\Sigma_b \in \mathbb{R}^{N_{bx} \times N_v^d}$ contains descending singular values.
2. Drop singular values in Σ_b that are smaller than some tolerance (e.g., 10^{-10}). Suppose there are r_b singular values remaining, set $\tilde{Q}_b = Q_b(:, 1:r_b) \in \mathbb{R}^{N_v^d \times r_b}$.
3. Concatenate a random matrix $X_h \in \mathbb{R}^{N_x^d \times r_b}$ to X^n , $\tilde{Q}_b$ to V^n and extend S^n with zero padding:

$$\begin{aligned}\hat{X} &= [X^n, X_h] \in \mathbb{R}^{N_x^d \times (r+r_b)}, \quad \hat{S} = \begin{bmatrix} S^n \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(r+r_b) \times (r+r_b)}, \\ \hat{V} &= [V^n, \tilde{Q}_b] \in \mathbb{R}^{N_v^d \times (r+r_b)}.\end{aligned}\quad (3.2)$$

4. Perform the QR decomposition of $\hat{X}$ and $\hat{V}$ to orthonormalize new basis as $\hat{X} = X_q S_x$ and $\hat{V} = V_q S_v$. Set $S_q = S_x \hat{S} S_v^T$.

Then (X_q, S_q, V_q) are the new low-rank factors and we proceed as in Sect. 2.4. If $f^b(\mathbf{x}, \mathbf{v}, t) = f^b(\mathbf{v}, t)$ is spatially homogeneous, we can directly start at step 3 and concatenate F_b to V^n .

Remark 3.1 In step 3 above, we explicitly add boundary basis into velocity basis such that there is no information loss in representing the boundary condition. In order to keep size consistent, we add random basis to spatial basis, and pad zeros to the middle matrix S^n . After this step, f^n remains unchanged due to the zero padding in S^n .

3.2 Dropping Basis Adaptively

To avoid the rank accumulation from the above procedure, we can decrease the rank r by dropping some small singular values of matrix $(S_{ij})_{1 \leq i, j \leq r}$.

At the end of time step t^n as described in Sect. 2.4, we proceed as follows to adjust the rank:

1. Compute the SVD of $S^{n+1} = (S_{ij}^{n+1})_{1 \leq i, j \leq r}$ to obtain $S^{n+1} = U \Sigma Q^T$, where $U, Q \in \mathbb{R}^{r \times r}$ are orthonormal and $\Sigma \in \mathbb{R}^{r \times r}$ is diagonal with descending singular values.
2. Drop singular values in Σ that are less than some tolerance **drop_tol**. Suppose there are r' singular values remaining, we set $\tilde{U} = U(:, 1:r')$, $\tilde{\Sigma} = \Sigma(1:r', 1:r')$ and $\tilde{Q} = Q(:, 1:r')$. Define $\tilde{S}^{n+1} = \tilde{\Sigma}$.
3. Update the basis as $[\tilde{X}_1^{n+1}, \tilde{X}_2^{n+1}, \dots, \tilde{X}_{r'}^{n+1}] = [X_1^{n+1}, X_2^{n+1}, \dots, X_r^{n+1}] \tilde{U}$ and $[\tilde{V}_1^{n+1}, \tilde{V}_2^{n+1}, \dots, \tilde{V}_{r'}^{n+1}] = [V_1^{n+1}, V_2^{n+1}, \dots, V_r^{n+1}] \tilde{Q}$ where $\{\tilde{X}_i^{n+1}\}_{i=1, \dots, r'}$ and $\{\tilde{V}_i^{n+1}\}_{i=1, \dots, r'}$ are the updated spatial and velocity basis functions respectively.

drop_tol plays an important role in overall computational efficiency and accuracy. Large **drop_tol** causes low accuracy for some high-rank solutions and small **drop_tol** suffers from heavy computation by large computational rank. We dynamically choose **drop_tol** according to the accuracy of the current solution. More details are given in Sect. 5.1.

4 Normal Shock Problem and Low Rank Property of the Solution

Generally speaking, it is hard to predict or analyze the rank of the solution to the Boltzmann equation due to its highly nonlinear structure. As such, the dynamical low rank method introduced above is really like a black box solver since one cannot tell in advance the rank of the solution until the actual simulation is run. If the rank turns out to be high, the method becomes slow and might not be competitive to the full tensor method. Nevertheless, in this section we identify a class of problems whose solutions are indeed low rank so that we have confidence about the efficiency of the low rank method.

The normal shock problem [4] is a classical benchmark test in rarefied gas dynamics and has been used to validate all kinds of numerical methods for the nonlinear Boltzmann equation. Consider a plane shock wave perpendicular to a flow. The flow is in the x_1 direction. The gas is uniform at upstream infinity ($x_1 \rightarrow -\infty$) and downstream infinity ($x_1 \rightarrow +\infty$) and the whole flow is stationary. We are interested in the shock profile developed in this setup with various Mach numbers.

The governing equation is the following 1D stationary Boltzmann equation:

$$v_1 \partial_{x_1} f = \mathcal{Q}(f, f), \quad (4.1)$$

with boundary condition

$$\begin{aligned} \lim_{x_1 \rightarrow -\infty} f(x_1, \mathbf{v}) &= f_L(\mathbf{v}) = \mathcal{M}(\rho_L, \mathbf{u}_L, T_L)(\mathbf{v}) \\ &= \frac{\rho_L}{(2\pi RT_L)^{d/2}} \exp\left(-\frac{(v_1 - u_L)^2 + v_2^2 + \cdots + v_d^2}{2RT_L}\right), \\ \lim_{x_1 \rightarrow +\infty} f(x_1, \mathbf{v}) &= f_R(\mathbf{v}) = \mathcal{M}(\rho_R, \mathbf{u}_R, T_R)(\mathbf{v}) \\ &= \frac{\rho_R}{(2\pi RT_R)^{d/2}} \exp\left(-\frac{(v_1 - u_R)^2 + v_2^2 + \cdots + v_d^2}{2RT_R}\right), \end{aligned} \quad (4.2)$$

where $\mathcal{M}(\rho, \mathbf{u}, T)$ is the Maxwellian distribution; $(\rho_L, \mathbf{u}_L, T_L)$ and $(\rho_R, \mathbf{u}_R, T_R)$ are the density, bulk velocity and temperature of the upstream and downstream flows; and R is the gas constant.

The net flow of mass, momentum and energy into the shock must be equal to the ones out of the shock:

$$\int v_1 f_L(\mathbf{v}) \begin{bmatrix} 1 \\ v_1 \\ \mathbf{v}^2 \end{bmatrix} d\mathbf{v} = \int v_1 f_R(\mathbf{v}) \begin{bmatrix} 1 \\ v_1 \\ \mathbf{v}^2 \end{bmatrix} d\mathbf{v}. \quad (4.3)$$

Rewriting Eq. (4.3) in terms of macroscopic quantities $\rho_{L,R}$, $u_{L,R}$ and $T_{L,R}$, we have the following Rankine-Hugoniot relations

$$\begin{aligned} \rho_L u_L &= \rho_R u_R, \\ \rho_L u_L^2 + \rho_L RT_L &= \rho_R u_R^2 + \rho_R RT_R, \\ \rho_L u_L (u_L^2 + (d+2)RT_L) &= \rho_R u_R (u_R^2 + (d+2)RT_R). \end{aligned} \quad (4.4)$$

Given the upstream quantities (ρ_L, u_L, T_L) and using the upstream flow Mach number defined by

$$M_L = \frac{u_L}{(\gamma RT_L)^{\frac{1}{2}}}, \quad \gamma = \frac{d+2}{d}, \quad (4.5)$$

we can solve Eq. (4.4) to obtain

$$\rho_R = \rho_L \frac{(d+1)M_L^2}{M_L^2 + d}, \quad u_R = u_L \frac{M_L^2 + d}{(d+1)M_L^2}, \quad T_R = T_L \frac{((d+2)M_L^2 - 1)(M_L^2 + d)}{(d+1)^2 M_L^2}. \quad (4.6)$$

In the following, we consider two scenarios where one can obtain some low rank approximation to the solutions of (4.1)–(4.2).

4.1 Weak Shock Wave: $M_L = \mathcal{O}(1)$

When $M_L = 1$, it is clear from (4.6) that there will be no jump hence no shock. When $M_L = \mathcal{O}(1)$ but bigger than 1, a weak shock will be developed. We assume

$$M_L = 1 + \varepsilon, \quad (4.7)$$

where ε is a small parameter. In fact, ε is on the same order of the mean free path [29]. We then rescale x_1 according to $\tilde{x}_1 = \varepsilon x_1$. The Eq. (4.1) thus becomes

$$v_1 \partial_{\tilde{x}_1} f = \frac{1}{\varepsilon} \mathcal{Q}(f, f). \quad (4.8)$$

On the other hand, we can see from (4.6) that the macroscopic quantities of upstream flow and downstream flow are very close:

$$\begin{aligned} \frac{\rho_R}{\rho_L} &= 1 + \frac{d(M_L^2 - 1)}{M_L^2 + d} = 1 + \mathcal{O}(\varepsilon), \\ \frac{u_R}{u_L} &= 1 - \frac{d(M_L^2 - 1)}{(d+1)M_L^2} = 1 + \mathcal{O}(\varepsilon), \\ \frac{T_R}{T_L} &= 1 + \frac{(d+1)(M_L^4 - 1) + (M_L^2 - 1)^2}{(d+1)^2 M_L^2} = 1 + \mathcal{O}(\varepsilon). \end{aligned} \quad (4.9)$$

Hence

$$\frac{f_R}{f_L} = 1 + \mathcal{O}(\varepsilon). \quad (4.10)$$

Therefore, it is reasonable to assume

$$f(\tilde{x}_1, \mathbf{v}) = f_L(\mathbf{v}) + \varepsilon f_1(\tilde{x}_1, \mathbf{v}) + \mathcal{O}(\varepsilon^2), \quad (4.11)$$

where $f_1(\tilde{x}_1, \mathbf{v})$ is yet to be determined.

The rest of the analysis is similar to the Hilbert expansion. Substituting (4.11) into (4.8) and matching orders, we obtain at order $\mathcal{O}(\varepsilon)$:

$$\mathcal{Q}(f_1, f_L) + \mathcal{Q}(f_L, f_1) = v_1 \partial_{\tilde{x}_1} f_L(\mathbf{v}) \equiv 0. \quad (4.12)$$

Using the linearized Boltzmann collision operator [3] defined by

$$L_{\mathcal{M}}(f) := \frac{1}{\mathcal{M}} (\mathcal{Q}(\mathcal{M}, \mathcal{M}f) + \mathcal{Q}(\mathcal{M}f, \mathcal{M})), \quad \mathcal{M} \text{ is a Maxwellian}, \quad (4.13)$$

we can write (4.12) as

$$L_{f_L} \left(\frac{f_1}{f_L} \right) (\tilde{x}_1, \mathbf{v}) = 0. \quad (4.14)$$

The kernel property of $L_{\mathcal{M}}$ implies that $\frac{f_1}{f_L}$ must be a linear combination of collision invariants $1, \mathbf{v}, |\mathbf{v}|^2$ and we may write

$$f_1(\tilde{x}_1, \mathbf{v}) = f_L(\mathbf{v}) (a(\tilde{x}_1) + \mathbf{b}(\tilde{x}_1) \cdot \mathbf{v} + c(\tilde{x}_1)|\mathbf{v}|^2), \quad (4.15)$$

where $a, \mathbf{b}$ and c are functions of x_1 only. Together with (4.11), we have

$$f(\tilde{x}_1, \mathbf{v}) = f_L(\mathbf{v})(1 + \varepsilon a(\tilde{x}_1) + \varepsilon \mathbf{b}(\tilde{x}_1) \cdot \mathbf{v} + \varepsilon c(\tilde{x}_1)|\mathbf{v}|^2) + \mathcal{O}(\varepsilon^2). \quad (4.16)$$

Therefore, up to order $\mathcal{O}(\varepsilon)$, the solution $f(\tilde{x}_1, \mathbf{v})$ is a low rank separated function in $\tilde{x}_1$ and $\mathbf{v}$.

We mention that the derivation of $\mathcal{O}(\varepsilon)$ term does not require specific properties of the collision kernel B . One can continue this process to derive $\mathcal{O}(\varepsilon^2)$ term, which is a low rank function as well and depends on the kernel B , see [29] for details.

4.2 Strong Shock Wave: $M_L \rightarrow \infty$

When M_L is very large, a strong shock wave will develop and one cannot hope for any asymptotic expansion as in the previous subsection. Over the years, people have tried to find various approximations to the solution in this regime and it turns out many heuristic solutions match well with the experiments, yet are low rank [4, 15]. Here we present one such approximation due to Mott-Smith, who obtained the first solution of Boltzmann's equation for the shock structure problem in 1951. More sophisticated approximations exist but they more or less follow a similar idea as Mott-Smith.

The starting point is a bimodal distribution (and low rank) approximation of f as

$$f(x_1, \mathbf{v}) = a_1(x_1)f_L(\mathbf{v}) + a_2(x_1)f_R(\mathbf{v}). \quad (4.17)$$

To satisfy the Rankine–Hugoniot equations, we must have $a_1(x_1) + a_2(x_1) \equiv 1$. We thus write $a(x_1) = a_1(x_1)$ and $a_2(x_1) = 1 - a(x_1)$. In order to determine $a(x_1)$, one additional condition is needed. The simplest way is to enforce the moment equation by multiplying Eq. (4.1) by $\int \cdot v_1^2 d\mathbf{v}$:

$$\int v_1^3 \partial_{x_1} f d\mathbf{v} = \int v_1^2 \mathcal{Q}(f, f) d\mathbf{v}, \quad (4.18)$$

which reduces to

$$a'(x_1) (\rho_L u_L (u_L^2 + 3RT_L) - \rho_R u_R (u_R^2 + 3RT_R)) = \alpha a(x_1)(1 - a(x_1)), \quad (4.19)$$

with

$$\alpha = \int v_1^2 (\mathcal{Q}(f_L, f_R) + \mathcal{Q}(f_R, f_L)) d\mathbf{v}. \quad (4.20)$$

Using (4.4), (4.19) can be further simplified to

$$(d-1)\rho_L u_L R(T_L - T_R)a'(x_1) = -\alpha a(x_1)(1 - a(x_1)). \quad (4.21)$$

This equation easily integrates to

$$a(x_1) = \frac{1}{\exp(\beta x_1) + 1}, \quad \beta = \frac{\alpha}{(d-1)\rho_L u_L R(T_L - T_R)}. \quad (4.22)$$

Therefore, we have found a closed form solution in the form of (4.17). Note that to evaluate α , we need to make use of specific properties of the collision kernel B . Accordingly, we can

see that the spatial change in ρ across the shock wave increases with increasing Mach number M_L of the upstream:

$$\frac{\rho(x_1)}{\rho_L} = \frac{1 + \frac{(d+1)M_L^2}{M_L^2+d} \exp(\beta x_1)}{1 + \exp(\beta x_1)}. \quad (4.23)$$

5 Numerical Examples

In this section, we evaluate the accuracy and efficiency of the proposed dynamical low rank method by several classical benchmark tests in rarefied gas dynamics, including normal shock wave (1D), Fourier flow (1D), lid driven cavity flow (2D), and thermally driven cavity flow (2D). All these examples concern the steady-state solution of the Boltzmann equation and we use the first order method in both time and space as described in Sect. 2, and Fourier spectral method for 2D Maxwell molecules [26] to evaluate the collision operator. The results are compared with full tensor method using the same discretization.

5.1 Convergence Criterion

Unlike time dependent problems, we need a proper stopping criterion for solving the steady state solutions.

For the full tensor method, we define the error as

$$\begin{aligned} \text{err}_{\text{full tensor}} &= \|f_{\text{full tensor}}^{n+1} - f_{\text{full tensor}}^n\|_{L^2} \\ &= \left\langle f_{\text{full tensor}}^{n+1} - f_{\text{full tensor}}^n, f_{\text{full tensor}}^{n+1} - f_{\text{full tensor}}^n \right\rangle_{\mathbf{x}, \mathbf{v}}^{\frac{1}{2}}. \end{aligned} \quad (5.1)$$

For the low rank method, we define the error similarly as

$$\text{err}_{\text{low rank}} = \|f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n\|_{L^2} = \left\langle f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n, f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n \right\rangle_{\mathbf{x}, \mathbf{v}}^{\frac{1}{2}}, \quad (5.2)$$

where $f_{\text{low rank}}^n = \sum_{i,j=1}^r X_i^n S_{ij}^n V_j^n$. Rather than reconstructing $f_{\text{low rank}}^n$, the above error term can be broke into three pieces:

$$\begin{aligned} f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n &= \sum_{i,j=1}^r X_i^{n+1} S_{ij}^{n+1} V_j^{n+1} - \sum_{i,j=1}^r X_i^n S_{ij}^n V_j^n \\ &= \sum_{j=1}^r \left(K_j^{n+1} - K_j^n \right) V_j^n \\ &\quad + \sum_{i,j=1}^r X_i^{n+1} \left(S_{ij}^{(2)} - S_{ij}^{(1)} \right) V_j^n + \sum_{i=1}^r X_i^{n+1} \left(L_i^{n+1} - L_i^n \right) \\ &:= \sum_{j=1}^r \Delta K_j V_j^n + \sum_{i,j=1}^r X_i^{n+1} \Delta S_{ij} V_j^n + \sum_{i=1}^r X_i^{n+1} \Delta L_i \end{aligned} \quad (5.3)$$

where the notation follows Sect. 2.4. By orthogonality of $\{X_i\}_{1 \leq i \leq r}$ and $\{V_j\}_{1 \leq j \leq r}$, (5.2) can be simplified as

$$\begin{aligned} \text{err}_{\text{low rank}}^2 &= \left\langle f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n, f_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n \right\rangle_{\mathbf{x}, \mathbf{v}} \\ &= \sum_{j=1}^r \langle \Delta K_j, \Delta K_j \rangle_{\mathbf{x}} + \sum_{i,j=1}^r \Delta S_{ij}^2 + \sum_{i=1}^r \langle \Delta L_i, \Delta L_i \rangle_{\mathbf{v}} + \text{I} + \text{II} + \text{III}, \end{aligned} \quad (5.4)$$

where I, II and III are cross terms:

$$\begin{aligned} \text{I} &= 2 \sum_{i,j=1}^r \left\langle \Delta K_j, X_i^{n+1} \right\rangle_{\mathbf{x}} \Delta S_{ij}, \\ \text{II} &= 2 \sum_{i,j=1}^r \left\langle \Delta L_i, V_j^n \right\rangle_{\mathbf{v}} \Delta S_{ij}, \\ \text{III} &= 2 \sum_{i,j=1}^r \left\langle \Delta K_j, X_i^{n+1} \right\rangle_{\mathbf{x}} \cdot \left\langle \Delta L_i, V_j^n \right\rangle_{\mathbf{v}}. \end{aligned} \quad (5.5)$$

We emphasize that it is crucial to evaluate $\text{err}_{\text{low rank}}$ using (5.4)–(5.5), since the cost of reconstructing $f_{\text{low rank}}^n$ is $\mathcal{O}(r^2 N_{\mathbf{x}}^d N_{\mathbf{v}}^d)$ which is comparable to a full tensor method.

In general, we set a fixed convergence tolerance **res_tol** and terminate the time iteration whenever $\text{err}_{\text{low rank}}, \text{err}_{\text{full tensor}} \leq \mathbf{res_tol}$ for both the full tensor method and low rank method.

For the adaptive low rank method discussed in Sect. 3, we have

$$|\text{err}_{\text{low rank}} - \text{err}_{\text{low rank}}^{\text{ada}}| \leq \|\tilde{f}_{\text{low rank}}^{n+1} - f_{\text{low rank}}^{n+1}\|_{L^2} \leq (r - r')^{\frac{1}{2}} \cdot \mathbf{drop_tol}, \quad (5.6)$$

where $\text{err}_{\text{low rank}}^{\text{ada}} = \|\tilde{f}_{\text{low rank}}^{n+1} - f_{\text{low rank}}^n\|_{L^2}$, $\tilde{f}_{\text{low rank}}^{n+1}$ is the solution at the end of time step t^n after adding and removing basis. We dynamically set $\mathbf{drop_tol} = c \cdot \text{err}_{\text{low rank}}^{\text{ada}}$ and control $\text{err}_{\text{low rank}}^{\text{ada}}$ through

$$\frac{1}{1 + c(r - r')^{\frac{1}{2}}} \text{err}_{\text{low rank}} \leq \text{err}_{\text{low rank}}^{\text{ada}} \leq \frac{1}{1 - c(r - r')^{\frac{1}{2}}} \text{err}_{\text{low rank}}. \quad (5.7)$$

In the following tests, we set $c = 0.2$ and always use the adaptive dynamical low rank method with convergence criterion $\text{err}_{\text{low rank}}^{\text{ada}} \leq \mathbf{res_tol}$.

5.2 Normal Shock Wave

We first consider the normal shock problem (4.1)–(4.2) with several different Mach numbers M_L . We take $R = 1$, $d = 2$, hence $\gamma = 2$, $M_L = \frac{u_L}{(2T_L)^{1/2}}$. In the following, the spatial domain is chosen as $x_1 \in [-30, 30]$ with $N_{\mathbf{x}} = 1000$; and the velocity domain is $(v_1, v_2) \in [-L_{\mathbf{v}}, L_{\mathbf{v}}]^2$.

We choose the upstream and downstream condition as

$$\begin{aligned} (\rho_L, \rho_R) &= \left(1, \frac{3M_L^2}{M_L^2 + 2}\right), \quad (u_L, u_R) = \left(\sqrt{2}M_L, \frac{\rho_L u_L}{\rho_R}\right), \\ (T_L, T_R) &= \left(1, \frac{4M_L^2 - 1}{3\rho_R}\right), \end{aligned}$$

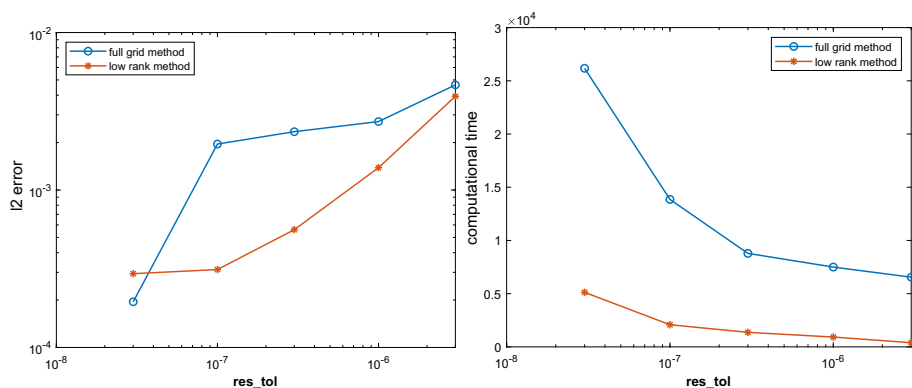


Fig. 1 Normal shock wave (Mach 1.4). Left: error of the full grid method and the adaptive low rank method for different convergence criterion **res_tol**. Right: computational time in seconds for both methods

and the initial condition as

$$\rho_0(x_1) = \frac{\tanh(\alpha x_1) + 1}{2(\rho_R - \rho_L)} + \rho_L, \quad T_0(x_1) = \frac{\tanh(\alpha x_1) + 1}{2(T_R - T_L)} + T_L,$$

$$\mathbf{u}_0(x_1) = \left(\frac{\tanh(\alpha x_1) + 1}{2(u_R - u_L)} + u_L, 0 \right),$$

with $\alpha = 0.5$.

When showing the numerical results, we are mainly interested in the macroscopic quantities: density $\rho(x_1)$, bulk velocity $u(x_1)$ (in first dimension) and temperature $T(x_1)$. Their normalized values will be plotted, which are defined by

$$\hat{\rho}(x_1) = \frac{\rho(x_1) - \rho_L}{\rho_R - \rho_L}, \quad \hat{u}(x_1) = \frac{u(x_1) - u_R}{u_L - u_R}, \quad \hat{T}(x_1) = \frac{T(x_1) - T_L}{T_R - T_L}.$$

5.2.1 Weak Shock Wave: Mach 1.4

In this subsection we consider Mach number to be $M_L = 1.4$ and set $N_v = 32$, $L_v = 13.11$. We set the reference solution f_{ref} as the solution from the full grid method with convergence criterion **res_tol** = 4×10^{-10} .

We check both the full grid method and adaptive low rank method by varying convergence criterion **res_tol**. The error is defined as $\|f_{\text{ref}} - f_{\text{num}}\|_{L^2}$ where f_{num} is the solution from either the full grid or low rank method. At the same time, we record the computational time needed for both methods to reach the same convergence criterion.

From Fig. 1, we can see that the low rank method can achieve the same accuracy much more efficiently compared to the full grid method. From Fig. 2, we can see that both methods match well with the reference solution. The rank in the adaptive low rank method grows slowly as time evolves and is stabilized to 16 before reaching the convergence criterion.

5.2.2 Strong Shock Wave: Mach 3.8 & Mach 6.5

In this subsection we consider the strong shock wave with two different Mach numbers $M_L = 3.8$ and $M_L = 6.5$. We compare the full grid method and the adaptive low rank method using the same convergence criterion **res_tol** = 4.6×10^{-7} .

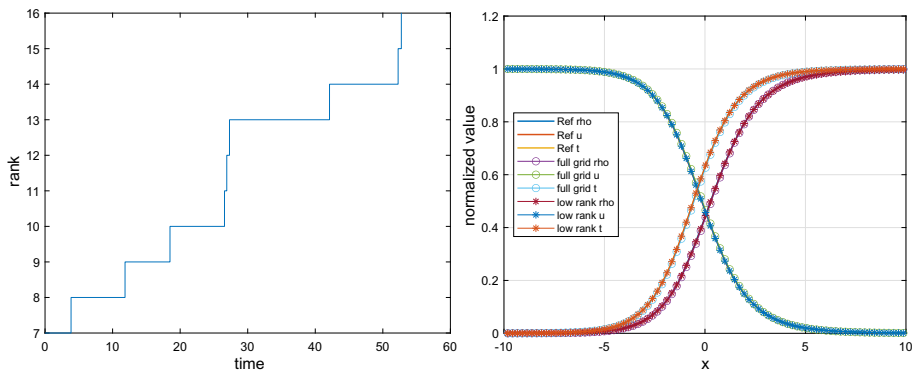


Fig. 2 Normal shock wave (Mach 1.4). Left: rank evolution of the adaptive low rank method. Right: normalized density, bulk velocity and temperature of the full grid method and the adaptive low rank method using $\text{res_tol} = 3 \times 10^{-7}$, in comparison to the reference solution

For the case $M_L = 3.8$, we use $N_v = 32$ and $L_v = 20.97$. The full grid method needs 18540 seconds to converge; and the adaptive low rank method needs 7556 seconds to converge. For the case $M_L = 6.5$, we use $N_v = 48$ and $L_v = 34.08$. The full grid method needs 44379 seconds to converge; and the adaptive low rank method needs 16157 seconds to converge.

The results of both cases are reported in Fig. 3. The full grid method and the adaptive low rank method match well. On the other hand, the rank in the adaptive low rank method behaves similarly as in the weak shock wave: the numerical rank is a bit higher but still quite low rank and stabilized before reaching the convergence criterion.

5.3 Fourier Flow

We next consider a Fourier heat transfer problem. The spatial domain is 1D: $x_1 \in [0, 2]$ with $N_x = 200$; and the velocity domain is 2D: $(v_1, v_2) \in [-L_v, L_v]^2$ with $L_v = 7.86$ and $N_v = 32$. The Maxwell diffusive boundary condition is assumed at $x_1 = 0$ with wall quantities $\mathbf{u}_w = (0, 0)$, $T_w = 1$ and $x_1 = 2$ with $\mathbf{u}_w = (0, 0)$, $T_w = 1.2$. For the initial condition, we use a spatially homogeneous Maxwellian with $\rho_0 = 1$, $\mathbf{u}_0 = (0, 0)$ and $T_0 = 1$.

The convergence criterion is set as $\text{res_tol} = 2 \times 10^{-7}$ for both the full grid method and the adaptive low rank method. For the full grid method, we need 925 seconds to reach convergence, while for the low rank method, we only need 509 seconds. The temperature profile as shown in Fig. 4 matches well for both methods. Furthermore, we can see that the numerical rank in the adaptive low rank method is stabilized to 11 in a very short time.

5.4 Lid Driven Cavity Flow

We now consider the 2D lid driven cavity flow problem. The spatial domain is rectangular $(x_1, x_2) \in [0, 0.5]^2$ with $N_x = 100$ in each dimension; and the velocity domain is $(v_1, v_2) \in [-L_v, L_v]^2$ with $L_v = 7.86$ and $N_v = 32$. The Maxwell diffusive boundary condition is assumed at all boundaries. The wall quantities at $x_2 = 0.5$ are $\mathbf{u}_w = (1, 0)$, $T_w = 1$, while at all other boundaries we set $\mathbf{u}_w = (0, 0)$, $T_w = 1$. For the initial condition, we use a spatially homogeneous Maxwellian with $\rho_0 = 1$, $\mathbf{u}_0 = (1, 1)$ and $T_0 = 1$.

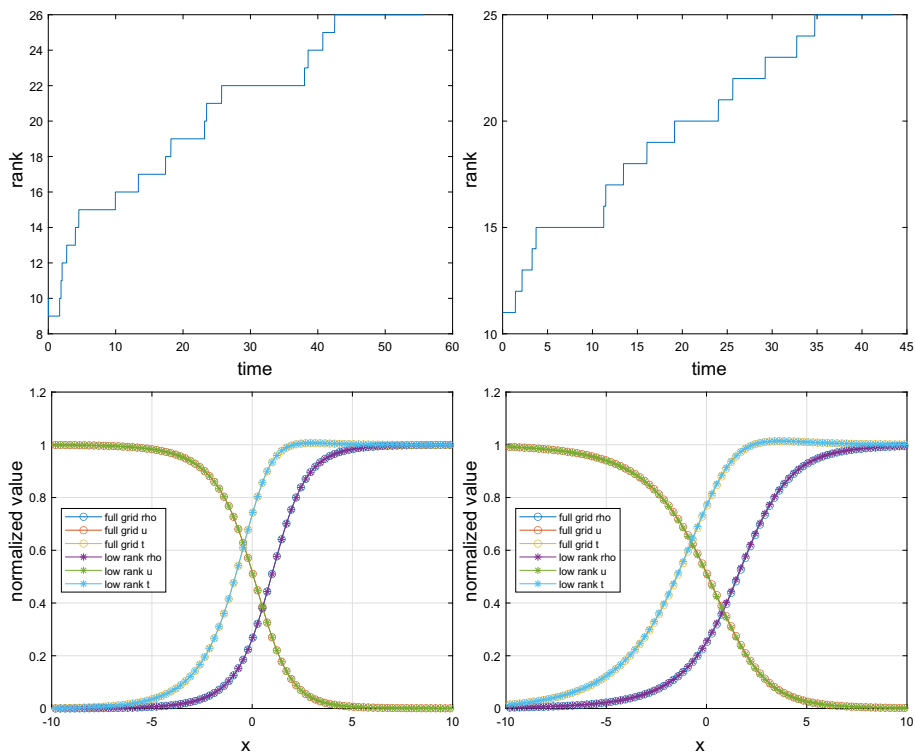


Fig. 3 Normal shock wave (Mach 3.8 & Mach 6.5). Top: rank evolution of the adaptive low rank method with Mach 3.8 (Left) and Mach 6.5 (Right); Bottom: normalized density, bulk velocity and temperature of the full grid method and the adaptive low rank method using $\text{res_tol} = 4.6 \times 10^{-7}$ with Mach 3.8 (Left) and Mach 6.5 (Right)

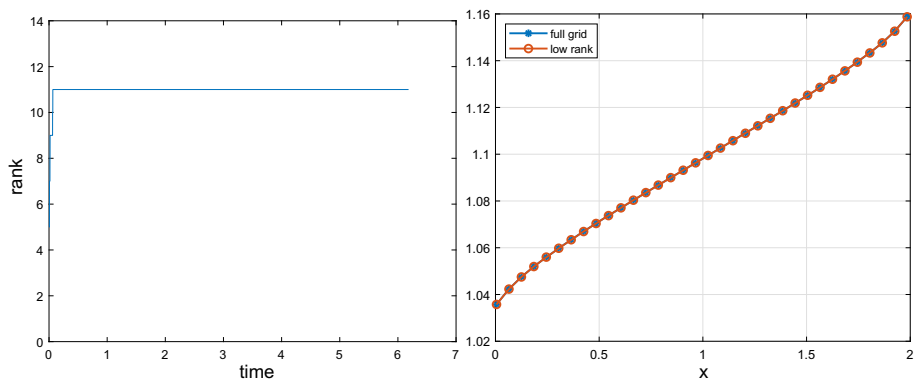


Fig. 4 Fourier flow. Left: rank evolution in the adaptive low rank method; Right: temperature profile of the full grid method and the adaptive low rank method using $\text{res_tol} = 2 \times 10^{-7}$

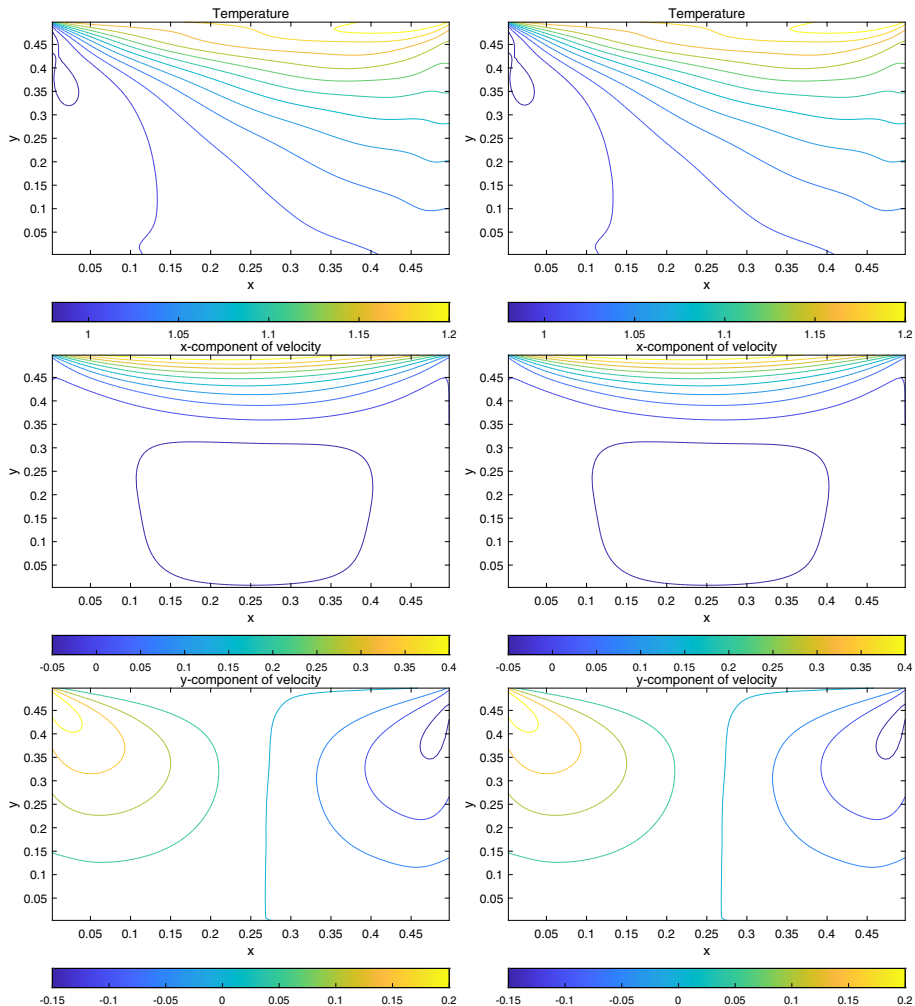


Fig. 5 Lid driven cavity flow. Top: temperature profile of the full grid method (Left) and low rank method (Right); Middle: x_1 -component velocity of the full grid method (Left) and low rank method (Right); Bottom: x_2 -component velocity of the full grid method (Left) and low rank method (Right). Convergence criterion is $\text{res_tol} = 2 \times 10^{-7}$ for both methods

The convergence criterion is set as $\text{res_tol} = 2 \times 10^{-7}$ for both the full grid method and the adaptive low rank method. For the full grid method, we need 29043 seconds to reach convergence, while for the low rank method, we only need 8323 seconds. We compare the temperature and velocity profile in Fig. 5 and a good match is obtained.

From Fig. 6, we can see that the rank in the adaptive low rank method is increasing with time and no stabilization is observed here, which implies this is an intrinsically high rank problem. Nevertheless, the error decay in the adaptive low rank method behaves similarly as in the full grid method (so our adaptive procedure does reasonable things in the actual simulation).

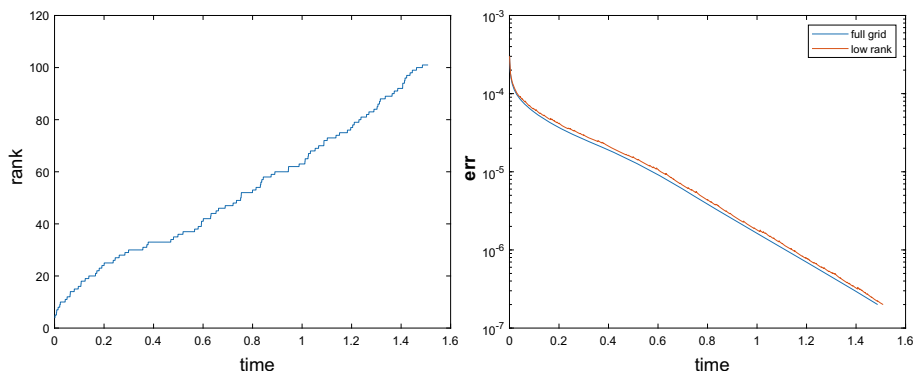
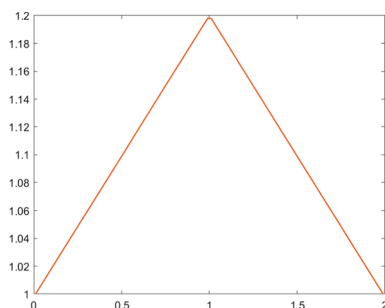


Fig. 6 Lid driven cavity flow. Left: rank evolution in the adaptive low rank method; Right: error decaying behaviors of the full grid method ($\text{err}_{\text{full tensor}}$) and low rank method ($\text{err}_{\text{ada low rank}}$)

Fig. 7 Thermally driven cavity flow. Wall temperature profile at $x_2 = 0$ and $x_2 = 2$



5.5 Thermally Driven Cavity Flow

We finally consider the 2D flow induced by thermal gradients. The spatial domain is rectangular $(x_1, x_2) \in [0, 2]^2$ with $N_x = 100$ in each dimension; and the velocity domain is $(v_1, v_2) \in [-L_v, L_v]^2$ with $L_v = 6.55$ and $N_v = 32$. The Maxwell diffusive boundary condition is assumed at all boundaries. We set the wall quantities at $x_2 = 0, x_2 = 2$ with $\mathbf{u}_w = (0, 0)$ and T_w follows a linear function ranging from 1 to 1.2 as in Fig. 7. At $x_1 = 0, x_1 = 2$, the wall quantities are set with $\mathbf{u}_w = (0, 0)$ and $T_w = 1$. For the initial condition, we use a spatially homogeneous Maxwellian with $\rho_0 = 1, \mathbf{u}_0 = (0, 0)$ and $T_0 = 1$.

The convergence criterion is set as $\text{res_tol} = 2 \times 10^{-7}$ for both the full grid method and the adaptive low rank method. For the full grid method, we need 19011 seconds to reach convergence criterion, while for the low rank method, we only need 7112 seconds. We plot the temperature and velocity profile for both methods as in Fig. 8 where we can see a good match.

Similarly as in the previous test, we track the rank evolution in the adaptive low rank method and the error decay behavior of both methods in Fig. 9. For this problem, the rank increases more rapidly, yet the low rank method can still produce reasonable solution more efficiently compared to the full grid method.

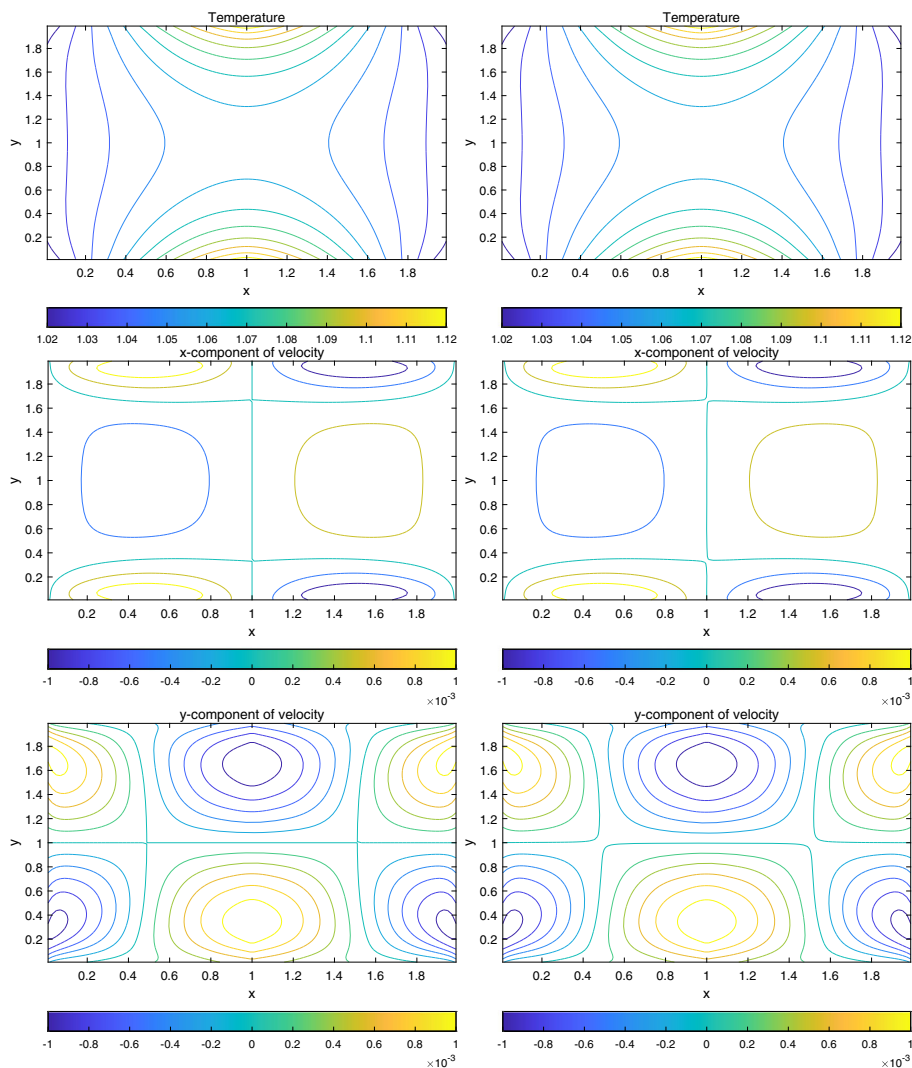


Fig. 8 Thermally driven cavity flow. Top: temperature profile of the full grid method (Left) and low rank method (Right); Middle: x_1 -component velocity of the full grid method (Left) and low rank method (Right); Bottom: x_2 -component velocity of the full grid method (Left) and low rank method (Right). Convergence criterion is $\text{res_tol} = 2 \times 10^{-7}$ for both methods

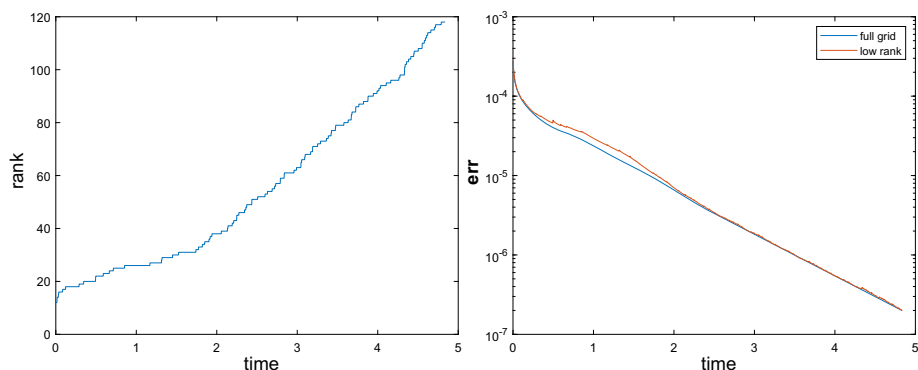


Fig. 9 Thermally driven cavity flow. Left: rank evolution in the adaptive low rank method; Right: error decaying behaviors of the full grid method ($\text{err}_{\text{full tensor}}$) and low rank method ($\text{err}_{\text{low rank}}^{\text{ada}}$)

6 Conclusions

We have introduced an adaptive dynamical low rank method for the nonlinear Boltzmann equation, concerning in particular the steady state computation. This method employs the fast Fourier spectral method (for the collision operator) and the dynamical low rank method to obtain computational efficiency. An adaptive strategy was introduced to incorporate the boundary information and control the computational rank by monitoring the residual error. A series of benchmark tests were performed to demonstrate the efficiency and accuracy of the proposed method in comparison to the full tensor grid method.

Funding This work is partially supported under the NSF CAREER grant DMS-2153208, NSF CDS & E grant CBET-1854829, and AFOSR grant FA9550-21-1-0358.

Data availability This manuscript has no associated data.

Declarations

Competing interests The authors declare that they have no conflict of interest.

References

1. Bird, G.A.: Molecular gas dynamics and the direct simulation of gas flows. Molecular gas dynamics and the direct simulation of gas flows, (1994)
2. Birdsall, C.K., Langdon, A.B.: Plasma Physics Via Computer Simulation. CRC Press, Boca Raton (2004)
3. Cercignani, C.: The Boltzmann Equation and Its Applications. Springer, New York (1988)
4. Cercignani, C.: Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations, vol. 21. Cambridge University Press, Cambridge (2000)
5. Ceruti, G., Kusch, J., Lubich, C.: A rank-adaptive robust integrator for dynamical low-rank approximation. BIT Numerical Mathematics, 1–26 (2022)
6. Dektor, A., Rodgers, A., Venturi, D.: Rank-adaptive tensor methods for high-dimensional nonlinear pdes. J. Sci. Comput. **88**(2), 1–27 (2021)
7. Deshpande, S. M.: Kinetic theory based new upwind methods for inviscid compressible flows. AIAA Paper 86-0275, (1986)
8. Dimarco, G., Pareschi, L.: Numerical methods for kinetic equations. Acta Numer **23**, 369–520 (2014)
9. Einkemmer, L.: A low-rank algorithm for weakly compressible flow. SIAM J. Sci. Comput. **41**(5), A2795–A2814 (2019)

10. Einkemmer, L., Jingwei, H., Wang, Y.: An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation. *J. Comput. Phys.* **439**, 110353 (2021)
11. Einkemmer, L., Jingwei, H., Ying, L.: An efficient dynamical low-rank algorithm for the Boltzmann-BGK equation close to the compressible viscous flow regime. *SIAM J. Sci. Comput.* **43**, B1057–B1080 (2021)
12. Einkemmer, L., Lubich, C.: A low-rank projector-splitting integrator for the Vlasov-Poisson equation. *SIAM J. Sci. Comput.* **40**(5), B1330–B1360 (2018)
13. Einkemmer, L., Lubich, C.: A quasi-conservative dynamical low-rank algorithm for the Vlasov equation. *SIAM J. Sci. Comput.* **41**(5), B1061–B1081 (2019)
14. Gamba, I.M., Haack, J.R., Hauck, C.D., Hu, J.: A fast spectral method for the Boltzmann collision operator with general collision kernels. *SIAM J. Sci. Comput.* **39**(4), B658–B674 (2017)
15. Harris, S.: *An Introduction to the Theory of the Boltzmann Equation*. Dover Publications, New York (2004)
16. Hesthaven, J., Pagliantini, C., Ripamonti, N.: Rank-adaptive structure-preserving model reduction of Hamiltonian systems. *ESAIM: M2AN*. **56**, 617–650 (2022)
17. Hu, J.: Fourier spectral methods for nonlinear Boltzmann equations (in Chinese). *Math. Numer. Sin.* to appear (2022)
18. Jaiswal, S., Alexeenko, A., Hu, J.: A discontinuous Galerkin fast spectral method for the full Boltzmann equation with general collision kernels. *J. Comput. Phys.* **378**, 178–208 (2019)
19. Koch, O., Lubich, C.: Dynamical low-rank approximation. *SIAM J. Matrix Anal. Appl.* **29**(2), 434–454 (2007)
20. Kusch, J., Einkemmer, L., Ceruti, G.: On the stability of robust dynamical low-rank approximations for hyperbolic problems. *arXiv preprint [arXiv:2107.07282](https://arxiv.org/abs/2107.07282)* (2021)
21. Kusch, J., Stammer, P.: A robust collision source method for rank adaptive dynamical low-rank approximation in radiation therapy. *arXiv preprint [arXiv:2111.07160](https://arxiv.org/abs/2111.07160)* (2021)
22. Lubich, C., Oseledets, I.V.: A projector-splitting integrator for dynamical low-rank approximation. *BIT Numer. Math.* **54**(1), 171–188 (2014)
23. Mandal, J.C., Deshpande, S.M.: Kinetic flux vector splitting for Euler equations. *Computers & fluids* **23**(2), 447–478 (1994)
24. Markowich, P.A., Ringhofer, C.A., Schmeiser, C.: *Semiconductor Equations*. Springer Science & Business Media, Berlin (2012)
25. Mouhot, C., Pareschi, L.: Fast algorithms for computing the Boltzmann collision operator. *Math. Comp.* **75**, 1833–1852 (2006)
26. Mouhot, C., Pareschi, L.: Fast algorithms for computing the Boltzmann collision operator. *Math. Comput.* **75**(256), 1833–1852 (2006)
27. Naldi, G., Pareschi, L., Toscani, G.: *Mathematical Modeling of Collective Behavior in Socio-economic and Life Sciences*. Springer Science & Business Media, Berlin (2010)
28. Nanbu, K.: Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent gases. *J. Phys. Soc. Jpn.* **49**(5), 2042–2049 (1980)
29. Ohwada, T.: Structure of normal shock waves: direct numerical analysis of the Boltzmann equation for hard-sphere molecules. *Phys. Fluids* **5**, 217–234 (1993)
30. Pareschi, L., Perthame, B.: A Fourier spectral method for homogeneous Boltzmann equations. *Transport Theory Statist. Phys.* **25**, 369–382 (1996)
31. Pareschi, L., Russo, G.: Numerical solution of the Boltzmann equation I: spectrally accurate approximation of the collision operator. *SIAM J. Numer. Anal.* **37**, 1217–1245 (2000)
32. Peng, Z., McClarren, R.G., Frank, M.: A low-rank method for two-dimensional time-dependent radiation transport calculations. *J. Comput. Phys.* **421**, 109735 (2020)
33. Rodgers, A., Dektor, A., Venturi, D.: Adaptive integration of nonlinear evolution equations on tensor manifolds. *J. Sci. Comput.* **92**(2), 1–31 (2020). [arXiv:2008.00155](https://arxiv.org/abs/2008.00155)
34. Villani, C.: A review of mathematical topics in collisional kinetic theory. In S. Friedlander and D. Serre, editors, *Handbook of Mathematical Fluid Mechanics*, vol. I, pp. 71–305. North-Holland (2002)