Journal of Computational Mathematics, Vol.27, No.2-3, 2009, 184–195.

THREE WAY DECOMPOSITION FOR THE BOLTZMANN EQUATION*

Ilgis Ibragimov
Elegant Mathematics Ltd., Hanauer Muehle 2, 66564 Ottweiler, Germany
Email: ii@elegant-mathematics.com
Sergej Rjasanow
Department of Mathematics and Computer Science, University of Saarland, Saarbrücken, Germany
Email: rjasanow@num.uni-sb.de

Abstract

The initial value problem for the spatially homogeneous Boltzmann equation is considered. A deterministic numerical scheme for this problem is developed by the use of the three way decomposition of the unknown function as well as of the collision integral. On this way, almost linear complexity of the algorithm is achieved. Some numerical examples are presented.

Mathematics subject classification: 82C40, 82C80, 65R20.
Key words: Boltzmann equation, Deterministic scheme, Three way decomposition.

1. Introduction

The object of our considerations is the initial value problem for the classical spatially homogeneous Boltzmann equation

$$\frac{\partial}{\partial t} f(t,v) = Q(f,f)(t,v), \quad t \in \mathbb{R}_+, \qquad f(0,v) = f_0(v), \quad v \in \mathbb{R}^3, \tag{1.1}$$

which describes the time evolution of the particle density

$$f : \mathbb{R}_+ \times \mathbb{R}^3 \to \mathbb{R}_+$$

from its initial value f_0 to the final Maxwell distribution

$$\lim_{t\to\infty} f(t,v) = f_M(v) = \frac{\varrho_0}{(2\pi\, T_0)^{3/2}}\, e^{-\frac{|v-V_0|^2}{2\, T_0}}\ . \tag{1.2}$$

The right-hand side of the equation (1.1), known as the collision integral or the collision term, is of the form

$$Q(f,f)(t,v) = \int\limits_{\mathbb{R}^3}\int\limits_{S^2} B(v,w,e)\big(f(t,v')f(t,w') - f(t,v)f(t,w)\big)\, de\, dw. \tag{1.3}$$

The following notations have been used in (1.3): $v, w \in \mathbb{R}^3$ are the pre-collision velocities, $e \in S^2 \subset \mathbb{R}^3$ is a unit vector, $v', w' \in \mathbb{R}^3$ are the post-collision velocities, and $B(v,w,e)$ is the collision kernel. The operator $Q(f,f)$ represents the change of the distribution function f due to the binary collisions between particles. A single collision results in the change of the velocities of the colliding partners

$$v, w \to v', w'. \tag{1.4}$$

* Received January 9, 2008 / Revised version received April 25, 2008 / Accepted May 4, 2008 /

The collision transformation (1.4) conserves the momentum and the energy

$$v + w = v' + w', \quad |v|^2 + |w|^2 = |v'|^2 + |w'|^2.$$

It can be written in the following form

$$v' = \frac{1}{2}\Big(v + w + |u|e\Big), \quad w' = \frac{1}{2}\Big(v + w - |u|e\Big), \quad e \in S^2,$$

where $u = v - w$ denotes the relative velocity of the colliding particles. We will deal with an isotropic cut-off kernel B, namely with the Variable Hard Spheres model (VHS) [3]

$$B(v, w, e) = C_\lambda\, |u|^\lambda, \quad -3 < \lambda \leq 1. \tag{1.5}$$

The model includes, as particular cases, the hard spheres model for $\lambda = 1$ and a special case of the Maxwell pseudo–molecules with $\lambda = 0$.

All relevant physical values of the gas flow are computed as the first 13 moments of the distribution function or their combinations. These moments are: the density

$$\varrho(t) = \int\limits_{\mathbb{R}^3} f(t, v)\, dv, \tag{1.6}$$

the momentum

$$m(t) = \int\limits_{\mathbb{R}^3} v\, f(t, v)\, dv, \tag{1.7}$$

the momentum flow

$$M(t) = \int\limits_{\mathbb{R}^3} vv^\top f(t, v)\, dv, \tag{1.8}$$

and the energy flow

$$r(t) = \frac{1}{2}\int\limits_{\mathbb{R}^3} v|v|^2 f(t, v)\, dv. \tag{1.9}$$

Note that the matrix M is symmetric and therefore defined by its upper triangle. Using these moments, we define the bulk velocity

$$V(t) = m(t)/\varrho(t), \tag{1.10}$$

the internal energy and the temperature

$$e(t) = \frac{1}{2\,\varrho(t)}\big(\mathrm{tr}M(t) - \varrho(t)|V(t)|^2\big)\,, \quad T(t) = \frac{2}{3}e(t), \tag{1.11}$$

the pressure

$$p(t) = \varrho(t)T(t), \tag{1.12}$$

the stress tensor

$$P(t) = M(t) - \varrho(t)V(t)V(t)^\top,$$

and the heat flux vector

$$q(t) = r(t) - \left(M(t) + \Big(\frac{1}{2}\mathrm{tr}M(t) - \varrho(t)|V(t)|^2\Big)I\right)V(t).$$

Note that in the spatially homogeneous case we consider here, the following important conservation properties hold. The density, the momentum, and the trace of the momentum flow remain constant during the relaxation

$$\varrho(t) = \varrho_0 = \int\limits_{\mathbb{R}^3} f_0(v)\,dv, \quad m(t) = m_0 = \int\limits_{\mathbb{R}^3} v\, f_0(v)\,dv,$$
$$\mathrm{tr}M(t) = \mathrm{tr}M_0 = \mathrm{tr}\int\limits_{\mathbb{R}^3} vv^\top f_0(v)\,dv.$$

Thus, corresponding to (1.10), (1.11), and (1.12), the bulk velocity, the internal energy, the temperature and the pressure are conserved quantities

$$V(t) = V_0, \quad e(t) = e_0, \quad T(t) = T_0, \quad p(t) = p_0.$$

The density ϱ_0, the bulk velocity V_0, and the temperature T_0 are the parameters of the final Maxwell distribution f_M in (1.2).

In [10], we have used the following form of the collision integral for its numerical computation

$$Q(f,f)(t,v) = \mathcal{F}_{y\to v}\left(\int\limits_{\mathbb{R}^3} T(u,y)\mathcal{F}^{-1}_{z\to y}\Big(f(t,z-u)f(t,z+u)\Big)(t,u,y)du\right)(t,v), \tag{1.13}$$

where $\mathcal{F}$ denoted the Fourier transform. The time independent kernel T is defined as follows

$$T(u,y) = 8\int\limits_{S^2} B(2v,2w,e)\Big(e^{-\imath|u|(y,e)} - e^{-\imath(y,u)}\Big)de. \tag{1.14}$$

For the VHS model of interaction (1.5), the integral in (1.14) can be computed analytically and the kernel T takes the form

$$T(u,y) = 2^{5+\lambda}\pi C_\lambda |u|^\lambda\Big(\mathrm{sinc}(|u|\,|y|) - e^{-\imath(y,u)}\Big), \tag{1.15}$$

where the abbreviation

$$\mathrm{sinc}z = \frac{\sin z}{z}, \quad z \in \mathbb{R}$$

has been used.

Before we begin the detailed description of our new numerical method, we refer to the papers [10] and [6], where an overview of the results known from the literature is given.

The paper is organised as follows. In Section 2, we give a motivation for our approach based on the Three Way Decomposition (TWD). The numerical algorithm for the Boltzmann equation will be formulated in Section 3. In the fourth and final section, we present the results of some numerical tests. Here, we use the analytically known time relaxation of the moments (1.6)-(1.9) for the Maxwell pseudo-molecules model for a careful check of accuracy.

2. Motivation

The numerical solution of the Boltzmann equation starts with the definition of the set of discrete velocities

$$C_v = \big\{v_j = V + h_v j,\ \ j\in C_n\big\}, \quad V\in\mathbb{R}^3, \quad h_v\in\mathbb{R}_+, \tag{2.1}$$

where the set C_n of three-dimensional indices j is defined with a natural even number n as follows

$$C_n = \big\{j\in\mathbb{Z}^3, \quad -n/2 \le j_\ell \le n/2, \quad \ell = 1,2,3\big\}.$$

In [10], we solve the initial value problem (1.1) on this grid with the use of the Fast Fourier Transform (FFT) on the grid (2.1) and with the help of the Runge-Kutta methods in time. The memory requirement is $9/16n^4+\mathcal{O}(n^2)$ for the values of the kernel (1.15) on the grid. This value is close to the optimal $\mathcal{O}(n^3)$. However the numerical work is $n^6/8+\mathcal{O}(n^5\log_2 n)$ for every time step, which is far from being optimal.

On the grid (2.1), a given density function f will be represented in a tensor like form

$$F(t)=\big(f_j(t)\big)_{j\in C_n}=\big(f_{j_1,j_2,j_3}(t)\big)_{j_1,j_2,j_3=-n/2}^{n/2}\in\mathbb{R}^{(n+1)\times(n+1)\times(n+1)}, \tag{2.2}$$

with $f_j(t)=f(t,v_j)$ and, therefore, will require $\mathcal{O}(n^3)$ words of memory. However, if the function f is degenerate in the variable v

$$f(t,v)=\sum_{k=1}^{r(t)}\beta_k(t)\prod_{\ell=1}^{3}f_k^{(\ell)}\big(t,v^{(\ell)}\big),\quad v=\big(v^{(1)},v^{(2)},v^{(3)}\big)^\top \tag{2.3}$$

with $r(t)\le r$, then its discretisation (2.2)

$$F(t)=\big(f_j(t)\big)_{j\in C_n},\quad f_j(t)=\sum_{k=1}^{r(t)}\beta_k(t)\prod_{\ell=1}^{3}f_k^{(\ell)}\big(t,v_j^{(\ell)}\big),\quad j\in C_n \tag{2.4}$$

will require at most $3r(n+1)+3$ words of memory, i.e. a linear amount for $n\to\infty$. We will refer to the number r in (2.3)–(2.4) as the rank of the function f or of the tensor F. As we will see later, numerical work can also be significantly reduced if the distribution function is degenerate. It is clear that the majority of realistic distribution functions is not degenerate. However, some of them, related to the Boltzmann equation, are degenerate (Maxwell distribution, BKW solution) or can be approximated up to the accuracy ε by a degenerate function f_ε, i.e.

$$\|F-F_\varepsilon\|_F\le\varepsilon\|F\|_F,\quad \|F\|_F=\sqrt{\sum_{j\in C_n}\big(f_j\big)^2}, \tag{2.5}$$

where now F_ε is of the form (2.4). Exactly this approximation is called Three Way Decomposition (TWD). If the accuracy of this approximation is correctly related to the accuracy of the discretisation in time and velocity space, then this additional error can not be seen in the final numerical result.

Note, that in two-dimensional case, i.e. if F is a matrix, the problem of approximation (2.5) is completely solved in terms of truncated Singular Value Decomposition (SVD). In the Frobenius norm, this result is usually referred to Eckard and Young in 1939, [5]. However, it was first proved by Schmidt in 1907, [15]. It was generalised to all unitarily invariant norms by Mirsky in 1960, [13]. There is no generalisation of these results for three- or multi-dimensional case. However, we expect that the so called Tucker decomposition [18], first introduced by Sokolov in 1960, [16], which is more flexible than the three way decomposition (2.3), can be a good approximation

$$f(t,v)\approx f_\varepsilon(t,v)=\sum_{k_1=1}^{r_1(t)}\sum_{k_2=1}^{r_2(t)}\sum_{k_3=1}^{r_3(t)}\alpha_k(t)\prod_{\ell=1}^{3}f_{k_\ell}^{(\ell)}\big(t,v^{(\ell)}\big),\quad k=(k_1,k_2,k_3)^\top. \tag{2.6}$$

The one-dimensional functions involved in (2.6) should fulfil the following orthogonalities

$$\int_{\mathbb{R}} f_{i_\ell}^{(\ell)}\big(t,v^{(\ell)}\big)\,f_{j_\ell}^{(\ell)}\big(t,v^{(\ell)}\big)dv^{(\ell)}=\delta_{i_\ell,j_\ell},\quad \ell=1,2,3.$$

Obviously,

$$\max\big(r_1(t),r_2(t),r_3(t)\big)\le r(t), \tag{2.7}$$

where $r(t)$ is the rank of the three way decomposition, see [18]. A method for the computation of the Tucker decomposition can be found in [18]. It is based again on the SVD. The memory requirements of the three way decomposition (2.3) and the Tucker decomposition (2.6) for the function discretised on a $n \times n \times n$ grid are

$$r(t)(3n+1) \quad \text{and} \quad \big(r_1(t)+r_2(t)+r_3(t)\big)n + r_1(t)r_2(t)r_3(t)\,.$$

Corresponding to (2.7), for large n the Tucker decomposition will be memory effective. We use the notation $\max\big(r_1(t), r_2(t), r_3(t)\big)$ as the Tucker rank, as usually refers in the literature, cf. [18]. In the recent paper [11], the Tucker decomposition of the kernel (1.14) was investigated. An additional sinc-interpolation was used there together with hierarchically organised block tensor-product formats.

In what follows, we will give some motivation for an approximate representation of the solution of the Boltzmann equation in degenerate form.

2.1. Maxwell distribution

The most prominent example for degenerate function is the Maxwell distribution (1.2)

$$f_M(v) = \varrho_0 \prod_{\ell=1}^{3} \frac{1}{(2\pi T_0)^{1/2}}\, e^{-\frac{\left(v^{(\ell)}-V_0^{(\ell)}\right)^2}{2T_0}}\,.$$

Thus, the Maxwell distribution is of the three way rank 1. Its Tucker rank is equal to 1 too. Since the time relaxation of the distribution function leads to the Maxwell distribution (1.2), we can expect that the rank or/and the norm of the difference will decrease during the time.

2.2. BKW solution

The famous exact solution found by Bobylev [4] and Krook and Wu [12] for the constant collision kernel

$$B(v,w,e) = \frac{1}{4\pi} \tag{2.8}$$

is

$$f(t,v) = \frac{\varrho_0}{(2\pi T_0)^{3/2}}(\beta(t)+1)^{3/2}\left(1+\beta(t)\Big(\frac{\beta(t)+1}{2T_0}|v|^2-\frac{3}{2}\Big)\right)e^{-\frac{\beta(t)+1}{2T_0}|v|^2}.$$

Here, $\varrho_0, T_0 > 0$ are some constant parameters and the function β is defined as follows

$$\beta(t) = \frac{\beta_0\, e^{-\varrho_0\, t/6}}{1+\beta_0\left(1-e^{-\varrho_0\, t/6}\right)},$$

where $0 \le \beta_0 \le 2/3$ denotes its the initial value. This solution is a quadratic polynomial in three variables multiplied by the Maxwell distribution

$$f(t,v) = \Big(a(t) + b(t)\big((v^1)^2+(v^2)^2+(v^3)^2\big)\Big)e^{-c(t)\big((v^1)^2+(v^2)^2+(v^3)^2\big)}$$

and can be written in the form (2.3) with $r(t)=3,\ \beta_k(t)=1,\ k=1,2,3$, and

$$f_k^{(\ell)}\big(t,v^{(\ell)}\big) = \begin{cases} e^{-c(t)(v^{(\ell)})^2}, & \text{for } k \neq \ell \\ \big(a_k(t)+b(t)(v^{(k)})^2\big)e^{-c(t)(v^{(k)})^2}, & \text{for } k = \ell \end{cases} \qquad k,\ell = 1,2,3,$$

where $a_1(t)+a_2(t)+a_3(t) = a(t)$. Obviously, the three way decomposition of f is not unique, its three way rank is 3, and it remains constant during the relaxation. However, the difference to the final Maxwell distribution has the function $\beta(t)$ as a factor and, therefore, disappears exponentially. Thus, the numerical rank will decrease from 3 to 1 during the relaxation for any $\varepsilon > 0$.

However, the Tucker rank of the BKW solution is always 2. A possible decomposition is

$$f^{(\ell)}_{k_\ell}(t, v^{(\ell)}) = \begin{cases} e^{-c(t)(v^{(\ell)})^2}, & \text{for } k_\ell = 1 \\ (v^{(\ell)})^2 e^{-c(t)(v^{(\ell)})^2}, & \text{for } k_\ell = 2 \end{cases} \qquad \ell = 1, 2, 3,$$

with the core

$$\alpha_k(t) = \begin{cases} a(t), & \text{for } k = (1,1,1) \\ b(t), & \text{for } k = (2,1,1), (1,2,1), (1,1,2) \\ 0, & \text{otherwise.} \end{cases}$$

2.3. Local equilibrium

In spatially non-homogeneous case, if the distribution function f is close to a Maxwell distribution, then one can expect that the description of the flow by the Boltzmann equation is close to its description by the system of Euler equations. The numerical solution of the Boltzmann equation is, in general, much more complicated than the numerical solution of the Euler equations, because the distribution function depends on seven variables. In contrast, the system of Euler equations contains five unknown functions depending on four variables. In [14, 17] the criterion of local equilibrium is developed which is based on the approximation of the function f by the function

$$\big(a + (b, v) + (C\,v, v) + (d, v)\,|v|^2 + e\,|v|^4\big) f_M(v), \tag{2.9}$$

where the parameters $a \in \mathbb{R}$, $b, d \in \mathbb{R}^3$, and $C = C^\top \in \mathbb{R}^{3\times 3}$ are determined in such a way, that all 13 moments and, in addition, the fourth "moment"

$$\gamma = \int\limits_{\mathbb{R}^3} |v - V|^4\, f(v)\, dv - 15\,T^2$$

of the function (2.9) coincide with those of the function f. Thus the Tucker rank of this function is at most 5, we skip the details. The weighted $\mathbb{L}_2$- norm of the polynomial in (2.9)

$$Crit = \frac{1}{T}\sqrt{\frac{1}{2}\,\|\tau\|_F^2 + \frac{2}{5\,T}\,|q|^2 + \frac{1}{120\,T^2}\,\gamma^2},$$

where $\tau = P - T\,I$ denotes the trace free pressure tensor, and $\|\cdot\|_F$ the Frobenius norm, can be successfully used as a criterion of local equilibrium (see [14] for more details). The two-dimensional plot of the function $Crit$ is shown in Fig. 2.1 for a supersonic flow over an ellipse for two different Knudsen numbers. It is clear to see that in big parts of the flow picture the deviation from Maxwell distribution is rather small, i.e. it is of the small rank or/and of the small norm.

Thus, even in spatially non-homogeneous case, we can assume that the exact numerical solution (the tensor F) can be successfully approximated by a low rank tensor F_ε, cf. (2.5).

3. Numerical Scheme

In the section we describe our new numerical scheme in details.

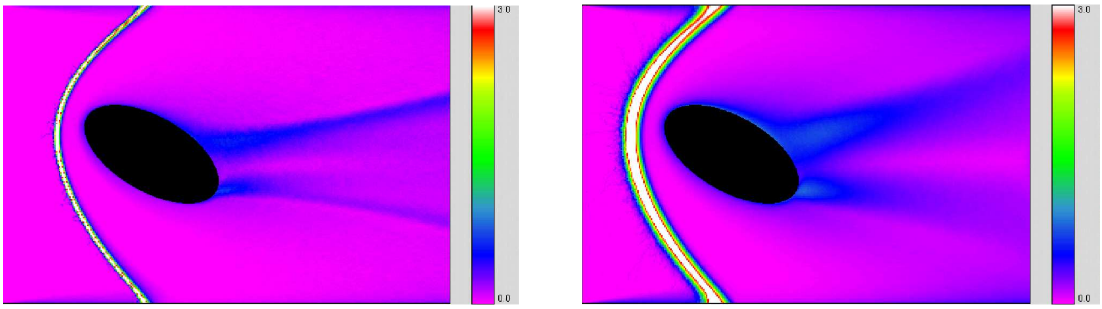

Fig. 2.1. Criteria of local equilibrium, $Kn = 0.02, 0.08$

3.1. Initial condition

The procedure starts with the Tucker decomposition of the given initial condition

$$F(0) = \big(f_0(v_j)\big)_{j\in C_n}, \tag{3.1}$$

i.e.

$$f_0(v_j) \approx f_\varepsilon(0,v_j) = \sum_{k_1=1}^{r_1}\sum_{k_2=1}^{r_2}\sum_{k_3=1}^{r_3} \alpha_k \prod_{\ell=1}^{3} f^{(\ell)}_{k_\ell}\big(v_j^{(\ell)}\big), \quad k=(k_1,k_2,k_3)^\top.$$

To obtain this approximation, we first generate the tensor (3.1) leading to $\mathcal{O}(n^3)$ arithmetical operations. Then, the approximation algorithm from [18] is applied: discretised functions f_{k_ℓ} are computed as eigenvectors corresponding to the largest r_ℓ eigenvalues of matrices:

$$A^{(\ell)} = [a_{i_\ell,i'_\ell}], \quad a_{i_\ell,i'_\ell} = \sum_{\ell\neq k,\ell'\neq k'} f_0(v_k)f_0(v_{k'}), \quad \ell=1,2,3, \quad k=(k_1,k_2,k_3)^T,$$

$$\alpha_k = \sum_{k'} f_0(v_{k'}) \prod_{\ell=1}^{3} f^{(\ell)}_{k_\ell}(v_j^{(\ell)}).$$

The $r_\ell{+}1$ eigenvalue corresponds to the approximation error in Frobenius norm up to a constant. In general, of course, we cannot expect a low Tucker rank of f_0. Thus, the existence of such low rank decomposition is rather an assumption on the initial condition.

3.2. Collision integral

For a given tensor $F_\varepsilon(t)$, we have to compute the collision integral (1.3) in the form(1.13). In addition to the set of discrete velocities introduced in (2.1), we consider two additional sets C_u and C_y for the discrete variables u and y involved in (1.13) as follows

$$C_u = \big\{u_k = h_v k\,,\ k\in C_n\big\}, \quad C_y = \big\{u_k = h_y k\,,\ k\in C_n\big\}, \quad h_y = \frac{2\pi}{n\,h_v}.$$

Then, we employ the algorithm from [10] for numerical computation of $Q(f,f)$. Note, that the time dependence and the subindex ε are omitted for simplicity.

Algorithm 3.1.

1. *Computation of the product:*
   $g^{(1)}(u_k, z_l) := f(z_l - u_k) f(z_l + u_k),\ z_l \in C_v,$
2. *Inverse Fourier Transform:*
   $g^{(2)}(u_k, y_j) := \frac{h_v^3}{(2\pi)^3} \sum_{z_m \in C_v} e^{-\imath \big(y_j,\, z_m \big)} g^{(1)}(u_k, z_m),\ y_j \in C_y,$
3. *Integration with respect to u:*
   $g^{(3)}(y_j) := \sum_{u_k \in C_u} T(u_k, y_j)\, g^{(2)}(u_k, y_j),\ y_j \in C_y,$
4. *Fourier transform:*
   $Q(f,f)(v_k) := h_y^3 \sum_{y_j \in C_y} e^{\imath \big(v_k,\, y_j \big)} g^{(3)}(y_j),\ v_k \in C_v.$

The first step of Algorithm 3.1 leads to a function $g^{(1)}$ which is again of the low rank. Indeed, if the function f is of the form

$$f(v) = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} \alpha_k \prod_{\ell=1}^{3} f_{k_\ell}^{(\ell)}\big(v^{(\ell)}\big), \quad k = (k_1, k_2, k_3)^\top,$$

then

$$f(z-u) f(z+u) = \sum_{k_1, k_1'=1}^{r_1} \sum_{k_2, k_2'=1}^{r_2} \sum_{k_3, k_3'=1}^{r_3} \alpha_k \alpha_{k'} \prod_{\ell=1}^{3} f_{k_\ell}^{(\ell)}\big(z^{(\ell)} - u^{(\ell)}\big) f_{k_\ell'}^{(\ell)}\big(z^{(\ell)} + u^{(\ell)}\big).$$

Constructing the Tucker factors

$$g_{k_\ell, k_\ell'}^{(1),(\ell)}(u^{(\ell)}, z^{(\ell)}) = f_{k_\ell}^{(\ell)}\big(z^{(\ell)} - u^{(\ell)}\big) f_{k_\ell'}^{(\ell)}\big(z^{(\ell)} + u^{(\ell)}\big),$$

we obtain the (six-dimensional) Tucker decomposition of the rank $r_1^2 \times r_2^2 \times r_3^2$ instead of $r_1 \times r_2 \times r_3$ by the function f. Note that the core $\alpha_k \alpha_{k'}$ is now six-dimensional too. The numerical complexity of this step is $\mathcal{O}\big(n^2(r_1^2 + r_2^2 + r_3^2) + r_1^2 r_2^2 r_3^2\big)$ instead of $\mathcal{O}(n^6)$ in the original algorithm.

In the second step of of Algorithm 3.1, the Fourier transform obviously retains the rank of the function $g^{(1)}$ due to

$$\begin{aligned} &\frac{h_v^3}{(2\pi)^3} \sum_{z_m \in C_v} e^{-\imath \big(y_j,\, z_m \big)} g^{(1)}(u, z_m) \\ &= \frac{h_v^3}{(2\pi)^3} \sum_{k_1, k_1'=1}^{r_1} \sum_{k_2, k_2'=1}^{r_2} \sum_{k_3, k_3'=1}^{r_3} \alpha_k \alpha_{k'} \prod_{\ell=1}^{3} \Big(\sum_{z_m \in C_v} e^{-\imath y_j^{(\ell)} z_m^{(\ell)}} g_{k_l, k_l'}^{(1),(\ell)}\big(u^{(\ell)}, z_m^{(\ell)}\big) \Big). \end{aligned}$$

The numerical complexity of this step is $\mathcal{O}\big(n^2 \log_2 n(r_1 + r_2 + r_3)\big)$. However, for moderate n, our numerical tests show that this step requires more computational time than the direct "Fourier matrix times vector" multiplication with $\mathcal{O}(n^3)$.

The most crucial step of Algorithm 3.1 is the third. Here, we have no explicit proof for the low rank property of the function $g^{(3)}$. However, this low rank approximation will be obtained numerically by the use of the adaptive algorithm described in [8]. To this end, the sum

$$\sum_{u_k \in C_u} T(u_k, y_j)\, g^{(2)}(u_k, y_j)$$

will be evaluated for only few points y_j which stay on some one-dimensional lines. The total amount of these points is of the order $\mathcal{O}(nr + r^3)$, where r denotes a priori unknown Tucker rank of the function $g^{(3)}$. The algorithm works similar to the Adaptive Cross Approximation introduced in [1,2]. It determines automatically both, the rank r and the approximation. Since one evaluation of the above sum requires $\mathcal{O}(n^3)$ arithmetical operations, the total numerical work in the third step will be $\mathcal{O}(n^4 r + n^3 r^3)$.

The final Fourier transform (the fourth step of Algorithm 3.1) does not change the low rank approximation.

Hence, Algorithm 3.1 computes $Q_\varepsilon(f_\varepsilon, f_\varepsilon)$ for a f_ε given in the Tucker form in this form again. After a recompression, this form of $Q_\varepsilon(f_\varepsilon, f_\varepsilon)$ can be used for a time integration.

3.3. Time integration

The most simple choice is the first order Euler scheme

$$F_\varepsilon(t_{i+1}) = F_\varepsilon(t_i) + \tau\, Q_\varepsilon(f_\varepsilon, f_\varepsilon), \quad i = 0, 1, \dots, \quad \tau > 0. \tag{3.2}$$

Thus, one step of the time integration leads to increase of the rank of the approximation. It makes a recompression of the tensor $F_\varepsilon(t_{i+1})$ necessary in every time step. A generalisation of the scheme (3.2) to Runge-Kutta methods of f.e. order 2 or 4 is obvious, we skip the details.

3.4. Conservation properties

In continuous case, the collision integral (1.3) has the following five conservation properties

$$\int\limits_{\mathbb{R}^3} Q(f,f)(t,v)\varphi(v)\,dv = 0, \quad \text{for} \quad \varphi(v) = 1,\ \varphi(v) = v,\ \varphi(v) = |v|^2 \tag{3.3}$$

which guarantees the conservation of mass, momentum and energy. It is also extremely important to conserve these quantities numerically, i.e. in every time step the condition

$$C\, F_\varepsilon(t_{i+1}) = C\, F_\varepsilon(0) \in \mathbb{R}^5, \quad i = 0, 1, \dots$$

should be fulfilled. Here the matrix $C \in \mathbb{R}^{5\times(n+1)^3}$ contains the values of the functions φ from (3.3) on the grid (2.1) and $F_\varepsilon \in \mathbb{R}^{(n+1)^3}$ is considered as a vector. Thus, instead of (3.2) we solve the following minimisation problem

$$\min\Big(\big\|F_\varepsilon(t_{i+1}) - F_\varepsilon(t_i) - \tau\, Q_\varepsilon(f_\varepsilon, f_\varepsilon)\big\|_2^2 + \lambda\,\big\|C\, F_\varepsilon(t_{i+1}) - C\, F_\varepsilon(0)\big\|\Big), \tag{3.4}$$

where the minimum is taken with respect to the new (non-negative) vector $F_\varepsilon(t_{i+1})$ and Lagrange multiplier λ. In order to save computational effort, we suggest to make the conservation together with the recompression. The minimisation problem 3.4 can be solved by the use of an alternate least squares minimisation which is similar to the Parallel Factor Algorithm, cf. [7]. Hence, we take an initial approximation, for instance, $F_\varepsilon(t_i)$ from the previous time step, and fix three from four matrices and tensors α_k, $f_{k_\ell}^{(\ell)}\big(v_{k_\ell}^{(\ell)}\big)$. Then, the minimisation problem (3.4) becomes quadratic with respect to the non-fixed matrix, and, therefore, can be transformed to a linear system with positive definite matrix. This algorithm converges monotonically. In the practical experiments,$F_\varepsilon(t_i)$ was always a good initial approximation.

The arithmetic complexity of this step is $\mathcal{O}\big(nr^2 + r^3\big)$, which is smaller than the complexity required for $g^{(3)}(y_j)$. However, for small n, it consumes most of the computational time due to a large number of operations with rather “short” vectors.

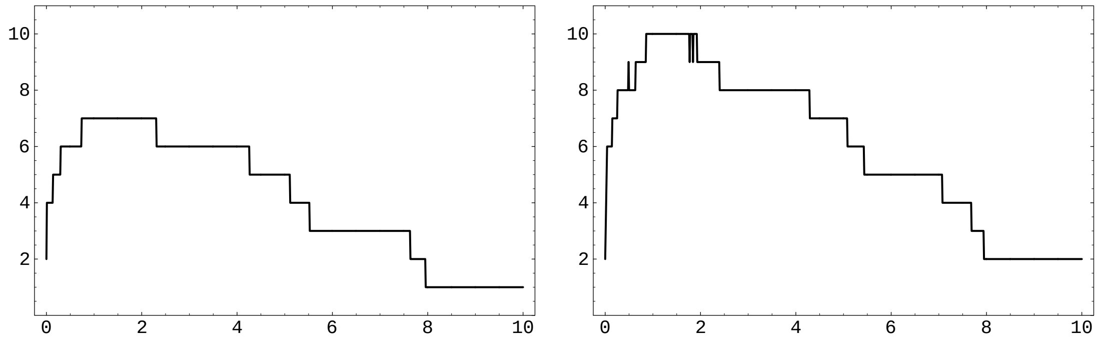

Fig. 4.1. Time relaxation of the three way rank, $\varepsilon = 10^{-6}$, $\varepsilon = 10^{-8}$

4. Numerical Experiments

As an example we consider the initial distribution $f_0(v)$ as a mixture of two different Maxwell distributions

$$f_0(v) = \alpha f_{M_1}(v) + (1-\alpha) f_{M_2}(v), \quad 0 \le \alpha \le 1.$$

Thus, its exact three way or Tucker rank is equal to 2. The parameters of the Maxwell distributions are V_1, T_1 and V_2, T_2. For the following choice

$$V_1 = (-2,2,0)^\top, \quad V_2 = (2,0,0)^\top, \quad T_1 = T_2 = 1, \quad \alpha = 1/2,$$

we obtain $\varrho_0 = 1$, $V_0 = (0,1,0)^\top$ and $T_0 = 8/3$. Furthermore, for the kernel (2.8), the time relaxation of the momentum flow and of the energy flow is

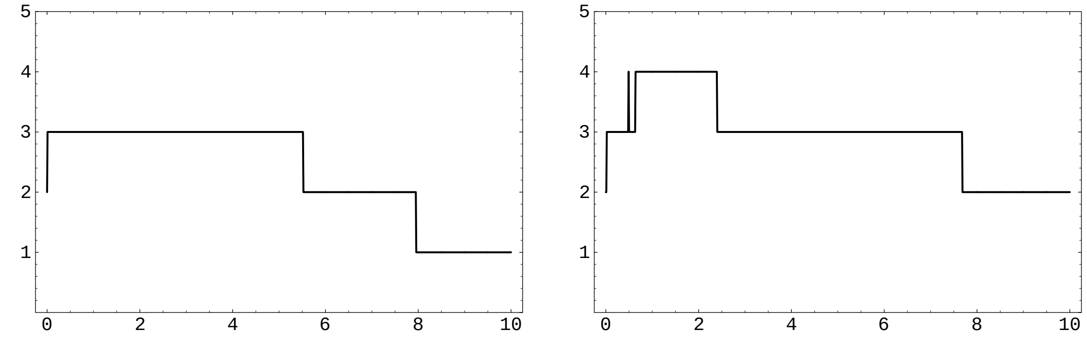

Fig. 4.2. Time relaxation of the Tucker rank, $\varepsilon = 10^{-6}$, $\varepsilon = 10^{-8}$

$$M(t) = \begin{pmatrix} 5 & -2 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} e^{-t/2} + \frac{1}{3} \begin{pmatrix} 8 & 0 & 0 \\ 0 & 11 & 0 \\ 0 & 0 & 8 \end{pmatrix} \Big(1 - e^{-t/2}\Big),$$

$$r(t) = \frac{1}{2} \begin{pmatrix} -4 \\ 13 \\ 0 \end{pmatrix} e^{-t/3} + \frac{1}{6} \begin{pmatrix} 0 \\ 43 \\ 0 \end{pmatrix} \Big(1 - e^{-t/3}\Big) - \frac{1}{6} \begin{pmatrix} 12 \\ 4 \\ 0 \end{pmatrix} \Big(e^{-t/2} - e^{-t/3}\Big),$$

see [10,14] for more details. The accuracy of the numerical solution obtained in [10] was of the order $10^{-3} - 10^{-4}$ for already moderate values of n. Our new numerical results show that the approximation accuracy of 10^{-4} of the Tucker decomposition is not sufficient in order to keep the accuracy of the whole scheme or even leads to divergence. Thus, we choose two different

Table 4.1: Memory requirements

Grid	MB(f), $\varepsilon = 0$	MB(f), $\varepsilon = 10^{-6}$	MB(f), $\varepsilon = 10^{-8}$	MB(T)
$16 \times 16 \times 16$	0.02	0.001	0.002	0.3
$32 \times 32 \times 32$	0.19	0.003	0.003	4.5
$64 \times 64 \times 64$	1.50	0.005	0.005	72.0
$128 \times 128 \times 128$	12.00	0.009	0.012	1100.0

Table 4.2: Computational times

Grid	$\varepsilon = 0$		$\varepsilon = 10^{-6}$		$\varepsilon = 10^{-8}$	
$16 \times 16 \times 16$	$0.2s$	$8.0s$	$0.4s$	$19.0s$	$0.5s$	$22.0s$
$32 \times 32 \times 32$	$11.0s$	$12.0m$	$0.8s$	$70.0s$	$1.0s$	$81.0s$
$64 \times 64 \times 64$	$7.0m$	$15.0h$	$3.3s$	$7.0m$	$3.4s$	$7.0m$
$128 \times 128 \times 128$	$6.0h$	$--$	$49.0s$	$3.6h$	$49.0s$	$3.6h$

values for the approximation accuracy of $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-8}$ leading to stable numerical results.

The most interesting numerical experiment is the time evolution of the numerical rank for a given accuracy ε of approximation. First, we solve the problem without three way approximation ($\varepsilon = 0$) and compute the three way rank of the functions $F(t_i)$ for all time steps by the use of the Parallel Decomposition Algorithm, see [9]. The results are shown in Figure 4.1. The left plot in this figure corresponds to $\varepsilon = 10^{-6}$ while the right plot to $\varepsilon = 10^{-8}$.

Afterwards, we solve the problem by the use of the Tucker approximation as described. The time evolution of the Tucker rank is shown in Figure 4.2 for $\varepsilon = 10^{-6}$ (left plot) and for $\varepsilon = 10^{-8}$ (right plot).

The memory requirements are summarised in Table 4.1. The number of discretisation points is listed in the first column of this table. The second column contains the memory requirement in MByte for the function f without approximation, while in the third and in the fourth columns of Table 4.1, the memory requirements for f_ε are given. The last column of this table shows the memory requirements in MByte for the kernel T, see (1.14) on the grid. It is clear to see that the memory is completely dominated by the kernel. In this sense, the drastic reduction of the memory requirements for the function f while using three way approximation does not lead to any significant reduction of the whole memory requirements.

However, the situation changes if we consider the computational times presented in Table 4.2. Here we can see drastic reduction of the computational time due to the Tucker approximation of the distribution function f and of collision kernel Q.

5. Conclusions

In the present paper we develop a new deterministic numerical method for the Boltzmann equation. This method uses a special form of the of the Boltzmann collision operator introduced in [10] which is available for all cut-off kernels and involves Fourier transforms. The discretisation uses a uniform grid in the velocity space, so the algorithm of Fast Fourier Transform can be applied to increase the efficiency of the method. The main new idea is an approximation of the discrete distribution function with the help of the three way or Tucker decomposition. This leads to a drastic reduction of the computational time of the algorithm. The memory requirements in the current version of the algorithm keep practically the same due to the kernel T.

The numerical results are obtained for analytically known curves for the time relaxation of the moments for the Maxwell pseudo-molecules.

The efficiency of the new algorithm is so high that soon we will be able to solve spatially non-homogeneous problems, at least for low Mach number problems.

References

[1] M. Bebendorf, Approximation of boundary element matrices, *Numer. Math.*, **86**:4 (2000), 565-589.

[2] M. Bebendorf and S. Rjasanow, Adaptive low-rank approximation of collocation matrices, *Computing*, **70** (2003), 1-24.

[3] G.A. Bird, Monte Carlo simulation in an engineering context, *Progr. Astro. Aero*, **74** (1981), 239-255.

[4] A.V. Bobylev, Fourier transform method in the theory of the Boltzmann equation for Maxwell molecules, *Doklady Akad. Nauk SSSR*, **225** (1975), 1041-1044.

[5] C. Eckard and G. Young, A principal axis transformation for non-hermitian matrices, *Bull. Amer. Math. Soc.*, **45** (1939), 118-121.

[6] F. Filbet, C. Mouhot and L. Pareschi, Solving the Boltzmann equation in $N \log_2 N$, *SIAM J. Sci. Comput.*, **28**:3 (2006), 1029-1053.

[7] R. Harshman, Foundations of PARAFAC procedure: models and conditions for an "exploratory" multi-mode analysis, *UCLA Working Papers in Phonetics*, **16** (1970), 1-84.

[8] I. Ibragimov, Application of the three-way decomposition for matrix compression, *J. Num. Lin. Alg.*, **9**:6 (2002), 551-565.

[9] I. Ibragimov, The Three-Way Decomposition, Preprint 66, University of Saarland, 2002.

[10] I. Ibragimov and S. Rjasanow, Numerical solution of the Boltzmann equation on the uniform grid, *Computing*, **69**:2 (2002), 163-186.

[11] B.N. Khoromskij, Structured data-sparse approximation to high order tensors arising from the deterministic Boltzmann equation, *Math. Comput.*, **76** (2007), 1291-1315.

[12] M. Krook and T.T. Wu, Exact solutions of Boltzmann equation, *Phys. Fluids*, **20**:10 (1977).

[13] L. Mirsky, Symmetric gage functions and unitarily invariant norms, *Quarterly Journal of Mathematics*, **11** (1960), 50-59.

[14] S. Rjasanow and W. Wagner, Stochastic Numerics for the Boltzmann Equation. Number 37 in Springer Series in Computational Mathematics, Springer, Berlin-Heidelberg-NewYork, 2005.

[15] E. Schmidt, Zur Theorie der linearen und nichtlieaeren Integralgleichungen. I Teil. *Math. Ann.*, **63** (1907), 433-476.

[16] N.P. Sokolov, Spatial Matrices and Their Applications (Russian), Gosudarstv. Izdat. Fiz-. Mat. Lit., Moscow, 1960.

[17] S. Tiwari, Coupling of the Boltzmann and Euler equations with automatic domain decomposition, *J. Comput. Phys.*, **144**:2 (1998), 710–726.

[18] L.R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika*, **31** (1966), 279-311.