A FAST ALGORITHM FOR THE ENERGY SPACE BOSON BOLTZMANN COLLISION OPERATOR

JINGWEI HU AND LEXING YING

ABSTRACT. This paper introduces a fast algorithm for the energy space boson Boltzmann collision operator. Compared to the direct $O(N^3)$ calculation and the previous $O(N^2 \log N)$ method [Markowich and Pareschi, 2005], the new algorithm runs in complexity $O(N \log^2 N)$, which is optimal up to a logarithmic factor (N is the number of grid points in energy space). The basic idea is to partition the 3-D summation domain recursively into elementary shapes so that the summation within each shape becomes a special double convolution that can be computed efficiently by the fast Fourier transform. Numerical examples are presented to illustrate the efficiency and accuracy of the proposed algorithm.

1. INTRODUCTION

The quantum Boltzmann equation or Nordheim-Uehling-Uhlenbeck equation [16, 23], describes the nonequilibrium dynamics of quantum gases. These are the low density gases consisting of bosons or fermions which, when cooled to certain temperatures, evolve and interact in ways that reveal the quantum mechanical nature of the particles. Let $F(t, \mathbf{x}, \mathbf{v})$ be the phase space distribution function of time t, position $\mathbf{x}$, and particle velocity $\mathbf{v}$, then the equation reads (assume a unit mass for all particles):

$$(1.1) \qquad \frac{\partial F}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} F - \nabla_{\mathbf{x}} V(\mathbf{x}) \cdot \nabla_{\mathbf{v}} F = \tilde{Q}(F)(\mathbf{v}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^3, \ \mathbf{v} \in \mathbb{R}^3,$$

where $V(\mathbf{x})$ is the external potential. $\tilde{Q}(F)$ is the collision operator modeling the interaction of bosons (although most of the discussion also applies to the Fermi gas, in this paper we will only focus on the Bose gas since it covers the interesting phenomenon of the Bose-Einstein condensation (BEC)):

$$\tilde{Q}(F)(\mathbf{v}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} W(\mathbf{v}, \mathbf{v}_*, \mathbf{v}', \mathbf{v}'_*) \delta\left(\mathbf{v} + \mathbf{v}_* - \mathbf{v}' - \mathbf{v}'_*\right) \delta\left(\frac{\mathbf{v}^2}{2} + \frac{\mathbf{v}_*^2}{2} - \frac{\mathbf{v}'^2}{2} - \frac{\mathbf{v}'^2_*}{2}\right)$$

$$(1.2) \qquad \cdot \left[F' F'_* (1 + F)(1 + F_*) - F F_* (1 + F')(1 + F'_*)\right] d\mathbf{v}_* d\mathbf{v}' d\mathbf{v}'_*.$$

Here $(\mathbf{v}, \mathbf{v}_*)$ and $(\mathbf{v}', \mathbf{v}'_*)$ are the velocity pairs before and after collision. F, F_*, F', and F'_* are shorthand notations for $F(t, \mathbf{x}, \mathbf{v})$, $F(t, \mathbf{x}, \mathbf{v}_*)$, $F(t, \mathbf{x}, \mathbf{v}')$, and $F(t, \mathbf{x}, \mathbf{v}'_*)$

respectively. The collision kernel W is a nonnegative function determined by the underlying interaction law.

Under the spatially homogeneous and velocity isotropic assumptions, one can derive from (1.1) the following energy space boson Boltzmann equation [3, 11, 14, 20–22]:

$$\rho(\varepsilon)\frac{\partial f}{\partial t} = Q(f)(\varepsilon), \quad \varepsilon \geq 0, \tag{1.3}$$

with $f(t, \varepsilon)$ being the distribution function of time t and particle energy ε. $\rho(\varepsilon)$ is the density of states:

$$\rho(\varepsilon) := \int_{\mathbb{R}^3} \delta\left(\varepsilon - \frac{\mathbf{v}^2}{2}\right) d\mathbf{v} = 4\pi\sqrt{2\varepsilon}. \tag{1.4}$$

$Q(f)$ is the boson Boltzmann collision operator:

$$Q(f)(\varepsilon) = \int_0^\infty \int_0^\infty \int_0^\infty w(\varepsilon, \varepsilon_*, \varepsilon', \varepsilon'_*)\delta(\varepsilon + \varepsilon_* - \varepsilon' - \varepsilon'_*)$$
$$\cdot [f'f'_*(1 + f)(1 + f_*) - ff_*(1 + f')(1 + f'_*)] \, d\varepsilon_* d\varepsilon' d\varepsilon'_*, \tag{1.5}$$

where $(\varepsilon, \varepsilon_*)$ and $(\varepsilon', \varepsilon'_*)$ are the particle energies before and after collision. f, f_*, f', and f'_* stand for $f(t, \varepsilon)$, $f(t, \varepsilon_*)$, $f(t, \varepsilon')$, and $f(t, \varepsilon'_*)$. For the simple hard sphere model, the collision kernel w is given by

$$w(\varepsilon, \varepsilon_*, \varepsilon', \varepsilon'_*) = \rho(\min(\varepsilon, \varepsilon_*, \varepsilon', \varepsilon'_*)). \tag{1.6}$$

A brief derivation of (1.3)–(1.6) and the basic properties of the energy space equation can be found in Appendix A.

Numerically solving the phase space quantum Boltzmann equation is challenging mainly due to the multidimensional structure of the collision integral (1.2). The traditional approach is the Monte Carlo simulation [7]. Over the past decade, the deterministic scheme, especially the spectral method [2, 6, 9, 10, 15, 17, 18], has drawn much attention for its high accuracy and low computational cost. However, all these works were developed for the classical Boltzmann equation. New difficulty arises when it comes to the quantum case — the collision operator is cubic instead of quadratic. In the spirit of [15], a fast spectral algorithm [8] was proposed for the full quantum Boltzmann operator (1.2).

Compared to the phase space description, the energy space equation is greatly simplified. Nevertheless, many interesting properties of the solution are retained and both analysis and numerics (none of the above fast algorithms apply) are nontrivial. For the theoretical work of equation (1.3) and related models, see, for instance, [1, 4, 5, 13, 22] and references therein. For numerical simulations, we refer to [3, 11, 14, 20, 21].

The goal of this paper is to design an efficient algorithm for the energy space boson Boltzmann collision operator (1.5). Our starting point is the following truncated version of $Q(f)$ used in [14]:

$$Q^R(f)(\varepsilon) = \int_0^R \int_0^R \int_0^R \rho(\min(\varepsilon, \varepsilon_*, \varepsilon', \varepsilon'_*))\delta(\varepsilon + \varepsilon_* - \varepsilon' - \varepsilon'_*)$$
$$\cdot [f'f'_*(1 + f)(1 + f_*) - ff_*(1 + f')(1 + f'_*)] \, d\varepsilon_* d\varepsilon' d\varepsilon'_*, \quad \varepsilon \in [0, R]. \tag{1.7}$$

How to choose the upper bound R will be made precise in the numerical examples. Here we only mention that in order to capture the physics R is usually not small.

We then introduce N uniform discrete points $\varepsilon_0 < \varepsilon_1 < \ldots < \varepsilon_{N-1}$ on $[0, R]$ with mesh size $\Delta \varepsilon = R/N$. Thus a consistent discretization of (1.7) is written as

$$Q_i^R = \Delta \varepsilon^2 \sum_{\substack{m,n,j=0 \\ m+n=i+j}}^{N-1} \rho(\varepsilon_{\min})[f_m f_n (1+f_i)(1+f_j) - f_i f_j (1+f_m)(1+f_n)]$$

$$(1.8) \qquad = \Delta \varepsilon^2 (1+f_i) \sum_{\substack{m,n,j=0 \\ m+n=i+j}}^{N-1} \rho(\varepsilon_{\min}) f_m f_n (1+f_j)$$

$$- \Delta \varepsilon^2 f_i \sum_{\substack{m,n,j=0 \\ m+n=i+j}}^{N-1} \rho(\varepsilon_{\min}) f_j (1+f_m)(1+f_n),$$

for $i = 0, \ldots, N-1$, and

$$(1.9) \qquad\qquad \varepsilon_{\min} := \min(\varepsilon_i, \varepsilon_j, \varepsilon_m, \varepsilon_n) = \varepsilon_{\min(m,n,i,j)}.$$

This is just a simple numerical quadrature rule that takes into account the approximation of the delta function. Depending on the application, one can either choose integer grid points (first order method), or half-integer grid points (second order method). It is not difficult to verify that the scheme (1.8) preserves the main physical features of the continuous problem: conservation of mass and energy, the entropy inequality, and the Bose-Einstein distribution as steady state. See [14] for more details.

Despite its simple form the efficient evaluation of (1.8) still presents a challenge. Clearly a direct calculation of Q_i^R (for all i) requires cubic complexity $O(N^3)$, which can be quite expensive for large N. Furthermore, it is well known that a singularity occurs at the origin when the BEC happens, thus a finer grid is necessary to maintain the resolution.

In [14] by exploiting the special form of (1.9), Markowich and Pareschi were able to reduce the above cost to $O(N^2 \log N)$ (all "log" in this paper refers to logarithm to base 2). Their approach is based on a 2-D domain decomposition that allows one to use the fast Fourier transform (FFT) to speed up the inner summation — a convolution.

In this work, we propose a faster algorithm for (1.8) that runs in only $O(N \log^2 N)$ steps, which is optimal up to a logarithmic factor. The main idea is to partition the 3-D summation domain recursively into elementary shapes such that the FFT can be applied to both inner and outer summations — a special double convolution.

The rest of the paper is organized as follows. In the next section we describe the fast algorithm in detail and analyze its complexity. Numerical results of computing the collision operator and solving the time-evolution equation are shown in Section 3. Finally, the concluding remarks are given in Section 4.

2. Fast algorithms for the boson Boltzmann collision operator

We first briefly review the previous $O(N^2 \log N)$ method in [14], since it provides a basis for constructing the new algorithm.

2.1. The previous $O(N^2 \log N)$ algorithm — 2-D domain decomposition.
The key observation behind the method [14] is: if one divides the grid $\{0, \ldots, N-1\}^2$ in the index mn-domain into four parts according to a fixed $j \in \{0, \ldots, N-1\}$:
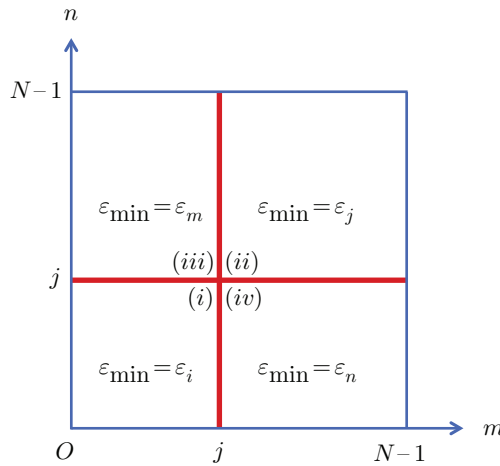
FIGURE 1. 2-D summation domain decomposition.

- region (i): $\{(m,n): 0 \le m \le j,\ 0 \le n \le j\}$;
- region (ii): $\{(m,n): j < m \le N-1,\ j < n \le N-1\}$;
- region (iii): $\{(m,n): 0 \le m \le j,\ j < n \le N-1\}$;
- region (iv): $\{(m,n): j < m \le N-1,\ 0 \le n \le j\}$,

(see Figure 1 for an illustration), then $\varepsilon_{\min}$ (1.9) takes a unique value in each region, i.e.,

$$\min(m,n,i,j) = \begin{cases} i & \text{in region } (i); \\ j & \text{in region } (ii); \\ m & \text{in region } (iii); \\ n & \text{in region } (iv). \end{cases}$$

Now the computation of (1.8) can be performed in each region separately. We write

$$Q_i^R = Q_i^{(i)} + Q_i^{(ii)} + Q_i^{(iii)} + Q_i^{(iv)},$$

where $Q_i^{(i)}$ denotes the summation of m, n in region (i) for each j, and so forth. Let us take region (iii) for example,

$$Q_i^{(iii)} = \Delta\varepsilon^2 (1+f_i) \sum_{j=0}^{N-1} \left(\sum_{\substack{0 \le m \le j, j < n \le N-1 \\ m+n=i+j}} \rho(\varepsilon_m) f_m f_n \right) (1+f_j)$$

$$- \Delta\varepsilon^2 f_i \sum_{j=0}^{N-1} \left(\sum_{\substack{0 \le m \le j, j < n \le N-1 \\ m+n=i+j}} \rho(\varepsilon_m)(1+f_m)(1+f_n) \right) f_j,$$

for $i = 0, \ldots, N-1$. For the gain term, if we treat $\rho(\varepsilon_m)f_m$ as a single function, then for each fixed j the inner summation is a convolution of functions ρf and f defined on truncated portions according to j. Similarly, the inner sum of the loss term is a convolution of truncated functions $\rho(1+f)$ and $(1+f)$. Either of them can be computed effectively by the FFT in $O(N \log N)$ operations, resulting a function

g with index $i + j$. Since g itself depends on j, the outer summation has to be carried out directly. Therefore, the total cost is $O(N^2 \log N + N^2) = O(N^2 \log N)$.

2.2. **The new $O(N \log^2 N)$ algorithm — 3-D domain decomposition.** We observe from (1.8) that the main computation task is of the following general form

$$u_i = \sum_{m+n=i+j} \rho(\varepsilon_{\min}) e_m g_n h_j, \quad \text{for} \ \ 0 \le i, j, m, n \le N - 1,$$

where $\{e_m\}$, $\{g_n\}$, and $\{h_j\}$ are sequences indexed by $\{0, \ldots, N - 1\}$. It turns out that in order to speed up the calculation, it is more convenient to consider all possible i that satisfy $m + n = i + j$ for some $0 \le m, n, j \le N - 1$ and then truncate the result to $0 \le i \le N - 1$. This allows us to consider the *3-D summation domain*

$$D := \{(m, n, j) : \ 0 \le m, n, j \le N - 1\}.$$

As pointed out earlier, the approach of [14] partitions the 2-D summation domain $\{(m, n) : \ 0 \le m, n \le N - 1\}$ into four regions for each fixed j. We instead partition the whole summation domain D into four regions — two *pyramids* and two *simplexes* (the 3-D counterparts of 2-D regions (i)–(iv), see Figure 2):

- region (I): $\{(m, n, j) : \ m \le j, \ n \le j, \ 0 \le m, n, j \le N - 1\}$;
- region (II): $\{(m, n, j) : \ j < m, \ j < n, \ 0 \le m, n, j \le N - 1\}$;
- region (III): $\{(m, n, j) : \ m \le j, \ j < n, \ 0 \le m, n, j \le N - 1\}$;
- region (IV): $\{(m, n, j) : \ j < m, \ n \le j, \ 0 \le m, n, j \le N - 1\}$,

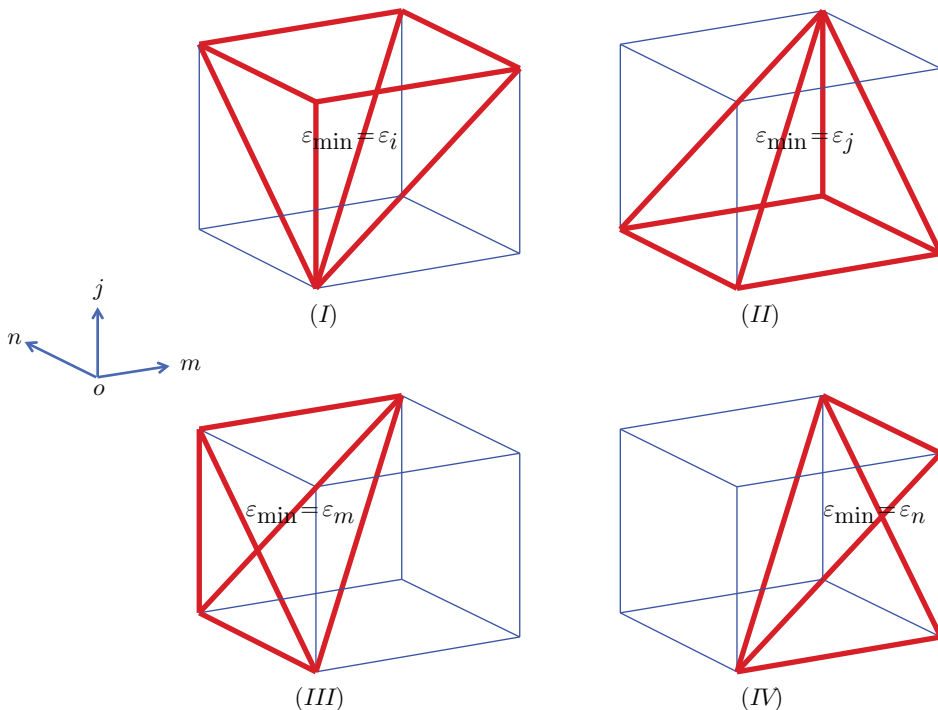and compute the contribution from each region one by one.

FIGURE 2. 3-D summation domain decomposition.

Within each 3-D region, $\min(m, n, i, j)$ takes a fixed value

$$
\min(m,n,i,j) = \begin{cases}
i & \text{in region } (I); \\
j & \text{in region } (II); \\
m & \text{in region } (III); \\
n & \text{in region } (IV),
\end{cases}
$$

and it is possible to combine the troublesome term $\rho(\varepsilon_{\min})$ with one of the sequences $\{e_m\}$, $\{g_n\}$, $\{h_j\}$ or treat it as an extra multiplication. Hence, we only need consider the following simple problem of the form

$$
u_i = \sum_{m+n=i+j} e_m g_n h_j
$$

with the summation taken over a single region and $\{e_m\}$, $\{g_n\}$, and $\{h_j\}$ are again defined on $\{0, \ldots, N-1\}$ (but with slightly different content).

Before discussing the algorithm in detail, we note that the next three summations will be encountered often later on. Given three sequences $\{e_m\}$, $\{g_n\}$, and $\{h_j\}$ defined on three independent consecutive integer intervals $\mathcal{M}$, $\mathcal{N}$, and $\mathcal{J}$ respectively, let

$$
(2.1) \qquad p_i = \sum_{\substack{m \in \mathcal{M}, n \in \mathcal{N} \\ i=m+n}} e_m g_n, \quad i \in \mathcal{M} + \mathcal{N},
$$

$$
(2.2) \qquad q_i = \sum_{\substack{m \in \mathcal{M}, n \in \mathcal{N} \\ i=m-n}} e_m g_n, \quad i \in \mathcal{M} - \mathcal{N},
$$

$$
(2.3) \qquad r_i = \sum_{\substack{m \in \mathcal{M}, n \in \mathcal{N}, j \in \mathcal{J} \\ i=m+n-j}} e_m g_n h_j, \quad i \in \mathcal{M} + \mathcal{N} - \mathcal{J},
$$

where $\mathcal{M} + \mathcal{N} = \{m + n : m \in \mathcal{M}, n \in \mathcal{N}\}$, $\mathcal{M} - \mathcal{N} = \{m - n : m \in \mathcal{M}, n \in \mathcal{N}\}$, and $\mathcal{M} + \mathcal{N} - \mathcal{J} = \{m + n - j : m \in \mathcal{M}, n \in \mathcal{N}, j \in \mathcal{J}\}$. We claim that the sequences $\{p_i\}$, $\{q_i\}$, and $\{r_i\}$ can all be evaluated by fast algorithms. In fact:

- (2.1) is nothing but a simple convolution of $\{e_m\}$ and $\{g_n\}$, which can be calculated by the FFT.
- To compute (2.2), we first reverse the ordering of $\{g_n\}$, i.e., introduce a new sequence $\{\tilde{g}_n = g_{-n}\}$ with $n \in -\mathcal{N}$, and rewrite q_i as

$$
q_i = \sum_{\substack{m \in \mathcal{M}, n \in -\mathcal{N} \\ i=m+n}} e_m g_{-n} = \sum_{\substack{m \in \mathcal{M}, n \in -\mathcal{N} \\ i=m+n}} e_m \tilde{g}_n, \quad i \in \mathcal{M} - \mathcal{N}.
$$

  This again falls into the form (2.1).
- For (2.3), we introduce the sequence $\{\tilde{h}_j = h_{-j}\}$ with $j \in -\mathcal{J}$, then

$$
r_i = \sum_{\substack{m \in \mathcal{M}, n \in \mathcal{N}, j \in -\mathcal{J} \\ i=m+n+j}} e_m g_n \tilde{h}_j, \quad i \in \mathcal{M} + \mathcal{N} - \mathcal{J}.
$$

  This is a double convolution and can also be sped up by the FFT.

Clearly, for any of the above subproblems the computational cost is bounded by

$$
\max(|\mathcal{M}|, |\mathcal{N}|, |\mathcal{J}|) \log(\max(|\mathcal{M}|, |\mathcal{N}|, |\mathcal{J}|)),
$$

where $|\cdot|$ denotes the cardinality of a set.

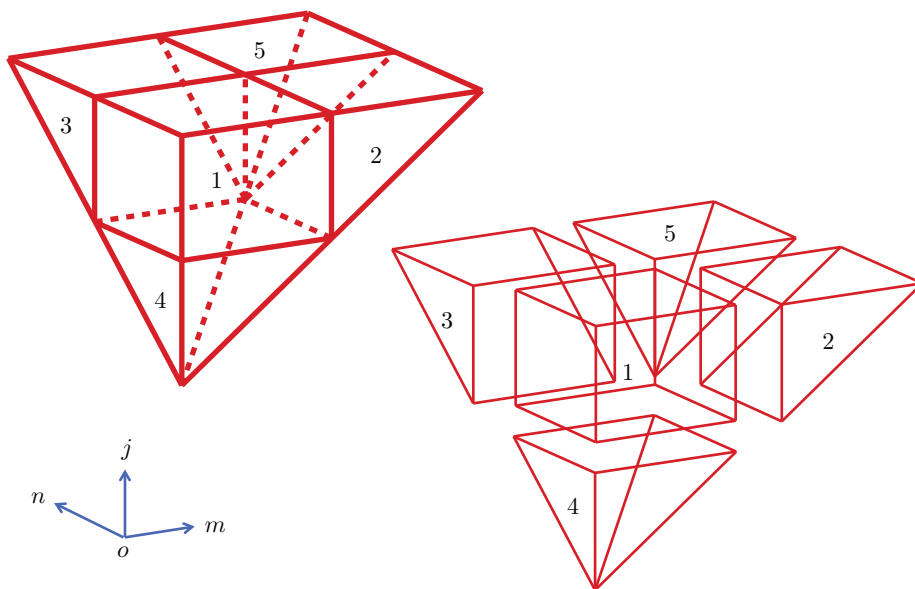We are ready to describe the final algorithm.

FIGURE 3. Decomposition of pyramid (I).

Computation of region (I)**.** For region (I), the relevant sum is of the form

$$u_i^I = \sum_{\substack{m \leq j, n \leq j, 0 \leq m,n,j \leq N-1 \\ m+n=i+j}} e_m g_n h_j.$$

In order to compute it efficiently, we partition this pyramid into five parts as shown in Figure 3.

- Part 1 is a cube:
$$\left\{ (m,n,j) : \ 0 \leq m, n \leq \frac{N}{2} - 1, \ \frac{N}{2} \leq j \leq N-1 \right\}.$$

- Part 2 is a wedge:
$$\left\{ (m,n,j) : \ 0 \leq n \leq \frac{N}{2} - 1, \ \frac{N}{2} \leq m \leq j \leq N-1 \right\}.$$

- Part 3 is another wedge:
$$\left\{ (m,n,j) : \ 0 \leq m \leq \frac{N}{2} - 1, \ \frac{N}{2} \leq n \leq j \leq N-1 \right\}.$$

- Part 4 is a pyramid:
$$\left\{ (m,n,j) : \ m \leq j, \ n \leq j, \ 0 \leq m,n,j \leq \frac{N}{2} - 1 \right\}.$$

- Part 5 is another pyramid:
$$\left\{ (m,n,j) : \ m \leq j, \ n \leq j, \ \frac{N}{2} \leq m,n,j \leq N-1 \right\}.$$

We now write $u_i^I = u_i^{I,1} + u_i^{I,2} + u_i^{I,3} + u_i^{I,4} + u_i^{I,5}$, where each $u_i^{I,k}$ stands for the summation related to the k-th part.

- For part 1 (cube), the relevant sum is

$$
u_i^{I,1} = \sum_{\substack{0 \le m,n \le \frac{N}{2}-1, \frac{N}{2} \le j \le N-1 \\ m+n=i+j}} e_m g_n h_j.
$$

  This double convolution can be computed as for (2.3) and the cost is $O\left(\frac{N}{2}\log\frac{N}{2}\right)$.
- For part 2 (wedge), the relevant sum is

$$
u_i^{I,2} = \sum_{n=0}^{\frac{N}{2}-1} \underbrace{\left(\sum_{\substack{\frac{N}{2} \le m \le j \le N-1 \\ m-j=i-n}} e_m h_j \right)}_{t_{i-n}} g_n.
$$

  The inner sum corresponds to the triangular side of the wedge. At first sight, this summation is difficult to compute due to the existence of the constraint $m \le j$. However, notice that $m \le j$ is equivalent to $i - n \le 0$ since $m - j = i - n$. Therefore, all one needs is to compute the inner sum as for (2.2) and set the resulting vector t_{i-n} to zero at indices greater than 0. The outer summation is computed as for (2.1). Thus the total cost for this part is also $O\left(\frac{N}{2}\log\frac{N}{2}\right)$.
- For part 3 (wedge), the relevant sum is

$$
u_i^{I,3} = \sum_{m=0}^{\frac{N}{2}-1} \left(\sum_{\substack{\frac{N}{2} \le n \le j \le N-1 \\ n-j=i-m}} g_n h_j \right) e_m.
$$

  This is computed exactly the same as above for part 2.
- Parts 4 and 5 (pyramids) have the same shape as region (I), but only half its size. Hence it is natural to perform the computation of these two parts using recursion.

Let us estimate the total complexity of region (I). Assume the computational cost is $T(N)$, where N stands for the size of the summation domain in each dimension. We have

$$
T(N) = \underbrace{O\left(\frac{N}{2}\log\frac{N}{2}\right)}_{\text{one cube}} + \underbrace{2O\left(\frac{N}{2}\log\frac{N}{2}\right)}_{\text{two wedges}} + \underbrace{2T\left(\frac{N}{2}\right)}_{\text{two pyramids}} = O(N\log N) + 2T\left(\frac{N}{2}\right)
$$

$$
= \ldots = O\left(\underbrace{N\log N + N\log\frac{N}{2} + N\log\frac{N}{4} + \ldots}_{\log N \text{ terms}}\right) + NT(1)
$$

$$
= O(N\log^2 N) + O(N) = O(N\log^2 N).
$$

In the actual implementation, we terminate the recursion whenever the size of the pyramid is smaller than a certain threshold (e.g. 64 or 128), since then the quadratic-complexity algorithm [14] becomes competitive.

Computation of region (II)**.** As the shape of region (II) is similar to region (I), it can be decomposed as a disjoint union of five parts: one cube, two wedges, and two self-similar pyramids of half the original size (see Figure 4). Therefore, the approach discussed for region (I) works with minor modifications. As a result, the computational cost of region (II) is also $O(N \log^2 N)$.
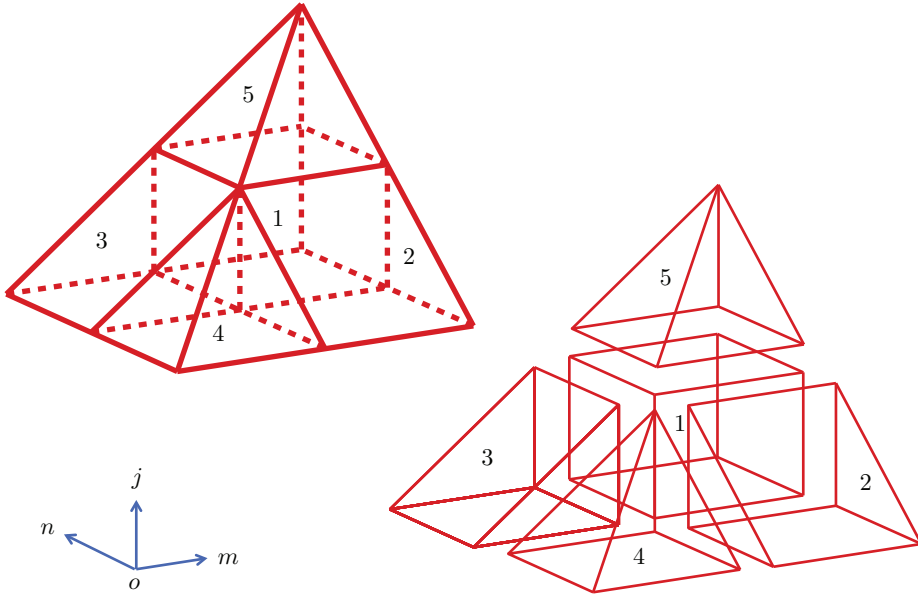

FIGURE 4. Decomposition of pyramid (II).

Computation of region (III)**.** For region (III), the relevant sum is of the form

$$u_i^{III} = \sum_{\substack{m \le j, j < n, 0 \le m,n,j \le N-1 \\ m+n=i+j}} e_m g_n h_j.$$

We partition this simplex into four parts as shown in Figure 5.

- Part 1 is a wedge:
$$\left\{ (m,n,j): \ \frac{N}{2} \le n \le N-1, \ 0 \le m \le j \le \frac{N}{2}-1 \right\}.$$

- Part 2 is another wedge:
$$\left\{ (m,n,j): \ 0 \le m \le \frac{N}{2}-1, \ \frac{N}{2} \le j < n \le N-1 \right\}.$$

- Part 3 is a simplex:
$$\left\{ (m,n,j): \ m \le j, \ j < n, \ 0 \le m,n,j \le \frac{N}{2}-1 \right\}.$$

- Part 4 is another simplex:
$$\left\{ (m,n,j): \ m \le j, \ j < n, \ \frac{N}{2} \le m,n,j \le N-1 \right\}.$$
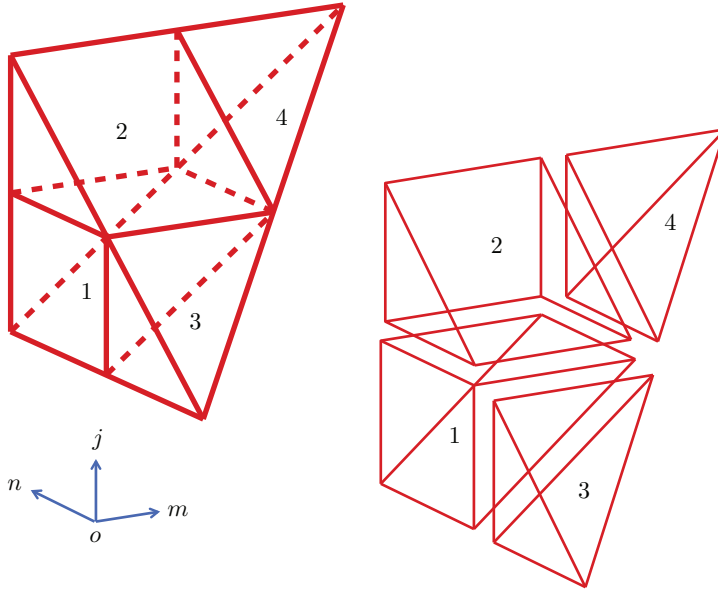
FIGURE 5. Decomposition of simplex (III).

We write $u_i^{III} = u_i^{III,1} + u_i^{III,2} + u_i^{III,3} + u_i^{III,4}$ accordingly, so that each $u_i^{III,k}$ stands for the summation associated with the k-th part.

- In part 1 (wedge), the relevant sum is

$$u_i^{III,1} = \sum_{n=\frac{N}{2}}^{N-1} \left(\sum_{\substack{0 \le m \le j \le \frac{N}{2}-1 \\ m-j=i-n}} e_m h_j \right) g_n.$$

  Similarly to part 2 of region (I), we compute the inner sum as for (2.2) and set the resultant vector to zero at indices greater than 0. The outer sum is computed as for (2.1). Thus the total cost of this part is $O\left(\frac{N}{2}\log\frac{N}{2}\right)$.
- In part 2 (wedge), the relevant sum is

$$u_i^{III,2} = \sum_{m=0}^{\frac{N}{2}-1} \left(\sum_{\substack{\frac{N}{2} \le j < n \le N-1 \\ n-j=i-m}} g_n h_j \right) e_m.$$

  This one is computed almost the same as above for part 1, with the only exception that we set the result of the inner sum to zero at indices less than or equal to 0 (since now $i - m = n - j > 0$).
- Parts 3 and 4 (simplexes) have the same shape as region (III) but only half its size, so the computation associated with them can be done using recursion again.

Assume the computational cost for region (III) is $T(N)$, where N is the size of the summation domain in each dimension. Then

$$T(N) = \underbrace{2O\left(\frac{N}{2}\log\frac{N}{2}\right)}_{\text{two wedges}} + \underbrace{2T\left(\frac{N}{2}\right)}_{\text{two simplexes}},$$

which again yields an algorithm of $O(N\log^2 N)$.

Computation of region (IV)**.** As the shape of region (IV) is similar to that of region (III), it can be decomposed as a disjoint union of four parts: two wedges, and two self-similar simplexes of half the original size (see Figure 6). Therefore, the approach discussed for region (III) works with minor modifications. As a result, the cost of region (IV) is also $O(N\log^2 N)$.
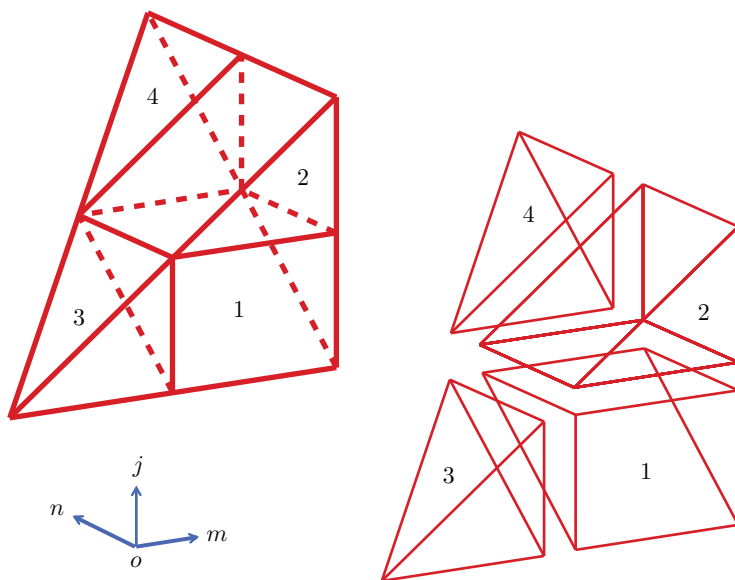

FIGURE 6. Decomposition of simplex (IV).

3. NUMERICAL RESULTS

In this section we provide several numerical examples to demonstrate the efficiency and accuracy of the new algorithm. We first test its performance on the collision operator, and then use it to solve the time-evolution equation. In all the examples, the grid points are chosen as $\varepsilon_0 = \Delta\varepsilon/2$, $\varepsilon_1 = 3\Delta\varepsilon/2$, ..., $\varepsilon_{N-1} = R - \Delta\varepsilon/2$ (second-order quadrature rule).

3.1. **Computing the collision operator.** We first examine the proposed algorithm by simply applying it to the distribution function

$$f(\varepsilon) = 3\exp\left(-(\varepsilon - 10)^2\right), \quad \varepsilon \in [0, R], \quad R = 30.$$

Remark 3.1. $R = 30$ is chosen such that $R \geq 2R_0$, where supp$(f) \approx [0, R_0]$. Indeed, if $\varepsilon > R$, then $f(\varepsilon) = 0$; also by conservation of energy $\varepsilon' + \varepsilon'_* = \varepsilon + \varepsilon_* \geq \varepsilon > R \geq 2R_0$, so one of ε', ε'_* must be bigger than R_0, i.e., $f(\varepsilon') = 0$ or $f(\varepsilon'_*) = 0$; either way $Q^R(f)(\varepsilon) \equiv 0$ (see Figure 7).
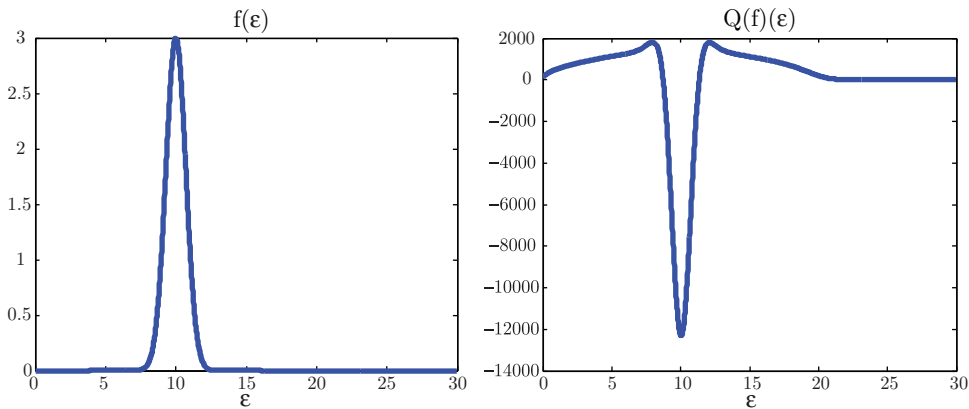

FIGURE 7. Left: the distribution function $f(\varepsilon)$. Right: the numerical $Q^R(f)(\varepsilon)$ computed by the new algorithm. $N = 2048$.

The results for different N are reported in Table 1. For comparison, the method in [14] is referred to as *quadratic algorithm*, and our method as *linear algorithm*. Column 5 confirms the linear complexity of the new algorithm. The speedup factor over the quadratic algorithm is about 24 for $N = 2048$.

TABLE 1. The average running time and the relative error between the quadratic algorithm in [14] and our linear algorithm.

N	quadratic algorithm T_q	$\frac{T_q(N)}{T_q(N/2)}$	linear algorithm T_l	$\frac{T_l(N)}{T_l(N/2)}$	$\|Q_l - Q_q\|_2 / \|Q_q\|_2$
128	0.386s	–	0.278s	–	3.2620e-16
256	1.450s	3.76	0.510s	1.83	8.2386e-16
512	5.579s	3.85	0.926s	1.82	1.2436e-15
1024	22.334s	4.00	1.816s	1.96	1.9140e-15
2048	88.980s	3.98	3.696s	2.04	2.0821e-15

3.2. **Solving the boson Boltzmann equation.** We now solve the equation (1.3) using the new fast algorithm. The time derivative is approximated by an explicit second-order Runge-Kutta method. We will only consider the Bose gas in the nondegenerate regime ($z < 1$). Modeling the degenerate Bose gas ($z > 1$) is a very complicated issue due the singularity of the distribution function (A.7) at the origin and is beyond the scope of this paper.

Suppose the initial condition is given by

$$f_0(\varepsilon) = e^{-(\varepsilon - 10)^2 / 10}, \quad \varepsilon \in [0, R], \quad R = 120.$$

The corresponding equilibrium takes the form (A.5) with $z \approx 0.6336$, $\beta \approx 0.1236$.

Remark 3.2. The reason to choose such a large R is because the final equilibrium $\mathcal{M}_{z,\beta}(\varepsilon)$ has a larger compact support than $f_0(\varepsilon)$. Moreover, the functions $\rho(\varepsilon)\mathcal{M}_{z,\beta}(\varepsilon)$ and $\rho(\varepsilon)\varepsilon\mathcal{M}_{z,\beta}(\varepsilon)$ (the integrands in the mass and energy (A.2), (A.3)) spread even wider (see Figure 8). In order to capture the real physics, R has to be large enough to include all mass and energy.
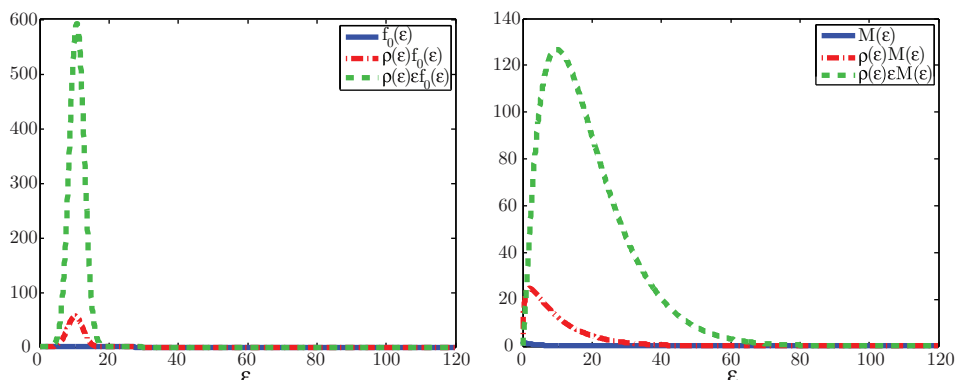

FIGURE 8. Left: the initial distribution $f_0(\varepsilon)$, $\rho(\varepsilon)f_0(\varepsilon)$, and $\rho(\varepsilon)\varepsilon f_0(\varepsilon)$. Right: the exact equilibrium $\mathcal{M}_{z,\beta}(\varepsilon)$, $\rho(\varepsilon)\mathcal{M}_{z,\beta}(\varepsilon)$, and $\rho(\varepsilon)\varepsilon\mathcal{M}_{z,\beta}(\varepsilon)$.

Figure 9 shows the distribution function $f(t,\varepsilon)$ at different times together with the exact equilibrium $\mathcal{M}_{z,\beta}(\varepsilon)$. The conservation of mass and energy, and the entropy growth can be observed from Figure 10. Here the number of grid points is $N = 2048$. The time step size is $\Delta t = 0.005$.

At certain time t, we expect the numerical solution $f(t,\varepsilon)$ converges to the exact equilibrium $\mathcal{M}_{z,\beta}(\varepsilon)$ as N increases. This can be seen from Table 2.

TABLE 2. The relative error between f and $\mathcal{M}_{z,\beta}$ at fixed time $t = 0.5$ ($\Delta t = 0.005$) and the cubic spline extrapolated value of $f(0)$ (Ref: the exact value $\mathcal{M}_{z,\beta}(0) = 1.7294$).

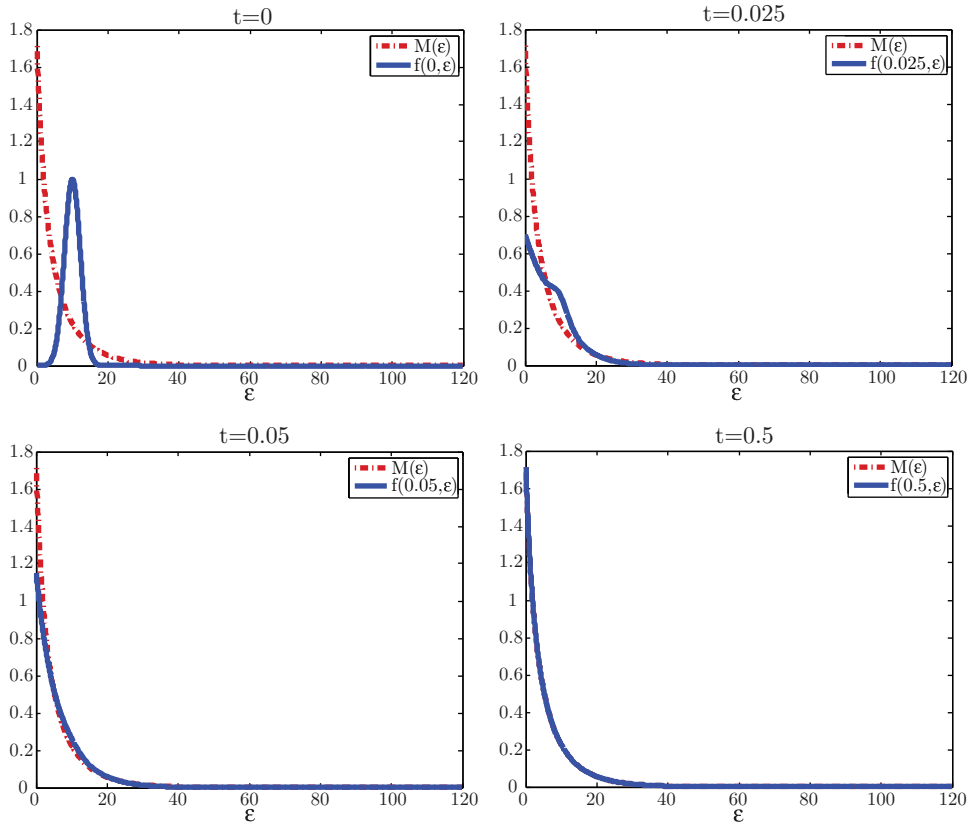| N	$\|f - \mathcal{M}_{z,\beta}\|_2/\|\mathcal{M}_{z,\beta}\|_2$	extrapolated value of $f(0)$
128	0.0121	1.6894
256	0.0047	1.7168
512	0.0017	1.7252
1024	0.0006	1.7279
2048	0.0002	1.7289

FIGURE 9. The numerical solution f at $t = 0$, 0.025, 0.05, and 0.5, and the exact equilibrium $\mathcal{M}_{z,\beta}$. $N = 2048$. $\Delta t = 0.005$.
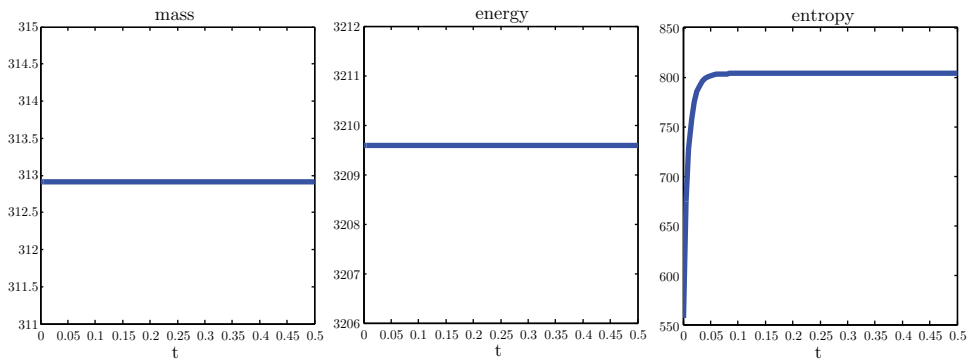

FIGURE 10. The time evolution of mass, energy, and entropy. $N = 2048$. $\Delta t = 0.005$.

4. CONCLUSIONS AND FUTURE WORK

A fast linear (optimal up to a logarithmic factor) algorithm is constructed for the energy space boson Boltzmann collision operator. The idea is to decompose the 3-D summation domain into elementary shapes: cubes, wedges, and self-similar pyramids and simplexes such that the collision kernel acquires a single value within each region. The summations in the cubes and wedges are double convolutions that can be evaluated efficiently using the FFT, while the self-similar parts are treated recursively, yielding an algorithm of $O(N \log^2 N)$. Numerical results further confirm the linear complexity of the algorithm and demonstrate its robustness in solving the time-evolution equation.

So far we have not touched the degenerate Bose gas yet. It seems that a logarithmic grid is more suitable to describe the condensation at the origin. The design of fast algorithms in this framework will be investigated in the future.

APPENDIX A. THE ENERGY SPACE BOSON BOLTZMANN EQUATION

In this appendix, we give a formal derivation of the energy space boson Boltzmann equation from its original form in the phase space, and summarize its basic properties. Part of the arguments can be found in one place or another such as [4, 12, 14, 19, 21, 22].

We will start from the following spatially homogeneous equation (a similar derivation carries through for the spatially inhomogeneous case and when there is an external potential):

$$(A.1) \qquad \frac{\partial F}{\partial t} = \tilde{Q}(F)(\mathbf{v}), \quad \mathbf{v} \in \mathbb{R}^3,$$

where the distribution F only depends on time t and velocity $\mathbf{v}$. The collision operator $\tilde{Q}(F)$ is given by (1.2), and we consider the simple case of $W \equiv 1$ which corresponds to the hard sphere model.

We now make the assumption that $F(t, \mathbf{v})$ is isotropic in $\mathbf{v}$, i.e., one can define an energy space distribution $f(t, \varepsilon)$ such that $f(t, \varepsilon) = F(t, \mathbf{v})$ with $\varepsilon = \mathbf{v}^2/2$. By a change of variables, the total mass of particles (per unit volume) is

$$(A.2) \qquad M := \int_{\mathbb{R}^3} F(t, \mathbf{v}) \, d\mathbf{v} = \int_0^\infty 4\pi \sqrt{2\varepsilon} \, f(t, \varepsilon) \, d\varepsilon,$$

and the total energy is

$$(A.3) \qquad E := \int_{\mathbb{R}^3} \frac{\mathbf{v}^2}{2} F(t, \mathbf{v}) \, d\mathbf{v} = \int_0^\infty 4\pi \sqrt{2\varepsilon} \, \varepsilon f(t, \varepsilon) \, d\varepsilon.$$

Therefore, it is convenient to introduce the function $\rho(\varepsilon)$ as in (1.4).

Similarly the collision operator (1.2) can be transformed into

$$\tilde{Q}(F) = \frac{4\pi^2}{\rho(\varepsilon)} \int_0^\infty \int_0^\infty \int_0^\infty \rho(\min(\varepsilon, \varepsilon_*, \varepsilon', \varepsilon'_*)) \delta\left(\varepsilon + \varepsilon_* - \varepsilon' - \varepsilon'_*\right)$$

$$(A.4) \qquad \cdot \left[f' f'_*(1+f)(1+f_*) - f f_*(1+f')(1+f'_*) \right] d\varepsilon_* d\varepsilon' d\varepsilon'_*,$$

where the equality

$$\int_{\mathbb{S}^2} \int_{\mathbb{S}^2} \int_{\mathbb{S}^2} \delta\left(r\sigma + r_*\sigma_* - r'\sigma' - r'_*\sigma'_*\right) d\sigma_* d\sigma' d\sigma'_* = \frac{4\pi^2}{rr_* r' r'_*} \min(r, r_*, r', r'_*),$$

($\mathbf{v} = r\sigma$, $r = |\mathbf{v}|$ and σ is the surface element) and the definition of $\rho(\varepsilon)$ have been used.

Substituting (A.4) back into (A.1), we get the energy space boson Boltzmann equation (1.3) presented at the beginning of the paper (differ by a constant).

A.1. **Properties.** The quantum collision operator (1.5) has 1 and ε as collision invariants:

$$\int_0^\infty Q(f)\,d\varepsilon = \int_0^\infty \varepsilon Q(f)\,d\varepsilon = 0,$$

so the total mass and energy are conserved:

$$M = \int_0^\infty \rho(\varepsilon)\,f(t,\varepsilon)\,d\varepsilon \equiv \int_0^\infty \rho(\varepsilon)\,f_0(\varepsilon)\,d\varepsilon,$$

$$E = \int_0^\infty \rho(\varepsilon)\,\varepsilon f(t,\varepsilon)\,d\varepsilon \equiv \int_0^\infty \rho(\varepsilon)\,\varepsilon f_0(\varepsilon)\,d\varepsilon,$$

where $f_0(\varepsilon)$ is the initial condition.

$Q(f)$ also satisfies Boltzmann's H-theorem:

$$\frac{d}{dt}S(\varepsilon) = \int_0^\infty Q(f)[\ln(1+f) - \ln f]\,d\varepsilon \geq 0,$$

where

$$S(\varepsilon) := \int_0^\infty \rho(\varepsilon)[(1+f)\ln(1+f) - f\ln f]\,d\varepsilon$$

is the entropy.

The entropy is always nondecreasing, and reaches its maximum if and only if f attains the equilibrium (the Bose-Einstein distribution):

$$(A.5) \qquad \mathcal{M}_{(z,\beta)}(\varepsilon) = \frac{1}{z^{-1}e^{\beta\varepsilon} - 1}.$$

Here z is the fugacity, β is the inverse temperature ($\beta = 1/T$; $z = e^{\mu/T}$, μ is the chemical potential). Given $\mathcal{M}_{(z,\beta)}$, the corresponding mass and energy can be expressed as

$$\begin{cases} M = \left(\dfrac{2\pi}{\beta}\right)^{\frac{3}{2}} G_{\frac{3}{2}}(z), & z \leq 1, \\[3mm] E = \dfrac{3}{2\beta}\left(\dfrac{2\pi}{\beta}\right)^{\frac{3}{2}} G_{\frac{5}{2}}(z), & z \leq 1, \end{cases}$$

where $G_\nu(z)$ is the Bose-Einstein function of order ν:

$$(A.6) \qquad G_\nu(z) = \frac{1}{\Gamma(\nu)}\int_0^\infty \frac{x^{\nu-1}}{z^{-1}e^x - 1}\,dx, \quad 0 < z < 1,\ \nu > 0;\ z = 1,\ \nu > 1.$$

For small z, the integrand in (A.6) can be expanded in powers of z,

$$G_\nu(z) = \sum_{n=1}^\infty \frac{z^n}{n^\nu} = z + \frac{z^2}{2^\nu} + \frac{z^3}{3^\nu} + \dots.$$

Thus the Bose gas behaves like a classical gas when $z \ll 1$. On the other hand, it becomes degenerate as $z \to 1$.

The famous BEC happens when $z > 1$. The equilibrium state $\mathcal{M}_{(z,\beta)}$ is then composed of two parts (in the sense of maximizing entropy):

$$\text{(A.7)} \qquad \mathcal{M}_{(z,\beta)}(\varepsilon) = \frac{1}{e^{\beta\varepsilon} - 1} + \frac{\ln z}{\rho(\varepsilon)}\delta(\varepsilon),$$

with z being now an indicator of the condensate mass. Correspondingly, M and E are given by

$$\begin{cases} M = \left(\dfrac{2\pi}{\beta}\right)^{\frac{3}{2}} G_{\frac{3}{2}}(1) + \ln z, \quad z > 1, \\[3mm] E = \dfrac{3}{2\beta}\left(\dfrac{2\pi}{\beta}\right)^{\frac{3}{2}} G_{\frac{5}{2}}(1), \quad z > 1. \end{cases}$$

Note that $G_\nu(1)$ is just the Riemann-Zeta function $\zeta(\nu)$ convergent for $\nu > 1$. In particular, $G_{3/2}(1) \approx 2.6124$, $G_{5/2}(1) \approx 1.3415$.

Acknowledgments

References

[1] Leif Arkeryd and Anne Nouri, *Bose condensates in interaction with excitations: a kinetic model*, Comm. Math. Phys. **310** (2012), no. 3, 765–788, DOI 10.1007/s00220-012-1415-1. MR2891873

[2] A. V. Bobylev and S. Rjasanow, *Fast deterministic method of solving the Boltzmann equation for hard spheres*, Eur. J. Mech. B Fluids **18** (1999), no. 5, 869–887, DOI 10.1016/S0997-7546(99)00121-1. MR1728639 (2001c:76109)

[3] C. Connaughton and Y. Pomeau, *Kinetic theory and Bose-Einstein condensation*. C. R. Pysique, 5:91–106, 2004.

[4] M. Escobedo, S. Mischler, and M. A. Valle, *Homogeneous Boltzmann equation in quantum relativistic kinetic theory*. J. Differ. Equ., Monograph 4, 2003.

[5] Miguel Escobedo, Federica Pezzotti, and Manuel Valle, *Analytical approach to relaxation dynamics of condensed Bose gases*, Ann. Physics **326** (2011), no. 4, 808–827, DOI 10.1016/j.aop.2010.11.001. MR2771726 (2012c:82046)

[6] Irene M. Gamba and Sri Harsha Tharkabhushanam, *Spectral-Lagrangian methods for collisional models of non-equilibrium statistical states*, J. Comput. Phys. **228** (2009), no. 6, 2012–2036, DOI 10.1016/j.jcp.2008.09.033. MR2500671 (2009m:82068)

[7] A. L. Garcia and W. Wagner, *Direct simulation Monte Carlo method for the Uehling-Uhlenbeck-Boltzmann equation*. Phys. Rev. E, 68:056703, 2003.

[8] Jingwei Hu and Lexing Ying, *A fast spectral algorithm for the quantum Boltzmann collision operator*, Commun. Math. Sci. **10** (2012), no. 3, 989–999. MR2911206

[9] I. Ibragimov and S. Rjasanow, *Numerical solution of the Boltzmann equation on the uniform grid*, Computing **69** (2002), no. 2, 163–186, DOI 10.1007/s00607-002-1458-9. MR1954793 (2004i:82060)

[10] Boris N. Khoromskij, *Structured data-sparse approximation to high order tensors arising from the deterministic Boltzmann equation*, Math. Comp. **76** (2007), no. 259, 1291–1315, DOI 10.1090/S0025-5718-07-01901-1. MR2299775 (2008f:15097)

[11] Robert Lacaze, Pierre Lallemand, Yves Pomeau, and Sergio Rica, *Dynamical formation of a Bose-Einstein condensate*, Phys. D **152/153** (2001), 779–786, DOI 10.1016/S0167-2789(01)00211-1. Advances in nonlinear mathematics and science. MR1837939 (2002c:82070)

[12] E. M. Lifshitz and L. P. Pitaevskiĭ, *Course of theoretical physics ["Landau-Lifshits"]. Vol. 9*, Pergamon Press, Oxford, 1980. Statistical physics. Part 2. Theory of the condensed state; Translated from the Russian by J. B. Sykes and M. J. Kearsley. MR586944 (84m:82003a)

[13] Xuguang Lu and Xiangdong Zhang, *On the Boltzmann equation for 2D Bose-Einstein particles*, J. Stat. Phys. **143** (2011), no. 5, 990–1019, DOI 10.1007/s10955-011-0221-z. MR2811470 (2012i:82047)

[14] Peter A. Markowich and Lorenzo Pareschi, *Fast conservative and entropic numerical methods for the boson Boltzmann equation*, Numer. Math. **99** (2005), no. 3, 509–532, DOI 10.1007/s00211-004-0570-5. MR2117737 (2006a:82055)

[15] Clément Mouhot and Lorenzo Pareschi, *Fast algorithms for computing the Boltzmann collision operator*, Math. Comp. **75** (2006), no. 256, 1833–1852 (electronic), DOI 10.1090/S0025-5718-06-01874-6. MR2240637 (2007d:65095)

[16] L. W. Nordheim, On the kinetic method in the new statistics and its application in the electron theory of conductivity. *Proc. R. Soc. London, Ser. A*, 119:689–698, 1928.

[17] Lorenzo Pareschi and Benoit Perthame, *A Fourier spectral method for homogeneous Boltzmann equations*, Proceedings of the Second International Workshop on Nonlinear Kinetic Theories and Mathematical Aspects of Hyperbolic Systems (Sanremo, 1994), 1996, pp. 369–382, DOI 10.1080/00411459608220707. MR1407541 (97j:82133)

[18] Lorenzo Pareschi and Giovanni Russo, *Numerical solution of the Boltzmann equation. I. Spectrally accurate approximation of the collision operator*, SIAM J. Numer. Anal. **37** (2000), no. 4, 1217–1245, DOI 10.1137/S0036142998343300. MR1756425 (2001g:65175)

[19] R. K. Pathria, *Statistical Mechanics*. Butterworth-Heinemann, second edition, 1996.

[20] D. V. Semikoz and I. I. Tkachev, *Kinetics of Bose condensation*. Phys. Rev. Lett., 74:3093–3097, 1995.

[21] D. V. Semikoz and I. I. Tkachev, *Condensation of bosons in the kinetic regime*. Phys. Rev. D, 55:489–502, 1997.

[22] Herbert Spohn, *Kinetics of the Bose-Einstein condensation*, Phys. D **239** (2010), no. 10, 627–634, DOI 10.1016/j.physd.2010.01.018. MR2601928 (2011i:82038)

[23] E. A. Uehling and G. E. Uhlenbeck, *Transport phenomena in Einstein-Bose and Fermi-Dirac gases. I.* Phys. Rev., 43:552–561, 1933.

INSTITUTE FOR COMPUTATIONAL ENGINEERING AND SCIENCES (ICES), THE UNIVERSITY OF TEXAS AT AUSTIN, 1 UNIVERSITY STATION, C0200, AUSTIN, TEXAS 78712

E-mail address: `hu@ices.utexas.edu`

DEPARTMENT OF MATHEMATICS AND INSTITUTE FOR COMPUTATIONAL AND MATHEMATICAL ENGINEERING (ICME), STANFORD UNIVERSITY, 450 SERRA MALL, BLDG 380, STANFORD, CALIFORNIA 94305

E-mail address: `lexing@math.stanford.edu`