SIAM J. NUMER. ANAL.
Vol. 45, No. 1, pp. 54–67

TIME SPLITTING ERROR IN DSMC SCHEMES FOR THE SPATIALLY HOMOGENEOUS INELASTIC BOLTZMANN EQUATION*

SERGEJ RJASANOW† AND WOLFGANG WAGNER‡

Abstract. The paper is concerned with the numerical treatment of the uniformly heated inelastic Boltzmann equation by the direct simulation Monte Carlo (DSMC) method. This technique is presently the most widely used numerical method in kinetic theory. We consider three modifications of the DSMC method and study them with respect to their efficiency and convergence properties. Convergence is investigated with respect to both the number of particles and the time step. The main issue of interest is the time step discretization error due to various splitting strategies. A scheme based on the Strang-splitting strategy is shown to be of second order with respect to time step, while there is only first order for the commonly used Euler-splitting scheme. On the other hand, a no-splitting scheme based on appropriate Markov jump processes does not produce any time step error. It is established in numerical examples that the no-splitting scheme is about two orders of magnitude more efficient than the Euler-splitting scheme. The Strang-splitting scheme reaches almost the same level of efficiency as that of the no-splitting scheme, since the deterministic time step error vanishes sufficiently fast.

Key words. granular matter, Boltzmann equation, stochastic numerics

AMS subject classifications. 82C40, 82C80, 65R20

DOI. 10.1137/050643842

1. Introduction.

A basic tool for modeling low-density flows of granular materials is the inelastic Boltzmann equation. We refer to the conference proceedings [17], [16] and to the monograph [5] for details concerning applications and an appropriate physical justification. In this paper we consider the spatially homogeneous uniformly heated inelastic Boltzmann equation

$$\partial_t f - \beta \, \Delta_v \, f = Q_\alpha(f, f) \tag{1.1}$$

with initial condition

$$f(0, v) = f_0(v). \tag{1.2}$$

Equation (1.1) describes the time evolution of a function $f(t, v)$ representing the average number of particles at time t having a velocity close to v. The symbol Δ denotes the Laplace operator and the parameter $\beta > 0$ determines the strength of the random forcing. The collision integral is most conveniently written in the weak form

$$\begin{aligned} &\int_{\mathbb{R}^3} Q_\alpha(f, f)(v) \, \varphi(v) \, dv \\ &\quad = \frac{1}{2} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \int_{S^2} B(v, w, e) \, [\varphi(v'_\alpha) + \varphi(w'_\alpha) - \varphi(v) - \varphi(w)] \, f(v) \, f(w) \, de \, dw \, dv, \end{aligned} \tag{1.3}$$

*Received by the editors October 31, 2005; accepted for publication (in revised form) July 13, 2006; published electronically January 8, 2007.
http://www.siam.org/journals/sinum/45-1/64384.html

†Fachrichtung 6.1 – Mathematik, Universität des Saarlandes, Postfach 15 11 50, 66041 Saarbrücken, Germany (rjasanow@num.uni-sb.de).

‡Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstraße 39, D-10117 Berlin, Germany (wagner@wias-berlin.de).

where φ is a test function and S^2 denotes the unit sphere in the Euclidean space $\mathbb{R}^3$. The function B is called the collision kernel. The postcollisional velocities are defined by

$$
\begin{aligned}
v'_\alpha = v'_\alpha(v,w,e) &= \frac{1}{2}(v+w) + \frac{1-\alpha}{4}(v-w) + \frac{1+\alpha}{4}|v-w|\,e,\\
w'_\alpha = w'_\alpha(v,w,e) &= \frac{1}{2}(v+w) - \frac{1-\alpha}{4}(v-w) - \frac{1+\alpha}{4}|v-w|\,e.
\end{aligned}
\tag{1.4}
$$

The parameter $0 < \alpha \le 1$ is called the restitution coefficient. For $\alpha = 1$ the collisions are elastic and $Q_1(f,f)$ coincides with the classical Boltzmann collision operator. A discussion of the relevance of (1.1), as well as more references, can be found in [10].

In this paper we address the issue of the numerical treatment of (1.1) by the direct simulation Monte Carlo (DSMC) method. This technique is presently the most widely used numerical method in kinetic theory (cf. [2], [6]). It is based on a system of particles performing a random evolution that imitates the behavior of the underlying physical model. As to inelastic collisions, the homogeneous cooling state of a low-density granular flow was studied by the DSMC method in [4]. A DSMC method for uniformly heated granular fluids, described by (1.1), was introduced in [13]. Related studies were performed in [1], [21]. We refer to [8] for an account of deterministic numerical methods for the elastic Boltzmann equation and to [9] concerning a deterministic numerical approach to (1.1).

The purpose of this paper is to study three modifications of the DSMC method for the uniformly heated inelastic Boltzmann equation with respect to their efficiency and convergence properties. The main issue of interest is the time step discretization error due to various splitting strategies. The first method, from [13], implements a straightforward Euler-type splitting in analogy with the classical Bird scheme for the spatially inhomogeneous elastic Boltzmann equation. The second method follows the Strang-splitting strategy (cf. [20]). The third method, previously used in [11], avoids any time step discretization error, since no splitting is used. Convergence is studied with respect to both the number of particles and the time step. All methods are of first order with respect to the inverse number of particles. The Strang-splitting scheme is of second order with respect to the time step, while the Euler-splitting scheme is of first order. In the numerical examples, the no-splitting scheme is about two orders of magnitude more efficient than the Euler-splitting scheme. It is observed that the Strang-splitting scheme reaches almost the same level of efficiency compared to the no-splitting scheme, since the deterministic time step error vanishes sufficiently fast.

The paper is organized as follows. In section 2 we describe a Markovian particle system approximating (1.1). In section 3 we define the three DSMC algorithms mentioned above. In section 4 we introduce a test example and present the results of numerical experiments. Here we study efficiency and convergence properties of the algorithms both in the transient and in the steady state cases. Section 5 contains some concluding remarks.

2. The direct simulation process. Here we describe the time evolution of a Markovian particle system

$$
\Big(v_1(t),\dots,v_n(t)\Big), \qquad t \ge 0, \tag{2.1}
$$

where each particle is characterized by its velocity. The process (2.1) corresponds to (1.1) in the sense that the family of empirical measures

$$\nu^{(n)}(t, dv) = \frac{1}{n} \sum_{j=1}^{n} \delta_{v_j(t)}(dv) \tag{2.2}$$

converges (as $n \to \infty$) to the measures $f(t, v)\, dv$. We refer to [14], [12] concerning rigorous convergence results for a wide class of Boltzmann-type models (see also [19, section 2.3.3]).

Roughly speaking, the system interacts through binary inelastic collisions. In addition, the particles continuously gain kinetic energy due to Gaussian white noise forcing. More precisely, we assume

$$\int_{S^2} B(v, w, e)\, de \le B_{max} \qquad \forall\, v, w \in \mathbb{R}^3. \tag{2.3}$$

Then the evolution of system (2.1) is determined via the following steps.

Initial measure. System (2.1) at $t = 0$ is chosen in such a way that the empirical measure $\nu^{(n)}(0, dv)$ (cf. (2.2)) approximates the initial measure $f_0(v)\, dv$ (cf. (1.2)).

Time counter. Given system (2.1) at time t, the next interaction (collision) takes place at a random time $t + \tau$, where

$$\text{Prob}\,\{\tau \ge s\} = \exp\left(-\frac{n-1}{2} B_{max}\, s\right), \qquad s \ge 0. \tag{2.4}$$

Brownian motion. The particle velocities perform individual Brownian motions between the collisions. After some time τ without collisions, the particle velocities are given by

$$v_j(t + \tau) = v_j(t) + \sqrt{2\,\beta\,\tau}\;\xi_j, \qquad j = 1, \ldots, n, \tag{2.5}$$

where $\xi_j \in \mathbb{R}^3$ are independent standard Gaussian random variables.

Collision partners. The indices i and j of the collision partners are chosen uniformly on the set $\{1 \le i \ne j \le n\}$.

Fictitious collision. Given i and j, the collision is fictitious (the system does not change) with probability

$$1 - \frac{\int_{S^2} B(v_i, v_j, e)\, de}{B_{max}}. \tag{2.6}$$

Collision. With the remaining probability, a direction vector e is generated according to the density

$$\frac{B(v_i, v_j, e)}{\int_{S^2} B(v_i, v_j, e)\, de}, \qquad e \in S^2, \tag{2.7}$$

and the postcollisional velocities

$$v'_\alpha(v_i, v_j, e), \qquad w'_\alpha(v_i, v_j, e) \tag{2.8}$$

are computed according to the collision transformation (1.4).

3. DSMC algorithms. Here we describe three algorithms based on the Markov process introduced in the previous section. They differ in the way time splitting is carried out.

The algorithms perform the time evolution of a particle system $(v_1, \dots, v_n)$. At some observation points

$$s_m, \qquad m = 0, 1 \dots, M, \tag{3.1}$$

functionals of the system

$$\xi^{(n)} = \frac{1}{n} \sum_{j=1}^{n} \varphi(v_j) \tag{3.2}$$

are computed, where φ is an appropriate test function. The random variable (3.2) approximates the functional

$$\int_{\mathbb{R}^3} \varphi(v)\, f(s_m, v)\, dv \tag{3.3}$$

of the solution of (1.1).

In order to reduce the random fluctuations of the estimator (3.2), a number N of independent ensembles of particles is generated. The corresponding values of the random variable are denoted by $\xi_1^{(n)}, \dots, \xi_N^{(n)}$. The empirical mean value of the random variable (3.2)

$$\eta^{(n,N)} = \frac{1}{N} \sum_{j=1}^{N} \xi_j^{(n)} \tag{3.4}$$

is used as an approximation to the functional (3.3). The independent ensembles of particles are also used to estimate the random fluctuations by means of confidence intervals. For details we refer, e.g., to [19, section 3.1.4].

3.1. Euler-splitting scheme. First we describe the DSMC method introduced in [13]. It implements the idea of standard (elastic) DSMC, where the free flow and collision simulation are separated (cf. [2]). The simulation of random "kicking" and collisions is split over a time step Δt. The state of the particle system is calculated at the discrete time points

$$t_k = k\, \Delta t, \qquad k = 0, 1, \dots, \tag{3.5}$$

until all observation points (3.1) (assumed to be multiples of the time step) are reached.

ALGORITHM 3.1.

1. Initialization
   - 1.1 set system time $t = t_0$
   - 1.2 generate v_j, $j = 1, \dots, n$, according to $f_0(v)$
2. Simulation (for $k = 1, 2, \dots$)
   - 2.1 Collision step of length Δt
     - 2.1.1 compute τ according to (2.4)
     - 2.1.2 update the system time $t := t + \tau$
     - 2.1.3 if $t \geq t_k$ then go to Step 2.2

2.1.4 generate the indices $i \neq j$
2.1.5 go to Step 2.1.1 with probability (2.6)
2.1.6 generate e according to (2.7)
2.1.7 replace v_i and v_j according to (2.8)
2.2 Kicking step of length Δt
2.2.1 update all velocities (cf. (2.5))

$$v_i := v_i + \sqrt{2\beta\,\Delta t}\,\xi_i, \qquad i = 1, \dots, n$$

2.2.2 set system time $t = t_k$
3. Compute functional (3.2) at all s_m

3.2. Strang-splitting scheme. Next we describe a modification of the algorithm from the previous section. We apply the idea of the Strang splitting. This has been introduced in the context of the elastic Boltzmann equation in [15]. Its application to equations with rather general operators was studied in [3].

ALGORITHM 3.2.
1. Initialization
2. Simulation (for $k = 1, 2, \dots$)
2.1 Collision step of length $\Delta t/2$
2.2 Kicking step of length Δt
2.3 Collision step of length $\Delta t/2$
3. Computation of functionals

3.3. No-splitting scheme. Finally we recall a DSMC algorithm that was introduced in [11]. The symbols σ_j denote the last time, at which the particle j was kicked.

ALGORITHM 3.3.
1. Initialization
1.1 set system time $t = 0$
1.2 generate v_j, $j = 1, \dots, n$, according to $f_0(v)$
1.3 set $\sigma_j = 0$, $j = 1, \dots, n$
2. Simulation (for $m = 0, 1, \dots, M$)
2.1 compute τ according to (2.4)
2.2 update the system time $t := t + \tau$
2.3 if $t \geq s_m$ then go to Step 3
2.4 generate the indices $i \neq j$
2.5 update the velocities v_i and v_j (cf. (2.5))

$$v_i := v_i + \sqrt{2\beta\,(t - \sigma_i)}\,\xi_i, \quad v_j := v_j + \sqrt{2\beta\,(t - \sigma_j)}\,\xi_j$$

2.6 update the times of last kicking $\sigma_i = \sigma_j := t$
2.7 go to Step 2.1 with probability (2.6)
2.8 generate e according to (2.7)
2.9 replace v_i and v_j according to (2.8) and go to Step 2.1
3. Calculation of functionals
3.1 update the velocities of all particles (cf. (2.5))

$$v_i := v_i + \sqrt{2\beta\,(s_m - \sigma_i)}\,\xi_i, \qquad i = 1, \dots, n$$

3.2 compute (3.2)
3.3 set system time $t = s_m$ and go to Step 2.1

3.4. Comments. In Algorithms 3.1 and 3.2, the kicking step is computed accurately, i.e., without any further time discretization. Particles are just moved according to Brownian motion. The collision step contains the random interaction times distributed according to (2.4). Its accuracy depends on the number of particles. In Algorithm 3.3 particles perform Brownian motion between collisions so that any splitting errors are avoided.

One might use deterministic interaction times obtained as the expectation of the distribution (2.4). This would introduce another error, which is small for large n. We refer to [19, section 3.5.2] concerning a discussion of various time counting procedures.

Unbounded collision kernels. The *variable hard sphere model*

$$B(v,w,e) = C_\lambda\,|v-w|^\lambda, \qquad 0 \le \lambda \le 1, \tag{3.6}$$

is widely used in applications (cf. [2, Chapter 2]). Particular cases are the models of *hard spheres* ($\lambda = 1$) and of *pseudo-Maxwell molecules* ($\lambda = 0$). The kernel (3.6) does not satisfy condition (2.3), unless $\lambda = 0$. In order to fit into the framework of section 2, one has to truncate the kernel using some maximal relative velocity U_{max}. The truncated kernel

$$\hat{B}(v,w,e) = \begin{cases} B(v,w,e) & \text{if} \quad |v-w| \le U_{max}, \\ C_\lambda\, U_{max}^\lambda & \text{otherwise} \end{cases} \tag{3.7}$$

satisfies (2.3) with $B_{max} = 4\pi\, C_\lambda\, U_{max}^\lambda$. Correspondingly, the parameter of the waiting time distribution (2.4) takes the form

$$2\pi\,(n-1)\,C_\lambda\,U_{max}^\lambda. \tag{3.8}$$

The probability of a fictitious collision (2.6) is

$$1 - \left(\frac{|v_i - v_j|}{U_{max}}\right)^\lambda.$$

The density (2.7) is constant so that the vector e is distributed uniformly on the unit sphere.

Adapting majorants. There are two aspects related to the choice of the truncation parameter U_{max}. If it is small, then the solution of (1.1) for the kernel (3.7) will significantly differ from the solution for the original kernel B. If, on the other hand, the parameter U_{max} is big, then the time steps between collisions are small (inverse of parameter (3.8)) and the algorithms are time consuming. Therefore, the parameter U_{max} is usually derived from the particle system used in the simulation.

In the classical (elastic) DSMC algorithm (cf. [2]) the starting value of U_{max} is based on the temperature of the initial particle system. Then this value is adapted during the process of calculation each time the relative velocity of a pair of particles exceeds the stored quantity. This procedure works well in steady state calculations. The error related to this procedure in transient calculations was studied in [18].

A problem with finding the maximum relative velocity in a particle system is related to the fact that the effort is quadratic in the number of particles. However, this can easily be reduced to a linear effort by using the estimate

$$\max_{i,j} |v_i - v_j| \le \max_{i,j} \Big(|v_i - V| + |V - v_j| \Big) = 2 \max_i |v_i - V|,$$

where V is any fixed vector. A particular choice is the numerical bulk velocity

$$V = \frac{1}{n} \sum_{j=1}^{n} v_j.$$

Thus, one may start with $U_{max} = 2 \max_i |v_i - V|$ and update the majorant after each collision

$$U_{max} := \max \Big\{ U_{max}, |v_i - V|, |v_j - V| \Big\}. \tag{3.9}$$

This procedure is used in Algorithms 3.1 and 3.2. The situation in Algorithm 3.3 is slightly more difficult, since particle velocities change continuously as a result of the kicking process. Here we implemented the above procedure of adapting the majorant, but in addition the quantity U_{max} was updated according to (3.9) after each Step 2.5. The error caused by this truncation does not seem to be significant, as shown by the very precise tail calculations in [11].

4. Numerical examples. Here we test the algorithms introduced in the previous section with respect to their convergence properties and efficiency.

We consider the case of a constant collision kernel, namely, (3.6), with

$$\lambda = 0, \qquad C_0 = \frac{1}{\pi}. \tag{4.1}$$

Note that other values of the constant C_0 can be handled by an appropriate time scaling, since the function $f(c\,t, v)$ solves (1.1) with diffusion coefficient $c\,\beta$ and collision kernel $c\,B$, where $c > 0$ is some constant. Furthermore, we assume

$$\int_{\mathbb{R}^3} f_0(v)\, dv = 1, \qquad \int_{\mathbb{R}^3} v\, f_0(v)\, dv = 0.$$

Note the conservation properties

$$\int_{\mathbb{R}^3} f(t,v)\, dv = \int_{\mathbb{R}^3} f_0(v)\, dv, \qquad \int_{\mathbb{R}^3} v\, f(t,v)\, dv = \int_{\mathbb{R}^3} v\, f_0(v)\, dv,$$

which can be derived easily from the weak form of the equation (cf. (1.3)).

In this case the relaxation of the temperature

$$T(t) = \frac{1}{3} \int_{\mathbb{R}^3} |v|^2 f(t,v)\, dv \tag{4.2}$$

is known analytically. Assuming $0 < \alpha < 1$, one obtains (cf. [11])

$$\begin{aligned} T(t) = T_{\alpha,\beta}(t) &= T_0\, e^{-(1-\alpha^2)\,t} + T_{\alpha,\beta}(\infty) \left(1 - e^{-(1-\alpha^2)\,t} \right) \\ &= T_{\alpha,\beta}(\infty) + [T_0 - T_{\alpha,\beta}(\infty)]\, e^{-(1-\alpha^2)\,t}, \end{aligned} \tag{4.3}$$

where

$$T_0 = \frac{1}{3} \int_{\mathbb{R}^3} |v|^2 f_0(v)\, dv$$

and

$$T_{\alpha,\beta}(\infty) = \frac{2\,\beta}{1-\alpha^2}. \tag{4.4}$$

Note that

$$\lim_{\alpha\to 1} T_{\alpha,\beta}(t) = T_0 + 2\beta\, t.$$

According to (4.1), the collision kernel is bounded so that the only sources of error are the number of particles n, the time step Δt (in Algorithms 3.1 and 3.2), and the number of independent samples N (cf. (3.4)). First order of convergence with respect to n has been established under rather general assumptions (cf. [14], [12] concerning the transient case and [7] concerning the steady state case). Convergence with respect to Δt for Euler splitting (first order) and Strang splitting (second order) was studied in [3] in the context of rather general operator equations. We refer to [19, section 3.5.5] for more details.

4.1. Approximation on a finite time interval (transient case). Here we use the Maxwell distribution

$$f_0(v) = \frac{1}{(2\,\pi)^{3/2}}\, e^{-\frac{|v|^2}{2}} \tag{4.5}$$

as the initial condition. For the parameters

$$\alpha = \frac{1}{2}, \qquad \beta = 1, \tag{4.6}$$

one obtains from (4.3), (4.4)

$$T(t) = e^{-\frac{3}{4}\,t} + \frac{8}{3}\left(1 - e^{-\frac{3}{4}\,t}\right). \tag{4.7}$$

We approximate the evolution of the temperature (4.2) on the time interval $[0, 8.0]$, using (3.2) with $\varphi(v) = \frac{1}{3}\,|v|^2$ (cf. (3.3)). The time step in the splitting schemes (cf. (3.5)) is chosen in the form

$$\Delta t = \frac{8}{K}, \qquad K \geq 4.$$

Unless indicated otherwise, the results are averaged over $N = 10\,000$ independent ensembles (cf. (3.4)).

Particle number convergence. Figure 4.1 illustrates the approximation of the analytical solution (4.7) (dashed line) by confidence bands (solid lines) computed using the no-splitting scheme with two different values of n. A "zoom" on the time interval $[5.0, 8.0]$ of the no-splitting scheme for two higher values of n is shown in Figure 4.2. The analytical solution (4.7) is mostly covered by the confidence interval for $n = 4\,096$ and completely covered for $n = 65\,536$. More detailed results are given in Table 4.1. The errors are computed as

$$E_{end} = \left|\frac{T(t_K) - T_K}{T(t_K)}\right|, \qquad E_{max} = \max_{0\leq k\leq K}\left|\frac{T(t_k) - T_k}{T(t_k)}\right|,$$

where $T(t_k)$ are the exact values (4.7) of the temperature at time point t_k and T_k is the computed temperature. The convergence factors (quotients of subsequent values) are denoted by CF. The results in Table 4.1 clearly indicate the expected convergence order $O(n^{-1})$ of the error. Note that the width Conf_{end} of the confidence interval at t_K is proportional to $\frac{1}{\sqrt{n\,N}}$, since the variance of the estimator (3.2) has the order $\frac{1}{n}$.

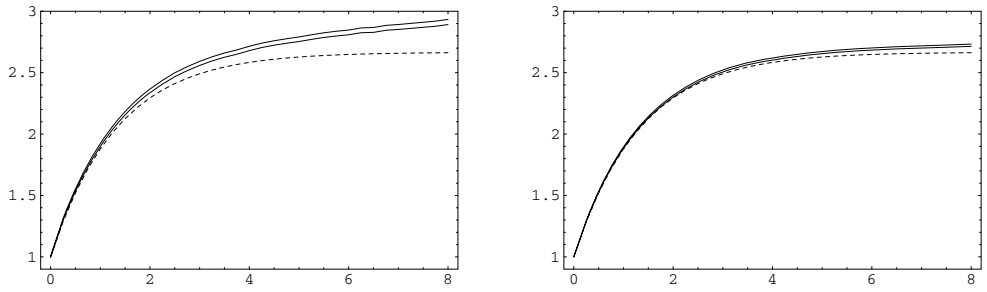

FIG. 4.1. *No-splitting scheme for $n = 64$ (left) and $n = 256$ (right).*

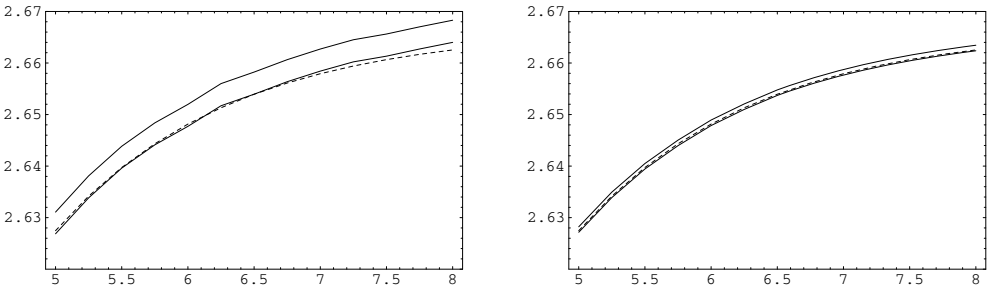

FIG. 4.2. *No-splitting scheme for $n = 4\,096$ (left) and $n = 65\,536$ (right).*

TABLE 4.1
No-splitting scheme for different n.

n	E_{end}	CF	Conf_{end}	CF	E_{max}	CF
16	0.359 E-00	-	0.119 E-00	-	0.359 E-00	-
64	0.939 E-01	3.82	0.420 E-01	2.83	0.939 E-01	3.82
256	0.229 E-01	4.10	0.183 E-01	2.30	0.229 E-01	4.10
1 024	0.584 E-02	3.92	0.865 E-02	2.11	0.584 E-02	3.92
4 096	0.136 E-02	4.29	0.430 E-02	2.01	0.136 E-02	4.29
16 382	0.293 E-03	4.64	0.215 E-02	2.00	0.332 E-03	4.10
65 536	0.141 E-03	2.08	0.108 E-02	1.99	0.141 E-03	2.35

TABLE 4.2
Euler- and Strang-splitting schemes for $n = 4\,096$.

K	E_{max}^{Euler}	CF	E_{max}^{Strang}	CF
4	0.931 E-00	-	0.863 E-01	-
8	0.421 E-00	2.21	0.195 E-01	4.42
16	0.200 E-00	2.11	0.501 E-02	3.89
32	0.977 E-01	2.05	0.997 E-03	5.03

Time step convergence. Similar convergence behavior with respect to the particle number is observed for the Euler- and Strang-splitting schemes, except that the n-limits contain an error depending on Δt. The corresponding numerical results are collected in Table 4.2. The linear convergence of the Euler-splitting scheme as well as the quadratic convergence of the Strang-splitting scheme are clearly indicated.

For $n = 4\,096$ and $K = 32$ the error of the Strang-splitting scheme is comparable to that of the no-splitting scheme, while the error of the Euler-splitting scheme is about 70 times larger. We note that for these parameters the numerical work of all

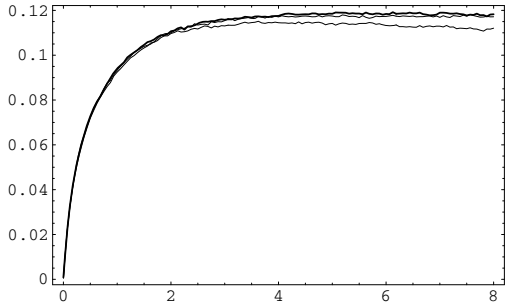

FIG. 4.3. *No-splitting scheme for* $n = 256$, 1 024, *and* 4 096 *(from below).*

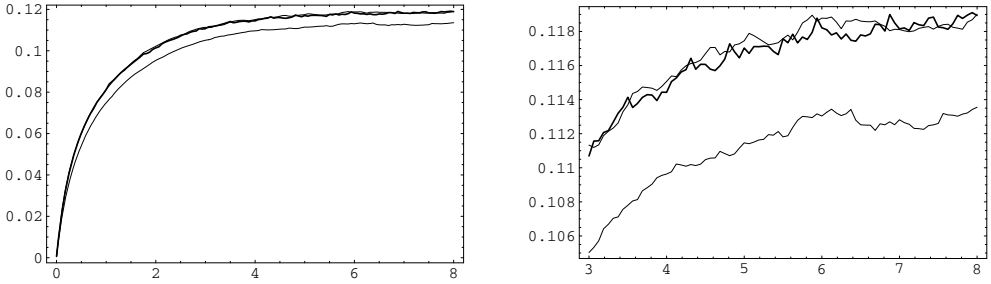

FIG. 4.4. *No-splitting scheme (thick line), Strang-splitting scheme (upper thin line), and Euler-splitting scheme (lower thin line) for* $n = 4\,096$.

schemes is roughly the same. A more detailed study of the efficiency will be made in section 4.2.

Deviation from the Maxwellian state. An interesting feature of the inelastic Boltzmann equation (1.1) is a non-Maxwellian steady state. A specific "criterion" for detecting deviations from the Maxwellian state has the form (cf. [19, section 1.8])

$$Crit(t) = \frac{1}{T(t)} \left(\frac{1}{2} \|P(t) - p(t)\, I\|_F^2 + \frac{2}{5\,T(t)} \|q(t)\|^2 + \frac{1}{120\,T(t)^2} \gamma(t)^2 \right)^{1/2}, \tag{4.8}$$

where $P(t)$ is the pressure tensor, $p(t)$ is the scalar pressure, $q(t)$ is the heat flux vector, I denotes the identity matrix, $\|A\|_F$ denotes the Frobenius norm of a matrix A, and

$$\gamma(t) = \int_{\mathcal{R}^3} \|v\|^4 f(t,v)\, dv - 15\, T(t)^2$$

is a fourth moment of the distribution function.

In Figure 4.3 we show the criterion (4.8) obtained by the no-splitting scheme with different values of n. The curves were calculated using, respectively, $N = 65\,536$, 16 192, and 4 096 repetitions. The functional starts from zero, according to (4.5), and tends to a strictly positive stationary value as $t \to \infty$. So it allows one to quantify the deviation from the Maxwellian state.

For comparison the corresponding curves for the splitting schemes with $K = 32$, $n = 4\,096$, and $N = 4\,096$ are provided in Figure 4.4. Similar to what was observed

TABLE 4.3
Euler-splitting scheme for $n = 65\,536$.

K	T_∞	Error	CF	Conf	CPU	CF
32	2.9226	0.2559	-	0.0046	1.09	-
256	2.6975	0.0308	8.3	0.0041	6.57	6.03
512	2.6817	0.0150	2.1	0.0040	12.8	1.95
2 048	2.6719	0.0052	2.9	0.0039	51.1	3.99
4 094	2.6710	0.0043	1.2	0.0053	100.4	1.96

TABLE 4.4
Strang-splitting scheme for $n = 65\,536$.

K	T_∞	Error	CF	Conf	CPU
8	2.6055	0.0612	-	0.0040	0.61
16	2.6520	0.0147	4.2	0.0047	0.86
32	2.6620	0.0047	3.1	0.0038	1.07
64	2.6655	0.0012	3.9	0.0045	1.87

TABLE 4.5
No-splitting scheme for $n = 65\,536$.

T_∞	Error	Conf	CPU
2.6684	0.0017	0.0040	1.0

for the temperature, the Strang-splitting scheme gives basically the same accuracy as the no-splitting scheme, while the error of the Euler-splitting scheme is significantly larger.

4.2. Approximation of the stationary value (steady state case). Here we consider the same example as in section 4.1 (cf. (4.7)). Figures 4.1 and 4.2 show that the stationary value $T(\infty) \sim 2.6667$ has almost been reached. So we start averaging at $s_0 = 8$ (cf. (3.1)). The quantity of interest is measured after each $\Delta t_{\max} = 2$ so that

$$s_m = s_0 + m\,\Delta t_{\max}, \qquad m = 1, \ldots, M.$$

This time step (corresponding to $K = 4$ in the context of the previous subsection) is big enough to assure almost independent observations. Confidence intervals are constructed over $M = 64$ observation points, so that $s_M = 136$.

Time step error. We choose $n = 65\,536$ so that the particle number error is negligible. Results for the different methods are given in Tables 4.3–4.5. As above, Conf denotes the width of the confidence interval and CF denotes the convergence factors (quotients of subsequent values). The CPU times for the splitting schemes are measured relative to the CPU time for the no-splitting scheme.

First order of the time step error is observed for the Euler-splitting scheme, while the Strang-splitting scheme provides second order. The no-splitting scheme avoids any time step error. Note that 0.0267 would be a 1% error.

Efficiency. The effort is roughly determined by the sum of the mean number of collisions N_{coll} and the mean number of kicks N_{kick}. These quantities can be predicted rather accurately.

The mean number of collisions is (cf. (3.8), (4.1))

$$N_{\text{coll}}(n) = 2\,(n-1)\,s_M.$$

TABLE 4.6
Example (4.9) *for* $n = 65\,536$.

Method	T_∞	Error	Conf
Euler	109.596	9.596	0.1711
Strang	99.739	0.261	0.1425
no-split	99.981	0.019	0.1877

This quantity does not depend on the particular splitting procedure. The number of kicks is easily calculated from the other parameters. In the no-splitting scheme one obtains

$$N_{\rm kick}^{\rm nosplit}(n) = 2\,N_{\rm coll}(n) + n\,M, \qquad M = \frac{s_M - s_0}{\Delta t_{\max}},$$

since at each collision both partners are kicked and, in addition, all particles are kicked before making a measurement. In the other methods one obtains

$$N_{\rm kick}^{\rm split}(n) = \frac{s_M}{\Delta t}\,n,$$

independently of the particular way of splitting. Accordingly, the effort is roughly the same for the splitting and no-splitting schemes if

$$4\,(n-1)\,s_M + n\,\frac{s_M - s_0}{\Delta t_{\max}} \;\sim\; \frac{s_M}{\Delta t}\,n$$

or

$$\frac{1}{\Delta t} \sim 4 + \frac{1}{\Delta t_{\max}}.$$

Thus, all methods have a similar effort for $\Delta t \sim \frac{1}{4}$, i.e., $K \sim 32$. In general, the effort for the splitting schemes increases inversely proportional to the time step. Note that these predictions are confirmed by the CPU measurements in Tables 4.3–4.5.

In conclusion, the Euler-splitting scheme needs running time about 100 times longer than the no-splitting scheme to cover the correct temperature by the confidence interval. The Strang-splitting scheme needs running time only about two times longer. Alternatively, with the same effort, the error for the Euler-splitting scheme is 100 times bigger than that for the no-splitting scheme, while the error for the Strang-splitting scheme is two times bigger.

These conclusions are qualitatively confirmed by a rough test for another parameter configuration, namely,

$$\alpha = 0.5, \qquad \beta = 37.5 \tag{4.9}$$

instead of (4.6). In this case the exact asymptotic value of the temperature is $T(\infty) = 100$ (cf. (4.4)). All other parameters are as above, in particular, $K = 32$, so that all three methods have approximately the same effort. The results are given in Table 4.6.

5. Concluding remarks. The direct simulation Monte Carlo (DSMC) method is one of the basic tools for the numerical treatment of nonlinear kinetic equations so that improvements of its efficiency are of significant practical importance. In this paper we considered a particular application, namely, the uniformly heated inelastic Boltzmann equation. We investigated the performance of two new DSMC algorithms

compared to a commonly used procedure. The order of convergence with respect to the numerical parameters (number of particles, time step) as well as the computational efficiency of the algorithms were studied both in the transient case (approximation of the solution on a finite time interval) and in the steady state case (approximation of the stationary solution). One scheme uses the Strang-splitting strategy instead of the Euler-splitting scheme. It provides second order time step convergence instead of first order. The other scheme is based on an appropriate Markov process avoiding any splitting procedure. It can be considered as providing infinite order time step convergence. All schemes are of first order with respect to the inverse particle number. In our particular numerical test cases, both the Strang-splitting scheme and the no-splitting scheme were up to two orders of magnitude more efficient than the Euler-splitting scheme.

Here we comment on the relevance of the results for more general applications. The first direction of generalization concerns the type of the driving force in (1.1). The adaptation of the schemes to other mechanisms instead of Brownian motion, e.g., to deterministic force terms as in [13], is rather obvious. We expect that the main messages of the paper concerning "Strang versus Euler" and "no-splitting scheme" remain valid.

It should be emphasized that in the case of inelastic collisions the spatially homogeneous situation is of independent interest, since there is some "nontrivial" behavior as, for example, a non-Maxwellian steady state. This issue has been intensively studied in recent years. The no-splitting scheme is very useful for investigating "fine" properties of the solution, as higher moments or tails of the steady state distribution. It is remarkable that the no-splitting scheme not only avoids the time step discretization error, but also is usually even more efficient than the other schemes. The quantitative value of the efficiency gain depends on the concrete example, in particular, on the level of "acceptable" time step error: if big time steps are sufficient, there is less or no efficiency gain; if small time steps are required, the efficiency gain may be rather significant.

Another interesting aspect of the present study is that it throws some additional light on the controversial issue about the order of the time step error in the elastic case ($\beta = 0$, $\alpha = 1$). Without going into detail, we refer to [19, section 3.5.5] concerning a discussion of this matter for the DSMC method in rarefied gas dynamics. Since temperature is not conserved, it provides a simple nontrivial test example in the inelastic situation. This is in contrast to the elastic case, where the time step issue can be studied only in spatially inhomogeneous examples.

A second direction of further study concerns the spatially inhomogeneous situation. In this case a term (v, ∇_x) is added to (1.1) and the solution $f(t, x, v)$ depends on three more variables (position coordinates x). The direct simulation process introduced in section 2 can be adapted to this situation. In addition to being accelerated by a random force, particles change their positions between collisions. However, DSMC algorithms in engineering applications [6] are based on splitting. The point is that it would be computationally too expensive to take into account the relative positions for the whole system at all times. The splitting should be performed by the Strang strategy, moreover, since the Strang and Euler schemes have basically the same effort per trajectory. The no-splitting approach provides alternatives for the splitting procedure in the spatially inhomogeneous situation. For example, the acceleration term might be combined either with the motion term or with the collision term. Additional time step errors should be avoided, whenever this is possible, and the no-splitting idea may help to do so.

REFERENCES

[1] A. Barrat, T. Biben, Z. Rácz, E. Trizac, and F. van Wijland, *On the velocity distributions of the one-dimensional inelastic gas*, J. Phys. A, 35 (2002), pp. 463–480.

[2] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, UK, 1994.

[3] A. V. Bobylev and T. Ohwada, *On the generalization of Strang's splitting scheme*, Riv. Mat. Univ. Parma (6), 2* (1999), pp. 235–243.

[4] J. J. Brey, M. J. Ruiz-Montero, and D. Cubero, *Homogeneous cooling state of a low-density granular flow*, Phys. Rev. E (3), 54 (1996), pp. 3664–3671.

[5] N. V. Brilliantov and T. Pöschel, *Kinetic Theory of Granular Gases*, Oxford University Press, Oxford, UK, 2004.

[6] M. Capitelli, ed., *Rarefied Gas Dynamics*, Proceedings of the 24th International Symposium (Bari, Italy, 2004), AIP Conf. Proc. 762, AIP, New York, 2005.

[7] S. Caprino, M. Pulvirenti, and W. Wagner, *Stationary particle systems approximating stationary solutions to the Boltzmann equation*, SIAM J. Math. Anal., 29 (1998), pp. 913–934.

[8] F. Filbet and G. Russo, *Accurate numerical methods for the Boltzmann equation*, in Modeling and Computational Methods for Kinetic Equations, Model. Simul. Sci. Eng. Technol., Birkhäuser Boston, Boston, MA, 2004, pp. 117–145.

[9] F. Filbet, L. Pareschi, and G. Toscani, *Accurate numerical methods for the collisional motion of (heated) granular flows*, J. Comput. Phys., 202 (2005), pp. 216–235.

[10] I. M. Gamba, V. Panferov, and C. Villani, *On the Boltzmann equation for diffusively excited granular media*, Comm. Math. Phys., 246 (2004), pp. 503–541.

[11] I. M. Gamba, S. Rjasanow, and W. Wagner, *Direct simulation of the uniformly heated granular Boltzmann equation*, Math. Comput. Modelling, 42 (2005), pp. 683–700.

[12] C. Graham and S. Méléard, *Stochastic particle approximations for generalized Boltzmann models and convergence estimates*, Ann. Probab., 25 (1997), pp. 115–132.

[13] J. M. Montanero and A. Santos, *Computer simulations of uniformly heated granular fluids*, Granular Matter, 2 (2000), pp. 53–64.

[14] V. V. Nekrutkin and N. I. Tur, *On the justification of a scheme of direct modelling of flows of rarefied gases*, Zh. Vychisl. Mat. i Mat. Fiz., 29 (1989), pp. 1380–1392 (in Russian).

[15] T. Ohwada, *Higher order approximation methods for the Boltzmann equation*, J. Comput. Phys., 139 (1998), pp. 1–14.

[16] T. Pöschel and N. Brilliantov, eds., *Granular Gas Dynamics*, Lecture Notes in Phys. 624, Springer-Verlag, Berlin, New York, 2003.

[17] T. Pöschel and S. Luding, eds., *Granular Gases*, Lecture Notes in Phys. 564, Springer-Verlag, Berlin, New York, 2001.

[18] S. Rjasanow and W. Wagner, *On time counting procedures in the DSMC method for rarefied gases*, Math. Comput. Simulation, 48 (1998), pp. 151–176.

[19] S. Rjasanow and W. Wagner, *Stochastic Numerics for the Boltzmann Equation*, Springer Ser. Comput. Math. 37, Springer-Verlag, Berlin, 2005.

[20] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 5 (1968), pp. 506–517.

[21] J. S. van Zon and F. C. MacKintosh, *Velocity distributions in dissipative granular gases*, Phys. Rev. Lett., 93 (2004), article 038001.