Acta Numerica (1995), *pp.* 417–457

Particle Methods for the Boltzmann Equation

Helmut Neunzert and Jens Struckmeier
Department of Mathematics
University of Kaiserslautern
Germany
E-mail: neunzert@mathematik.uni-kl.de
and
struckm@mathematik.uni-kl.de

CONTENTS

1. Introduction

In the following chapters we will discuss particle methods for the numerical simulation of rarefied gas flows.

We will mainly treat a billiard game, that is, our particles will be hard spheres. But we will also touch upon cases where particles have internal energies due to rotation or vibration, which they exchange in a collision, and we will talk about chemical reactions happening during a collision.

Due to the limited size of this paper, we are only able to mention the principles of these real-gas effects. On the other hand, the general concepts of particle methods to be presented may be used for other kinds of kinetic equations, such as the semiconductor device simulation. We leave this part of the research to subsequent papers.

Finally, this paper is written by mathematicians. Missing physical intuition needed to 'simulate the game of nature' (Bird, 1989), we have to describe rarefied gas flows by a kinetic equation – this is the modelling part – and then we have to solve this equation numerically.

In a first – a modelling – part we will describe how to get the 'correct' kinetic equation. In a second part we shall describe our basic ideas for solving these equations. They lead to particle methods or – as we sometimes prefer to call them in order to stress the principal similarity to finite differences or finite elements – finite point set methods (FPM).

In Section 4 we shall talk about the practical aspects of a realization of particle methods and the rôle of random numbers and give a comparison between existing codes. In the last part we shall touch on several techniques to improve particle codes, to accelerate the algorithm and to use particle methods on massively parallel machines.

Finally we present some numerical results obtained with particle codes.

2. Collision Integrals for the Boltzmann Equation

Our mathematical model will be a kinetic equation describing the time evolution of a density in position–velocity space

$$t \rightarrow f(t,x,v), \qquad x \in \Omega, v \in \mathbb{R}^3,$$

which may depend on internal energies too. A kinetic equation has the form

$$\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} + E \cdot \frac{\partial f}{\partial v} = \hat{I}(f),$$

where E is an exterior or self-consistent force field and $\hat{I}(f)$ denotes the collision term.

A prototypical kinetic equation is the Boltzmann equation stated in 1874 by Ludwig Boltzmann. The equation describes the microscopic behaviour of a dilute gas undergoing binary collisions. For the rest of the paper we assume that the force field E vanishes. Hence the main aspect in the modelling part is the derivation of the collision integral.

2.1. Collision Integral

Bobylev (1993) gives a systematic derivation of $\hat{I}(f)$ from several quite simple postulates. We shall shortly review these results since they seem to offer a new approach for collision modelling – the classical approach due to Boltzmann or improved versions of it as given by Cercignani (1989) are well known.

(a) We take into account only binary collisions – hence $\hat{I}$ is a quadratic, time-independent operator

$$\hat{I}(f)(v) = I(f,f)(v) = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} K(v \mid v_1, v_2) f(v_1) f(v_2)\, \mathrm{d}v_1\, \mathrm{d}v_2.$$

Remark 1 This assumption fails if one has to consider recombination in chemical reactions, where a third collision partner is needed as an energy source.

(b) The collision operator $\hat{I}$ is invariant under translation in the velocity space: if

$$f_a(v) := f(v+a),$$

then

$$\hat{I}(f_a) = \hat{I}(f)_a$$

Remark 2 This assumption is not true for semiconductor devices.

From (a) and (b) one gets

$$K(v \mid v_1, v_2) = Q(v_1 - v, v_2 - v)$$

and

$$\hat{I}(f)(v) = \int\limits_{\mathbf{R}^3}\int\limits_{\mathbf{R}^3} Q(u_1, u_2) f(v+u_1) f(v+u_2)\, \mathrm{d}u_1\, \mathrm{d}u_2.$$

(c) The collision operator $\hat{I}$ is invariant under rotations in v-space. Then

$$Q(u_1, u_2) = \tilde{Q}(|u_1|, |u_2|, < u_1, u_2 >).$$

(d) The collision operator $\hat{I}$ can be decomposed into a gain and a loss term

$$Q = Q^+ - Q^- \text{ with } Q^\pm \geq 0$$

and $\hat{I}^-(f) = 0$ if $f = 0$: nothing can be lost, if there is nothing. Then

$$Q^-(u_1, u_2) = \frac{1}{2}\left[g(|u_1|)\delta(u_1) + g(|u_2|)\delta(u_1)\right],$$

where $g(|u|)$ is an arbitrary function.
With q defined by $Q^+(u_1, u_2) = 2^3 q(2u_1, 2u_2)$ we get

$$\begin{aligned}\hat{I}(f)(v) &= \int\limits_{\mathbf{R}^3}\int\limits_{\mathbf{R}^3} q(u-u', u+u') f(v') f(w')\, \mathrm{d}u'\, \mathrm{d}w \\ &\quad -f(v) \int\limits_{\mathbf{R}^3} g(|u|) f(w)\, \mathrm{d}w,\end{aligned}$$

where $u = v - w$, $v' = v + \frac{1}{2}(u' - u)$, $w' = w - \frac{1}{2}(u' - u)$.

(e) We have conservation of mass (or particles):

$$\int\limits_{\mathbf{R}^3} \hat{I}(f)(v)\, \mathrm{d}v = 0.$$

Then

$$g(|u|) = \int\limits_{\mathbf{R}^3} q(u' - u, u' + u)\, \mathrm{d}u'$$

and with $p(u' \mid u) = q(u - u', u + u')$ (transition probability) we get

$$\hat{I}(f)(v) = \int\limits_{\mathbb{R}^3}\int\limits_{\mathbb{R}^3} [p(u' \mid u)f(v')f(w') - p(u \mid u')f(v)f(w)]\, \mathrm{d}u'\,\mathrm{d}w.$$

(f) Microreversibility means $p(-u' \mid -u) = p(u \mid u')$. If we include the symmetry (c), we get

$$p(u \mid u') = \tilde{p}\left(|u|, |u'|, < u, u' >\right)$$

and

$$p(u' \mid u) = p(u \mid u').$$

From (f) we get the H-theorem

$$\int\limits_{\mathbb{R}^3} \hat{I}(f)(v)\ln f(v)\,\mathrm{d}v \leq 0.$$

Remark 3 Assumption (b) implies also conservation of momentum

$$\int\limits_{\mathbb{R}^3} v\hat{I}(f)(v)\,\mathrm{d}v = 0.$$

(g) Conservation of energy implies $p(u \mid u') = 0$ if $|u| \neq |u'|$. Then

$$p(u \mid u') = 2\delta\left((|u'|)^2 - |u|^2\right)\sigma\left(|u|, \frac{< u, u' >}{|u|^2}\right).$$

With $u' = |u'| \cdot \eta = |u| \cdot \eta$, $u = v - w$ we get finally

$$\hat{I}(f)(v) = \int\limits_{\mathbb{R}^3}\int\limits_{S^2} |u|\sigma\left(|u|, \frac{< u, \eta >}{|u|}\right)[f(v')f(w') - f(v)f(w)]\,\mathrm{d}w\,\mathrm{d}\omega,$$

where $\sigma(|u|, \cos\theta)$ is now the only undetermined function, the differential cross section.

The differential cross section σ is now to be chosen in such a way that we are able to reproduce measurements. These measurements are mainly those on transport coefficients – for example, the dependence of the kinematic viscosity on temperature. The simplest idea for σ is given by considering a billiard gas (in the phenomenological derivation)

$$\sigma(|u|, \cos\theta) = d \cdot \cos\theta,$$

where d is a constant connected with the diameter of the molecules. But this gives wrong macroscopic laws; for example, the viscosity η does not depend on T as experiments tell us, which is reflected in the Sutherland formula:

$$\frac{\eta}{\sqrt{T}} \sim \frac{T}{T+T_s}.$$

A better agreement can be achieved by changing σ a bit, using the so-called variable hard sphere (VHS) model

$$\sigma(|u|, \cos\theta) = d\left(1 + \frac{\alpha}{|u|^2}\right)\cos\theta.$$

In this model, the diameter 'shrinks' if the relative velocity $|u| = |v - w|$ is larger; this is not microscopically realistic, but reasonable in the sense of modelling.

2.2. Real-Gas Effects

If one wants to include real-gas effects such as inelastic scattering or chemical reactions, the model gets much more complicated. We will sketch the approach to these phenomena.

Assume we have a mixture of molecules A_2 and the corresponding atoms A. There are essentially (neglecting ionization) five kinds of collision processes that we have to take into account:

$$\begin{aligned}
A + A &\rightleftharpoons A + A, && (i)\\
A + A_2 &\rightleftharpoons A + A_2, && (ii)\\
A_2 + A_2 &\rightleftharpoons A_2 + A_2, && (iii)\\
A + A_2 &\rightleftharpoons A + A + A, && (iv)\\
A_2 + A_2 &\rightleftharpoons A + A + A_2. && (v)
\end{aligned}$$

The equations (i)–(iii) describe scattering processes, where (i) corresponds to the classical Boltzmann case. The possibility of dissociation and recombination is stated in (iv) and (v). Note that in the case of recombinations we have to consider triple collisions in order to fulfil energy and momentum conservation.

In 1960 Ludwig and Heil formulated a system of generalized Boltzmann equations describing the aforementioned collision processes. Following Kuščer (1991) we reformulate these equations in terms of differential cross sections.

Let $f(v,t)$and $g(v,\epsilon,t)$ be the distribution functions for the components A and A_2 of the mixture, where ϵ represents the internal energy of the molecule. Then the Boltzmann (or 'Ludwig–Heil') equations for f (and g, respectively) have collision terms representing these 5 collision processes. The differential cross sections depend on the total energy E of the process (instead of $|u|$) and on internal energies. As an example we show just one expression occurring in the description of the dissociation of molecules:

$$\sum_{\epsilon_2,\epsilon',\epsilon_1'} \int \frac{m^2|u'|^2}{16E_{\text{tr}}^2}\sigma_{\text{dm}}\left(E,\eta',\epsilon',\epsilon_1' \to E_{\text{tr}},\omega,\epsilon_2\right)$$

$$\times\left[g'g_1' - \left(\frac{2h}{m}\right)^3 ff_1g_2\right] \mathrm{d}v_1\,\mathrm{d}v_2\,\mathrm{d}\omega(\eta')$$

and it would take some time to explain all the terms here. Recombination is shown in $\left(\frac{2h}{m}\right)^3 ff_1g_2$ and it is still an open question whether recombination plays a significant rôle in real applications.

The undetermined part is again σ; for the nonreactive part, where the molecules and atoms are just scattered, one uses a generalization of the so-called Larssen–Borgnakke model (Borgnakke–Larssen, 1975) that consists essentially in dividing the differential cross section into three parts and performing 'detailed balance'. For collisions among diatomic molecules the model is as follows:

$$\sigma_{\text{sm}}(E,\eta\cdot\eta',\epsilon',V_i,\epsilon_1',V_j \to \epsilon,V_k,\epsilon_1,V_\ell) = (1-a-b)\sigma_{\text{sm,el}} + a\sigma_{\text{sm,ve}} + b\sigma_{\text{sm,in}}$$

with

$$\begin{aligned}
\sigma_{\text{sm,el}} &= \frac{1}{4\pi}\sigma_{\text{sm}}^{\text{tot}}(E)\cdot\delta(\epsilon-\epsilon')\delta(\epsilon_1-\epsilon_1')\delta_{ik}\delta_{j\ell},\\
\sigma_{\text{sm,ve}} &= \frac{3}{2\pi E^3}\sigma_{\text{sm}}^{\text{tot}}(E)\cdot(E-\epsilon-\epsilon_1)\delta_{ik}\delta_{j\ell},\\
\sigma_{\text{sm,in}} &= C(E)\cdot(E-\epsilon-\epsilon_1-V_k-V_\ell)\sigma_{\text{sm}}^{\text{tot}}(E),
\end{aligned}$$

where ϵ is continuous rotational energy, V_i is discrete vibrational energy with level index i and $\sigma_{\text{sm}}^{\text{tot}}$ is total scattering cross section. Note that $\sigma_{\text{sm}}^{\text{tot}}$ depends on the collision energy E as in the VHS model.

In the generalized Larssen–Borgnakke model three kinds of scattering are considered:

(i) completely elastic ($\sigma_{\text{sm,el}}$),
(ii) vibrationally elastic but maximally inelastic with respect to rotation ($\sigma_{\text{sm,ve}}$),
(iii) completely inelastic ($\sigma_{\text{sm,in}}$).

The explicit form of the factor $C(E)$ (depending on the vibrational model) is somewhat lengthy and therefore not quoted here. The parameters a and b are chosen to reproduce measured transport coefficients.

For the dissociation reaction we assume (since we do not have enough measurements) for the differential cross section a uniform probability distribution over the energy shell in phase space. This concept is widely used in high-energy physics and often successful in describing decay processes. The differential cross sections for the dissociation reactions (iv) and (v) are the following:

$$\sigma_{\text{da}}(E', \eta', \epsilon' \to E^{\text{tr}}, \omega) = \frac{1}{4\pi^2}\sigma_{\text{da}}^{\text{tot}}(E', \epsilon'),$$

$$\begin{aligned}\sigma_{\text{dm}}(E', \eta', \epsilon', V_i', \epsilon_1', V_j' &\to \epsilon, V_k, \epsilon_{tr}, \omega) \\ &= C_{\text{vib}}(E)(E - \epsilon - V_k)^2\sigma_{\text{dm}}^{\text{tot}}(E', \epsilon', V_i', \epsilon_1', V_j')\end{aligned}$$

with ('threshold cross section')

$$\sigma_{\text{da,dm}}^{\text{tot}} = \sigma^{(n)}\frac{(E' - E_{\text{B}})^n}{E_{\text{tr}}'} \cdot \Theta(E' - E_{\text{B}}),$$

where E_{B} is the binding energy of the molecule and Θ is the Heaviside function. The parameters $\sigma^{(n)}$ and n have to be chosen to reproduce the measured 'rate coefficient' in equilibrium. This means that averaging of $|u| \cdot \sigma_{\text{da}(m)}^{\text{tot}}$ over Maxwell–Boltzmann distributions should lead to a form of the rate coefficient similar to the well known 'Arrhenius law':

$$K(T) = AT^s \exp\left(\frac{-E_{\text{B}}}{K_{\text{B}}T}\right),$$

where K_{B} is Boltzmann's constant and T is temperature. The modelling becomes complicated, but is still possible to handle. We finally want to mention that – besides recombination – ionization, radiative energy transfer and soforth are not yet included and much work remains to be done. We refer the reader to Kuščer (1991) and Bärwinkel and Wolters (1975).

3. Particle Methods for the Boltzmann Equation

There are two aspects of particle methods for the Boltzmann equation: the first one is the theoretical derivation of a particle method; the second the practical aspects of implementing such a simulation scheme.

In this chapter we will discuss the first aspect starting with the definition of particle approximations. The fundamental part in the time evolution of particles is the collision integral; hence we first consider in Subsection 3.2 the homogeneous Boltzmann equation. Finally we explain how to derive particle methods for the full inhomogeneous equation.

3.1. Particle Approximations

A particle is characterized by its position x, velocity v and mass (or charge) α. In order to simplify the notation we put $\boldsymbol{p} = (x, v)$. A particle ensemble (or finite point set) is given by

$$\omega_N = \{(\alpha_1, \boldsymbol{p}_1), \ldots, (\alpha_N, \boldsymbol{p}_N)\}$$

or – in another notation – by

$$\delta_{\omega_N} = \sum_{i=1}^{N} \alpha_i \delta_{p_i}.$$

We consider sequences of particle ensembles

$$\omega_N^N = \left\{(\alpha_1^N, \boldsymbol{p}_1^N), \ldots, (\alpha_N^N, \boldsymbol{p}_N^N)\right\}$$

or

$$\delta_{\omega_N^N} = \sum_{i=1}^{N} \alpha_i^N \delta_{\boldsymbol{p}_i^N}.$$

Often $\boldsymbol{p}_i^N$ are taken from a sequence of $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots$, that is, more and more particles are brought into the game; then

$$\left\{\boldsymbol{p}_1^N, \ldots, \boldsymbol{p}_N^N\right\} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}.$$

One can in general not expect as good results for sequences of velocities as for sequences of ensembles.

Now, for a given density $f \in \mathcal{L}_+^1(\mathbb{R}^3)$ we say that '$\delta_{\omega_N^N}$ converges to f' if

$$\lim_{N\to\infty} \sum_{i=1}^{N} \alpha_i^N \varphi(\boldsymbol{p}_i^N) = \int f \cdot \varphi \, \mathrm{d}v \, \mathrm{d}x \qquad \text{for all } \varphi \in C^b(\mathbb{R}^3 \times \mathbb{R}^3).$$

This means that the discrete measure $\delta_{\omega_N^N}$ weak* converges to $f \, \mathrm{d}v \, \mathrm{d}x$.

Remark 4

(a) We may interpret this as an integration rule, where we integrate the function φ with respect to the measure $f\mathrm{d}v\mathrm{d}x$. Knots and weights depend on f, not on φ. Estimates should distinguish between a distance between ω_N^N and f and a smoothness property of φ.

(b) We should be aware that if f does not have a bounded support, we are not able to include unbounded φ such as $|v|^2$ or $|v|^2 v$ etc. So we do not get the convergence of moments we need for physical reasons (as temperature or heat transfer). This is a serious problem, which we see also numerically, if we compute the heat transfer. Some improvements in this direction may be found in Struckmeier (1994).

We would like to measure the distance between ω_N^N and f. This might be done by any distance in measure spaces (such as the Prohorov metric or bounded Lipschitz distance), but also – since the limit $f \, \mathrm{d}v \, \mathrm{d}x$ is absolutely continuous with respect to the Lebesgue measure – with the help of the 'discrepancy'. Consider an axis-parallel 'rectangle' $R \subset \mathbb{R}^3 \times \mathbb{R}^3$ and the mass of ω_N^N in R:

$$\sum_{i=1}^{N} \alpha_i^N \mathcal{X}_R(\boldsymbol{p}_i^N) \qquad \text{with } \mathcal{X}_R(P) := \begin{cases} 1 & \text{if} \qquad P \in R, \\ 0 & \text{otherwise.} \end{cases}$$

Compare it with the mass in R as given by f, that is, $\int_R f \,\mathrm{d}v$. The largest possible deviation, that is,

$$\sup_R \left| \sum_{i=1}^{N} \alpha_i^N \mathcal{X}_R(\boldsymbol{p}_i^N) - \int_R f \,\mathrm{d}v\,\mathrm{d}x \right| =: D\left(\omega_N^N, f\right)$$

is called 'discrepancy'. It is a distance between ω_N^N and f and we have

$$\delta_{\omega_N^N} \to f \quad \text{iff} \quad D\left(\omega_N^N, f\right) \to 0.$$

There are other similar definitions of discrepancy using the class of convex sets and so forth instead of rectangles – but this does not change the situation.

There are two consequences of our definition – at least for equal weights $\alpha_i^N = \frac{M}{N}$:

(a) The Koksma–Hlawka inequality:

$$\left| \int \varphi f \,\mathrm{d}v\,\mathrm{d}x - \frac{M}{N} \sum_{j=1}^{N} \varphi(v_j^N) \right| \le \mathrm{Var}[\varphi] \cdot D(\omega_N^N, f).$$

We see that in fact $\delta_{\omega_N^N} \to f$ if $D(\omega_N^N, f) \to 0$ and that it goes linear with D. The variation of φ, which we denote by $\mathrm{Var}[\varphi]$, is for one-dimensional v the usual total variation and might be substituted by $\int |\varphi'(v)| \,\mathrm{d}v$, if φ is differentiable. In dimension 3 or higher it is the so-called 'Hardy–Krause' variation, a quite lengthy concept based on the Vitali variation. One realizes that the estimate separates the distance D from the properties of the test function. For f we assume nothing more than that it is a density.

(b) We are now able to discuss an optimal speed of convergence: How fast does $D(\omega_N^N, f)$ converge to zero? Clearly, the speed depends on the definition of D and we get mainly relative information. For $f = \mathcal{X}_{[0,1]^k}(v)$, the uniform distribution in the unit cube, there are very strong number-theoretic results:

With $D(\omega_N^N) = D(\omega_N^N, \mathcal{X}_{[0,1]^k})$ one gets that there exist constants C_k, C_k' with

$$D(\omega_N^N) \le C_k \frac{\ln N^{k-1}}{N} \text{ for some } \omega_N^N$$

and

$$D(\omega_N^N) > C_k' \frac{\ln N^{\frac{k-1}{2}}}{N} \quad \text{for all } \omega_N^N.$$

Since one can construct sequences of ω_N^N, which have a convergence rate given by $\ln N^{k-1}/N$, one may say that this is 'almost optimal' today and not much can be gained in principle. The convergence is slow, but faster than $N^{-\frac{1}{2}}$, which would be the rate for random numbers. And it grows relatively slowly with the dimension k – this is the reason why particle methods are useful for higher dimensions! We shall see that for us k will be typically $2 \times 3 + 2 = 8$. We shall come back to the question of how to construct this optimal convergence order in Section 4.

Remark 5

(a) Do we gain much by using weighted particles? We have more free parameters, but realize: we want to improve $D(\omega_N^N, f)$, not $|\int f\varphi \, dv - \sum \alpha_i^N \varphi(v_i^N)|$, for a concrete φ! The only answer that is yet known is for a very simple case: Take $k = 1$ and $f = \mathcal{X}_{[0,1]}$. Then the best we can get without weights is $\frac{1}{N}$, and with weights $\frac{1}{N+1}$ – but only if $\sum_{i=1}^N \alpha_i^N = \frac{N}{N+1}$. The order of convergence is not changed in this case.

(b) If we construct ω_N^N using a sequence $(\boldsymbol{p}_j)_{j \in N}$, by just adding a new particle when moving from N to $N+1$, we loose a bit of convergence speed: Now

$$\mathcal{O}\left(\frac{\ln N^k}{N}\right)$$

is the optimal order we can achieve.

3.2. The Homogeneous Boltzmann Equation

The spatially homogeneous Boltzmann equation is given by

$$f_t(t, x, v) = \hat{I}^+(f) - f \int \int k f(t, w) \, \mathrm{d}\omega(\eta) \, \mathrm{d}w.$$

We have to discretize this equation with respect to t, putting $f_j(v) = f(j\Delta t, v)$, and we may do that either by just a simple Euler step

$$f_{j+1} = \left(1 - \Delta t \int k f_j \, \mathrm{d}\omega(\eta) \, \mathrm{d}w\right) f_j + \Delta t \int k f_j(v') f_j(w') \, \mathrm{d}\omega(\eta) \, \mathrm{d}w \quad (3.1)$$

or by integrating

$$\frac{\partial f}{\partial t} = \hat{I}^+(f_j) - f \int k f_j \, \mathrm{d}\omega(\eta) \, \mathrm{d}w \quad (3.2)$$

over $j\Delta t \leq t \leq (j+1)\Delta t$ with f_j as initial value.

For the first idea we have as a price to pay a severe restriction on Δt – but we pay it, since the second idea is computationally very expensive without a restriction on Δt. There has been no investigation yet as to whether it might be occasionally cheaper to combine both methods.

Anyhow, we proceed with the simple explicit discretization and use a weak formulation, which we get by multiplying both sides by a bounded continuous test function $\varphi \in C^b$ and integrating over v (in principle, we should realize that $f(t, \cdot)$ is a density of a measure and measures are quite natural mathematical objects for dealing with mass or charge distributions etc.; we could derive a measure formulation of any kinetic equation, which would be a natural starting point for our particle approximations, but the weak formulation is equivalent to a measure formulation). We get using $\mathrm{d}v' \, \mathrm{d}w' = \mathrm{d}v \, \mathrm{d}w$, $|v' - w'| = |v - w|$ and $v = v' - \eta < v' - w', \eta >$

$$\int \varphi(v) f_{j+1}(v) \, \mathrm{d}v = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} (K_{v,w}\varphi) f_j(v) f_j(w) \, \mathrm{d}v \, \mathrm{d}w \tag{3.3}$$

with

$$K_{v,w}\varphi = \left(1 - \Delta t \int k(|v - w|, \theta) \, \mathrm{d}\omega(\eta)\right) \varphi(v) + \Delta t \int k(|v-w|, \theta) \varphi(v') \, \mathrm{d}\omega(\eta).$$

Equation (3.1) is equivalent to (3.3), if we use $\int f_j(v) \, \mathrm{d}v = 1$, which is guaranteed by the conservation of mass. The 'transition kernel' $K_{v,w}\varphi$ is here independent of f_j – this would be different for (3.2).

We need to transform $K_{v,w}\varphi$ into a form like

$$K_{v,w}\varphi = \int \varphi(\psi(v, w, x)) \chi(x) \, \mathrm{d}x \tag{3.4}$$

with an auxiliary k-dimensional variable x, since we then get

$$\int \varphi(v) f_{j+1}(v) \, \mathrm{d}v = \int \int \int \varphi(\psi(v, w, x)) f_j(v) f_j(w) \chi(x) \, \mathrm{d}x \, \mathrm{d}v \, \mathrm{d}w$$

and we shall see that a point approximation of the $(6+k)$-dimensional density $f_j(v) f_j(v) \chi(x)$ leads immediately to an approximation of f_{j+1}. Assuming that we have such an approximation for f_j, we have to construct one for $f_j(v) f_j(w) \chi(x)$ and get the approximation for the time evolution $j \to j+1$.

The representation (3.4) is due to Babovsky (1989):
Let B be a ball in $\mathbb{R}^2$ of area 1 (radius $\frac{1}{\sqrt{\pi}}$); then we can construct a function $\phi_{v,w}: B \to S^2_+$ such that

$$\psi(v, w, x) = T_{v,w} \left(\phi_{v,w}(x)\right)$$

and χ is the characteristic function of B; here $T_{v,w}(\eta)$ is just v', that is, $T_{v,w}(\eta) = v - \eta < v - w, \eta >$. So $\phi_{v,w}(x)$ is nothing but another representation of the 'impact parameter η'. But more is hidden: the formulation includes at the end ‘dummy collisions’, that is, collisions without effect – a useful strategy (as we shall see) originally used by Nanbu (1980), Neunzert, Gropengiesser and Struckmeier (1991) and Ivanov and Rogasinsky (1988).

We shall give the construction of ϕ, since it is the basis of our simulation code: We fix v, w and take $v-w$ as polar axis in a polar coordinate system (α, β) for η, where α is the angle between η and $v-w$, that is, θ. We get

$$k(\theta)\,\mathrm{d}\eta = k(\alpha)\sin\alpha\,\mathrm{d}\alpha\,\mathrm{d}\beta.$$

Choose a function $r(\alpha)$ such that

$$r(\alpha)\frac{\mathrm{d}r}{\mathrm{d}\alpha} = \Delta t \cdot k(\alpha)\sin\alpha.$$

Since $\eta \in S_+^2$, that is, $0 \le \alpha \le \frac{\pi}{2}$, the right-hand side is positive for $\alpha > 0$ and $r(\alpha)$ is invertible with inverse $\alpha(r)$. The maximal value of $r^2(\alpha)$ is $r^2(\frac{\pi}{2}) = 2\Delta t \int\limits_0^{\frac{\pi}{2}} k(\alpha)\sin\alpha\,\mathrm{d}\alpha$.

Now the restriction for the Euler scheme comes into play. We have to guarantee nonnegativity of f_{j+1} if f_j is nonnegative; this is achieved by

$$1 - \Delta t \int\limits_{S_2^+} k(|v-w|,\theta)\,\mathrm{d}\omega(\eta) \ge 0 \qquad \text{for all } v, w,$$

that is,

$$\Delta t \int\limits_0^{2\pi}\int\limits_0^{\frac{\pi}{2}} k(\alpha)\sin\alpha\,\mathrm{d}\alpha\,\mathrm{d}\beta \le 1$$

or

$$2\Delta t \int\limits_0^{\frac{\pi}{2}} k(\alpha)\sin\alpha\,\mathrm{d}\alpha = r^2\left(\frac{\pi}{2}\right) \le \frac{1}{\pi}.$$

This is a serious restriction on Δt! With $r_{\max} = r(\frac{\pi}{2})$ we get

$$\begin{aligned}\Delta t \int k(\theta)\varphi(v')\,\mathrm{d}\omega(\eta) &= \Delta t \int \varphi(T_{v,w}(\eta))k(\theta)\,\mathrm{d}\omega(\eta)\\ &= \int_0^{2\pi}\int_0^{r_{\max}} \varphi\left(T_{v,w}(\alpha(r),\beta)\right) r\,\mathrm{d}r\,\mathrm{d}\beta\\ &= \int_{B_{r_{\max}}} \varphi\left(T_{v,w}(\phi_{v,w}(x))\right)\,\mathrm{d}^2x,\end{aligned}$$

if $\phi_{v,w}(x)$ is just the mapping $x \sim (r,\beta) \to (\alpha(r),\beta)$ ((r,β) are the polar coordinates of the point x in the ball $B_{r_{\max}}$ with radius $r_{\max}$).

We have defined $\phi_{v,w}(x)$ for $x \in B_{r_{\max}} \subset B$; this describes the case when 'real' collisions happen – v' is different from v. The other part – corresponding to $(1 - \Delta t \int k \,\mathrm{d}\omega)\varphi(v)$ – reflects the probability that no collision occurs and so we define $\phi_{v,w}(x)$ as follows.

If $x = (r\cos\beta, r\sin\beta) \notin B_{r_{\max}}$, then

$$\phi_{v,w}(x) := \left(\frac{\pi}{2}, \beta\right).$$

If $\alpha = \frac{\pi}{2}$, $v - w$ is orthogonal to η and $v' = v$! Therefore, if x is in the annulus $r_{\max} \le r \le \frac{1}{\sqrt{\pi}}$ we have dummy collisions.

$\phi_{v,w}$ is now defined for all $x \in B$ and since

$$\left(1 - \Delta t \int k(\theta)\,\mathrm{d}\omega(\eta)\right)\varphi(v) = \int\limits_{r_{\max}\le\frac{1}{\sqrt{\pi}}} \varphi\left(T_{v,w}(\phi_{v,w}(x))\right)\mathrm{d}x,$$

it does what it should do:

$$K_{v,w}\varphi = \int\limits_{B} \varphi\left(\psi(v,w,x)\right)\mathrm{d}x$$

with $\chi(x) = 1$ for all $x \in B$.

What we have to solve numerically is

$$\int\limits_{\mathbf{R}^3} \varphi(v) f_{j+1}(v)\,\mathrm{d}v = \int\limits_{\mathbf{R}^3}\int\limits_{\mathbf{R}^3}\int\limits_{B} \varphi\left(\psi(v,w,x)\right) f_j(v) f_j(w)\,\mathrm{d}x\,\mathrm{d}v\,\mathrm{d}w. \tag{3.5}$$

Assume that we have an approximation $\left\{v_1^N(j), \ldots, v_N^N(j)\right\}$ of f_j and we want to construct an approximation of f_{j+1}.

The right-hand side of (3.5) tells us what we have to do: The measure over which we integrate is

$$f_j(v) f_j(w) \mathcal{X}_B(x)\,\mathrm{d}v\,\mathrm{d}w\,\mathrm{d}x,$$

where $\mathcal{X}_B$ is the characteristic function of B. We need therefore a 'finite point set' that approximates $f_j(v) f_j(w) \mathcal{X}_B(x)$, which is an 8-dimensional density of total 'mass' $1/M^2$.

If we construct a set $\left\{\left(v_1^N(*), w_1^N(*), x_1^N\right), \ldots, \left(v_N^N(*), w_N^N(*), x_N^N\right)\right\}$ (with weights M/N) approximating this density, then

$$\frac{M}{N}\sum_{i=1}^{N} \varphi\left(\psi\left(v_i^N(*), w_i^N(*), x_i^N\right)\right)$$

approximates $\int \varphi(v) f_{j+1}(v) dv$ and

$$v_i^N(j+1) = \psi\left(v_i^N(*), w_i^N(*), x_i^N\right)$$

is an approximation of f_{j+1}!

This gives the simulation procedure and a convergence criterion:

Given an approximation $\left\{v_i^N(j), \ldots, v_N^N(j)\right\}$ of f_j, construct from that an approximation

$$\left\{\left(v_1^N(*), w_1^N(*), x_1^N\right), \ldots, \left(v_N^N(*), w_N^N(*), x_N^N\right)\right\}$$

of $f_j(v) f_j(w) \mathcal{X}_B(x)$. Then

$$v_i^N(j+1) = \psi(v_i^N(*), w_i^N(*), x_i^N), \quad i = 1, \ldots, N,$$

approximates f_{j+1}.

The main question remains: how do we get $(v_i^N(*), w_i^N(*))$? We have only $v_i^N(j)$, $i = 1, \ldots, N$, but we have a lot of freedom – the only theoretical condition is the convergence condition. We will come back to this question in Subsection 4.1.

Practically speaking, we have more conditions – it is necessary to maintain all conservation properties (mass, momentum, energy) even for the evolution in the simulation process, which means for equal weights

$$\sum_{i=1}^N v_i^N(j) = \sum_{i=1}^N v_i^N(j+1)$$

and

$$\sum_{i=1}^N \|v_i^N(j)\|^2 = \sum_{i=1}^N \|v_i^N(j+1)\|^2.$$

All practical computations show the importance of the numerical conservation of these quantities (see also Greengard and Reyna (1992)).

3.3. Particle Methods for Inhomogeneous Problems

In the previous subsection we derived a particle method for the spatially homogeneous Boltzmann equation. If we solve an inhomogeneous problem we have to take into account the spatial location of a particle.

Concerning the discretization of the inhomogeneous equation we may use

$$\begin{aligned} \bar{f}((j+1)\Delta, x, v) &= f(j\Delta t, x - \Delta t v, v), \\ \frac{\partial \bar{f}}{\partial t} &= \hat{I}(\bar{f}), \end{aligned}$$

that is, there is a decoupling of the free flow of particles and the collisions among them.

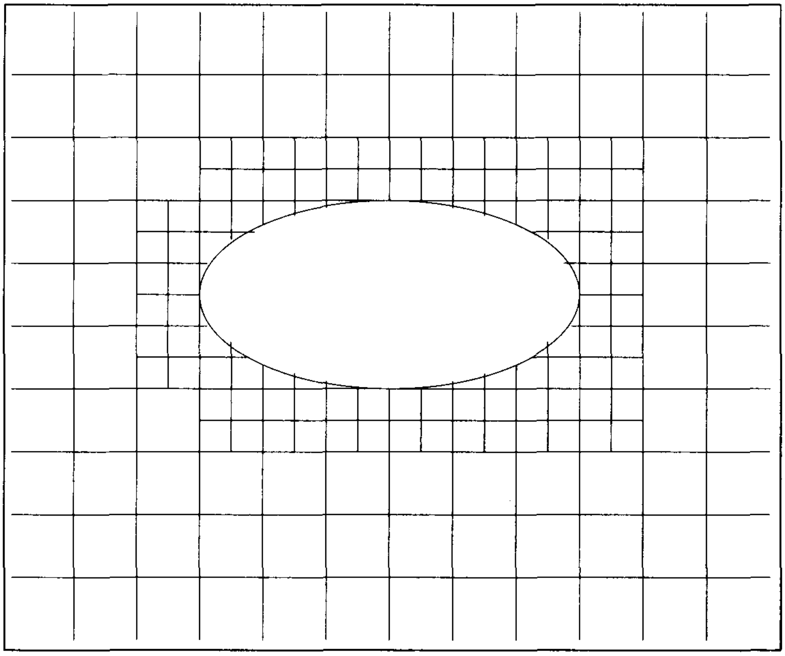

Fig. 1. Adaptive regular grid structure

Given an approximation of $f(j\Delta t, x, v)$ by a finite point set we have no problems with the first equation: we just move the particles over the time increment Δt with the particle velocity and no spatial discretization is required.

The second equation is much more complicated. Remember that $\bar{f}$ depends on x, which for finite point sets is the x-coordinate of the particle, but the collision operator is local in space.

The easiest way to get rid of the difficulties – this approach is used by nearly all methods – is

(a) to introduce a spatial cell structure, like that shown in Figure 1,

(b) to substitute $\bar{f}$ by a step function

$$\bar{f}(t,x,v) = f_{C_i}(t,v) \quad \text{if } x \in C_i$$

(c) and to consider in each cell the homogeneous Boltzmann equation and use the algorithm presented in Subsection 3.2.

Hence, only particles that are located in the same cell can form a collision pair.

Several important remarks have to be made here:

(a) One problem with this approach is that in every time step the particles have to be resampled, after the free flow, from the cell structure. If the cell structure is like in classical FEM given by triangles or tetrahedral cells this procedure requires an enormous computational effort. Hence one uses regular meshes like the one shown in Figure 1.

(b) The size of a cell has to be smaller than the local mean free path, the appropiate resolution scale. Now the local mean free path depends strongly on the local macroscopic gas density. This quantity may vary by orders of magnitude in different regions: in front of an obstacle the density may increase by a factor of 10 whereas on the lee side the density may decrease by the same factor. There are two ways to overcome this problem. The first is to use an adaptive grid structure, like in Figure 1 the second is to use different particle weights in different regions (this is discussed in Subsection 5.1).

(c) In the first part of the simulation process, the free flow of the particles, some particles may hit the spatial boundary, leave the domain or enter it. One has to take care of the corresponding boundary conditions:

- particles may leave the computational domain (absorption),
- particles may be re-emitted at a physical boundary (gas–surface interaction),
- particles may be reflected because of symmetry,
- particles may enter the spatial domain at parts of the boundary.

The gas–surface-interaction part is the most important phenomenon. The boundary condition is defined by a scattering kernel describing the velocity (respectively internal energy) change of particles hitting the surface. For monatomic gases the boundary condition is

$$|(v,n)|f(t,x,v) = \int\limits_{(v',n)<0} R(v' \longrightarrow v; t,x)|(v',n)|f(t,x,v')\,\mathrm{d}v'$$

for all times $t \in \mathbb{R}_+$, x on the spatial boundary and $(v,n) > 0$.

The classical model (for monatomic gases) is the diffuse reflection with complete thermal accommodation. Several other models, such the as Maxwell model (Cercignani, 1989), Cercignani–Lampis model (Lord, 1991) or Nocilla model (Nocilla, 1961), exist in the literature.

Different boundary models lead to different aerodynamic characteristics, whence a concrete knowledge of the real interaction law is fundamental for the description of rarefied gas flows.

4. Practical Aspects of Particle Methods

In Section 3 a description of the main idea was given. But still particle methods have enormous demands of computational time and storage. Therefore many minor tricks are needed to improve the efficiency and reliability of the method. These tricks are the treasure different groups accumulate during the development and the use of this code. In the following chapters we describe some of the details of our code.

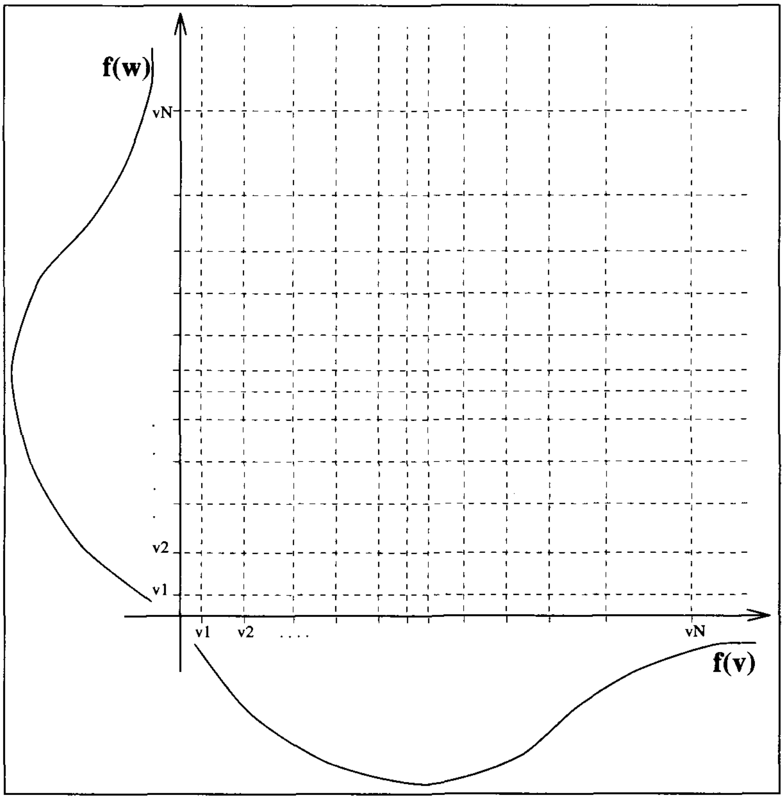

Fig. 2. Approximation of the product $f(0,v) \cdot f(0,w)$

4.1. Collision Selection and Conservation Quantities

The main part of the particle method given in the last chapter is the collision procedure which may be described as follows: Given N particles (of equal weights) at $v_1, \ldots, v_N$ (we omit the indices which are not necessary), determine N pairs $\{(v_1^*, w_1^*), \ldots (v_N^*, w_N^*)\}$ and 'impact parameters' $x_1^*, \ldots, x_N^*$ appropriately and get the new velocities by

$$\psi\left(v_1^*, w_1^*, x_1^*\right), \ldots, \psi\left(v_N^*, w_N^*, x_N^*\right).$$

There is no theoretical 'necessity' to form the pairs out of the set of particles already given – but it is quite natural. Then we have N^2 candidates for those pairs:

$$\left(v_i, v_j\right), \quad 1 \leq i, j \leq N.$$

Figure 2 gives a 1-dimensional impression. How do we select N pairs out of N^2 possible ones in order to get an approximation of $f_j(v) f_j(w)$? Denote the selected pairs by $\left(v_1, v_{j(1)}\right), \ldots, \left(v_N, v_{j(N)}\right)$.

If we have the pair $(v_i, v_{j(i)})$, we find an impact parameter x_i. $\{x_1, \ldots, x_N\}$

must approximate $\mathcal{X}_B$, that is, the uniform distribution in a ball, and they may do that independently of $\left(v_i, v_{j(i)}\right)$. This defines the new velocity $\psi\left(v_i, v_{j(i)}, x_i\right)$. So, where do we put our cross in the ith column of the (v_i, v_j)-diagram, that is, what is $j(i)$?

The first idea due to Nanbu (1980) was a stochastic one:

Select a random number r_i from a uniform distribution in $[0, 1]$ and put $j(i) = [Nr_i] + 1$; then $j(i) \in \{1, \dots, N\}$, but it might happen that two different i get the same partner $j(i)$. We distribute the crosses randomly in each column. We need to show that, for fixed velocities $\tilde{v}$, $\tilde{w}$ and $R_{\tilde{v}} \times R_{\tilde{w}} = \{(v, w) \mid v \leq \tilde{v}, w \leq \tilde{w}\}$,

$$\frac{1}{N} \sum \mathcal{X}_{R_{\tilde{v}} \times R_{\tilde{w}}} \left(v_i, v_{j(i)}\right) \rightarrow \int_{v \leq \tilde{v}} f(v)\, \mathrm{d}v \int_{w \leq \tilde{w}} f(w)\, \mathrm{d}w.$$

Using the central limit theorem, Babovsky (1989) showed that this is true for almost all sequences $(r_i)_{i \in N}$,that is, the procedure converges with probability 1.

In principle we are through – but only in principle: There are many necessary and possible improvements.

For example, in the Nanbu procedure described above, there is no conservation of total momentum or energy – this is true only 'on average'. The practical consequences were such that Nanbu's method could not compete with the so-called Direct Simulation Monte Carlo (DSMC) of Bird (1976), which we shall describe in Subsection 4.3.

Babovsky gave an improvement that does not have this drawback.

Assume that $N = 2n$. Then divide the set $\{v_1, \dots, v_N\}$ randomly into two subsets $\{v_1^1, \dots, v_n^1\}$ and $\{v_1^2, \dots, v_n^2\}$, each containing half of the particles. Now choose a permutation π of $\{1, \dots, n\}$ at random (i.e. each permutation has the same probability) and consider $\left(v_i^1, v_{\pi(i)}^2\right)$as well as $\left(v_{\pi(i)}^2, v_i^1\right)$ as pairs: we make our crosses symmetric with respect to the main diagonal. Finally, we choose the same impact parameter x_i for both pairs and get two new velocities

$$\psi\left(v_i^1, v_{\pi(i)}^2, x_i\right) \text{ and } \psi\left(v_{\pi(i)}^2, v_i^1, x_i\right).$$

This procedure keeps the idea of a binary collision and it conserves energy and momentum, since they are conserved 'pairwise'

$$v_i^1 + v_{\pi(i)}^2 = \psi\left(v_i^1, v_{\pi(i)}^2, x_i\right) + \psi\left(v_{\pi(i)}^2, v_i^1, x_i\right)$$

and similarly for the energy $\|v_i^1\|^2 + \|v_{\pi(i)}^2\|^2$.

So, symmetry guarantees these conservation laws – but only for equally weighted particles. Babovsky has also shown convergence in probability for this procedure.

If we have different weights for approximating different species in a mixture with great differences in the concentrations, then, in one cell, one might have particles with different weights α_i.

Conservation of momentum and energy in a cell would mean

$$\begin{aligned}\sum_{j=1}^{k}\int f^j(t,v)v\,\mathrm{d}v &= \text{constant},\\ \sum_{j=1}^{k}\int f^j(t,v)\|v\|^2\,\mathrm{d}v &= \text{constant},\end{aligned}$$

where $f^j(t,v)$ designates the distribution of the jth species, which is assumed to have a total mass M_j.

Approximating f^j by $M_j\alpha_j\sum_{i=1}^{N_j}\delta_{v_i^j}$, where α_j is the weight of the jth species, we would get the discrete conservation of total momentum C_M and total energy C_E

$$\sum_{j=1}^{k}M_j\alpha_j\sum_{i=1}^{N_j}v_i^j = C_M, \tag{4.1}$$

$$\sum_{j=1}^{k}M_j\alpha_j\sum_{i=1}^{N_j}\|v_i^j\|^2 = C_E. \tag{4.2}$$

If we would now consider binary collisions and try to conserve momentum and energy 'individually' in each of these binary collisions, we would fail if two particles representing different species were involved:

$$\alpha_j M_j v^j + \alpha_i M_i v^i = \alpha_j M_j v^{j'} + \alpha_i M_i v^{i'}$$

and

$$\alpha_j M_j\|v^j\|^2 + \alpha_i M_i\|v^i\|^2 = \alpha_j M_j\|v^{j'}\|^2 + \alpha_i M_i\|v^{i'}\|^2$$

are solvable only if $\alpha_i = \alpha_j$ or if no collision happens.

However, it is possible to conserve momentum and energy with weighted particles for the particle ensemble $\left\{(\alpha_1^1, v_1^1), \dots, (\alpha_k, v_{N_k}^k)\right\}$ – not 'pairwise', but by choosing the collision parameters x_i^k such that equations (4.1) and (4.2) are fulfilled for the post-collision velocities [27].

4.2. Random Numbers and the Generation of Random Variates

We discuss now the question of how much stochasticity is necessary in a particle code. What we need is to have

(a) a good approximation of the initial value $f_0(v)$ by a particle set;

(b) a selection of N pairs $(v_1, w_1), \ldots, (v_N, w_N)$ out of N^2 candidates (v_i, v_j) such that they are a good approximation of $f(v)f(w)$, if $v_1, \ldots, v_N$ is a good approximation of f;

(c) N 2-dimensional points $x_1, \ldots, x_N$ approximating $\mathcal{X}_B(x)$;

(d) for the case where there are stochastic boundary conditions (such as diffuse reflection etc.) an approximation of the distribution of the fluxes leaving the boundary.

One may use random number generators for all purposes, that is, one takes a 1D random number generator (for a uniform distribution in [0,1]), uses sections of length k to get k-dimensional points, which should be uniformly distributed in $[0, 1]^k$ and transforms them to get a sample distributed with the given density f – this is what we have to do for (a) and for (c). How we use random number generators in (b) was described in the previous pages.

The main question is: do we need the 'random property' of these generators and how should we define this property?

We give one version of stochasticity for uniformly distributed random numbers on $[0, 1]$:

If one has to construct a set of N points $x_1, \ldots, x_N$ approximating $\mathcal{X}_{[0,1]}(x)$ in an optimal way, the best solution is simply the set

$$\left\{\frac{1}{2N}, \frac{3}{2N}, \ldots, \frac{2N-1}{2N}\right\}.$$

The discrepancy is $\frac{1}{N}$ and this is optimal, but certainly not very random. We can see that by constructing 2D points from it, for example,

$$\left(\frac{1}{2N}, \frac{3}{2N}\right), \left(\frac{3}{2N}, \frac{5}{2N}\right), \ldots, \left(\frac{2N-3}{2N}, \frac{2N-1}{2N}\right);$$

all points are near the diagonal of the unit square $[0, 1]^2$ and therefore certainly not a good approximation of $\mathcal{X}_{[0,1]^2}$.

The best discrepancy we can get (for the 1D and 2D sets) is now of order $\ln N/N$. The points $x_1, \ldots, x_N$ seem now to be more stochastic – let's call them stochastic of order 1. We may realize that it is only pseudo random by looking at sections of length k: $(x_1, \ldots, x_k), (x_2, \ldots, x_{k+1}), \ldots$ and considering them as points in $[0, 1]^k$. If they are still good approximations of $\mathcal{X}_{[0,1]^k}$, we say they have stochasticity of order $k - 1$. A real random number generator should have stochasticity of order ∞ – if we use it for Monte Carlo methods in reactor physics, we need a stochasticity of very high order (the dimension is proportional to the number of collisions a neutron has with a nucleus).

In starting a simulation we should check how much stochasticity is needed – and only then can we decide how to generate our particles. For problems (a) and (c), we need just 3D or 2D approximations of $f(v)$ or $\mathcal{X}_B(x)$. We might do this by using sections of length 3 or 2 – or by other constructions, called low-discrepancy methods, which we shall describe now.

For (b) we need – for our permutation method – a stochastic separation of a $2n$-set into two n-sets and then a sequence $r_1, \dots r_N$ of [0,1] random numbers. Since the convergence proof shows convergence in probability, we need the independence of $r_1, \dots r_N$, in our language stochasticity of order $N-1$. But this is only due to our method of selecting N pairs out of N^2. We could do that completely deterministically but haven't done it yet. Why not select just one cross pattern $(i, j(i))$, which represents a uniform distribution of the crosses (one may play with introducing an index discrepancy just defined on $\{1, \dots, N\}^2$ and find an optimal $j(i)$), and apply it in each collision process? It would fulfil our convergence condition but would presumably insert a small but systematic error, which may accumulate during the evolution. This is the only risk in using as little stochasticity as possible: The fluctuations get smaller, but might be 'one-sided' and do not average out in the evolution.

Such a problem may occur in treating boundary conditions (Missmahl 1990), where one gets one-sided errors, which lead to a 'numerical cooling'. Changing the deterministic procedure just a bit, one may get rid of the effect – but this has to be done carefully.

But for (a) and (c) one may easily use low-discrepancy methods extensively described by H. Niederreiter (1992).

We want to construct point sequences (not ensemble sequences) $x_1, x_2, \dots$ such that $\omega_N = \{x_1, \dots, x_N\}$ has a low discrepancy against $\mathcal{X}_{[0,1]^k}$, that is,

$$D(\omega_N, \mathcal{X}_{[0,1]^k}) = D(\omega_N) = \mathcal{O}\left(\frac{\ln N^k}{N}\right)$$

(remember: sequences of ensembles could have $\mathcal{O}\left(\frac{\ln N^{k-1}}{N}\right)$).

For $k = 1$, we get as optimal order $\frac{\ln N}{N}$, not $\frac{1}{2N}$ as for $\left\{\frac{1}{2N}, \dots, \frac{2N-1}{2N}\right\}$, which is an ensemble sequence.

The starting point is an old idea by van der Corput, defining x_i as follows: Take the dual representation of $i = \ell_1 + \ell_2 2^1 + \cdots + \ell_m 2^{m-1}$, $\ell_k = 0, 1$, and put

$$x_i = \phi_2(i) := \ell_1 2^{-1} + \ell_2 2^{-2} + \cdots + \ell_m 2^{-m} \in [0, 1].$$

For it

$$D(\omega_N) = \frac{2}{3 \ln 2} \frac{\ln N}{N} + \mathcal{O}\left(\frac{1}{N}\right),$$

so it has optimal order. We can change the basis 2 and use any p-adic representation of i as well; this was done by Hammersley and is denoted by $\phi_p(i)$.

To get k-dimensional sequences, Halton proposed taking numbers $p_1, \ldots, p_k$ relatively prime and constructing

$$\boldsymbol{x}_i = \left(\phi_{p_1}(i), \ldots, \phi_{p_k}(i)\right), \quad i \in \mathbb{N}.$$

Here again $D(\omega_N) = \mathcal{O}\left(\frac{\ln N^k}{N}\right)$, that is, is optimal.

Please realize that we do not construct k-dimensional points by using sections of 1D sequences. Therefore we need only stochasticity 0.

There are several other methods of constructing k-dimensional low-discrepancy sequences, mainly by Faure (1982), Sobol (1969) and Niederreiter (1992). They differ in the $\mathcal{O}$-constants, which depend on the dimension k – and they may have especially low discrepancy for certain N. Since our k is never higher than 10, we do not care about it too much. There are many tests on the behaviour of different LD-sequences by G. Pagès (1992).

From a practical point of view it is fundamental to have fast algorithms for generating $\boldsymbol{x}_i$ – the algorithms should not be slower than the linear congruential methods used in normal random number generators.

A fast algorithm for a special class of low-discrepancy sequences can be found in Struckmeier (1993). It uses the p-adic Neumann–Kakutani transformations T_p: $[0,1] \to [0,1]$, which might be written as $T_p(x) = x \oplus \frac{1}{p}$ with a 'left addition $\oplus$' or as

$$T_p(x) = x + b_j^p$$

with

$$b_j^p = \frac{1}{p^j}(p + 1 - p^j)i$$

and

$$j = j(x) = \left[-\frac{\ln(1-x)}{\ln p}\right] + 1.$$

Now x_i defined by $x_i = T_p(x_{i-1})$, $x_0 \in [0,1]$ arbitrary, is a low-discrepancy sequence, called a generalized Halton sequence, and has the same optimal behaviour.

The algorithm is clear: one generates b_j^p $\forall j \in \mathbb{N}$ and then iterates as follows.

Given x_n, we compute $j(x_n)$ and then $x_{n+1} = x_n + b_{j(x_n)}^p$ (in practice it is sufficient to compute only the first 32 points of b_j^p). In k dimension, we use relatively prime numbers $p_1, \ldots, p_k$ and define the mth component, x_i^m, of $\boldsymbol{x}_i$ by

$$x_i^m = T_{p_m}\left(x_{i-1}^m\right), \; 1 \leq m \leq k.$$

Table 1. *CPU time in seconds to generate* 10^6 *numbers on* $[0,1]$

Hardware	g.H. (b=2)	LC (F77)	rand() (UNIX)
IBM 6000/530	1.9	2.8	1.6
HP 9000/835 SRX	4.8	25.8	12.9
HP 9000/710	1.0	3.1	2.0
nCUBE 2S 1 node	6.3	5.4	-

Table 2. *Discrepancy and variation of different sequences*

Sequence	D_N	V_M	D_N	V_M	D_N	V_M
Optimal	$1.72 \cdot 10^{-2}$		$5.15 \cdot 10^{-3}$		$2.89 \cdot 10^{-3}$	
rand()	$1.30 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	$7.76 \cdot 10^{-2}$	$6.7 \cdot 10^{-4}$	$6.40 \cdot 10^{-2}$	$3.0 \cdot 10^{-4}$
g.H. (b=2)	$3.97 \cdot 10^{-2}$	$7.1 \cdot 10^{-5}$	$1.25 \cdot 10^{-2}$	$5.7 \cdot 10^{-6}$	$9.71 \cdot 10^{-3}$	$8.1 \cdot 10^{-7}$
g.H. (b=3)	$3.50 \cdot 10^{-2}$	$6.1 \cdot 10^{-5}$	$1.64 \cdot 10^{-2}$	$8.1 \cdot 10^{-6}$	$8.99 \cdot 10^{-3}$	$3.6 \cdot 10^{-6}$
g.H. (b=5)	$3.43 \cdot 10^{-2}$	$6.1 \cdot 10^{-5}$	$1.57 \cdot 10^{-2}$	$1.1 \cdot 10^{-5}$	$9.63 \cdot 10^{-3}$	$2.3 \cdot 10^{-6}$
	$N = 29$	$M = 20$	$N = 97$	$M = 20$	$N = 173$	$M = 20$

This method works quite well in low dimensions, but not for very high dimensions k: then p_k becomes very large and T_p produces worse results for very large p (the $\mathcal{O}$-constant depends on p and tends to ∞ exponentially fast).

Here are some of the numerical results given in Struckmeier (1993): First the time needed to generate 10^6 numbers on different machines is given in Table 1. Then some discrepancies averaged over samples of size M – we average the discrepancy and compute the variation V_M – are given in Table 2.

Further numerical examples are given in Subsection 5.5.

Up to now, all the effort has been put into the generation of uniformly distributed sequences on $[0,1]^k$. But the densities in rarefied gas dynamics, which we want to approximate, are never constant; typical densities are, for example, Maxwellians. Therefore we have to transform uniformly distributed sequences into f-distributed ones, where f is a given density. This is easy for Maxwellians: the densities factorize, so the problem may be reduced to 1D problems. The 1D case is simple – especially since one can use the so-called Box–Muller algorithm.

If the k-dimensional density does not factorize, the problem is more complicated. Hlawka and Mück (1972) have constructed a transformation T whose inverse transforms uniformly distributed point sets into f-distributed ones. The transformation $T = (T_1, \ldots, T_k)$, which has to be inverted, has a diagonal structure

$$T_j(x_1, \ldots, x_k) = T_j(x_1, \ldots, x_j),\ j = 1, \ldots, k.$$

This can be used for a numerical inversion – an extensive study on the optimal numerical method was done by M. Hack (1993). The estimates for the discrepancy are worse in this case (Hlawka and Mück, 1972)

$$D\left(T^{-1}\omega_N^N, f\right) \leq C \cdot D\left(\omega_N^N\right)^{\frac{1}{k}},$$

but the computations show much better behaviour. Fortunately, the problems we have treated until now have not called for the construction of point sets with low discrepancy against an arbitrary f (the simulation algorithm did it).

4.3. Bird's DSMC Method

We shall now describe the DSMC version, originally developed by G. Bird, and compare it with our method.

One main difference is that the original DSMC method does not consider dummy collisions, that is, one checks whether a pair really performs a collision (i.e. if $x \in B_{r_{\max}}$). If so then we call it a 'collision pair'.

To decide whether a given pair (v_i, v_j) is a collision pair (cp), one uses an acceptance–rejection method with a parameter $V_{\max}$, which is supposed to be the maximum relative speed of all particles

$$V_{\max} = \max\left\{\|v_i - v_j\| \mid 1 \leq i, j \leq N\right\}.$$

Then a pair is a cp if a [0,1]-uniformly distributed random number r is larger than

$$\frac{\|v_i - v_j\|}{V_{\max}}.$$

In this case an impact parameter is chosen and a collision is performed. The computation of $V_{\max}$ requires N^2 operations; therefore one starts with a guess V of $V_{\max}$ and updates it if one finds a larger $\|v_i - v_j\|$. We get the following procedure:

$$\begin{tikzcd}
(v_1,\dots,v_N) \arrow[d, "\text{select}"] & \text{guess } V & \\
(v_i^1,v_j^1) \arrow[r] & \frac{\|v_i^1-v_j^1\|}{V} > r \arrow[r, "\text{yes}"] \arrow[ddl, "\text{no}"] & (v_i^1,v_j^1) = \mathrm{cp}(v_1,w_1) \\
 & & V_1 = \max\{V, \|v_1-w_1\|\} \\
(v_i^2,v_j^2) \arrow[r] & \dots &
\end{tikzcd}$$

If $\mathrm{cp}(v_1, w_1)$ is selected, we determine a time increment

$$\Delta\tau_1 = \frac{C}{N\|v_1 - w_1\|} \quad \text{(C is a gas-dependent constant).}$$

We substitute (v_i', v_j') for (v_i, v_j), that is, we update our particle ensemble after $\Delta\tau_1$, and we repeat the process until we reach Δt, that is, until

$$\Delta\tau_1 + \cdots + \Delta\tau_k \geq \Delta t.$$

(For our space-independent problem, Δt has lost its original meaning: our time step is $\Delta\tau$ and it is chosen such that only one collision happens during this interval; in this case, the time discretization is coupled with N – the time step tends to zero as N goes to ∞. In the finite point set method, N may go to ∞ without Δt tending to 0. In a space-dependent problem, Δt keeps its importance: we move the particles in space over Δt.)

For the correct procedure (with the real $V_{\max}$), Wagner (1992) has shown convergence as a stochastic process, that is, in probability. In practice, the results are sensitive to wrong initial guesses of $V_{\max}$.

The 'no time counter' version of Bird, mainly used today for computational reasons, seems similar: instead of changing time steps $\Delta\tau_i$ choose one fixed $\Delta\tau$, which is supposed to be the average time, in which one collision happens

$$\Delta\tau = \frac{C}{N \cdot V}$$

(i.e. V instead of $\|v - w\|$). V is updated at the end of Δt, not after $\Delta\tau$. The algorithm works quite well, again up to a sensitivity with respect to V.

To compare shortly our finite point set method with permutations, we have just

$$\begin{tikzcd}
v_1, \dots, v_N \arrow[rr, "\Delta t"] \arrow[dr] & & v'_1, \dots, v'_N \\
& \text{pairs } (v_1, w_1), \dots, (v_N, w_N) \arrow[ur, "x_1, \dots, x_N"'] &
\end{tikzcd}$$

Δt is restricted by

$$1 - \Delta t \int_{S_2^+} k(\|v - w\|, \theta) \, \mathrm{d}\omega(\eta) \geq 0$$

for all (possible) v, w!

Finally, we may also do updating during the collision process: We perform each collision immediately, that is, substitute (v'_i, v'_j) for (v_i, v_j) after $\Delta t/N$. The difference is that we keep N collisions (including the dummy ones) and work with the small timestep $\Delta t/N$, but do not need a guess of $V_{\max}$.

The differences in computing time are less than 10 per cent, the results are demonstrated by the examples given in Subsection 5.5.

5. Some Ideas on How to Improve and Extend the Code

In this last section we shall report on some ideas on how to improve the code, to accelerate the algorithm and to extend it to more realistic situations. These topics will be:

5.1 'Different weights for particles in different regions'.
This is different from 'different weights for different species' and does not create the same problem of conserving energy and momentum when particles of different weights collide. There is a detailed study of it by Schreiner (1991).

5.2 'The use of symmetry in particle codes'.
If point sets are considered in a physical way – as representations of real particle sets – it is not easy to take advantage of geometrical symmetries of the problem (and the solution). To do that we have to exploit the idea of approximation by discrete measure; for example, if the density has cylindrical symmetry, depending only on x_1, v_1, $\|\tilde{x}\|$, $\|\tilde{v}\|$ and $< \tilde{x}, \tilde{v} >$ (where $\tilde{x} = (x_2, x_3)$, $\tilde{v} = (v_2, v_3)$), then our measures will be measures in this 5D (instead of 6D) space. One can save a lot of computing time, as is shown by Struckmeier and Steiner (1993).

5.3 'Matching' of kinetic equations with diffusion or aerodynamic limits.
This must be promising: each kinetic equation has some singular limits ('Diffusion approximation', Euler or Navier–Stokes equation etc.), which hold at least in some parts of position space. Solving these simpler equations in these parts and matching the solutions with those of the kinetic equations, which one gets in the 'kinetic rest' of the domain, poses a new problem in domain decomposition. There have been attempts in this direction – see, for example, Illner and Neunzert (1993), Bourgat et al. (1994) and Klar (1994).

5.4 'Efficiency on massively parallel systems'.
During the past few years several authors have investigated the performance of a particle method on massively parallel systems (see e.g. Barteland Plimpton (1992), Dagum (1991), Struckmeier and Pfreundt (1993) and Wong and Long (1992)). In this section we will follow the approach given in Struckmeier and Pfreundt (1993).

5.1. Spatially Weighted Particles

'Different weights in different regions, but equal weights in each cell' is an easily solvable weighting problem. In Schreiner (1991) the author describes how to find an appropriate particle mass in each cell in position space and how to change our particles (by splitting them or pasting them: Splipa) so that each particle has this desired mass.

Clearly, this desired mass m^* has to be small if the density in a cell is small (e.g. behind a space vehicle) – and it has to be large if the density is high (in the bow shock). In this way one may control the number of particles in each cell. During the free flow, particles of different masses may enter the same cell – but since we want to perform collisions only with particles of the same mass, we have to homogenize them. We allow only integer values for particle masses and we assume that m^* is always of the form 2^j; therefore homogenization might be done by splitting particles of mass 2^{j+k} into 2^k particles of mass m^* or by pasting minor particles together (by first splitting them into particles of minimal mass and then unifying them two at a time again and again until they have grown enough). The only problem here is that one should do this in such a way that mass, momentum and energy are conserved in each Splipa procedure; in particular the velocities after pasting have to be chosen carefully and there are only some signs to be chosen freely.

One might save time by these ideas. For a 2D problem (flow around an ellipse), Schreiner (1991) used 25 or 64 particles per cell in the beginning; the simulation without any weighting is then called A25 or A64, and with 3 or 4 different weights we call it B25-3 or B25-4, respectively (see Table 3).

Table 3. *CPU times and number of particles in the stationary state*

	CPU	Partnr
A64	44'41"	706,000
A25	24'52"	275,000
B25-3	29'13"	334,000
B25-4	26'17"	248,000

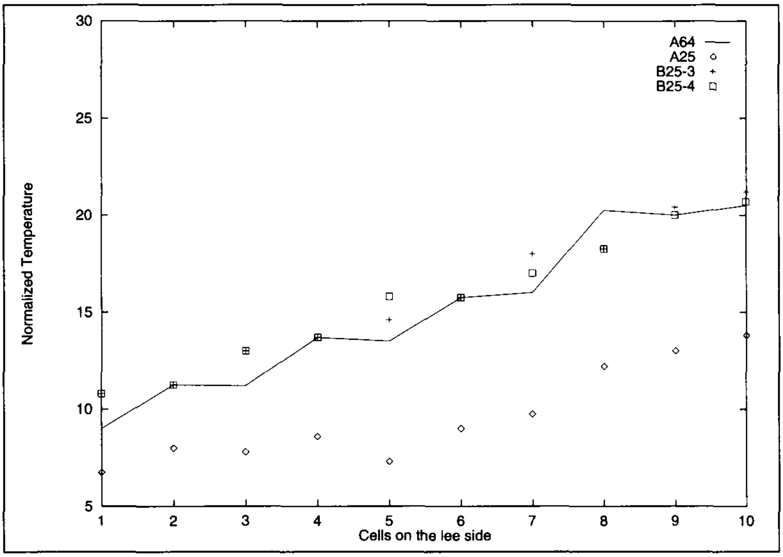

Fig. 3. Temperature along line in flow direction

The results differ – at least the temperature (it is a second moment) shows big changes from A25 to B25-3 behind the vehicle (see Figure 3). So, it is cheap and rewarding to use this weighting. But we want to recall that weights for different species, where homogenization is not possible, create much bigger problems.

5.2. Simulations with Axisymmetric Geometry

Symmetry reduces dimension in any numerical method, but normally not for particle methods. The reason is the usual one: particles are considered as physical quantities, not as approximations of densities.

Assume that we have cylindrical symmetry, that is, the boundary has a rotational symmetry with respect to the x-axis. Introducing cylindrical coordinates means to substitute (x, r, φ) for (x, y, z) and (v_x, v_r, v_φ) for (v_x, v_y, v_z).

Since v_r, v_φ depend on φ, we get a more complicated free-streaming term and have to transform $\hat{I}(f)$ to cylindrical coordinates (which was done in Niclot (1987)). A new collision strategy has to be defined – this way seems to be too elaborate. We may use (x, r, φ) together with (v_x, v_y, v_z) – things will not fit completely, but some important aspects remain unchanged. We get

$$\frac{\partial F}{\partial t} + v_x \frac{\partial F}{\partial x} + (\cos\varphi v_y + \sin\varphi v_z)\frac{\partial F}{\partial r} + \frac{-\sin\varphi v_y + \cos\varphi v_z}{r}\frac{\partial F}{\partial \varphi} = \hat{I}(F).$$

$\hat{I}(F)$ is not changed here. Free streaming means solving $\dot{x} = v_x$, $\dot{r} = (\cos\varphi v_y + \sin\varphi v_z)$, $\dot{\varphi}= (-\sin\varphi v_y + \cos\varphi v_z)/r$ with initial values (x_0, r_0, φ_0). The solution is

$$\begin{aligned} T_x(t, x_0, r_0, \varphi_0, v) &= x_0 + t v_x, \\ T_r(t, x_0, r_0, \varphi_0, v) &= \left(r_0^2 + 2tr_0(\cos\varphi_0 v_y + \sin\varphi_0 v_z) + t^2(v_y^2 + v_z^2)\right)^{1/2}, \\ T_\varphi(t, x_0, r_0, \varphi_0, v) &= \arctan\left(\frac{r_0 \sin\varphi_0 + t v_z}{r_0 \cos\varphi_0 + t v_y}\right). \end{aligned}$$

For $\hat{I}(f) = 0$ we get $F(t, x, r, \varphi, v) = F_0(T(-t, x, r, \varphi), v)$.

Now we define $G = r^{-1}F$ and consider the corresponding equation. If, for example, F is a uniform distribution in position space with respect to the Lebesgue measure (in polar coordinates $rdrd\varphi dx$), then G can be regarded as a uniform distribution with respect to the 'cartesian' measure $\mathrm{d}r\,\mathrm{d}\varphi\,\mathrm{d}x$, since $G\, r\, \mathrm{d}r\,\mathrm{d}\varphi\,\mathrm{d}x = F\ \mathrm{d}r\,\mathrm{d}\varphi\,\mathrm{d}x$.

To be more flexible, we consider

$$G(t, x, r, \varphi, v) = R(r)F(t, x, r, \varphi, v).$$

The equation for G is similar to that for F, but it has on the left-hand side an additional term $-(\cos\varphi v_y + \sin\varphi v_z)\partial_r(\ln R)g$ and instead of $\hat{I}(F)$ we have $R^{-1}\hat{I}(G)$. This additional term changes the solution of the free-streaming part into

$$G(t, x, r, \varphi, v) = \frac{R(T_r(-t))}{R(r)} G_0(T(-t), v)$$

and the factor $R(T_r(-t))/R(r)$ may be handled as a weight: a particle, moving from $P_i = (x_i, r_i, \varphi_i, v_i)$ to $P_i(\Delta t) = (T(\Delta t, P_i), v_i)$ changes its weight in proportion to $R(r_i)/R(r_i(\Delta t))$. For the natural choice $R(r) = r^{-1}$ the particles become heavier in moving away from the axis – the number of particles in a ring of thickness Δr remains unchanged (since the mass in a ring $(i-1)\Delta r \leq r \leq i\Delta r$ grows linearly with i, the weight of a particle has to grow linearly with i too in order to keep the particle numbers constant).

But now we have particles of different weights in the same cells – something we wanted to avoid. Even in the beginning, $R(r) = r^{-1}$ would give

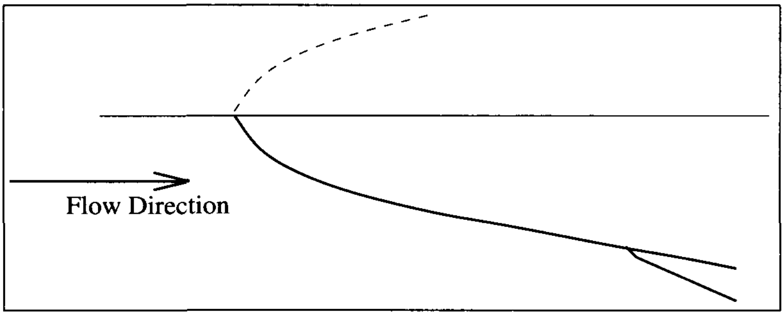

Fig. 4. Geometry of the hyperboloid flare

different weights. Therefore R is chosen as a step function approximating r^{-1}; but still differently weighted particles may enter a ring. Besides homogenization described under (b) one may follow a general idea by Bird: if the weight changes by a factor α less than one, just keep the particle with its old weight but with a survival probability of α. If α is larger than 1, say $\alpha = m + \alpha'$, $m \in \mathbb{N}$, $0 \leq \alpha' < 1$, create m new particles of the same weight and one other with probability α'. Again such a strategy does not work if we have different species of gas, but is successful here. No rigorous proof is yet available.

This reduces the computational costs drastically. Struckmeier and Steiner (1993) have done a study for HERMES with a flap at the leading edge (see Figure 4). Some results are shown in Table 4.

5.3. Domain Decomposition Techniques

We believe that the most promising prospect practically as well as theoretically is to use kinetic equations only where one is forced to use them – and to use the appropriate limits wherever it is possible. This idea materializes in two questions:

(a) The 'where' problem: What are the regions where the diffusion limit or the Euler equation is valid, but in their complements the kinetic equations are necessary.

(b) The other problem is the 'how' problem: how do we patch or match the solution of the kinetic equation with those of the limits.

Kinetic equations deal with position-velocity densities and the limits with macroscopic quantities, which can be interpreted as some moments of the kinetic density: What kind of boundary conditions for the two types of equations are the 'correct' ones? (Assuming the kinetic solution everywhere is the truth, which boundary conditions at the transition give a 'combined solution' as near as possible to the truth?) Until now, only the continuity of the macroscopic quantities across the transition boundary has been tried

Table 4. *Numerical parameters and global surface quantities*

Altitude[km]	Gas	T_∞[K]	Ma	T_w[K]	λ_∞[m]
120	N_2	368	20	1400	2.69
110	N_2	247	23	1400	0.60
100	N_2	194	25	1400	0.137
Altitude[km]	Partnr	Cellnr	Part/Cell	Timesteps	CPU[h]
120	570,000	11,264	64	1000	1.5
110	925,000	11,264	64	1000	2.5
100	2,000,000	40,960	36	1000	4.0
Altitude[km]	$C_{d,0^\circ}$	$C_{l,0^\circ}$	$(L/D)_{0^\circ}$	$C_{h,0^\circ}$	$C_{m,0^\circ}$
120	2.191	.890	.406	.868	.882
110	1.688	1.048	.621	.539	.641
100	1.360	1.170	.860	.313	.490
Altitude[km]	$C_{d,12^\circ}$	$C_{l,12^\circ}$	$(L/D)_{12^\circ}$	$C_{h,12^\circ}$	$C_{m,12^\circ}$
120	2.304	.941	.408	.901	.974
110	1.785	1.109	.621	.557	.727
100	1.461	1.246	.853	.325	.584
Coefficient	Drag	Lift	Lift/Drag	Heat	Pitching

to be realized; details are described in Lukshin, Neunzert and Struckmeier (1992).

Since we focus on collisions, we just want to stress one comparing the simulation of collisions with the solution of an Euler equation (we choose Euler since it is – as a singular limit – much better understood than Navier–Stokes). The Boltzmann equation is solved by a particle code that moves the particles in a free flow over Δt and then treats the collisions at the end of the time step. The Euler equation can be solved by a very similar procedure: move particles in a free flow over Δt, but then redistribute them according to a Maxwellian distribution whose moments are given by particles at the end of a time step. The Euler equation gives a time evolution that is a free flow with a constraint: stay on the manifold given by $\{f : \hat{I}(f) = 0\}$. The ordinary free flow, starting at this manifold, moves away – so we have to project back onto it; this is the redistribution (see figure 5). So, the difference

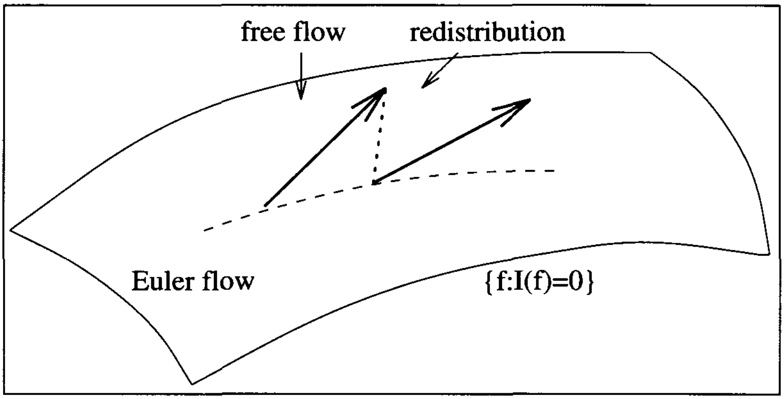

Fig. 5. Redistribution of the kinetic density

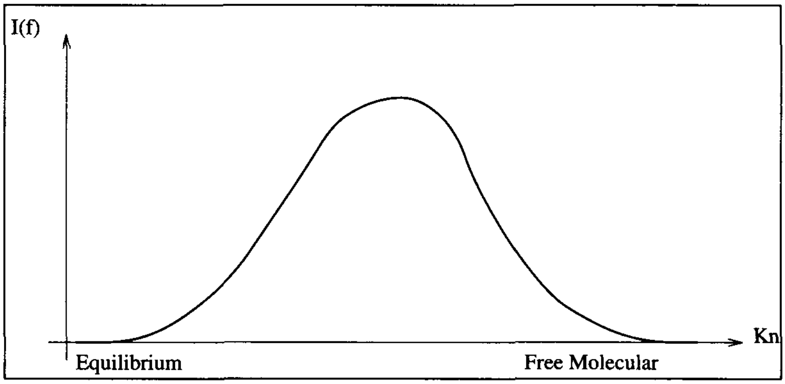

Fig. 6. Influence of $\hat{I}(f)$ in dependence on the Knudsen number

between the Boltzmann and Euler equations is the difference between the collision procedure and the projection. The projection is numerically much cheaper – so do projection whenever it is possible and collisions when it is necessary. What we try to use here is the fact that $\hat{I}(f)$ becomes small when f becomes very rarefied – and, when f becomes very dense and near to a Maxwellian, frequent collisions create an equilibrium distribution f for which $\hat{I}(f) = 0$. The denser f is the more expensive the collision procedure becomes – but at the same time, the smaller $\hat{I}(f)$ becomes (see Figure 6). To avoid this effect, we may use these projections. The key words here are 'kinetic schemes' and they may be converted into particle schemes for the Euler equations – see Schreiner (1994).

Using kinetic schemes, matching of the two codes is a minor problem: just do projection or collisions cellwise, but otherwise move freely without caring where you are.

5.4. Particle Methods on Parallel Computers

Still, realistic problems need enormous computational effort. Therefore it is reasonable to investigate the performance of Boltzmann simulation codes on massively parallel systems.

A parallelization of the code refers mainly to the grid structure on the spatial domain; the cells are, for example, cubes with a length smaller than the mean free path of the unperturbed gas. The collision process in a cell is independent of those in other cells – and it is the most time consuming part. One parallelizes the code by assigning a certain number of cells to each processor. In general one would have much more cells than processors, whence it is necessary to include a communication procedures.

In the first part of the time-iteration process particles may leave cells and enter others – if these cells belong to different processors, this means communication between processors.

The partition of cells has to be done such that this communication, that is, the number of particles crossing processor boundaries, is minimized. But a static partition, fixed at the beginning of the computation according to a priori information about the flow fields (most of the particles move essentially with the stream velocity), does not produce a good load balance of the processors – particle numbers per processor change and result in a very insufficient load balance; this reduces the speed-up factor as compared to single processors.

To get an adaptive procedure, we put cells lying in a row with respect to the main stream velocity together to form 'spatial sticks'. Several spatial sticks are assigned to processors – and the adaption consists simply of exchanging sticks from the minimally to the maximally loaded processors. This exchange creates an iteration procedure until we get near to the partition when the numbers of particles in the processor domains are near the average number. The procedure creates partitions where the local character of the stick–processor assignment is destroyed (see Table 5).

One may suspect that this gives rise to a high communication time; that this is not the case is shown in Figure 7.

The speed-up factor is constant near 30 (here the factor 32 is – using 32 processors – optimal) if one compares different Knudsen numbers, that is, different densities and therefore a different collision frequency (see Figure 8). A comparison of CPU times on the nCUBE2s with a VP100 shows that the higher peak performance of the vector machine does not lead to lower CPU time (Table 6).

5.5. Numerical Examples

First we come back to the comparison of the finite point set method and the DSMC method of Bird (see Subsection 4.3).

Table 5. *Final state of the adaptive processor partition (zy-plane)*

6	27	25	4	5	8	2	11	18	4	3	18
23	19	14	25	15	2	17	8	32	27	2	5
18	27	14	22	3	29	4	8	24	29	16	22
30	4	4	24	21	22	14	16	10	9	11	15
6	30	20	10	2	24	3	23	8	21	29	26
15	26	9	9	13	16	8	27	1	7	30	3
21	12	22	5	7	30	30	6	23	26	6	2
23	31	23	18	13	7	22	10	12	3	29	27
32	29	10	13	31	23	21	14	18	26	13	14
26	9	7	2	10	9	24	29	1	16	17	18
8	21	24	30	19	7	15	30	12	13	6	12
11	3	24	30	14	12	25	4	25	22	31	6
21	16	18	1	29	15	1	20	13	4	28	26
28	25	29	3	2	5	19	20	23	2	28	32
1	5	18	29	1	11	26	20	32	21	28	22
7	15	28	20	28	17	17	14	6	5	28	25
25	24	5	23	16	8	17	10	21	9	17	4
13	12	20	14	6	19	16	14	30	20	31	17
2	11	11	9	8	1	29	25	5	9	30	30
10	19	4	9	29	3	7	25	18	1	10	30
12	32	22	17	1	20	15	5	22	19	30	30
28	26	28	10	12	11	11	15	15	6	27	19
3	8	21	27	27	16	24	19	23	20	12	30
7	16	13	7	17	26	11	19	32	13	27	24

We consider the flow around a hyperboloid flare (see Figure 4) at high Mach number and an altitude around 100 km. Here one may use the axisymmetric version of the particle codes described in Subsection 5.2. We calculate 'global' quantities acting on the body such as the drag, lift or heat-transfer coefficient.

The main task is to investigate the sensitivity of the different approaches to the number of particles used in the simulation.

Looking at Figures 9 and 10, we realize that, except for the DSMC time-counter version, both methods, DSMC as well as the finite point set method, show nearly the same behaviour with respect to the particle number. A lot of more results can be found in Struckmeier and Steiner (1993).

The high degree of modelling necessary to describe real-gas effects requires a validation of the models used in a particle method. The following example has become a 'classic' test-case, mainly due to the establishment of the European Hypersonic Data Base (EHDB).

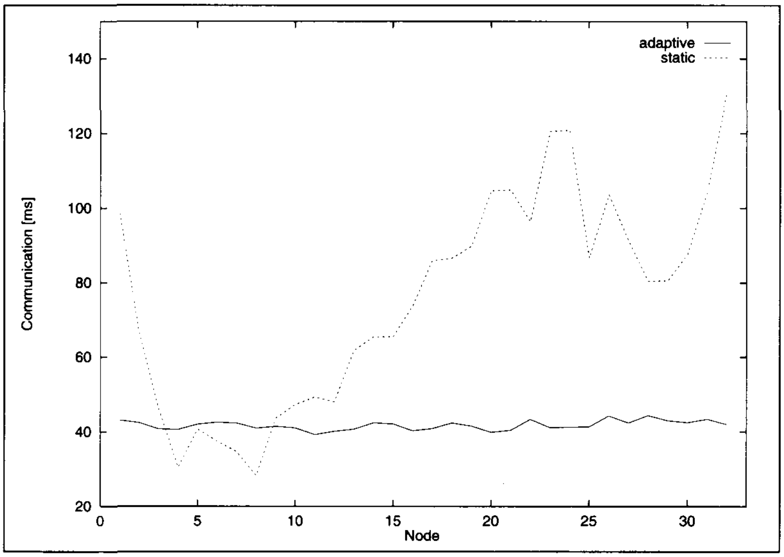

Fig. 7. Communication time vs. nodes at Kn = 0.5 and 45° angle of attack

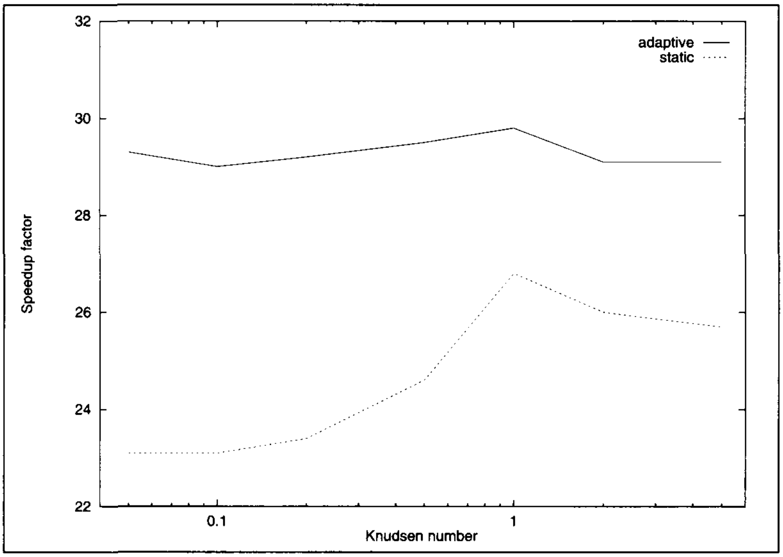

Fig. 8. Speed-up factor vs. Knudsen number at 45° angle of attack

We consider the flow of nitrogen gas around a 3D delta-wing at a high Mach number. The measured quantities are global surface quantities like the drag or the heat-transfer coefficient. Figure 11 shows the drag coefficient versus the Knudsen number at Mach 20.2 and 45° angle of attack, Figure 12 the heat-transfer coefficient.

Table 6. *CPU times for a 3D computation with $Kn = 0.5$ and 45° angle of attack*

	MFLOP	CPU[s]	ratio
nCUBE2s/8	35	579	3.7
nCUBE2s/16	70	297	1.9
nCUBE2s/32	140	156	1.0
Fujitsu VP100	285	1075	6.9

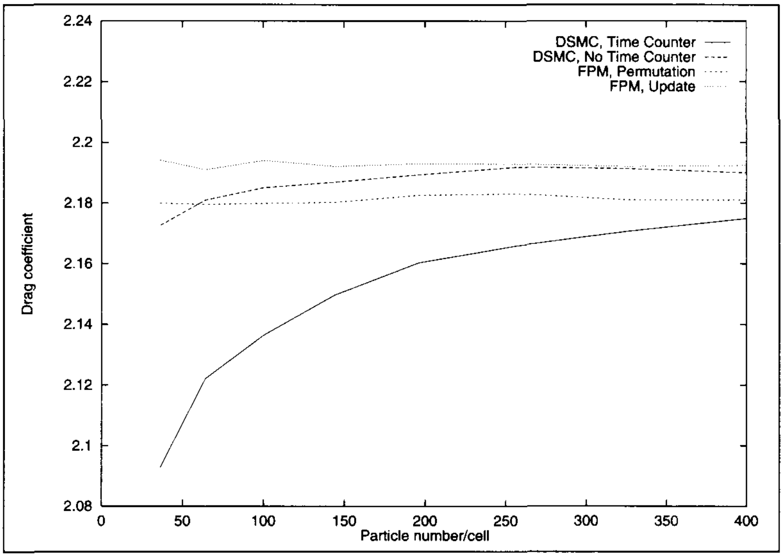

Fig. 9. Drag coefficient versus particle number

The agreement between numerical results and measurements is quite convincing; hence, the model for the exchange of different energy types like rotational and vibrational energies (in this case the Larsen–Borgnakke model) is accurate enough to reproduce the physical situation.

As a last example (see Figure 13) we compare the local pressure distribution along the surface line calculated by a particle method with the prediction given by the modified Newton theory. We consider again the hyperboloid flare (Figure 4) with flap angle of 0° respectively 12° at Mach 25 and an altitude of 100 km. The given altitude corresponds to the 'small' Knudsen number of $9 \cdot 10^{-3}$; hence, one may expect that the modified Newton theory gives reasonably accurate results.

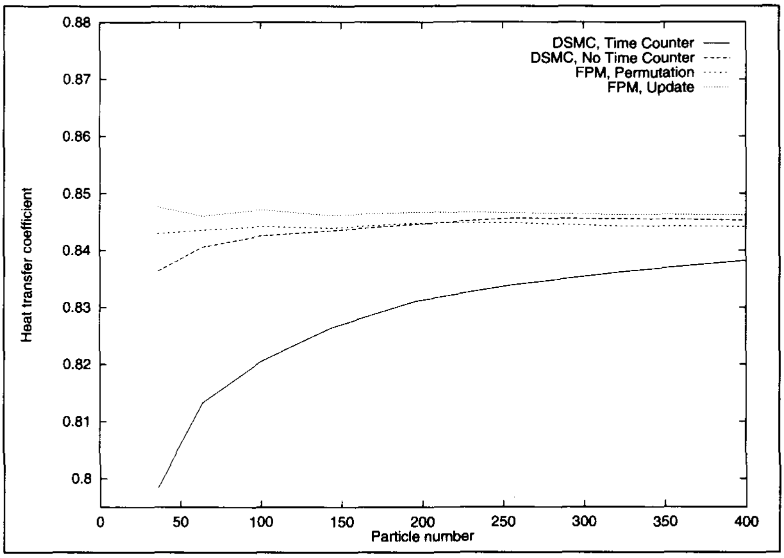

Fig. 10. Heat-transfer coefficient versus particle number

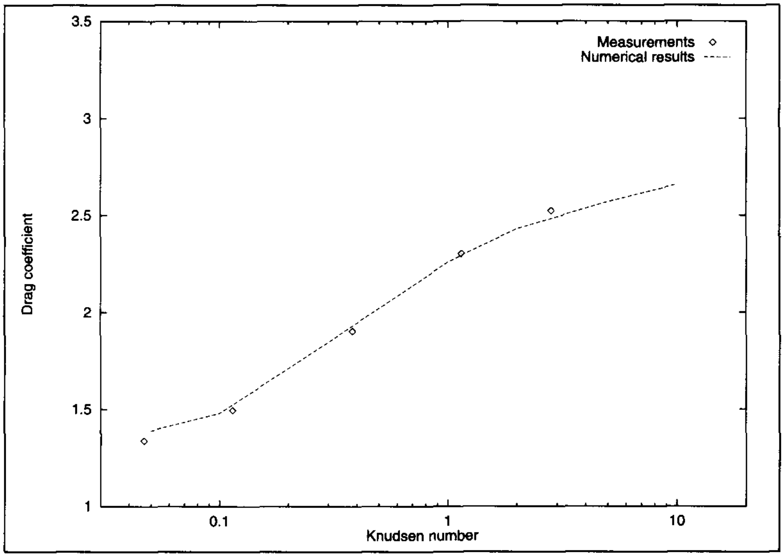

Fig. 11. Drag coefficient versus Knudsen number

6. Final Remarks

We believe that particle methods have become a reliable instrument in rarefied gas dynamics. Using massively parallel systems one can treat realistic problems with a reasonable effort. However, some physical effects like ionization or recombination are still neglected (or handled in an unreliable way); therefore further improvements are needed. The most promising ansatz is

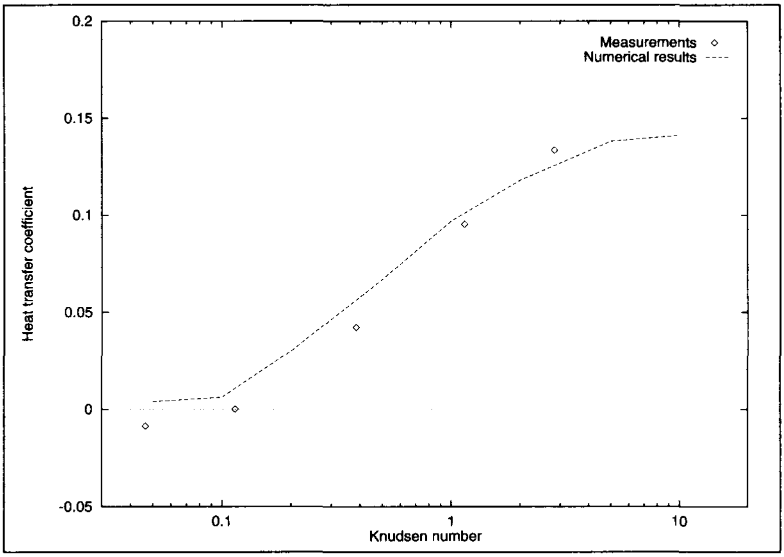

Fig. 12. Heat-transfer coefficient versus Knudsen number

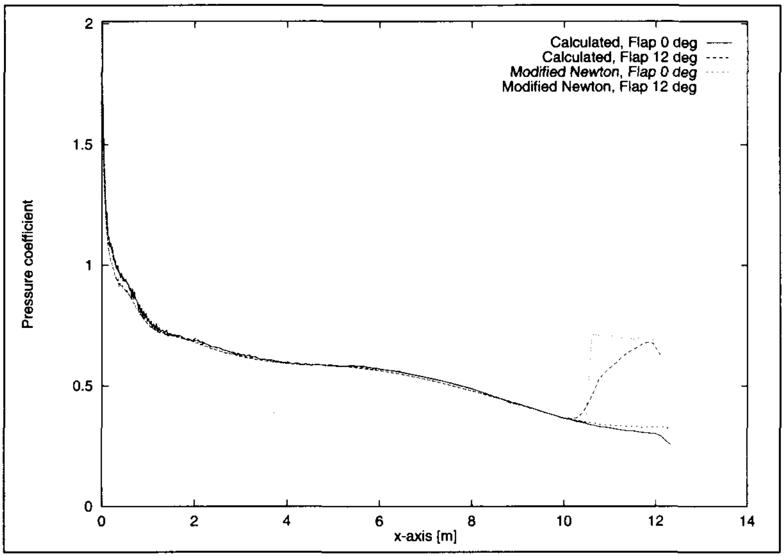

Fig. 13. Local pressure coefficient along the surface line

the combination of asymptotic analysis with numerical methods (as in many fields!); we will solve the simpler 'singular limit equations' whenever it is possible – and the more complicated kinetic equations when it is necessary.

Why are particle methods not just finite difference methods or something similar? Kinetic equations are high-dimensional. Approximations of densities by discrete measures are more robust with respect to dimensions. This might be the reason that competing methods until now have failed in treating 3D problems. Why are there still some stochastic elements and why is there still a lot of 'Monte Carlo'? The explanation for this may be given by the theory of information-based complexity; there are many hints that this theory provides ideas to answer the question of why Monte Carlo – correctly applied – is advantageous.

There are still gaps between existence theory and numerics; but the theory cannot provide us with uniqueness which is what we need to bridge the gap. There is still a long and exciting way to go.

REFERENCES

H. Babovsky (1989), 'Convergence proof for Nanbu's Boltzmann simulation scheme', *Europ. J. of Mech. B/Fluids* **1**, 41–55.

T. J. Bartel and S. J. Plimpton (1992), DSMC Simulation of Rarefied Gas Dynamics on a Large Hypercube Supercomputer, Paper–92–2860, Am. Inst. Aero. Astro., Washington.

K. Bärwinkel and H. Wolters (1975), Reaktionskinetik und Transport in relaxierenden Gasen, Report BMF-FB W W 75–28, Dornier, Friedrichshafen.

G. A. Bird (1976), *Molecular Gas Dynamics*, Clarendon Press, Oxford.

G. A. Bird (1989), 'Perception of Numerical Methods in Rarefied Gas Dynamics', *Progr. Astro. and Aero.* **118**, 211–226.

A. V. Bobylev (1993), 'The Boltzmann Equation and the Group Transformations', *Math. Models & Methods in Appl. Sciences* **3**, No. 4, 443–476.

C. Borgnakke and P. S. Larssen (1975), 'Statistical collision model for Monte Carlo simulation of polyatomic gas mixtures', *J. Comput. Phys.* **18**, 405–420.

J. F. Bourgat, P. LeTallec, B. Perthame and Y. Qiu (1994), 'Coupling Boltzmann and Euler Equations without Overlapping', *Contemporary Mathematics* **157**, 377–398.

C. Cercignani (1989), *The Boltzmann Equation and Its Applications*, Spinger, Berlin.

L. Dagum (1991), Three Dimensional Direct Particle Simulation on the Connection Machine, Paper–91–1365, Am. Inst. Aero. Astro., Washington.

H. Faure (1982), 'Discrepance des suites associées à une système de numération (en dimension s)', *Acta Arithmeticae* **41**, 337–351.

C. Greengard and L. G. Reyna (1992), 'Conservation of expected momentum and energy in Monte Carlo particle simulation', *Phys. Fluids A* **4**, 849–852.

M. Hack (1993), 'Construction of Particlesets to Simulate Rarefied Gases', Report No. 89, Lab. Technomathematics, University of Kaiserslautern.

J. H. Halton (1960), 'On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals', *Numer. Math.* **2**, 84–90.

J. M. Hammersley (1960), 'Monte Carlo methods for solving multivariate problems', *Ann. New York Acad. Sci.* **86**, 844–874.

E. Hlawka and R. Mück (1972), A transformation of equidistributed sequences, Appl. of Number Theory to Numerical Analysis, ed. S. K. Zaremba, Academic Press, New York, 371–388

R. Illner and H. Neunzert (1993), 'Domain Decomposition: Linking of Aerodynamic and Kinetic Descriptions', Report No. 90, Lab. Technomathematics, University of Kaiserslautern.

M. S. Ivanov and S. V. Rogasinsky (1988), 'Analysis of numerical techniques of the direct simulation Monte Carlo method in the rarefied gas dynamics', *Sov. J. Numer. Anal. Math. Modelling* **3**, No. 6, 453–465.

A. Klar (1994), 'Domain Decomposition for Kinetic and Aerodynamic Equations', PhD thesis, Dept. Math., University of Kaiserslautern.

I. Kuščer (1991), 'Dissociation and Recombination in an Inhomogeneous Gas', *Physica A* **176**, 542–556.

G. Lord (1991), 'Some extensions to the Cercignani–Lampis gas–surface scattering kernel', *Phys. Fluids A* **4**, 706–710.

G. Ludwig and M. Heil (1960), 'Boundary-layer theory with dissociation and ionization', in *Advances of Applied Mechanics* , vol. 6, Academic Press, New York.

A. Lukshin, H. Neunzert and J. Struckmeier (1992), 'Interim Report for the Project DPH 6473/91: Coupling of Navier–Stokes and Boltzmann Regions', Internal Report, Dept. Math., University of Kaiserslautern.

G. Missmahl (1990), 'Randwertprobleme bei der Boltzmann–Simulation', Diploma thesis, Dept. Math., University of Kaiserslautern.

K. Nanbu (1980), 'Direct Simulation Scheme Derived from the Boltzmann Equation', *J. Phys. Japan* **49**, 2042–2049.

H. Neunzert, F. Gropengiesser and J. Struckmeier (1991), 'Computational methods for the Boltzmann equation', in *Venice 1989: The State of Art in Appl. and Ind. Math.* (ed.), R. Spigler, Kluwer, Dordrecht, 111–140.

H. Neunzert, K. Steiner and J. Wick (1993), 'Entwicklung und Validierung eines Partikelverfahrens zur Berechnung von Strömungen um Raumfahrzeuge im Bereich verdünnter ionisierter Gase', Report DFG-FB Ne 269/8–1, Lab. Technomathematics, University of Kaiserslautern.

B. Niclot (1987), The Two Particle Boltzmann Collision Operator in Axisymmetric Geometry, Report No. 164, Centre de Mathèmatiques Appliquées, Ecole Polytechnique, Palaiseau.

H. Niederreiter (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia.

S. Nocilla (1961), 'On the Interactions between Stream and Body in Free-molecule Flow', *Proc. 2nd Int. Symp. on Rarefied Gas Mechanics*, (L. Talbot, ed.), Academic Press, New York.

G. Pagés (1992), 'Van der Corput sequences, Kakutani transform and one-dimensional numerical integration', *J. Comp. & Appl. Math.* **44**, 21–39.

K. Sobol (1969), *Multidimensional quadrature formulae and Haar functions*, Nauka, Moscow.

M. Schreiner (1991), 'Weighted Particles in the Finite Pointset Method', Report No. 62, Lab. Technomathematics, University of Kaiserslautern.

W. Schreiner (1994), Partikelverfahren für kinetische Schemata zu den Eulergleichungen, PhD thesis, Dept. Math., University of Kaiserslautern.

J. Struckmeier (1993), 'Fast Generation of Low-Discrepancy Sequences', Report No. 93, Lab. Technomathematics, University of Kaiserslautern, to appear in *J. of Comp. & Appl. Math.*

J. Struckmeier (1994), 'Die Methode der finiten Punktmengen: Neue Ideen und Anregungen', PhD thesis, Dept. Math., University of Kaiserslautern.

J. Struckmeier and F. J. Pfreundt (1993), 'On the efficiency of simulation methods for the Boltzmann equation on parallel computers', *Parallel Computing* **19**, 103–119.

J. Struckmeier and K. Steiner (1993), 'A Comparison of Simulation Methods for the Boltzmann Equation', Report No. 91, Lab. Technomathematics, University of Kaiserslautern, submitted to *J. Comp. Physics.*

J. G. Van der Corput (1935), 'Verteilungsfunktionen I,II', *Nederl. Akad. Wetensch. Proc. Ser. B* **38**, 813–821; 1058–1066.

W. Wagner (1992), 'A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation', *J. Stat. Phys.* **66**, 689–722.

B. C. Wong and L. N. Long (1992), Direct Simulation Monte Carlo (DSMC) on the Connection Machine, Paper–92–0564, Am. Inst. Aero. Astro., Washington.