COMPUTER SIMULATIONS OF TWO-DIMENSIONAL GRANULAR FLOW PAST A CIRCULAR OBSTACLE

by

IGOR V. BARYSHEV

A dissertation submitted to the Graduate Faculty in Physics in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2003

This manuscript has been read and accepted for the Graduate Faculty in Physics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/17/03
Date

Chair of Examining Committee Jan Kopik

1/21/03
Date

Executive Officer

Hernán A. Makse

Mark D. Shattuck

Gabriel I. Tardos

M. Silvina Tomassone

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

COMPUTER SIMULATIONS OF TWO-DIMENSIONAL GRANULAR FLOW PAST A CIRCULAR OBSTACLE

by

Igor V. Baryshev

Adviser: Professor Joel Koplik

I present event-driven simulations of the interaction of a dense stream (volume fraction 0.65) of hard smooth disks with a circular obstacle. Both "slip" and "no-slip" boundary conditions at the obstacle are studied. First, a reference "elastic" case is investigated for Mach numbers from 0.02 to 0.3 (corresponding to Reynolds numbers from 3 to 46). In simulations of slower flows, the temperature and density are nearly homogeneous. In the case of no-slip boundary conditions, the flow patterns show the presence of eddy-like structures formed and shed irregularly and asymmetrically during the run. For slip boundary conditions, the corresponding simulations show a "waving tail" pattern behind the obstacle. For faster flows, in the no-slip case the patterns change to high-intensity regularly-shed swirls, likely caused by large temperature and density variations. In the slip case, these swirls are present only at the initial stages of the flow development.

Introduction of inelasticity into the system leads to very fast cooling, and the flow patterns show a transition to a shock development phase. It is shown that the friction-like (no-slip) interaction between particles and the

obstacle is essential for the formation of a "static dune", while the absence of friction between the particles themselves did not prevent the dune from forming. To investigate an inelastic case in the presence of an energy source compensating a loss of fluctuational energy due to inelastic collisions, an adaptation of a well-known stochastic thermostat model is used to establish a regime which avoids the suppression of flow patterns. It is shown that the granular medium driven in this way can exhibit flow patterns similar to those observed in non-driven systems of elastic particles, i.e., eddy-like structures and a waving tail, with no-slip and slip boundary conditions, respectively.

Acknowledgments

I would like to thank my adviser, Joel Koplik, for the opportunity to work on this project under his supervision. I have learned a great deal from him. I would also like to thank the members of my thesis committee, Hernán A. Makse, Mark D. Shattuck, Gabriel I. Tardos, and M. Silvina Tomassone for taking the time to read my thesis and for all their important and insightful comments.

Mark D. Shattuck helped me become a better writer by his careful and thorough comments. His help in estimating the speed of sound in the simulated systems was indispensable. He also has provided many enlightening and interesting conversations and contributed to the thesis through his interest and extensive knowledge on the subject.

I am also grateful to German Drazer for many informative discussions and suggestions. In many cases, his knowledge and insight have opened the road to faster and more efficient solutions.

I would also like to thank Andrew Eng. He was always putting in extra effort and time to maximize the performance of our computer systems.

Thank you all.

Contents

List of Figures

Chapter 1

Introduction

1.1 Outline

Granular materials play a very important role in such industries as agriculture and pharmaceutics as well as mining and construction. They exist in nature in various forms such as powders, sands or even planetary rings. Huge amounts of materials in granular form are obtained, transported and then stored for further use on the daily basis. Even when materials are intended to be used in some "classic" (liquid, solid, etc.) form, very often they are processed into pellets, tablets, capsules, powders, etc. for storage or transportation purposes. At later stages, two or more different kinds of granular materials are often to be mixed for further processing. Granular mixing and de-mixing at the industrial level are of immense importance. One of the problems is maximizing the performance of huge industrial mixers. Another is providing better uniformity of the component distribution in the final prod-

uct. These are just few topics related to the necessity of studying granular flow patterns, especially the ones occurring in the presence of physical boundaries such as walls and obstacles. Since granular materials present such a rich plethora of interesting phenomena, academic attention to the problem is also not surprising.

If the material is dry then the effects of interstitial media (usually air) can often be neglected, for example, if individual grains composing the material are relatively heavy. Since granular matter is a large collection of solid particles, one way to study such an ensemble is to use methods similar to those used for studying and describing regular gases and fluids, i.e., methods of statistical mechanics and thermodynamics. Because the number of particles in a granular material can be very large, the possibility of using a continuum description is of great importance. The task is complicated by the fact that granular materials can exhibit a wide range of behavior. For the purpose of this discussion, it is convenient to characterize the granular material phenomena as belonging to the following groups:

1. Solid-like behavior, e.g. the description of the statics of heaps and silos.

2. Flow phenomena:

   (a) Slowly flowing granular material where the majority of interactions between the grains occurs through multiple contacts of relatively long duration (not collisions).

   (b) Granular gas state in which most interactions can be approximated as binary collisions between grains.

The first category covers topics which are relatively far from the present discussion (with maybe the exception of the phenomena of "static dunes" which could form in the obstructed flow in front of an obstacle [ABK01]). A lot of information concerning the static problems can be found in a special review dedicated to this class of phenomena [dG99].

The division of the *flow* phenomena into the two groups above is, of course, oversimplified. More strict classification can be made using shear stress vs. shear rate relations as the appropriate criteria (e.g., see [TM00]). Since the main discussion will primarily deal with the last category (gas-like state), other flow phenomena will be discussed only briefly, and the simple classification given above will suffice.

1.2 Interactions: multiple contacts

Systems in which the interaction of particles with each other involves multiple contacts, and the forces exerted on the particles are determined by the friction and deformations of the contact area, can be studied using soft particle molecular dynamics (MD, e.g., see [AT87]) computer simulations, also called DEM or discrete element method. Makse and Kurchan [MK02] studied simple shear flow using this technique and concluded that their "results strongly support the thermodynamic description". It is important to note that strong shearing of the system can lead to the appearance of regions where the interaction between grains would predominantly have a collisional (binary) character. This can be seen near the shearing walls in the 2D simula-

tions of Couette flow performed in [KH99]. The authors studied both binary mixture and monodisperse systems and measured the heat dissipation rate.

Experimental studies (3D) of Couette flow in densely packed material can be found in [MDK+00]. Using magnetic resonance imaging (MRI), the authors were able to obtain "flow velocities from the interior of a 3D system" and to show that "key characteristics of the granular microstructure determine the shape of the velocity profile". The torque on the inner cylinder in a Couette device was measured in [TKS97].

The important issue of the interaction of the flow with an obstacle was addressed in [APBS99] in which the drag force on a cylinder slowly moving through a dense granular medium was measured. The authors showed that in this limit the drag is independent of velocity of the media but "quadratically dependent on the depth of insertion". The authors also showed that analytically the quadratic dependence stems from the fact that "the grains in all layers must simultaneously meet the criteria to reorganize".

1.3 Interactions: binary collisions

Provided there is a sufficient energy input into the system (shearing, shaking, sliding in the presence of gravity, etc.), a granular system can be driven into a state where most interactions are binary collisions between grains. Even after the energy source is removed, the regular binary collisions might dominate in such freely developing system for some time. Because of the type of particle interactions in these systems, it would be natural to try to use a continuum

description similar to the one for regular gases (one of the classic sources is [CC70]).

One of the most important steps in this direction was done in [JR85] where a kinetic theory for "dense gas of identical, rough, inelastic, circular disks" was suggested. The authors used the description of a collision similar to the one given below (section 3.4) with the exception that the formal tangential restitution coefficient e_{tl} was considered to be constant.

On the other hand, many works indicate a difficulty in applying hydrodynamic approach to some granular systems. A study of a one dimensional system of grains driven at one end was done in [DLK95]. Using numerical simulations, the authors show that the system evolves into the state which "clearly violates equipartition of energy". A very similar geometry (one side driving) was used for a 2D system in [GZBN97]. Nevertheless, the generalization of "hydrodynamic treatments to situations where high and low density regions coexist" showed good agreement with event-driven (ED) simulations of the system (ED methods will be discussed in detail later).

Event driven simulations of granular systems lacking sufficient energy input had to deal with the issue of the so-called inelastic collapse. In [MY94] and [MY96] the authors performed 2D simulations of a freely evolving system, and showed that while the energy is dissipated from the system through collisions, clusters of particles were beginning to form, and eventually a state of collapse, characterized by the infinite number of collisions in a finite time, occurs within these clusters.

One of the ways to avoid the collapse "singularity" was suggested in

[LM98]. The authors introduce a cutoff time, and if for a given particle the time from one collision to the next is less than the cutoff, then the last collision should be treated as elastic. Another choice is based on a more realistic dependence of the normal restitution coefficient on velocity, i.e., the coefficient approaches unity for small normal velocities (e.g. see [GSB+98] and [BSSS99]).

It is important to note that the collapse state does not necessarily imply the absence of energy input in the system. A driven granular monolayer of stainless steel balls [OU98] is cooled (reduced driving) through different stages starting from a gas into a clustering regime and finally to the formation of stationary clusters. Similar transitions were studied using ED simulations in [EP97] in which the authors suggested a corresponding "granular phase diagram".

During the initial stages of granular material cooling before clusters form, the system is still in a homogeneous state. A detailed study [LHMZ98] of this "homogeneous cooling" process, using both ED simulations (2D and 3D) and kinetic theory, shows that while for short times the system energy decreases linearly with time, for large times the dependence is inverse quadratic (t^{-2}).

Flow past an obstacle is a traditional test case in fluid mechanics (e.g. see the classic source [Bat67]). Since the applicability of the continuum approach to the granular materials is of such great interest, it was only natural that researchers attempted to investigate similar geometries. For example, two works ([ABK01] and [APBS99]) were already mentioned above (pages 3 and 4). Using soft particle MD, Buchholtz and Pöschel [BP98] study

a 2D rapidly flowing stream of granular particles interacting with a fixed obstacle. They find that for high stream velocities, the force on the obstacle depends quadratically on the the stream velocity. In [RBSS02] the authors investigate fast "supersonic" flows using experiments, ED simulations, and Navier-Stokes-like continuum equations. The term "supersonic" is used in a way similar to the one for regular gases, i.e., it refers to the types of flows in which macro (i.e., the appropriate space averages) velocities are much larger then thermal velocities of particles. Specifically, it does not mean that the granular particles travel faster than sound in air. The velocity profiles for the shock formed around the wedge shaped obstacle show good agreement between different descriptions.

One of the most important features of granular materials is the inelastic nature of interactions between the grains. During an inelastic interaction, a portion of the kinetic energy of particles is converted to heat and dissipated. The problem of interparticle collisions was investigated at various levels. One of the most detailed theoretical approaches can be found in [BSHP96]. It is based on an analysis of viscoelastic deformations during the impact and creating the appropriate model describing the process of collision in terms of particle material properties such as the Young modulus and the Poisson ratio. Moreover, the connection can be made between two mechanical processes: rolling on a plane and interparticle collision (see [BP99]). The results can be used for soft particle simulations discussed above.

For hard sphere simulations, a different approach is used. The collision itself is considered to be instantaneous and is described by several coeffi-

cients relating pre-collision and post-collision velocities. The details on the models involved and derivations can be found e.g. in [FLCA94], [Lud95], [Lud98] and references therein (see also section 3.4). It is important to say that these derivations are based on the approximation that the two types of contact (slip with friction and no-slip) do not coexist during a collision. This assumption, as well as an assumption that the restitution coefficients can be considered constant for certain range of velocities, was tested, for example, in experiments on measuring the coefficients e, e_t, and μ (normal restitution coefficient, tangential restitution coefficient, and friction coefficient correspondingly) for soda lime glass spheres and cellulose acetate spheres [FLCA94]. The agreement of the experiments and the approximation in question was found to be satisfactory for a tested range of relative contact velocities.

Of course, there are many more important and complicated issues related to the topic of granular materials, discussion of which would lead too far from the particular subject at hand. More information on progress in studying the world of granular materials can be found, e.g. in comprehensive and interesting reviews [JNB96] and [dG99] and references therein.

Chapter 2

Algorithm and code

2.1 General approach

In contrast to "soft sphere" simulations where the whole system is being moved forward in time by a number of fixed time steps small enough to "resolve" the interaction between particles defined by some kind of a smooth potential, hard sphere systems are simulated by using the "event driven" (ED) approach. In this case the system develops in time from one collision (event) in the system to the next collision and so on. Generally, events should be understood in a very broad sense and not just as collisions. In principle, depending on the model and algorithm used in the simulation, any change of state (collision, crossing a boundary, creation or destruction of a particle, etc.) in a given system might need to be treated as an event. Collisions themselves, as a special type of event involving two particles (or more generally - objects), might also be of a nature very different from "regular"

billiard like collisions. For example, two particles, tied to each other by a fixed length link, will experience a "collision" whenever they try to part from each other for more than the length of the link. The general nature of events in the discrete-event systems allows simulation of various phenomena including navigation and transport processes. Games and combat models are other examples of the use of event driven algorithms.

The time until the next collision in the whole system can have in principal any value depending on the system. For example, if we consider a system composed of two or more isolated exact replicas of a single subsystem, then obviously there will be two or more collisions occurring simultaneously in the replicas. In this case the time from one collision to another will be zero until all simultaneous collisions are processed. Although this example is of course artificial, simultaneous (within numerical precision) collisions do occur in simulations of relatively large systems. The other extreme is an example of two or more particles moving in parallel to each other. Since in this case no collisions occur, no events are possible, and the next event time effectively becomes undefined unless crossing certain boundaries is treated as an event.

The basic straightforward algorithm for simulation of hard sphere systems is described, for example, in [AT87] which can be summarized by the following general scheme:

(a) Get the next event:

[(b) move all objects forward to the event time point;]

(c) implement the event and make all the necessary updates;

Steps (a), (b) and (c) imply taking into account existence of all boundaries in the configuration (walls, periodic, etc.). Although bracketed step (b) makes this scheme very intuitive and similar to a fixed time step procedure for "soft" potential molecular dynamics (MD), an efficient ED algorithm should use a different approach which will be discussed later.

In the very beginning (time point t_0) each particle i can be tested against each other particle j for a possible collision. Thus, for a fixed index i we go through all indices j and calculate time until collision t_{ij}. The quantities of greatest importance though are τ_i (the minimum collision time for particle i), which is initially made equal to some big number playing role of infinity, and p_i (partner of i) which is initially made equal to some non-existent index, e.g. -1. If for current index j the time t_{ij} is less than τ_i, then t_{ij} is stored by overriding the previous value of τ_i, i.e., $\tau_i = t_{ij}$ and $p_i = j$. Otherwise t_{ij} is thrown away. At the end of going through indices j, the variable τ_i will be equal to the smallest of all times, corresponding to possible collisions between the particle i and all other particles j, and variable p_i will be equal to the index of a particle j which is a partner of particle i in the collision with time τ_i. This time τ_i can now be stored in some sort of a table (timetable) which in a simplest case could be just an array with elements indexed by particle indices i. The partner particle index p_i should be stored too. Then the same procedure should be repeated for the next particle i. For clarity it was assumed here that for each fixed index i all other indices j should be checked for possible collision. That of course leads to checking each pair of particles in the system twice: (i, j) and (j, i). One can avoid this as given in [AT87]

by simply starting index j from $i+1$. In this case t_{ij} should be checked not only against τ_i but also against τ_j which is available from a timetable as an entry with an index j. On the other hand, the overhead caused by double checking of each pair during this timetable initialization is simply negligible compared to the duration of the simulation itself.

In case when particles are hard spheres moving freely (with no external forces) between collisions, the times t_{ij} are readily determined based on the following [AT87]. If at time t two spheres, i and j, with diameters d_i and d_j have positions $\vec{r}_i$, $\vec{r}_j$ and velocities $\vec{v}_i$, $\vec{v}_j$, then at time $t+t_{ij}$ the distance between the centers of these particles is

$$|\vec{r}_i + \vec{v}_i t_{ij} - (\vec{r}_j + \vec{v}_j t_{ij})| = |\vec{r}_{ij} + \vec{v}_{ij} t_{ij}| \,. \tag{2.1}$$

where $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ and $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$. If two particles are in collision this distance is $(d_i/2 + d_j/2) = \sigma$, or

$$|\vec{r}_{ij} + \vec{v}_{ij} t_{ij}| = \sigma \,. \tag{2.2}$$

And the square of it gives

$$v_{ij}^2 t_{ij}^2 + 2(\vec{r}_{ij} \cdot \vec{v}_{ij}) t_{ij} + r_{ij}^2 - \sigma^2 = 0 \,. \tag{2.3}$$

Before trying to find the roots t_{ij} of this quadratic equation, it is useful to check the sign of $(\vec{r}_{ij} \cdot \vec{v}_{ij})$. When the particles are moving apart, $(\vec{r}_{ij} \cdot \vec{v}_{ij})$ is greater than zero, and further calculation steps for the pair of particles (i, j) can be aborted. The next check is the sign of the discriminant:

$$D = (\vec{r}_{ij} \cdot \vec{v}_{ij})^2 - v_{ij}^2 \left(r_{ij}^2 - \sigma^2 \right) \,. \tag{2.4}$$

If $D < 0$, the equation (2.3) for t_{ij} does not have real roots and particles miss each other. If $D > 0$, there are two positive roots. The bigger one corresponds to a "parting point" which would occur if the particles went through each other without a collision. The smaller one,

$$t_{ij} = \frac{-(\vec{r}_{ij} \cdot \vec{v}_{ij}) - \sqrt{D}}{v_{ij}^2}. \tag{2.5}$$

corresponds to the collision and should be checked against τ_i and τ_j which were until now the smallest collision times for particles i and j.

It is easy to expand the result for freely moving particles to the system where particles move in a uniform external force field, e.g. gravity. Obviously, the terms $(gt^2/2)$ would cancel each other in (2.2) leaving the final result for t_{ij} unchanged. Thus, although in the presence of gravity particles move along parabolas, the time between particle collisions is exactly the same as if they were moving freely along straight lines. Of course, this result is valid only for the case when both particles in question are in "free fall" in the external field. For example, if one of the particles is fixed and plays a role of an obstacle, then (2.2) will lead to a quartic equation which still has an algebraic solution although it takes careful coding to implement numerical realizations of appropriate rather complicated formulas for the roots.

After initializing the timetable containing the collision times τ_i and corresponding partners p_i, the search for τ_{min} (the minimum of all τ_i's in the timetable) is done. The actual search procedure depends on the implementation of the timetable and will be discussed later. Then the system can be moved to a next time point $t + \tau_{min}$ (at this stage $t = t_0$ simulation start up

time) implementing step (b) as

$$\vec{r}_l\,(t+\tau_{min}) = \vec{r}_l + \vec{v}_l\tau_{min}\,. \tag{2.6}$$

In a simple algorithm discussed for now, index l goes through all the particles in the system. The times τ_i in the timetable should also be decreased by τ_{min} since each molecule moved closer to the designated partner. Then the collision of particles (I and its partner p_I) corresponding to τ_{min} is performed changing the variables of particles I and p_I (such as $\vec{r}, \vec{v}$, etc.) according to the collision rule which depends on the model being simulated. Before restarting the whole cycle and coming back to the step (a), the changes in the timetable caused by the current event must be taken into account.

At this point there is no need to reconstruct the whole timetable from scratch since only part of it is invalidated by the most recent collision. The actual entries which are invalidated are the entries of particles which participated in the collision (particles I and p_I) and all other particles for which either particle I or particle p_I is a partner. The group of all these particles will be called "affected" particles. Locating all the affected particles can be done, in principle, by going through all the system particles indexed by l and checking for the condition:

$$(l == I)\;||\;(l == p_I)\;||\;(p_l == I)\;||\;(p_l == p_I)\,. \tag{2.7}$$

where p_l denotes partner of particle l, "||" means logical "OR", "==" means logical "equal".

Every affected particle found is then checked against all other particles to get new collision times and partners which are to be put in the timetable.

After the timetable is revalidated, the search for the next collision time in the timetable can be done and the simulation cycle repeats.

Since limiting the revalidation of the timetable to only "affected" particles is a valuable optimization not just for the scheme just presented but also for further improvements to be discussed, it is important to notice the following. The simple and intuitive way of finding and updating "affected" particles is using an **if** condition similar to 2.7 as it is given in the discussion in [AT87]. But let's assume that, when going through particles indexed by l, the first "affected" particle encountered is particle $l = I$, i.e. one of the collided particles. Then updating this particle usually will establish new particle-partner relationships resulting in particles for which particle I is a partner. For this newly "generated" partners the condition 2.7 will be satisfied too and then they will have to be updated just as the actual "affected" particles. Of course, the other collided particle p_I might cause the same effect. To avoid this kind of false "affected" particles, it is important to update all the "partners" first and only then the "main parties" I and p_I. To go closer to the actual code developed and used in the present work, the main parties (objects which are directly responsible for the current event) will be called p1, p2. These variables (pointers) are actually referring to some data of a particle grouped in a structure, and to access any of these data, the following C-language notation will be used: `p->data`. For example, the coordinates x, y, z of a particle "p" are: `p->r[0]`, `p->r[1]`, `p->r[2]` (C array indices start from zero). The partner of a particle "p" is `p->partnr` and its partner is `p->partnr->partnr` and so on. The code for updating the "affected"

particles is similar to

```
if( ((p->partnr ==p1)||(p->partnr ==p2)) && p!=p1 && p!=p2 )
{
  ptcl_update(p);
}
ptcl_update(p1);
ptcl_update(p2);
```

Here `ptcl_update(p)` is a function call to perform the actual update procedure for a particle "p", and variable "p" goes through all the particles which could be affected (for now it is all the particles in the system) by the current event.

The algorithm which was discussed in the present section provides only the basic framework and understanding of the main steps involved in any ED simulation. Without optimizations which are going to be discussed below this algorithm should be used only for systems with a small number of particles.

2.2 Cells

Even if we assume that all the "affected" particles are available to us for an immediate update, the computational cost of checking one of the "affected" particles against all other particles in the system is $O(N)$ per event. It can be reduced to $O(1)$ by splitting simulation domain into cells of a size more than the diameter of a largest particle in a system [Rap95]. At the beginning

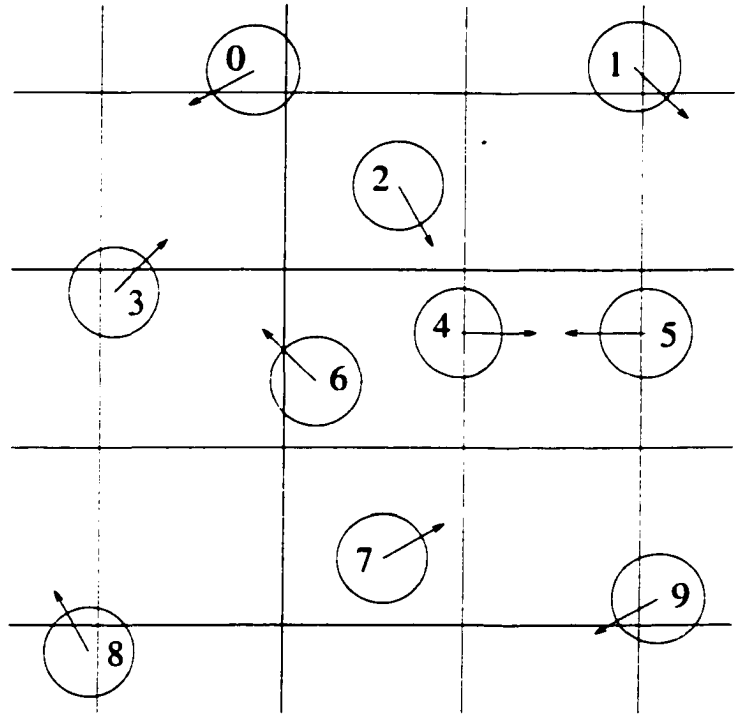

Figure 2.1: Part of the simulation domain and cells

of a simulation each particle is associated with its cell based on particle's position and velocity. Since the number of particles in cells is not constant through the run, the linked list [Knu68] data structure is usually used for placing and removing particles in cells. Now when searching for the possible partners of a particle belonging to a given cell "c", one need to consider only particles in a block of 9 ($3^{\mathcal{D}}$, $\mathcal{D}$ is a system dimension) cells the center of which is cell "c" (Fig. 2.1). This block will be called the neighborhood of cell "c" or of any particle belonging to cell "c". For example, the only candidates for partners of particles **6** and **4** belonging to the central cell in Fig. 2.1 will

be particles **2**, **3** and **7**. Crossing any of the boundaries of a cell is now a separate event. Particles **4** and **5** are on their way to a head on collision, but since the size of the cell is larger than the sum of their radii, the collision between them can occur only after one of them crosses their cell boundary. This collision event is prevented by cell boundary crossing event and need not be considered. This is a manifestation of the fact that particle **5** does not belong to the neighborhood of particle **4** and thus would be excluded from an update procedure for particle **4** at present time.

It is also clear that the search for "affected" particles needs to be done only among the particles belonging to overlapping neighborhoods of particles participating in an event. If the event particles belong to the same cell, the two neighborhoods coincide, otherwise the overlap is partial. Thus, the total cost of the update procedure is $O(1)$ per collision and it is determined by the number of particles in a neighborhood.

It is easy to see that the cell approach for ED simulations is very similar to the algorithm used in soft particle simulations [AT87]. In both cases the idea is to localize the update procedure by splitting the simulation domain into cells, but for soft particles the size of the cells is determined by the cutoff distance of interaction potential. The hard sphere model deals with discontinuous, singular potential, but it also has a cutoff equal to the diameter of a hard particle and this diameter determines the minimum size of the cells.

The actual optimal size of the cells depends on the mean free path in the system. If the cells are too small compared to the mean free path, then particles cross cell boundaries often which increases the cost. On the other

hand. an increase in the size of cells leads to rather fast (to the $\mathcal{D}$ power) growth in number of particles in a cell and the associated with this cost. It is very easy to find the optimal size experimentally making few runs for small systems using different cell sizes. For example, for the system of monodisperse particles with volume fraction 0.65, the optimal value of a cell size is about 1.2 diameter of a particle. The situation changes if the system of particles is moving as a whole with an average (stream) velocity which is rather high compared to thermal velocities of particles. In this case the particles will often cross cell boundaries between collisions, but usually the time scale of interest for the whole run is also much smaller since the stream velocity is high, and hence the cell crossing overhead is of no concern. If it is not the case and optimal performance is still needed, it is relatively straightforward to implement the cell grid which is moving with constant velocity along with the stream. Placing and updating particles will basically remain the same since all these operations can be performed the same way in a reference frame moving with a stream, and in that frame the cell grid is at rest. Of course, there still be a difficulty caused by objects (boundaries, obstacles) which are at rest in the lab frame. If these objects just as regular particles are also considered as local to certain cells, then special care would need to be taken to make sure they always belong to the appropriate cells.

2.3 Moving particles forward

Although the idea of bringing all the particles with their positions and velocities up to-date whenever an event occurs is very intuitive, the cost associated with this approach is $O(N)$ per event. Besides since the time between consecutive events in large systems is very small (sometimes zero within numerical precision), the incremental changes in positions and velocities of particles would introduce abnormal numerical errors. To address this issue, the following approach was suggested [Lub91]. Each object is given a data field 'time' corresponding to the time of the last event this particular object participated in. Initially all 'time' fields are set to the current simulation time point. When the next event is processed, only the few particles participating in the event are brought up to-date including their 'time' which becomes equal to the current time in the system. All other objects continue "living in their past" until an event involving them occurs. The updating procedure would involve solving the equation (2.2) for particles with parameters corresponding to two different times in the past. For this purpose the synchronization of parameters (positions, velocities, etc.) can be done by either calculating the "older" particle parameters at the "newer" particle time (which would involve the cost of an **if** statement needed to determine which one is "newer") or by simply calculating parameters of both particles at present ("newest") time. The cost for the latter is not doubled compared to the cost for a single particle because of the nature of the update procedure itself. In a function `ptcl_update(p)` particle "p" is fixed and synchroniza-

tion part is done for it only once. Then particle "p" is checked against every other object in the neighborhood of "p", and synchronization for those objects is done independently of "p". This latter approach was used in the present work. Once parameters of particles in question are calculated at the same time point, the rest of the procedure of solving (2.2) is exactly the same.

Since all particles in the system are effectively at different times, the `p->coltim` (collision time) data field should not be used as a sorting parameter for the timetable. The collision time (`p->coltim`) gives time until the next event for a particle "p" counted from particle time (`p->time`). Hence an absolute time (`p->time + p->coltim`) of the next event of a particle "p" is appropriate as a sorting parameter for the timetable.

As a result of this optimization only few particles directly involved in the current event are moved forward in time which brings the cost of this step down to $O(1)$.

2.4 Timetable

The search for the smallest element in a regular unsorted array would involve the cost of $O(N)$ per event. Since each event invalidates only small part of the timetable, it is useful to keep the timetable ordered in such a way that updated entries could be brought to their new places fast. Also the minimum entry of the timetable should be available at low cost. Since the problem of search and sorting is general and well studied, there are many data structures

which satisfy the necessary conditions [Knu73]. The classic example of such data structure is a binary tree which is actually used as a timetable realization in [Rap95]. Skipping the details and strict definitions, it is useful to say that in binary trees each data entry (a node) can have one parent node and two child nodes (left and right). The particular implementation of these relationships can be done for example by using pointers to the corresponding nodes. Together with these three pointers, the nodes also contain the sorting parameter called "key" and any other user data. The ordering in the tree is maintained by the constraint that the key of the left child is less (in some sense) than the key of the parent and the key of the right child is more than the key of the parent. Thus, in a completely ordered binary tree, for any node the left branch (left subtree) will contain the nodes with the smaller keys and the right branch will have the bigger keys. It is important that for a given set of nodes there can be many ordered trees depending on the order the nodes were supplied.

One of the examples is given in Fig. 2.2. The other rather extreme example would be if the nodes were supplied already in order, i.e., 9, 8, 7, etc. The resulting tree would have only left nodes, and the height of the tree (the number of levels) would be equal to the number of nodes. Such a degenerate tree is not suitable for searching, deleting and inserting the nodes since it has basically a regular linked list structure. The tree would be much more balanced if the nodes are supplied in a random order. For a balanced tree each level contains twice as much nodes as previous one, and the total number of

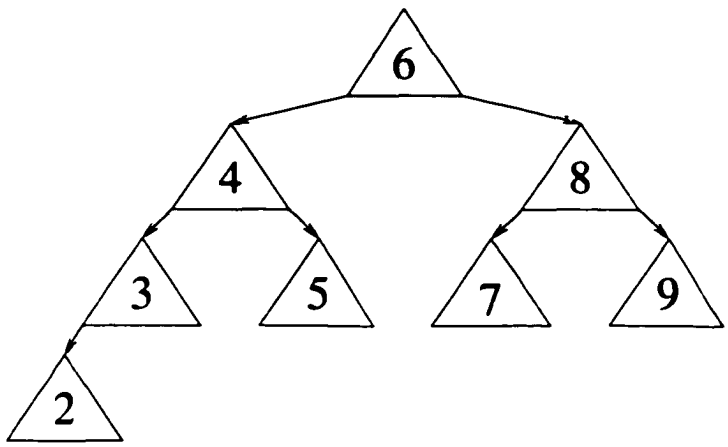

Figure 2.2: An example of a binary tree

nodes is given by a geometrical progression:

$$N = 2^0 + 2^1 + 2^2 + \ldots + 2^{h-1} = \frac{2^h - 1}{2 - 1} = 2^h - 1 . \tag{2.8}$$

where h is the height of the tree and N is the number of nodes. Therefore,

$$h = \log_2(N + 1) . \tag{2.9}$$

The cost of operations of inserting and deleting a node is based on the cost of finding the node. To insert a node, one must search for a place in the tree where the node can be inserted and then simply rearrange the appropriate pointers. To delete a node, one must find a place where to reinsert the node's children. And the cost of finding a node is proportional to the height of the tree since all that is needed is to travel down the tree (at worst starting from the root) making lefts or rights based on the value of the key in search. As an example, consider adding the node with a key 4.5 to the tree in Fig. 2.2.

First comparison with the root (key = 6) will point to the left (`p =p->left`) node **4**, then comparison with **4** will point to the right (`p =p->right`) node **5**, then to the left, and since the left node of **5** is `NULL`, the new node **4.5** becomes the left child of **5**. On the other hand, inserting a node into a "linear like" structure such as a linked list, array or degenerated tree generally would involve comparisons with most of the nodes. Thus, the cost of all operations with the timetable based on a binary tree is, in principle, $O(\log(N))$ per event provided that the tree remains reasonably balanced.

The binary tree timetable is used in [Rap95] where it is pointed out that since from the point of view of the timetable the hard sphere collisions occur randomly, the tree remains balanced. On the other hand, [Sch96] warns that regular binary trees have a tendency to become unbalanced in which case the tree would need to be re-balanced periodically. Since all operations with the timetable have a significant cost, the timetable abstraction layer can be introduced with a negligible penalty. In this case all operations with a timetable are done using functions of the kind `time_table_oper(...)` which will hide the actual implementation from main code. This makes possible to use various timetable implementations without changing other parts of code.

In the present work the binary tree timetable was implemented based on the algorithms for inserting, deleting and searching discussed in [Sch96]. The test runs have confirmed that the binary tree remained far from its degenerate state, but they have also shown that the nearly balanced state, observed at the beginning of a simulation, had become rather unbalanced just after several events per particle. Of course, this does not rule out the suitability

of the binary tree data structure for the timetable. More information on this issue could be obtained by using various binary tree implementations.

Nevertheless, the fact that the height of the binary tree can vary rather strongly, and that it also depends on how "random" events are in the system, shows that this particular data structure might not be the best for the task at hand. To continue along the same line, one could use other types of binary trees which involve additional constraints on the nodes serving to limit the tree height variations, e.g. so called "red-black" trees [CLR90].

But the important fact is that the major goal of binary trees is to provide a quick search for *any* element in the tree. And the only element the key of which is actually needed to be searched for in the timetable is the minimum one. The *heap* data structure can be used for this purpose. While in a regular binary tree the ordering is: (left child) $<$ parent $\leq$ (right child), in a heap the ordering is: parent $\leq$ (both left and right), without specifying the relation between the children. In addition, every node must have both left and right children except maybe the nodes at the lowest level. This guarantees triangular (heap-like) structure of the collection of the nodes with the height of about $\log_2(N+1)$. By the very design the minimum (maximum) element is always available as the top of the heap.

Similarly to binary trees, there can be many representations of a given set of nodes in the form of a heap. For example, the nodes **5** and **7** (Fig. 2.3) can be swapped (or **7** and **6**) and the resulting structure would still satisfy the constraints for a heap. The nodes of each layer are not necessarily "less" than any node in the lower layer, but the top of each "sub-heap" is always

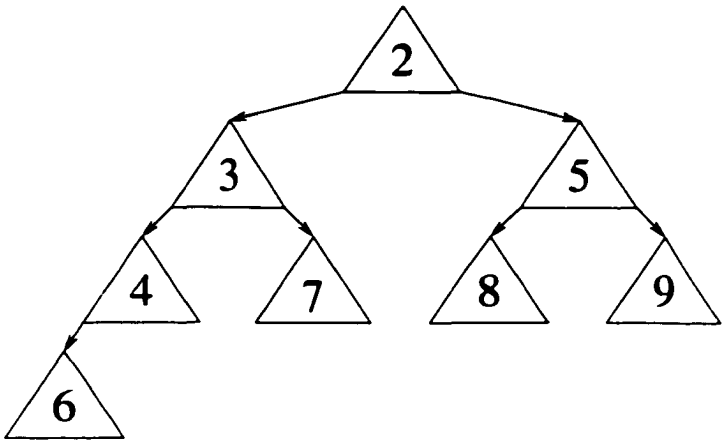

Figure 2.3: An example of a heap

its smallest (biggest) element.

When events in the system are being processed, the top element and possibly some other "affected" elements become updated, their keys corresponding to (`p->time +p->coltim`) get new values and the heap would need to be adjusted. The actual assigning of the new values to the keys should be done one node at a time. Every time any single node has its key changed, the immediate readjustment procedure should follow to keep the resulting heap valid.

Suppose the next event corresponding to the node **2** has been processed and the new value of the key for this node is **10**. Since the smallest out of the trinity (parent and two children) should go on top, the node **10** will be swapped down with **3**, then by the same token with **4** and then with **6**. The nodes are being swapped only and the height remains constant. When there

is an updated node in the "depth" of the heap, it will be swapped up or down in a similar fashion, but the number of swaps is always defined by the height of the heap. As an example, for a system of 2^{20} ($2^{10} \cdot 2^{10} = 1024 \cdot 1024 \approx 10^6$) particles the height is just 20 layers. Thus, the cost of basic operations needed for the timetable maintenance is always $O(\log(N))$. One of the classic and compact implementations of the heap data structure is based on a regular array. In such arrays an element with an index i ($i = 1, 2, 3 \ldots$) has children indexed by $2i$ and $2i + 1$. The actual data stored in each element of the array is a pointer to a specific node which is a container for all the pertinent information. This way not the containers itself but only pointers to them are getting swapped during the heap operations.

Whenever some object in the system was updated and its (`p->time` + `p->coltim`) entry was changed, then the timetable node associated with this object will need to be located for the revalidation of the timetable (heap, binary tree, etc.). For this purpose, each object which has an entry in the timetable stores the location of this entry as part of its data. This entry can be a pointer to the corresponding node in a binary tree or an index of a corresponding element in an array based heap. For example, if each particle has an entry in the timetable, then together with its position, velocity and other parameters it would also store its place in the timetable. This kind of "particle based" timetables is used in [Rap95] and [Lub91].

However in the present work objects associated with the timetable are not particles, but the cells. First of all, while the number of particles can vary, the number of cells is fixed which makes it very suitable for compact

array based heap implementation. More important is the fact that during updates of the "affected" particles any other particle in the neighborhood can change its (`p->time + p->coltim`) value and not just once. If its location in the timetable is immediately adjusted, then during the same update it might need to be readjusted again. To avoid this, all particles which changed their future event time at least once should be stored away. Then, as the final step of the current update, all of them can be recalled one by one and the timetable revalidated. But the cells already store all the particles involved. Whenever there is a particle which changes its future event, the host cell of this particle is tagged as "affected" by using a flag in the cell data structure. After all the particles involved in the updating procedure are processed, the "affected" cells can be easily collected by direct inspection of one or two neighborhoods involved. Then, since the number of particles in each cell is very small, the nearest future event in a cell can be located fast simply by going through the particles of a cell and checking their (`p->time + p->coltim`). Profiling confirms that the overhead for this search is not significant. The end result is stored in "`next_event_time`" and "`next_event_ptcl`" entries of a cell data structure. The value of (`c->next_event_time`) should now be used as a new key in the timetable and the timetable is finally readjusted. This approach allows completion of all operations locally without accessing the "main" timetable until it is necessary.

It is interesting to notice the following property of the heap based timetables. Suppose in the system there is a source of events which generates events very often. For example, this can be a virtual "thermostat particle" which

causes with a certain frequency random kicks to some particles. If these events happen very often, then the timetable entry for this event source is going to be always located near the top of the heap. Every time an event of this source is processed, the corresponding timetable entry is going to be swapped only very few layers down to its new position. Thus, the timetable overhead for maintaining this kind of frequent events would be negligible.

2.5 Code design

The straightforward way to organize the scheduling and processing of various types of events is to store the event type information and then process an event based on its type. For example, in [Rap95] the event type is stored in the timetable. Then using either **if** or **switch** operators the particular change of state and update procedures can be chosen:

```
switch(event_type_id)
{
case PTCL_PTCL_EVENT:
  //Do a collision and necessary updates
  break;
case PTCL_CELL_EVENT:
  //Move the ptcl to the appropriate cell, do the updates
  break;
case PTCL_OBSTACLE_EVENT:
  //Do a collision and necessary updates
```

```
  break;
case BOX_SOLID_BOUNDARY_EVENT:
  //Bump the ptcl of the wall, do the updates
  break;
case ...
...
}
```

The `switch()` statement is more appropriate here since it jumps directly to the correct `case` block whereas (`if else if`) chain might go through until `event_type_id` matches one of the type id's. The more feature rich the simulated system becomes, the more `case` blocks should be included (in the present work the number of various event sources was about 10). This does not affect the performance but suggests that better encapsulation of each event source and associated functions would be helpful when modifying, debugging and maintaining code.

Ideally adding a new feature (event source) should mainly involve the development of the appropriate module with minimum and consistent changes for the rest of code. As an attempt to reach this goal, the following abstractions and design were developed. The basic object for the event source is a "particle". This data structure encapsulates the parameters of an object such as coordinates, velocities, mass, etc. Most event sources are active agents in a sense that they are acting upon regular particles changing their state. Regular particles play the role of both agents and passive objects. To allow many

objects to share the same functionality the following data structure is used:

```
struct interface_s
{
  int type;
  INTERFACE_ACTION action;
  INTERFACE_EVNT_TIME event_time;
  double rstu;   //restitution
  double ttur;   //temperature
  //More data members follow ...
};
```

A pointer to it (**infc**) is included in a basic object data structure:

```
struct ptcl_s
{ int ID;
  double r[SPACE_DIM]; //position
  double time;
  double v[SPACE_DIM]; //velocity
  struct cell_s *home; //pointer to a host cell
  //Time until the next event:
  double coltim;
 //The agent of the next event:
  ptcl_s *partnr;
  double rad;              //radius
  double mass;
```

```
  struct interface_s *infc;        //interface info.
  //More data members follow ...
};
```

The member "action" of **interface_s** is a pointer to a function which does all the necessary work for the event processing which usually involves calls to functions responsible for the change of state and the appropriate for each type of events updates. This pointer gets its value during the initialization procedure of the particular event source. Then during the simulation the correct routine is called through this pointer without any additional checking of an event type.

The common code for an event procedure is simply similar to

```
event_get();
p1 =glob_next_event_ptcl;
p0 =p1->partnr;
p0->infc->action(p0, p1);
```

Here global variable **glob_next_event_ptcl** is set by **event_get()** call which simply takes the next event from the timetable.

A similar approach is useful when calculating the possible future event times for a given particle during its update. The particle should be checked against all possible agents that can generate an event involving this particle. Different agents would have different routines for calculation of event times. One example considered before is particle-particle event. Others are

cell crossing event, obstacle-particle event, etc. During the initialization of every agent's interface, its member "**event_time**" is set to the pointer to the function which performs the particular calculation of the future event time. Then during an update of a particle "p0" the check against possible agent "p" is performed by

```
p->infc->event_time(p, p0);
```

Every object in the system, once properly initialized and placed in the appropriate cells (some should be kept global), starts functioning on its own through the methods (member functions) of its interface, i.e., "**event_time**" and "**action**". To add a new feature (event source) to the simulated system, one can create a separate module containing code for the routines implementing "**event_time**" and "**action**" and maybe some other functionality, and provide the initialization calls for each new object corresponding to the feature being added.

2.6 Other options

To further improve the performance of the serial algorithm discussed, one can store all possible partners of a given particle rather than only the partner with the best future event time. The maintenance of such a list for each particle would involve keeping "back" pointers for fast location of a given particle in the list of its possible partner. Then, once a particle changes its state through an event, it can easily be removed from the lists of its possible partners. The

details of this optimization are discussed in [Rap95]. However in [Lub91] it is pointed out that keeping all the possible partners and "rating" them for the "best candidate" might not benefit that much the overall performance due to the additional complexity overhead. The discussion, comparison and some benchmarks of various flavors of the serial algorithm can be found in some details in [Kra96].

An approach aimed at eliminating cell crossing events is proposed in [Iso99]. In this Extended Exclusive Particle Grid Method (EPGM) the simulation domain is covered by a grid with a cell size smaller than a particle diameter. Only a single particle can belong to any cell. The optimum set of cells comprising a neighborhood (mask) depends on the density of the system. The time interval after which the "re-population" of a neighborhood with particles is needed depends on the maximum velocity in the system. This interesting variant of the serial algorithm benefits from the techniques developed for the soft particle simulations (e.g. see [BP93]) and hopefully further works will confirm its performance advantages.

Certain level of parallelization can be achieved by using multi-threading approach. When two CPU's sharing the same memory are available, then the updates for both main parties involved in particle-particle event can be done by two threads (light weight processes) running in parallel. Of course, since both threads would have to operate on the same set of objects (particles in the common part of two neighborhoods), the proper object locking mechanism is needed. Careful object locking is an intrinsic part of multi-threaded programming and it is usually done using mutex or semaphore mechanisms.

Since the whole idea of discrete event simulations is based on processing one event after another, it is rather difficult to create a parallel version of these simulations. If you split the simulation region in several domains and each CPU would process the events in its own domain, then each domain will have its own time. At some point a particle will cross the boundary between domains and the parallel algorithm has to handle the situation when a particle shows up having its time in the past. A lot of work has been done to develop efficient parallel algorithms for event driven simulations. The description of one of them and references to some other developments in this direction can be found in [Mar97].

Chapter 3

Boundary conditions; collision rule

3.1 Periodic boundary conditions

Very often the simplest way to simulate the stream of particles is to use the periodic boundary conditions in the direction of the flow. If there is an agent (an obstacle, a wall with friction, etc.) in the flow that is taking the momentum away from the stream, such an agent will be slowing the stream down until the flow stops. Thus, to maintain the motion some driving mechanism must be provided. External force such as gravity can be used and its value can be chosen such that the momentum loss is balanced by the driving force. This approach was used for the simulation of the flow of soft elastic disks in [RC86] and [Rap87]. Among other things important for future development, Rapaport [Rap87] pointed out studying non-slip boundary conditions for the

obstacle and creating a "velostat" system which would not require trial and error search for the appropriate gravity value. It would also shorten or eliminate the stabilizing period during which the flow changes its initial velocity until it reaches its final value. The difference between the two values went as high as 40% in [Rap87]. Certain steps in this direction (modification of periodic boundary conditions) were already made by Rapaport [Rap87] since when particles were crossing the down stream boundary back into the upstream boundary, their velocities were changed (randomized) to correspond to the velocities of a stream with the given temperature. This also partially isolated the "up" and "down" sides from each other since any velocity patterns on the down side were destroyed by randomizing during the re-insertion at the up side. This isolation is only partial because of the non-zero particle interaction range, i.e. particles which were already reinserted and randomized would "feel" the influence of the down side until they move far enough down the stream. This can be improved by expanding the velocity control region from a single line (up side boundary) to a whole strip located somewhere upstream as in [LGIK01]. This approach also provides an alternative driving mechanism. Nevertheless, since the up and down sides are still connected through the spatial periodic boundary conditions, the total number of particles in the system is fixed. If density changes downstream are strong enough they might persist on the upstream side causing positive feedback in the system. Even far away from such a drastic regime, it is not obvious beforehand if the simulation region is large enough and the flow patterns are not affected by the fact that the system can not "shed" or "pick up" some

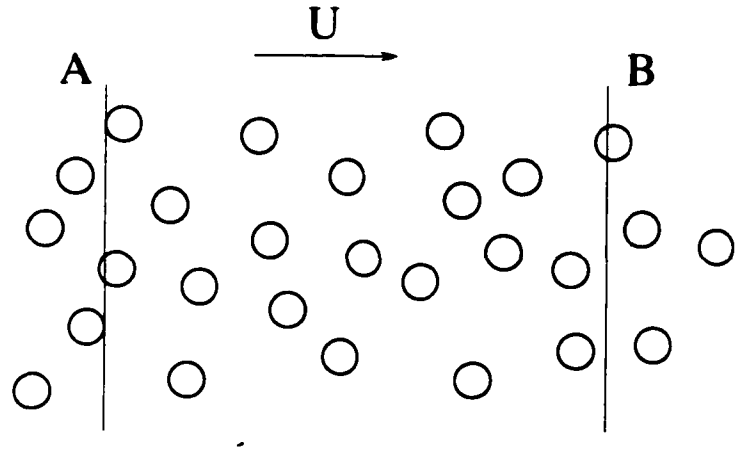

Figure 3.1: Part of an infinite stream

particles. To address these issues the following boundary conditions were proposed.

3.2 Stream boundary conditions at low densities

Consider an infinite uniform flow moving with velocity $\vec{U}$. To simulate part of this stream between planes **A** and **B**, one can try to mimic the behavior of particles crossing these planes (Fig. 3.1). If thermal velocities $\vec{v}$ of the particles are much less than $\vec{U}$ (high Mach number, Ma), then the task is relatively simple. One must produce (inject) particles at the plane **A** at a certain rate corresponding to the flux of the flow through the plane. If the volume fraction is not too high, there will always be a spot to insert a new

particle at the boundary **A**. To find an opening, one can try several random positions in the plane **A** until there is no overlap with any other objects in the neighborhood. Since thermal velocities are so small, the flux across the plane **A** in the direction opposite to $\vec{U}$ is negligible, and a rare particle trying to go out of the region through **A** can be simply bounced back. There is still a question of assigning appropriate new particle velocities corresponding to the temperature and the velocity of the flow. If the upstream temperature at **A** is assumed to be zero, then all the new particles at **A** can simply get velocity $\vec{U}$. For very small temperatures, a thermal component can be added on top of $\vec{U}$. For higher temperatures (or equivalently slower flows), the velocities should be sampled from the distributions to be discussed later.

The situation at the plane **B** is even simpler. Since the flux into the region through **B** is small, all the particles leaving **AB** through **B** can be deleted.

For slower flows when thermal velocities are comparable to $\vec{U}$, the approach described above is no longer valid. The "in" flux across **B** is substantial, and in the limit case when $\vec{U} = 0$ it is equal to the "out" flux. To imitate the "back" flux, the particles which try to leave the region through **B** must be turned back with a certain probability. For the particles turned back new velocities should be assigned. A similar problem was addressed in [LF00] and [FL02] where the appropriate velocity distributions at the inflow and outflow boundaries were needed for the DSMC (direct simulation Monte Carlo) simulations of the low density flows in microelectromechanical systems (MEMS). Since the corresponding ideas and distributions used in the

present work were developed at approximately the same time and completely independent of [LF00], the detailed description follows.

3.2.1 Velocity distribution of particles crossing the given plane.

Consider the particles with velocities $\left[\vec{W}, \vec{W} + d\vec{W}\right]$. The number of such particles going through the area dA of the parallel to YZ plane **P** per time dt is the number of them within a cylinder with the base dA and height $|W_x|dt$:

$$|W_x|dt \cdot dA \cdot dn_{\vec{W}}. \tag{3.1}$$

where $dn_{\vec{W}}$ is number density of these particles. The same number of particles taken per unit area per unit time is just

$$|W_x| \cdot dn_{\vec{W}}. \tag{3.2}$$

Let

$$\vec{W} = \vec{U} + \vec{v}. \tag{3.3}$$

where $\vec{U}$ is a stream velocity and $\vec{v}$ is a velocity of a particle in the reference frame moving along with the stream. In the stream frame, assuming local thermodynamic equilibrium, the number of particles per unit volume (number density) with velocities in the interval $\left[\vec{v}, \vec{v} + d\vec{v}\right]$ is given by Maxwell distribution,

$$dn_{\vec{v}} = n \cdot d\omega_{v_x} d\omega_{v_y} d\omega_{v_z}. \tag{3.4}$$

where

$$d\omega_{v_i} = \left(\frac{c}{\pi}\right)^{\frac{1}{2}} \exp\left(-cv_i^2\right) dv_i \,. \quad i = (x, y, z). \quad c = \frac{m}{2kT} \,. \tag{3.5}$$

$$\int d\omega_{v_i} = 1 \,. \tag{3.6}$$

But in the lab frame the *same* group of particles will have the velocities $\left[\vec{U} + \vec{v}, \vec{U} + \vec{v} + d\vec{v}\right]$ which is the same as $\left[\vec{W}, \vec{W} + d\vec{W}\right]$ if $\vec{U}$=const at least locally. That means

$$dn_{\vec{W}} \;=\; dn_{\vec{v}} = n \cdot d\omega_{v_x} d\omega_{v_y} d\omega_{v_z} \,. \tag{3.7}$$

or taking into account that $d\vec{v} = d\vec{W}$:

$$dn_{\vec{W}} = n \cdot d\omega_{W_x} d\omega_{W_y} d\omega_{W_z} \,. \tag{3.8}$$

where

$$d\omega_{W_i} = \left(\frac{c}{\pi}\right)^{\frac{1}{2}} \exp\left(-cv_i^2\right) dW_i = \left(\frac{c}{\pi}\right)^{\frac{1}{2}} \exp\left(-c\left(W_i - U_i\right)^2\right) dW_i \,. \tag{3.9}$$

$$\int d\omega_{W_i} = 1 \,. \tag{3.10}$$

Thus (3.2) is

$$|W_x| \cdot dn_{\vec{W}} = |W_x| \cdot n \cdot d\omega_{W_x} d\omega_{W_y} d\omega_{W_z} \,. \tag{3.11}$$

Probability that a particle going through dA will have a velocity $\left[\vec{W}, \vec{W} + d\vec{W}\right]$ is the number of such particles divided by the total number of particles going through dA. This total number (per unit area per unit time) of particles is equal to (3.11) integrated over $\vec{W}$:

$$n \cdot \int_{W_x} |W_x| d\omega_{W_x} \cdot \int_{W_y} d\omega_{W_y} \cdot \int_{W_z} d\omega_{W_z} = n \cdot \int_{W_x} |W_x| d\omega_{W_x} \,. \tag{3.12}$$

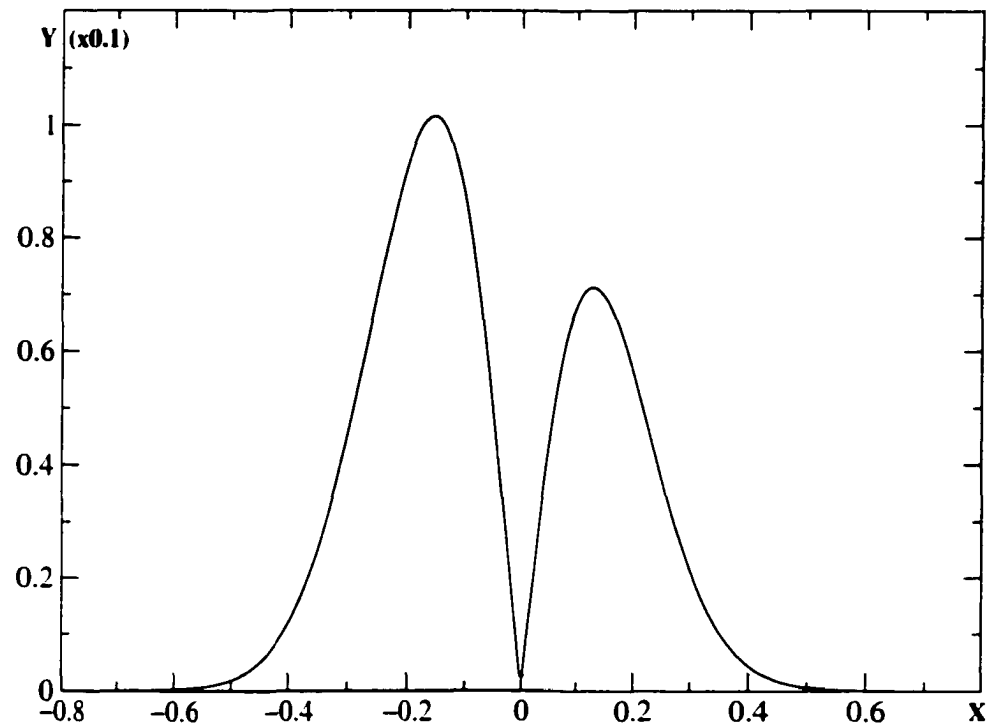

Figure 3.2: $y = |x|e^{-c(x-a)^2}$. $c = \frac{m}{2kT} = \frac{1}{2 \cdot 1 \cdot 0.02}$. $a = -0.025$

After integration it gives

$$n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot \left[f(-\sqrt{c}U_x) + f(\sqrt{c}U_x) \right] = \frac{n}{\sqrt{\pi}\sqrt{c}} \left[e^{-cU_x^2} + \sqrt{\pi}\, \sqrt{c}U_x \operatorname{erf}(\sqrt{c}U_x) \right]. \tag{3.13}$$

where

$$f(t) = \left[e^{-t^2} + \sqrt{\pi}\, t \cdot (\operatorname{erf}(t) + 1) \right].$$
$$f(-t) = \left[e^{-t^2} + \sqrt{\pi}\, t \cdot (\operatorname{erf}(t) - 1) \right]. \tag{3.14}$$

The absolute value of the flux of particles crossing the plane with *negative*

velocity component of interest is

$$F_- = n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot f(-\sqrt{c}U_x) \tag{3.15}$$

and, if divided by the total number of particles in question, it gives the probability for a particle to have a negative velocity x-component:

$$\frac{f(-\sqrt{c}U_x)}{f(-\sqrt{c}U_x) + f(\sqrt{c}U_x)} = \frac{1}{1 + \frac{f(\sqrt{c}U_x)}{f(-\sqrt{c}U_x)}} . \tag{3.16}$$

The flux of particles crossing the plane with *positive* velocity component of interest is

$$F_+ = n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot f(\sqrt{c}U_x) , \tag{3.17}$$

and the probability for a particle to have a positive velocity x-component is

$$\frac{f(\sqrt{c}U_x)}{f(-\sqrt{c}U_x) + f(\sqrt{c}U_x)} = \frac{1}{1 + \frac{f(-\sqrt{c}U_x)}{f(\sqrt{c}U_x)}} . \tag{3.18}$$

As one would expect, the resulting flux is

$$n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot \left[-f(-\sqrt{c}U_x) + f(\sqrt{c}U_x) \right] = n \cdot U_x . \tag{3.19}$$

It is also interesting to consider limiting cases of, for example, F_+. For a large positive stream velocity, F_+ flux becomes total flux (its x-component) $n \cdot U_x$:

$$\sqrt{c}U_x \to \infty, \; F_+ \to n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot \left[0 + \sqrt{\pi c}\, U_x (1 + 1) \right] = n \cdot U_x . \tag{3.20}$$

For a large negative stream velocity, F_+ flux vanishes:

$$\sqrt{c}U_x \to -\infty, \; F_+ \to n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot \left[0 - \sqrt{\pi c}\, |U_x| (1 - \mathrm{erf}(\sqrt{c}|U_x|)) \right] \to$$
$$n \cdot \sqrt{\frac{c}{\pi}} \frac{1}{2c} \cdot \left[0 - \sqrt{\pi c}\, |U_x| (1 - 1) \right] \to 0 . \tag{3.21}$$

Finally, for zero stream velocity,

$$U_x = 0,\ F_+ = n \cdot \sqrt{\frac{c}{\pi}}\frac{1}{2c} \cdot [1+0] = \frac{n}{\sqrt{4\pi c}}. \tag{3.22}$$

One can also expect that $F_+ \geq n \cdot U_x$ is always true for all positive U_x ($n \cdot U_x$ plus some thermal part). Indeed,

$$n \cdot \sqrt{\frac{c}{\pi}}\frac{1}{2c} \cdot f(\sqrt{c}U) \Big/ n \cdot U = \sqrt{\frac{c}{\pi}}\frac{1}{2c}\left[\frac{e^{-cU^2}}{U} + \sqrt{\pi}\sqrt{c}\ (\mathrm{erf}(\sqrt{c}U)+1)\right] =$$
$$\frac{1}{2}\left[\frac{e^{-t^2}}{t\sqrt{\pi}} + (\mathrm{erf}(t)+1)\right] = \frac{1}{2}\left[\frac{e^{-t^2}}{t\sqrt{\pi}} + \frac{2}{\sqrt{\pi}}\int_0^t e^{-t^2}dt\ + 1\right] \overset{t\to\infty}{\longrightarrow} 1. \tag{3.23}$$

Its derivative,

$$\frac{1}{2}\left[\frac{-2t^2e^{-t^2} - e^{-t^2}}{t^2\sqrt{\pi}} + \frac{2}{\sqrt{\pi}}e^{-t^2}\right] = -\frac{1}{2}\frac{e^{-t^2}}{t^2\sqrt{\pi}}. \tag{3.24}$$

is always negative, and thus the limit $\left(\overset{t\to\infty}{\longrightarrow} 1\right)$ is reached from above which proves that $F_+ > n \cdot U_x$.

The number of particles (per unit area per unit time) going through the plane with velocity x-components $[W_x, W_x + dW_x]$ is equal to (3.11) integrated over W_y and W_z:

$$|W_x| \cdot n \cdot d\omega_{W_x} \int_{W_y} d\omega_{W_y} \int_{W_z} d\omega_{W_z} = |W_x| \cdot n \cdot d\omega_{W_x}. \tag{3.25}$$

This number divided by the total number of particles in question is the probability of a particle going through the plane to have a velocity x-component $[W_x, W_x + dW_x]$ and it is equal to

$$|W_x|\sqrt{\frac{c}{\pi}}\ \exp\left(-c\,(W_x - U_x)^2\right) dW_x \Big/ \sqrt{\frac{c}{\pi}}\frac{1}{2c}\left[f(-\sqrt{c}U_x) + f(\sqrt{c}U_x)\right]. \tag{3.26}$$

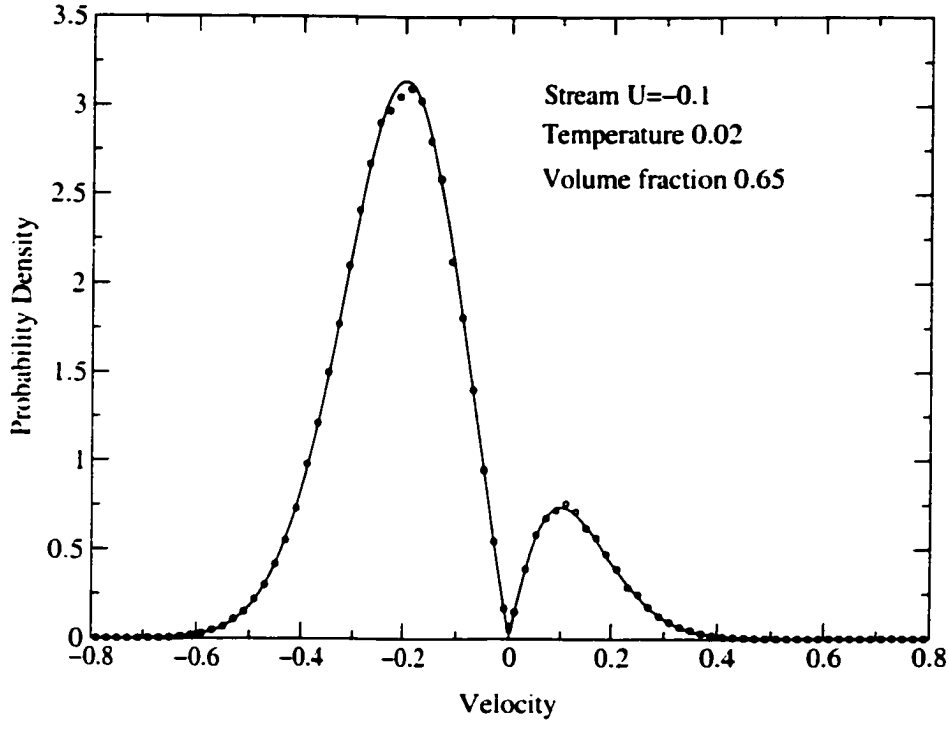

Figure 3.3: Simulation and analytical expression (3.27)

This probability divided by the velocity x-component interval $[W_x, W_x + dW_x]$ is the probability density and is equal to (see Figs. 3.2 and 3.3)

$$p_{W_x}(W_x) = \frac{2c}{[f(-\sqrt{c}U_x) + f(\sqrt{c}U_x)]}|W_x| \exp\left(-c\,(W_x - U_x)^2\right). \quad (3.27)$$

If we consider only the particles crossing the plane with *negative* velocity component of interest, the corresponding velocity distribution is

$$p_-(W_x) = \frac{2c}{f(-\sqrt{c}U_x)}|W_x| \exp\left(-c\,(W_x - U_x)^2\right). \quad (3.28)$$

And the particles crossing the plane with *positive* velocity component of

interest have velocity distribution given by (see Fig. 3.4)

$$p_+(W_x) = \frac{2c}{f(\sqrt{c}U_x)} W_x \exp\left(-c(W_x - U_x)^2\right). \tag{3.29}$$

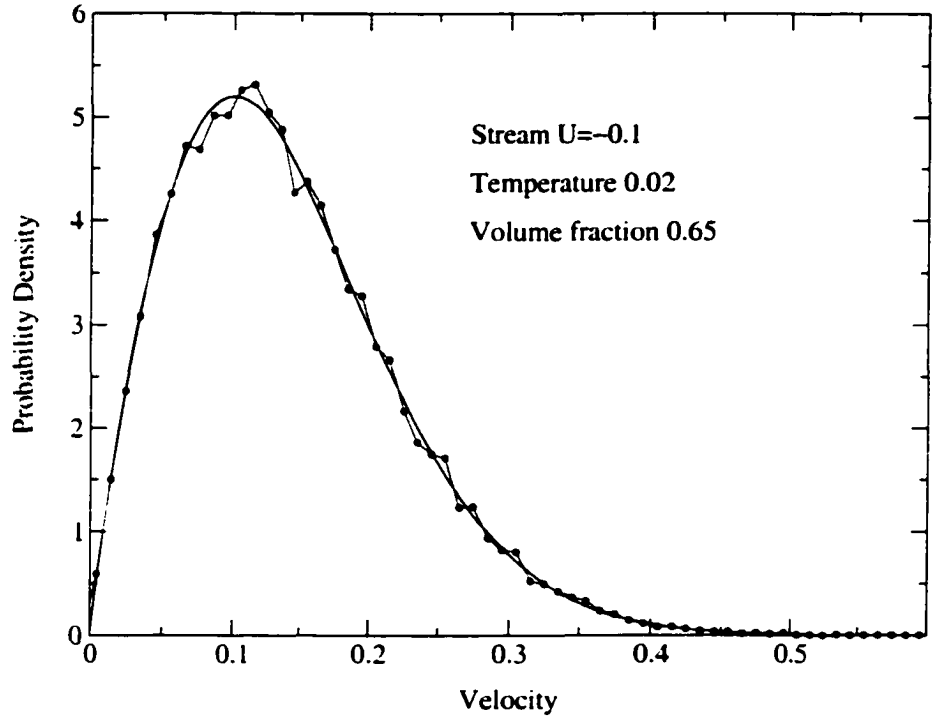

Figure 3.4: Simulation and analytical expression (3.29)

3.2.2 Generating a deviate according to its given distribution.

To generate velocities according to the distributions derived above, one can use the following. If random variable y has the given probability density

$P(y)$ and y is the function of the uniform deviate x,

$$y = y(x), \tag{3.30}$$

then the probability of x to be in the interval $[x, x + dx]$ is equal to dx (x is a uniform deviate) and at the same time it is equal to the probability of y to be in the interval $[y, y + dy]$ which is equal to $P(y)dy$ (e.g. see [PTVF92]). Thus,

$$dx = P(y)dy\,. \tag{3.31}$$

The integration gives

$$\int_{x_0}^{x} dt = x - x_0 = \int_{y_0}^{y} P(t)dt\,. \tag{3.32}$$

The function $P(t)$ is non-negative, so

$$x = \int_{y_0}^{y} P(t)dt + x_0 \tag{3.33}$$

is a non-decreasing function of y, and

$$x_{min} = \int_{y_0}^{y_{min}} P(t)dt + x_0\,. \tag{3.34}$$

Thus, if we choose y_0 to be equal y_{min}, then $x_{min} = x_0$, i.e. x_0 is the lower boundary of the x-range the choice of which remains arbitrary. On the other hand, after fixing y_0 and x_0, x_{max} is determined from the normalization of $P(y)$:

$$\int_{y_{min}}^{y_{max}} P(t)dt = 1\,. \tag{3.35}$$

$$x_{max} = \int_{y_{min}}^{y_{max}} P(t)dt + x_{min} = 1 + x_{min}\,. \tag{3.36}$$

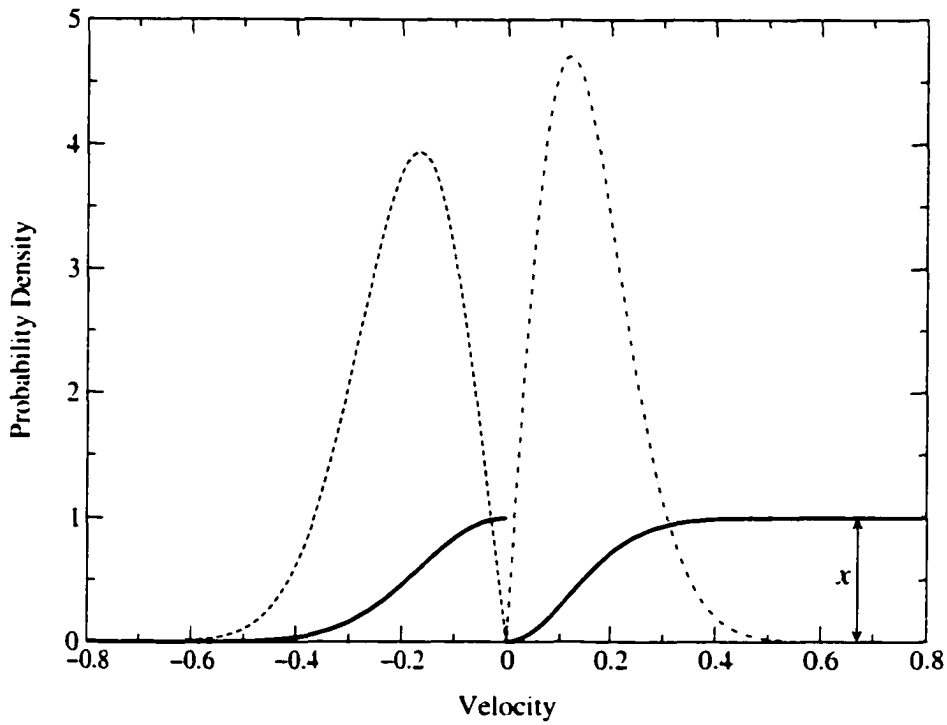

Figure 3.5: (3.47), (3.40) plotted for $c = \frac{m}{2kT} = \frac{1}{2 \cdot 1 \cdot 0.02}$, $a = -0.05$

It is natural to choose $x_{min} = x_0 = 0$. Then the solution in question is just

$$x = \int_{y_{min}}^{y} P(t)dt\,. \tag{3.37}$$

where x is a uniform deviate from the range $[0, x_{max}] = [0, 1]$.

3.2.3 Velocity generation

Velocity x-component distribution (3.29) has the form:

$$P_+(y) = \frac{2c}{f(\sqrt{c}a)} \cdot y \cdot e^{-c(y-a)^2},\ y \in [0, \infty)\,. \tag{3.38}$$

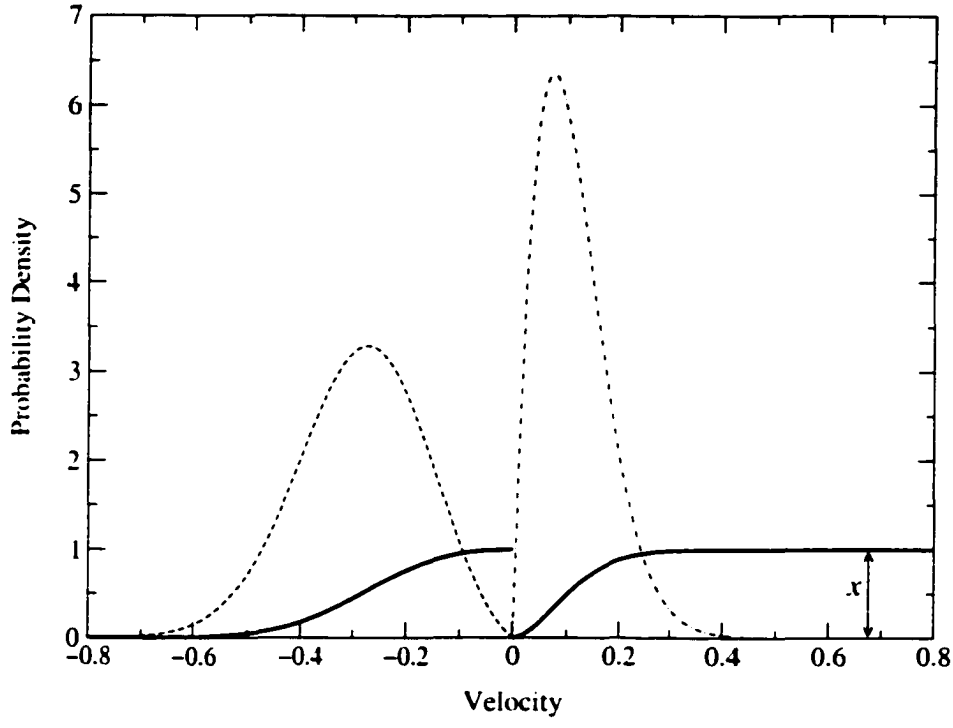

Figure 3.6: (3.47), (3.40) plotted for $c = \frac{m}{2kT} = \frac{1}{2 \cdot 1 \cdot 0.02}$, $a = -0.2$

Thus,

$$x = \frac{2c}{f(\sqrt{c}a)} \int_0^y t \cdot e^{-c(t-a)^2} dt . \tag{3.39}$$

The integral is similar to (3.12) and it gives (see Figs. 3.5 and 3.6)

$$\frac{1}{f(\sqrt{c}a)} \left[e^{-ca^2} - e^{-c(y-a)^2} + \sqrt{\pi}(\sqrt{c}a) \left(\mathrm{erf}(\sqrt{c}a) + \mathrm{erf}(\sqrt{c}(y-a)) \right) \right] =$$

$$\frac{1}{f(\sqrt{c}a)} \left\{ \left[e^{-ca^2} + \sqrt{\pi}(\sqrt{c}a) \left(\mathrm{erf}(\sqrt{c}a) + 1 \right) \right] - \right.$$

$$\left. \left[e^{-c(y-a)^2} + \sqrt{\pi}(\sqrt{c}a) \left(1 - \mathrm{erf}(\sqrt{c}(y-a)) \right) \right] \right\} =$$

$$\frac{1}{f(\sqrt{c}a)} \left\{ f(\sqrt{c}a) - \left[e^{-c(a-y)^2} + \sqrt{\pi}(\sqrt{c}a) \left(\mathrm{erf}(\sqrt{c}(a-y) + 1) \right) \right] \right\} =$$

$$1 - \frac{1}{f(\sqrt{c}a)}\left[e^{-c(a-y)^2} + \sqrt{\pi}(\sqrt{c}a)\left(\mathrm{erf}(\sqrt{c}(a-y)+1)\right)\right] =$$
$$1 - \frac{f_+(\sqrt{c}a, \sqrt{c}(a-y))}{f_+(\sqrt{c}a, \sqrt{c}a)} = x\,. \tag{3.40}$$

where

$$f_+(\xi_0, \xi_1) = \left[e^{-\xi_1^2} + \sqrt{\pi}\,\xi_0 \cdot (\mathrm{erf}(\xi_1)+1)\right]. \quad f_+(\xi, \xi) = f(\xi)\,. \tag{3.41}$$

For $a = 0$ (stream velocity $U_x = 0$) this becomes

$$x = \frac{1}{f(0)} \cdot \left[1 - e^{-cy^2}\right] = 1 - e^{-cy^2}. \tag{3.42}$$

and can be solved for y:

$$e^{-cy^2} = 1 - x: \quad y = \sqrt{-\frac{1}{c} ln(1-x)}\,. \tag{3.43}$$

Now to generate a random variable y, we just have to generate a uniform deviate $x \in [0, 1]$ and then calculate $y(x)$ according to the above formula.

In more general case when $U_x \neq 0$, one has to create a table of $x \in [0, 1]$ and $y \in [0, y_{large}]$ using (3.40) for a given U_x. Then during the simulation we again have to generate a uniform deviate $x \in [0, 1]$ and to look up the correspondent y value in the table.

For negative velocity component, the distribution has the form:

$$P_-(y) = \frac{2c}{f(-\sqrt{c}a)} \cdot (-y) \cdot e^{-c(y-a)^2}. \quad y \in (-\infty, 0]\,, \tag{3.44}$$

and

$$x = \frac{2c}{f(-\sqrt{c}a)} \int_{-\infty}^{y} (-t) \cdot e^{-c(t-a)^2} dt\,. \tag{3.45}$$

The integral is of the same type as before and the appropriate terms in its value can be regrouped as

$$\begin{aligned} & e^{-ca^2} - e^{-c(y-a)^2} + \sqrt{\pi}(\sqrt{ca})\left(\mathrm{erf}(\sqrt{ca}) + \mathrm{erf}(\sqrt{c}(y-a))\right) = \\ & \left[e^{-ca^2} + \sqrt{\pi}(\sqrt{ca})\left(\mathrm{erf}(\sqrt{ca}) - 1\right)\right] - \\ & \qquad \left[e^{-c(y-a)^2} + \sqrt{\pi}(\sqrt{ca})\left(-1 - \mathrm{erf}(\sqrt{c}(y-a))\right)\right] = \\ & f(-\sqrt{ca}) - \left[e^{-c(a-y)^2} + \sqrt{\pi}(\sqrt{ca})\left(\mathrm{erf}(\sqrt{c}(a-y) - 1)\right)\right] = \\ & \qquad f(-\sqrt{ca}) - f_-(\sqrt{ca}, \sqrt{c}(a-y)) . \end{aligned} \tag{3.46}$$

So (see Figs. 3.5 and 3.6)

$$\begin{aligned} & -\frac{2c}{f(-\sqrt{ca})}\left[-\frac{1}{2c}f(-\sqrt{ca}) + \frac{1}{2c}\left(f(-\sqrt{ca}) - f_-(\sqrt{ca}, \sqrt{c}(a-y))\right)\right] = \\ & \qquad \frac{f_-(\sqrt{ca}, \sqrt{c}(a-y))}{f_-(\sqrt{ca}, \sqrt{ca})} = x . \end{aligned} \tag{3.47}$$

where (combining with (3.41))

$$f_\pm(\xi_0, \xi_1) = \left[e^{-\xi_1^2} + \sqrt{\pi}\,\xi_0 \cdot (\mathrm{erf}(\xi_1) \pm 1)\right] . \quad f_-(\xi, \xi) = f(-\xi) . \tag{3.48}$$

The expressions for x as a function of y allow generation of velocities for the particles which are either being turned back into the simulation domain or injected at the upstream boundary.

The acceptance-rejection method [Bir94] can also be used to generate the correct distribution as pointed out in [FL02].

3.3 High volume fractions

One problem with the boundary conditions at high volume fractions is the fact that the upstream and downstream conditions must be a "matching

pair". The correct mechanism at the upstream boundary will not be able to function properly unless the outlet mechanism also does its job. Since at low volume fraction the insertion (placing without overlap) and deletion of particles did not present a problem, the use of the appropriate velocity distributions (3.28) and (3.29) at both upstream and downstream boundaries was sufficient to provide such a "matching pair". At high volume fractions the empirical search for the correct conditions was done through many numerical experiments. The upstream and downstream boundaries were placed sufficiently far from each other, so that the effect of a particular trial mechanism (specifics of placing and deleting particles) used at only one of the boundaries could be observed separately before the other boundary would change the stream.

The simulations at volume fraction 0.65 showed (Fig. 3.3 and Fig. 3.4) that the velocity distributions of particles can still be described by the expressions derived above. The outlet boundary mechanism discussed above is turning particles back with a probability determined by the stream velocity and temperature, assigning the appropriate velocities to those particles and deleting the rest of the particles coming to the outlet boundary. It works perfectly well for volume fraction (vf) 0.04, but when used for vf = 0.6, it leads to a persistent leak of particles out of the system (see Fig. 3.7).

The intuitively attractive idea to increase the probability of turning the particles back leads, of course, to the desired containment. But it also leads to an increase in temperature near the boundary of the stream. Besides, instead of choosing the theoretical value for the probability, one would have

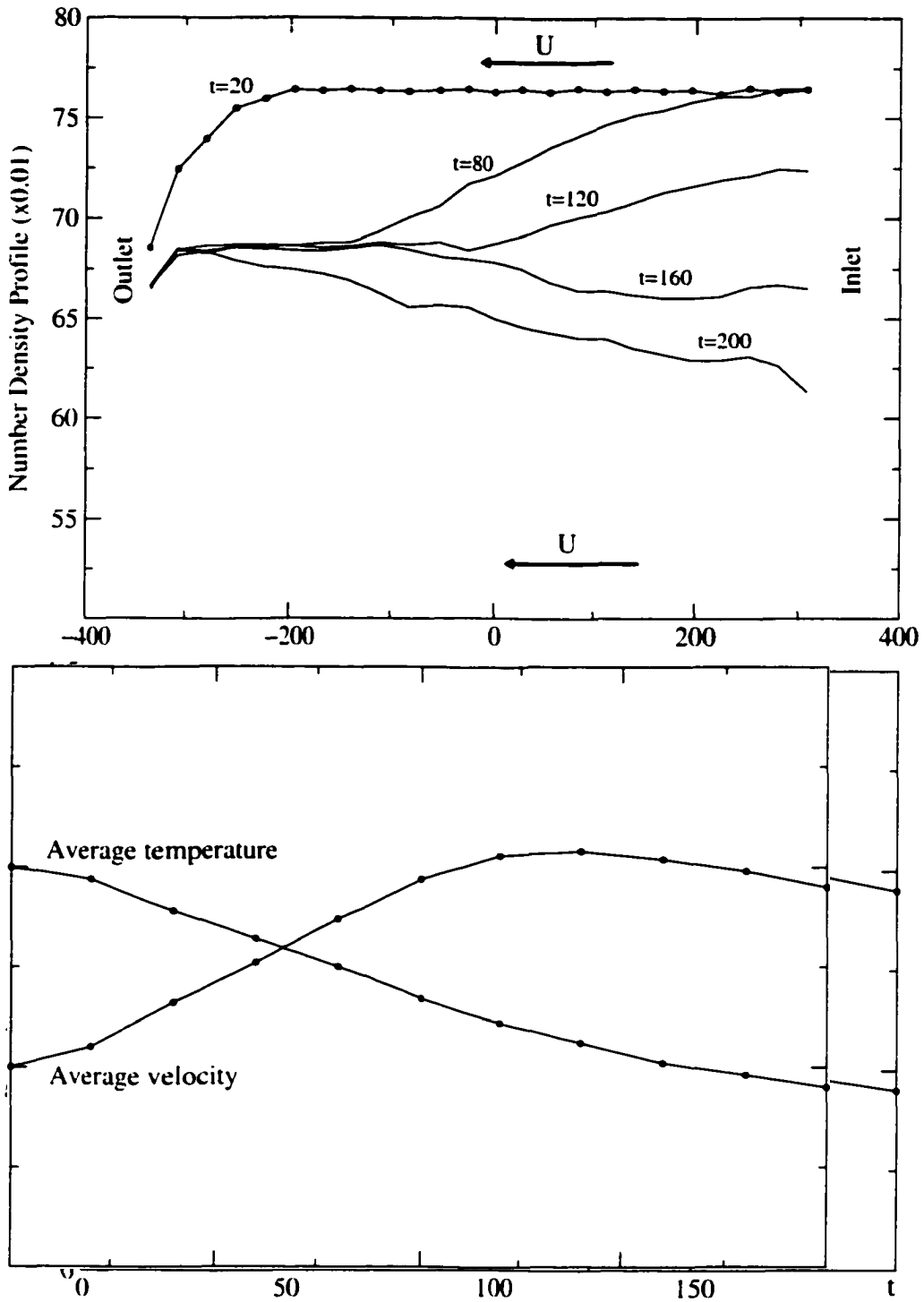

Figure 3.7: The density drop propagates to the rest of the region

to use the value based on trial and error.

The analysis of the leak shows that it is caused by two factors. First is the fact that the particles near the outlet boundary do not "feel" any resistance from the material which is supposed to be in front of them farther downstream beyond the outlet boundary. At low volume fractions, mean free path is larger than the radius of particles, and most of the particles near the outlet would leave the simulated part of an infinite stream without experiencing a collision with particles beyond the simulated boundary. In this case, only the back flow needs to be simulated which is done by turning some particles back with a certain probability (F_+/F_- when the back flow occurs in the positive direction, see (3.15) and (3.17)). At high volume fractions, special measures need to be taken to simulate the collisions of near outlet particles with particles which would be present in the actual stream extending beyond the simulated boundary. Secondly, there is also an issue caused by the fact that the particles which are deleted at the boundary leave the empty space at their location (Fig. 3.8). In spite of the microscopic nature of these openings, the nearby particles "feel" and use the openings almost immediately. The particles rush to the outlet region and some of them are deleted again creating new openings. This process continues until the volume fraction drops to the level of the resulting steady state.

To solve this problem, the following approach was used. A special interface plane f called "filter" was placed in front of the outlet boundary at a distance more than a particle radius (Fig. 3.8). The "turning back" part of the filter mechanism works exactly as in case of a low density outlet boundary

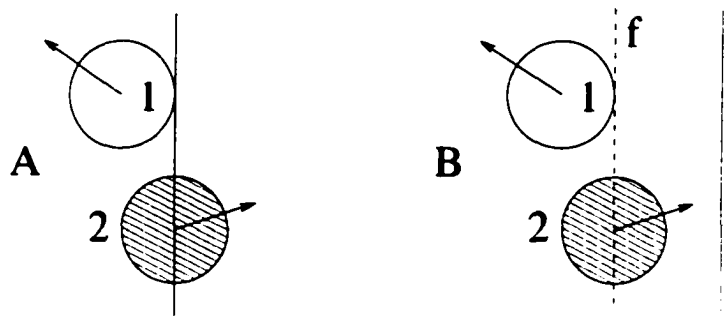

Figure 3.8: Placing a filter near the outlet

conditions. But if a particle should not be turned back, then instead of deleting it the filter changes the parameters of the particle. The particle mass becomes infinite and its velocity equal to the given velocity of the stream (no thermal component). The modified particle proceeds and will be later deleted at the "real" boundary of the simulation region. The fact that the particle does not vanish at the filter prevents the creation of vacancies. Other particles will still collide with the modified one until it completely clears the region of the filter. Of course, the vacancies will still be created at the actual downstream boundary (not the filter) of the system, but these vacancies will be isolated from the simulated part of the stream (regular particles) by the layer (or more, depending on how far the filter is placed) of modified particles. Since they have infinite mass, the modified "frozen" particles will not be heated by collisions with regular particles. The presence of "heavy" particles will have a heat insulating effect. Instead of this, one can choose a constant temperature condition. Then a new collision handler should be assigned for modified particles, and this handler would process collisions with

regular particles and assign appropriate thermal velocities.

The situation at the inlet is more complicated. While at the outlet one would need to avoid the creation of openings, at the inlet it is necessary to provide space for placing new particles. For this purpose the following "dispenser" mechanism was used (Fig. 3.9). A hard plane (a piston) starts

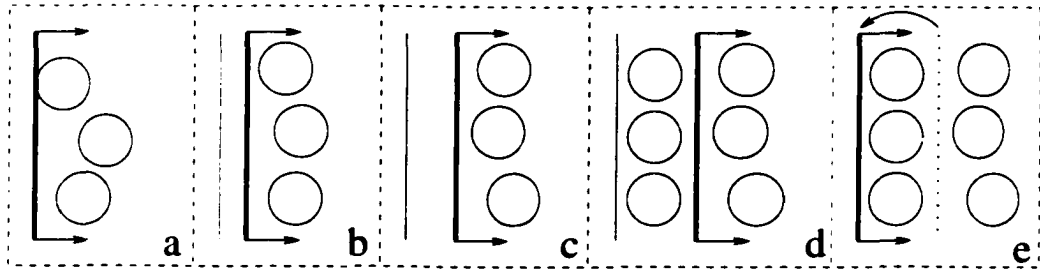

Figure 3.9: Dispenser

moving with the velocity equal to the velocity of the stream. During its motion it clears a space behind it large enough for one mono-layer of particles of the given volume fraction (stages **a**, **b**, **c**). Then the mono-layer is placed and the velocities of the particles are set to the velocity of the stream plus a random thermal component corresponding to the temperature of the stream (stage **d**). Then the piston instantaneously jumps back to its initial position, i.e., vanishes from its last position (the dotted line) and reappears at the upstream boundary (stage **e**). Thus, the new portion of particles is "dispensed" and the cycle repeats. The piston can be chosen to be elastic (heat insulating) or thermal wall ([TTKB98]). Since the piston is moving with the velocity of the stream, it should not introduce any density changes

in front of it unless the flow of particles in this region is somehow impeded. In principle, the piston can move in any way allowing to calculate collision times for particles. For example, constant acceleration (i.e. gravity) can be incorporated virtually without any changes.

It is also possible to produce particles at the inlet with velocities corresponding to a given profile. This would require several dispensers along the inlet boundary. Each of them would operate with a velocity corresponding to the value of the profile at the location of the dispenser. To clarify coding difficulties, it is important to notice that there are two agents (event sources) associated with a dispenser. One is a piston itself and the other is a timer set to trigger the re-charge. Naturally, the "action" part of the timer source will have to place the mono-layer and do the appropriate updates of the whole strip which would be affected by the piston new state and by all the new particles placed.

The matching pair of "dispenser" and "filter" mechanisms produces a stable stream with the given parameters (volume fraction, velocity, temperature). It is interesting to notice that even in the limiting case of a stream velocity equal to zero the mechanism still works. The inlet dispenser piston velocity would be zero in this case and no re-charge would ever occur. At the outlet filter, the particles which are supposed to go through will have their velocities set to zero and eventually they will form an impenetrable wall of particles.

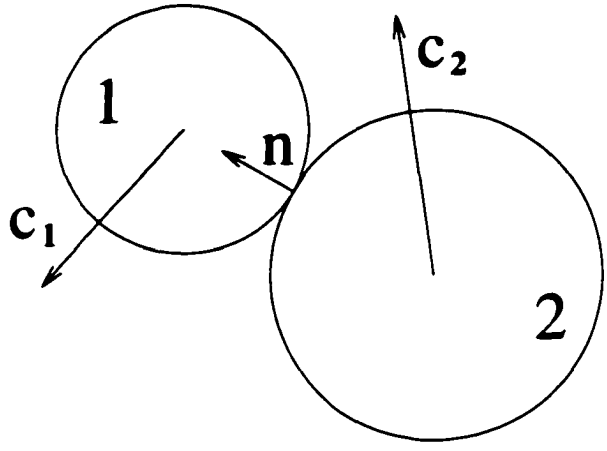

Figure 3.10: Two disks right before the impact

3.4 Collision rule

As it was mentioned in the introduction, an approach used for hard sphere simulations is that a collision between two particles is considered to be instantaneous and described by several coefficients relating pre-collision and post-collision velocities. Since these collision rules were used in all the simulations done in the present work (smooth particles only, i.e., no friction), it seems only appropriate to provide some of the calculation details.

It will be instructive and useful to split the calculation into two stages. First, the post collisional velocities will be calculated for smooth particles, and then the additional expressions will be obtained for rough particles. Let two particles have coordinates $\vec{r}_1, \vec{r}_2$ at the moment of collision. Denote their translational velocities right before the collision as $\vec{c}_1, \vec{c}_2$. The corresponding parameters after the collision will be marked by prime ($'$) symbols. The

position and velocity of particle **1** with reference to particle **2** are

$$\vec{r} = \vec{r}_1 - \vec{r}_2, \quad \vec{c} = \vec{c}_1 - \vec{c}_2 . \tag{3.49}$$

Interaction between the two particles in collision is assumed to occur instantaneously: thus there is no effect of "soft" external forces (gravity, etc.) on the system and the total momentum is conserved:

$$m_1\vec{c}_1' + m_2\vec{c}_2' = m_1\vec{c}_1 + m_2\vec{c}_2 \Leftrightarrow m_1(\vec{c}_1' - \vec{c}_1) = -m_2(\vec{c}_2' - \vec{c}_2) \equiv \vec{j} . \tag{3.50}$$

where $\vec{j}$ is an impulse exerted by particle **2** on particle **1** in the collision. And the velocities after the collision are simply

$$\begin{aligned} \vec{c}_1' &= \vec{c}_1 + \vec{j}/m_1 . \\ \vec{c}_2' &= \vec{c}_2 - \vec{j}/m_2 . \end{aligned} \tag{3.51}$$

Thus, the problem of calculating the post collision velocities is simply reduced to finding the impulse $\vec{j}$.

For inelastic collisions, instead of energy conservation equation a relation describing partial restitution of the normal component of the relative velocity can be used:

$$\vec{c}_n' = -e \cdot \vec{c}_n . \tag{3.52}$$

where e is a restitution coefficient which can take values in the interval $[0, 1]$ (1 corresponds to a complete restitution in the elastic case). For spherical particles, the unit normal vector can simply be written as

$$\vec{n} = (\vec{r}_1 - \vec{r}_2) \; / \; |\vec{r}_1 - \vec{r}_2| . \tag{3.53}$$

Finally, to obtain $\vec{j}$, the normal component of (3.51) can be substituted in (3.52) giving

$$\vec{j}_n = -(1+e)\,\vec{c}_n \,/\, (1/m_1 + 1/m_2) = -(1+e)\,M\,\vec{c}_n\,. \tag{3.54}$$

where $M \equiv (1/m_1 + 1/m_2)^{-1}$ is the reduced mass. In the case of smooth particles the task of calculating the post collision velocities (3.51) is completed since the tangential component of the impulse is $\vec{j}_t = 0$ and hence $\vec{j} = \vec{j}_n$. It might also be useful to note here for computational purposes that obtaining a normal component of a vector does not require a square root operation:

$$\vec{c}_n = \left(\vec{c}\cdot\frac{\vec{r}}{|r|}\right)\frac{\vec{r}}{|r|} = (\vec{c}\cdot\vec{r})\,\frac{\vec{r}}{r^2}\,. \tag{3.55}$$

For rough particles, consider the relative velocity of the contact point on the surface of particle **1** with reference to the same point on the surface of particle **2** right before the collision:

$$\vec{g} = \vec{c}_1 - \vec{c}_2 - \left(\frac{\sigma_1}{2}\vec{\omega}_1 + \frac{\sigma_2}{2}\vec{\omega}_2\right)\times\hat{n}\,. \tag{3.56}$$

where σ is the diameter of a particle and $\vec{\omega}$ is its angular velocity. The equation for partial restitution of the normal component of the relative velocity,

$$\vec{g}_n{}' = -e\cdot\vec{g}_n\,. \tag{3.57}$$

simply reduces to (3.52) used for smooth particles and does not add anything to the previous derivations.

To complete the computation (3.51) of post collision translational velocities, the value of $\vec{j}_t$ is needed. When the tangential component of the impulse

is not zero, it causes a change of angular momentum of the particles:

$$I_1(\vec{\omega}_1' - \vec{\omega}_1) = -(\sigma_1/2)\hat{n} \times \vec{j} = -(\sigma_1/2)\hat{n} \times \vec{j}_t .$$
$$I_2(\vec{\omega}_2' - \vec{\omega}_2) = -(\sigma_2/2)\hat{n} \times \vec{j} = -(\sigma_2/2)\hat{n} \times \vec{j}_t . \qquad (3.58)$$

where

$$I = q \cdot m(\sigma/2)^2 \qquad (3.59)$$

is the moment of inertia about the center of a particle ($q = 2/5$ for a sphere, $q = 1/2$ for a disk), and the post collision angular velocities are

$$\vec{\omega}_i' = \vec{\omega}_i - \frac{\sigma_i}{2I_i}\hat{n} \times \vec{j} = \vec{\omega}_i - \frac{\sigma_i}{2I_i}\hat{n} \times \vec{j}_t . \qquad (3.60)$$

Using (3.51) and (3.60) in the expression (3.56) for $\vec{g}$ and $\vec{g}'$, one easily finds:

$$\vec{g}' - \vec{g} = \vec{j}\left(\frac{1}{m_1} + \frac{1}{m_2}\right) + \left(\frac{\sigma_1^2}{4I_1} + \frac{\sigma_2^2}{4I_2}\right)\left(\hat{n} \times \vec{j}\right) \times \hat{n} . \qquad (3.61)$$

The double cross product can be reduced to $(\vec{j} - \vec{j}_n)$, and after substituting (3.59) for the moment of inertia, the previous expression becomes

$$\vec{g}' - \vec{g} = \frac{\vec{j}}{M}\left(1 + \frac{1}{q}\right) - \frac{1}{q}\frac{\vec{j}_n}{M} . \qquad (3.62)$$

For contacts without sliding, a model somewhat similar to the one for normal component can be used. That is, if the tangential component of the relative pre-collision velocity $\vec{g}$ is not zero, then the collision causes the deformation in the contact region such that upon the release of the energy

stored in this deformation partial restitution of the tangential component of the relative velocity occurs:

$$\vec{g}_t' = -e_t \cdot \vec{g}_t \,. \tag{3.63}$$

where $0 \leq e_t \leq 1$ is the tangential restitution coefficient. Substituting this in the tangential part of (3.62) immediately gives

$$\vec{j}_t = -M \frac{q}{1+q}(1+e_t)\,\vec{g}_t \,. \tag{3.64}$$

which is a perfectly well defined expression and does not involve any singularities for g_t approaching zero. The tangential component of the impulse simply goes to zero too just as one would expect.

If the tangential component of the relative contact velocity $\vec{g}$ is "large" enough compared to its normal component, which is responsible for the normal forces (i.e. $\vec{j}_n$) acting during the contact, then sliding occurs. In principle, "large" enough means that the tangential forces exceed the maximum value which can be provided by static friction. This would happen when

$$M \frac{q}{1+q}(1+e_t)\,|g_t| \;\geq\; \mu_s |j_n| = \mu_s (1+e)\,M\,|c_n| = \mu_s (1+e)\,M\,|g_n| \,. \tag{3.65}$$

where μ_s is a static friction coefficient. For sliding contacts, $\vec{j}_t$ can be assumed to be caused by Coulomb friction:

$$|j_t| = \mu\,|j_n| = \mu\,(1+e)\,M\,|g_n| \,. \tag{3.66}$$

where μ is the coefficient of friction. This expression is also well behaved for all possible $\vec{g}$, and since the direction of $\vec{j}_t$ can simply be taken as opposite

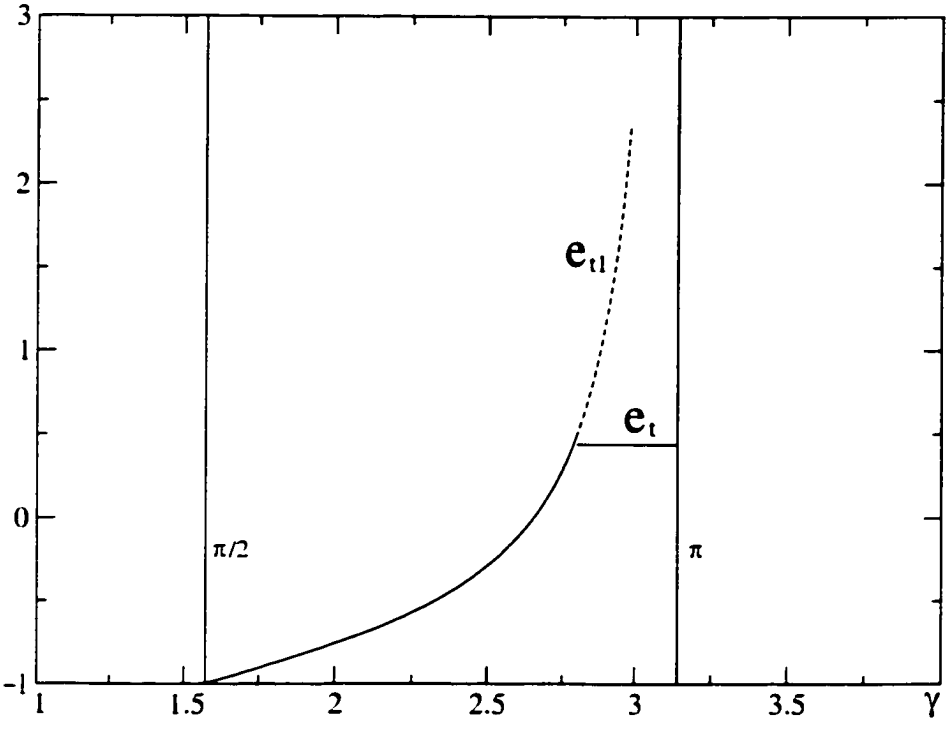

Figure 3.11: Formal tangential restitution coefficient (e=0.97, e_t=0.44, μ=0.09, q=1/2)

to the direction of $\vec{g}_t$, the problem of calculating post collision translational and angular velocities is solved provided the constants involved are known.

The expressions above are sufficient for computer simulations and well behaved. Nevertheless, an attempt to write down a single expression for vector $\vec{j}_t$ gives

$$\vec{j}_t = -|j_t| \frac{\vec{g}_t}{|g_t|} = \mu(1+e)M\cot(\gamma)\,\vec{g}_t\,. \tag{3.67}$$

where $\gamma \in [\pi/2, \pi]$ is the angle between $\vec{g}$ and $\hat{n}$ ($\cot(\gamma) = -|g_n|/|g_t|$). Con-

tinuing along the same line. [Lud98] introduces

$$e_{tl} = -1 - \mu(1+e)\cot(\gamma)\left(1+\frac{1}{q}\right). \tag{3.68}$$

which can formally be plugged in (3.64) to get (3.67). Expression (3.64) could now be used for both types of contact provided that e_t is properly chosen for a specific type. These expressions were obtained by using $\vec{g}_t/|g_t|$ as the appropriate unit vector in the tangential direction. Thus, numerically it might not be well behaved for small $\vec{g}_t$ ($\gamma \to \pi$, $\cot(\gamma) \to -\infty$), but that does not mean that $\vec{j}_t$ becomes large in the above limit. Rather confusingly, the opposite statement is made in at least two works ([Lud95] and [Lud98]), and the tangential restitution model is referred to as a solution removing this "singularity".

Under the assumption of $\mu \approx \mu_s$ (unfortunately, not clearly specified in [FLCA94]), the condition (3.65) for sliding occurring below the critical angle γ_0 can be written as

$$(1+e_t) = -\mu\left(1+\frac{1}{q}\right)(1+e)\cot(\gamma_0). \tag{3.69}$$

Below γ_0 the value of the formal tangential restitution coefficient should be taken as e_{tl} and above γ_0 as actual e_t. Since e_{tl} is the increasing function of γ (see Fig. 3.11), the formal coefficient in question is simply the minimum of e_{tl} and e_t.

Chapter 4

Elastic smooth spheres

4.1 Model parameters

As a starting point for studying systems of inelastic hard spheres, it is important to investigate numerically the behavior of systems composed of elastic smooth hard spheres. The inelasticity can later be included in the simulation model by choosing a restitution coefficient less than unity, and the inelastic behavior and transition phenomena can be investigated. One of the previous studies most relevant to the present work was presented in [RC86] and [Rap87]. Rapaport [Rap87] (see also section 3.1) studied a $2D$ flow past a cylindrical obstacle (slip boundary conditions) for a system composed of soft elastic spheres described by a shifted and truncated Lennard-Jones potential (MD units, Fig. 4.1):

$$V(r) = \begin{cases} r^{-12} - r^{-6} + \frac{1}{4}, & r < r_c = 2^{1/6} \approx 1.12 \\ 0. & r \geq r_c \end{cases} \tag{4.1}$$

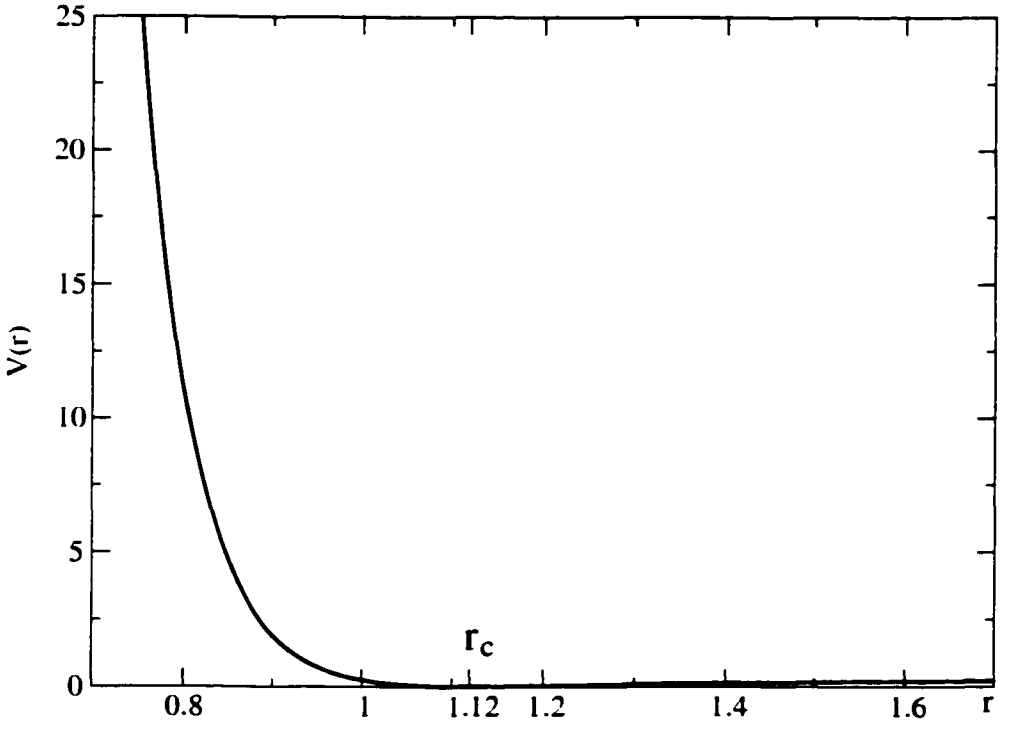

Figure 4.1: Function $r^{-12} - r^{-6} + \frac{1}{4}$

It was shown that at sufficiently high flow velocities two symmetrical eddies appear behind the obstacle, and eventually the flow pattern transforms into an oscillating wake. The phenomena were discussed in connection with the framework of a well-known change of flow patterns as the Reynolds number ($\mathrm{Re} = UD/\nu$, U-flow velocity, D-obstacle diameter, ν-kinematic viscosity) becomes larger [Bat67].

For the purely repulsive potential (4.1), one can expect a behavior similar to the case of smooth elastic hard spheres. To facilitate the comparison, the flow parameters in the simulations discussed below were chosen to be equal to

the corresponding parameters of the simulations in [Rap87], i.e., temperature $kT = 0.02$ (thermal velocities $v = 0.2$), obstacle diameter $D = 74$, flow velocities U order of 0.1, number density $n \approx 0.83$. In the case of hard spheres the time scale can not be determined from the interaction potential, so not the velocities themselves but rather the ratio of the flow velocity to thermal velocities (U/v) is the control parameter of the problem for constant density. The number density is already specified and the diameter of the particles is needed to determine the volume fraction. From the potential shown in Fig. 4.1 it can be seen that the diameter $\sigma = 1$ is a reasonable choice which also provides a convenient length scale. The alternative and more rigorous approach would be to use a hard sphere diameter estimate based on the viscosity of the soft particle system [Bir94]. In $2D$, the volume (packing) fraction is given by $\eta = (\pi\sigma^2/4)n \approx 0.65$. Other relevant values of the $2D$ volume fraction are $\eta_{\max} = \pi/\sqrt{12} \approx 0.9069$ (triangular packing), $\eta_{\mathrm{RP}} \sim 0.82 - 0.89$ (random packings, e.g. see [Ber83] and references therein) and $\eta_{\mathrm{rect}} = \pi/4 \approx 0.7854$ (rectangular packing). For discussions on the issue of packings one can also see [MJS00] and the references therein. The mean free time, corresponding to the chosen volume fraction and temperature, was measured to be 0.43, which also gives the relevant time scale of the problem.

In the present work, before an obstacle was placed in the flow, the free stream had been simulated for a time corresponding to at least 100 collisions per particle. Slip boundary conditions for the obstacle were implemented using specular reflection of particles when they collided with the obstacle. No-slip boundary conditions were realized by "re-emitting" the collided par-

ticle with a random thermal velocity corresponding to the temperature of the boundary [TTKB98]. Another variant of no-slip boundary conditions is based on "randomizing" the direction of the velocity tangential component upon a particle collision. This variant was used for the simulations involving inelasticity.

4.2 Slower flows — nearly homogeneous temperature and density

Since the subject of the present study is the behavior of the system as it develops in time, the best way to present the results of the simulations being discussed would be in the form of video clips. Such video clips composed of hundreds of individual plots were indeed created for all the simulations performed. In a text discussion only a few frames can be shown although it is not always easy to pick the ones representing the flow behavior in its fullness. In addition, since each frame is a result of a time average over a rather long time interval, the picture becomes "blurred". Of course, the word "blurring" is used not in the sense that the resulting plots are not sharp enough, but that certain fast changing pattern details "average out" while the remaining features may become distorted.

For slip boundary conditions, simulations with velocities U=0.025, 0.05, 0.1 showed the presence of a distinct "waving tail" right behind the obstacle. This pattern establishes itself during the first pass of the flow past the

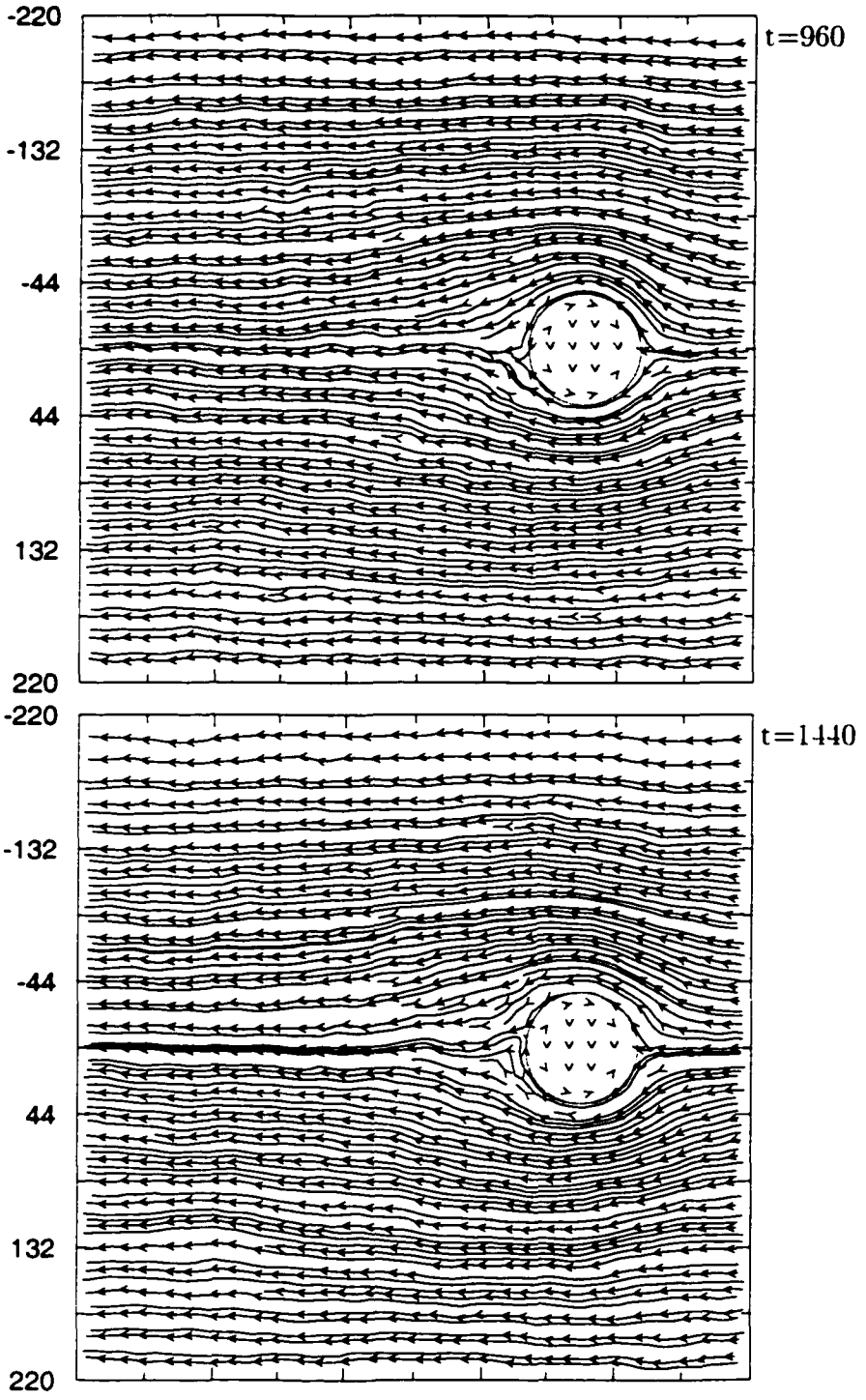

Figure 4.2: "Waving tail" formation (C=0.05, Re=6, Ma=0.036, slip)

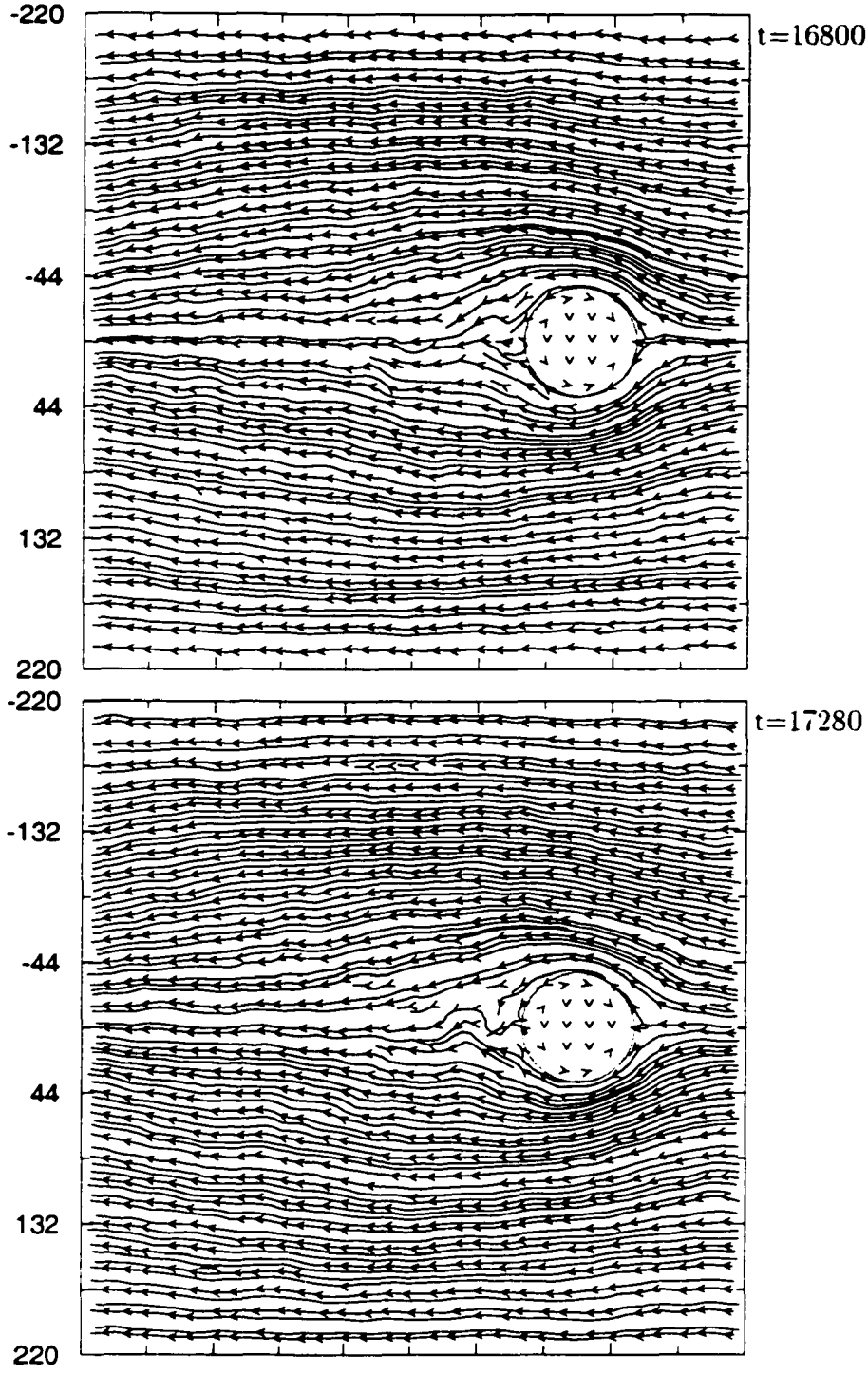

Figure 4.3: "Waving tail" (U=0.05, Re=6, Ma=0.036, slip)

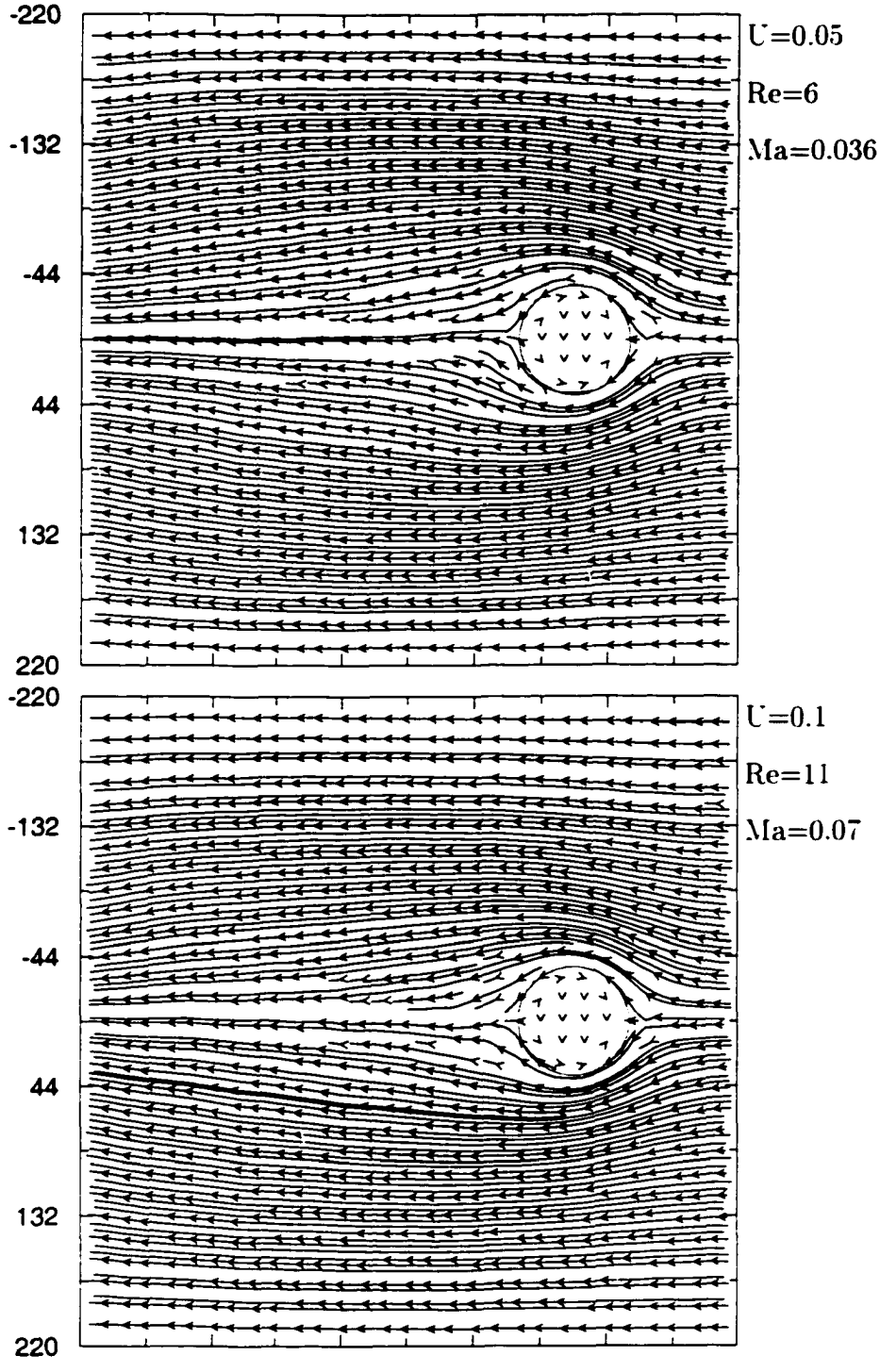

Figure 4.4: Long time averages (slip)

cylinder and persists during the whole simulation. The parameters of the simulated systems were $N \approx 1.5 \times 10^4$ (number of particles). $S_x \times S_y = 440 \times 440$ (simulation domain size). $B_{nx} \times B_{ny} = 60 \times 60$ bins (data grid size). The obstacle was placed at 110 length units from the domain center in the upstream direction.

Consider the simulation for $U = 0.05$. Fig. 4.2 shows two sequential frames (2nd and 3rd from the simulation start) each representing a time average over the interval ($\tau_E = 480$), corresponding to the mean flow passing approximately 1/3 of the cylinder diameter. This average corresponds to about 540 clsn/ptcl (collisions per particle) or 2.4×10^4 clsn/bin (collisions per data bin). The "waving tail" pattern becomes more clear after the flow passes about two obstacle sizes. This can be seen in Fig. 4.3 which shows flow patterns after the time during which the flow traveled the distance equal to 11 obstacle sizes. For the sake of brevity, the time will occasionally be specified directly in units of length which should be understood in the above given context.

The long time average pictures show the lack of upstream-downstream symmetry expected in fluid flows for the Reynolds number greater than zero. The top part of Fig. 4.4 shows the flow averaged over "exposure" $\tau_E = 17280$ or about 12 sizes of the obstacle (SO) for $U = 0.05$ at time $t = 69120$. The frame at the bottom corresponds to parameters $U = 0.1$, $t = 17280$ and $\tau_E = 4320$ ($\approx$ 6 SO). The long time averages for flows with velocity $U = 0.025$ have similar appearance.

It is rather clear that the patterns discussed here for $U = 0.1$ and down

to $U = 0.025$ are quite different from the ones in [Rap87] where the deviation from Stokes flow was not noted even for $U = 0.1$.

The simulations for no-slip boundary conditions ($U = 0.025, 0.05, 0.1$) show the presence of eddy-like structures behind the obstacle, formed and shed irregularly and asymmetrically (unlike [Rap87]) during the run. There is some weak evidence for the structures to form alternately above and below the center line. Frames representing averages over about 1/3 of the obstacle size are shown in Figs. 4.5, 4.6 and 4.7. After being shed, the structures dissipate quite fast and completely vanish within roughly one obstacle size leaving behind only a waving trail. The Knudsen number Kn = λ/L (λ-mean free path, L -characteristic dimension) is equal to Kn = 10^{-3} when calculated with L taken to be the size of the obstacle. This value is small enough to justify the applicability of continuum hydrodynamics when considering the interaction of the fluid with the obstacle. On the other hand, smaller structures in the fluid such as eddies might not involve enough particles to persist farther down the stream. It is also important to note that the temperature and density fields are nearly homogeneous. The largest temperature variations are observed in the simulation with U=0.1. These variations are localized primarily behind the obstacle and are less than 10% above the base level.

To estimate the viscosity of the media, simple planar 2D Couette flow simulations were performed. The size of the system was $S_x \times S_y = 128 \times 128$, temperature $kT = 0.02$ and the velocities of the moving boundary were 0.025, 0.05 and 0.1. The resulting value for the kinematic viscosity was

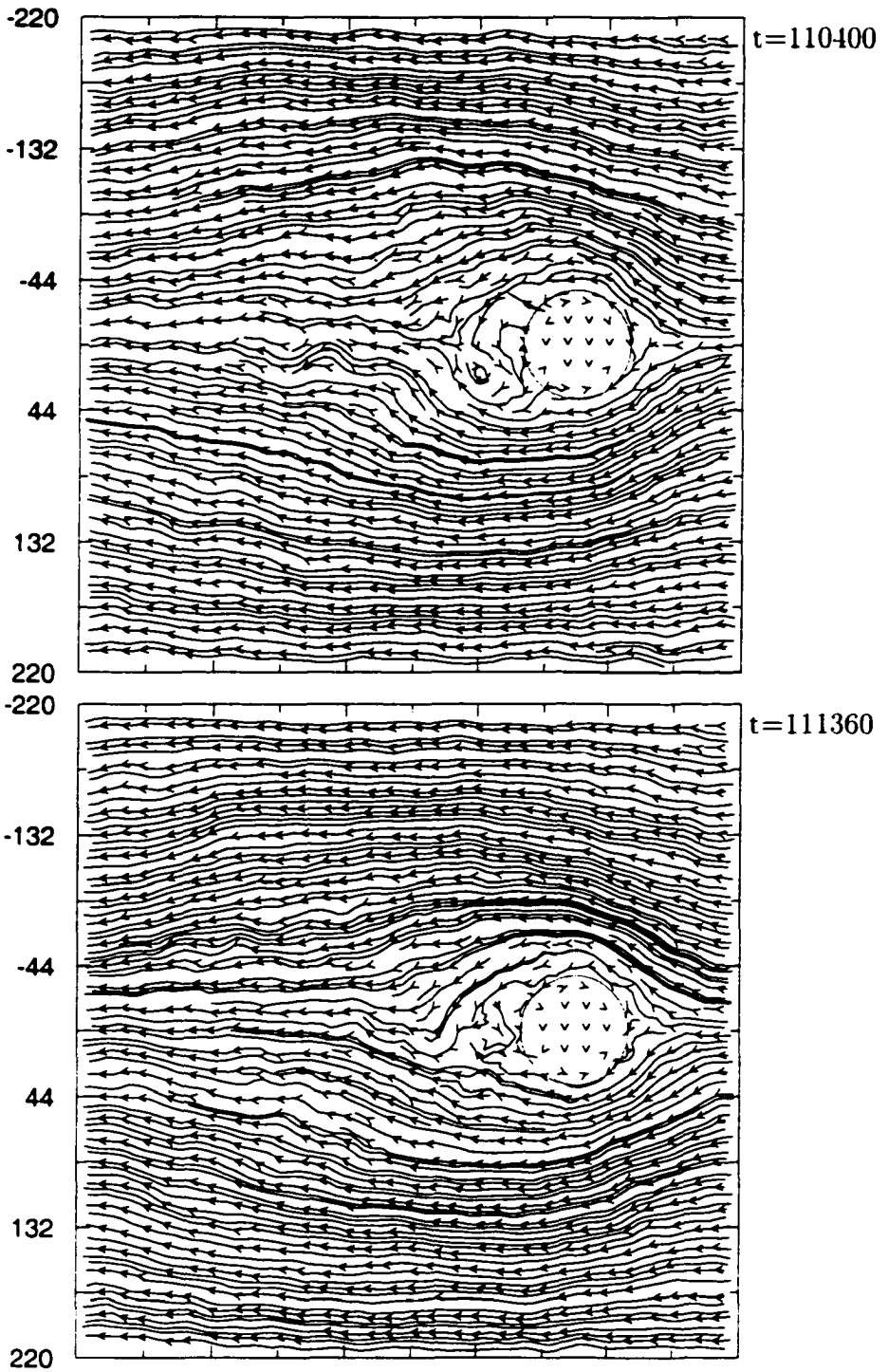

Figure 4.5: Eddy-like structures (U=0.025, Re=3, Ma=0.018, no-slip)

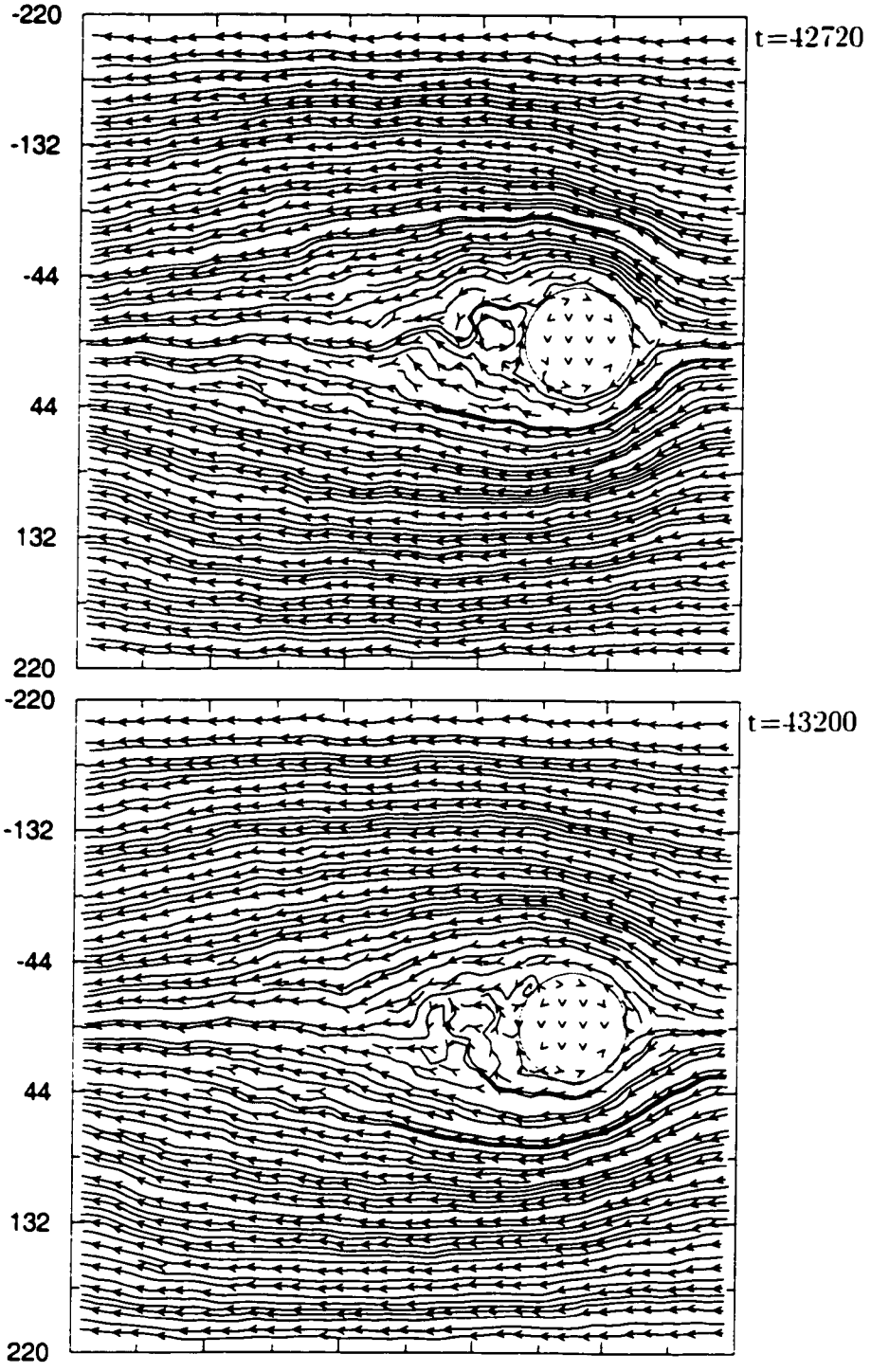

Figure 4.6: Eddy-like structures (U=0.05, Re=6, Ma=0.036, no-slip)

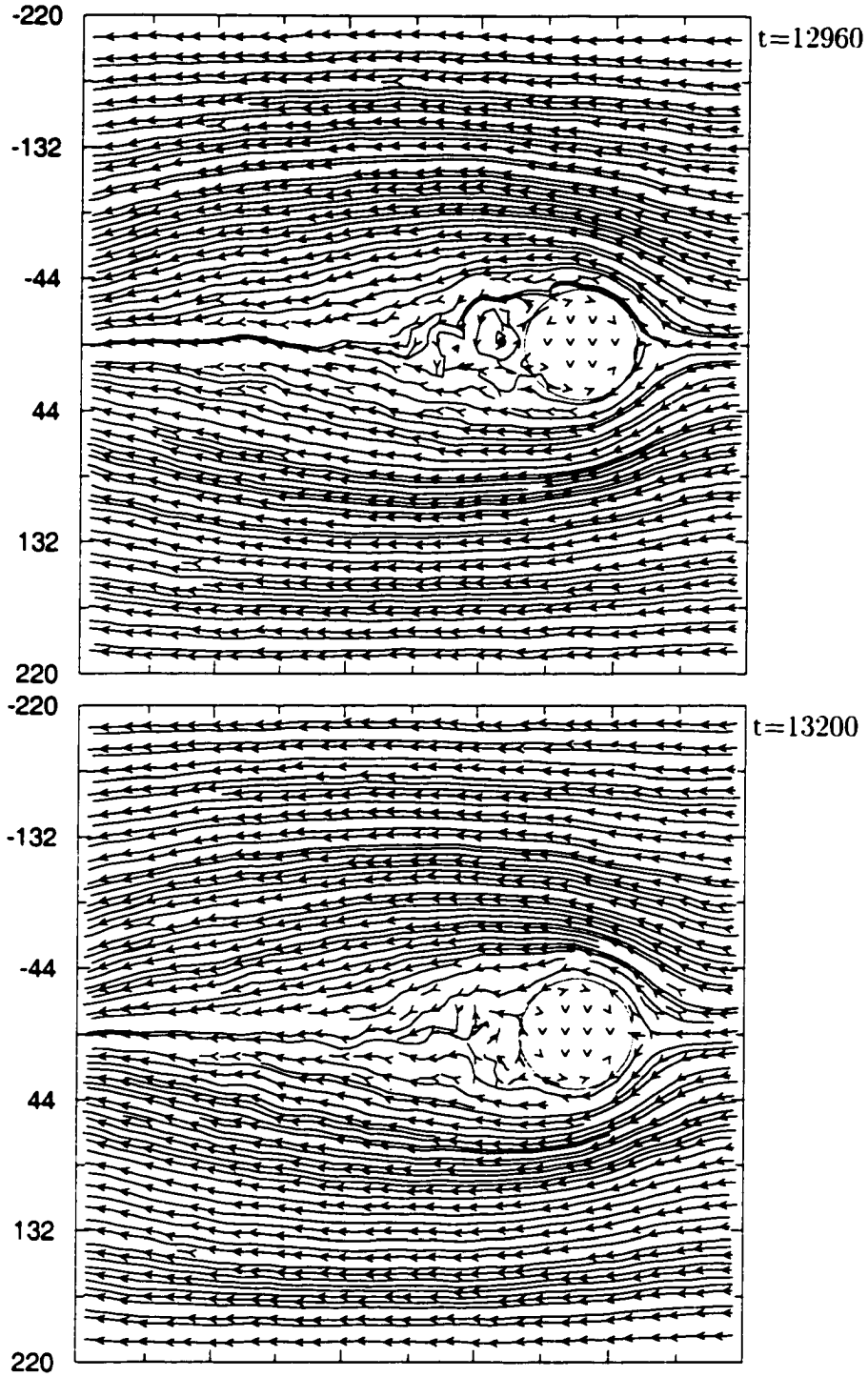

Figure 4.7: Eddy-like structures (U=0.1, Re=11, Ma=0.07, no-slip)

found to be $\nu \approx 0.65$. For the obstructed flow simulations discussed above (with $U = 0.025, 0.05, 0.1$), this viscosity estimate gives the corresponding Reynolds number range from 3 to 11.

Speed of sound Us (Mach number Ma=U/Us) was estimated by measuring the frequency of the first and second harmonics of a standing wave excited in a system (vf=0.65, T=0.02) bound by two elastic hard walls in "y" direction (periodic boundary conditions in "x" direction).

It is interesting that in "no-slip" simulations for smaller flow velocity ($U = 0.025$), the eddy-like structures appear to be farther apart in the direction perpendicular to the flow (i.e., shift sideways) compared to the simulations with $U = 0.05$ and $U = 0.1$. Of course, it would be very informative to reduce the Reynolds number even further and to see if this trend eventually leads to the well known pattern with two symmetric eddies. Unfortunately, the noise level is already rather high even for simulations with $U = 0.025$. The measurement of the velocity field is being done in the presence of the fluctuations corresponding to the given temperature $kT = 0.02$ (thermal velocities $v = 0.2$). The reduction of the Reynolds number by simply lowering the flow velocity would bring down the signal being measured and decrease the signal-to-noise ratio below an acceptable level.

The computational penalty associated with reducing the Reynolds number while keeping the signal-to-noise ratio constant can be estimated based on the following arguments. The cost of a measurement of the velocity value is determined by the size S of the velocity statistical ensemble. The ensemble can be considered to consist of a number of samples taken during the time

averaging procedure while each sample represents the average over the particles belonging to the same data bin (space average). Increasing the time average interval n times would increase the size of the ensemble (number of samples) by the same factor provided that the sampling for the average occurs at the same rate. Lets assume for simplicity that if the flow velocity is reduced by n the physics of the simulation does not change substantially and it will take n times longer to capture the same phenomena of interest. Then, to provide maximum signal without a loss of a time resolution, the time average interval can be increased by a factor of n too, i.e:

$$(U \to U/n) \Rightarrow (S \to n \cdot S) \tag{4.2}$$

If the characteristic length L is scaled by a factor of m, then it will take m times longer for the flow to pass the distance of interest (e.g. one obstacle size). Also, assuming that the space resolution (number of data bins) is kept the same, the volume of a data bin will scale by a factor of $m^{\mathcal{D}}$ ($\mathcal{D}$ -system dimension). Thus,

$$(L \to m \cdot L) \Rightarrow (S \to m^{\mathcal{D}} \cdot m \cdot S) = m^{\mathcal{D}+1} S\,. \tag{4.3}$$

Finally,

$$(\mathrm{Re} \to \frac{\mathrm{m}}{\mathrm{n}}\mathrm{Re} = \alpha \mathrm{Re}) \Rightarrow (\mathrm{S} \to \mathrm{m}^{\mathcal{D}+1}\mathrm{nS})\,. \tag{4.4}$$

When the size of the ensemble grows as N, the relative fluctuation $\sqrt{\langle(\Delta f)^2\rangle}/\bar{f}$ of some additive measurable f changes as $1/\sqrt{N}$ (assuming that there are no correlations). Thus,

$$\frac{\sqrt{\langle(\Delta U)^2\rangle}}{\bar{U}} \to \frac{1}{\sqrt{m^{\mathcal{D}+1}n}} \frac{\sqrt{\langle(\Delta U)^2\rangle}}{\bar{U}/n} = \sqrt{\frac{n}{m^{\mathcal{D}+1}}} \frac{\sqrt{\langle(\Delta U)^2\rangle}}{\bar{U}}\,. \tag{4.5}$$

Keeping signal-to-noise ratio constant would mean

$$\sqrt{\frac{n}{m^{\mathcal{D}+1}}} = 1, \; n = m^{\mathcal{D}+1}. \tag{4.6}$$

And combined with $(m/n) = \alpha$ it gives

$$m = (1/\alpha)^{1/\mathcal{D}}, \; n = (1/\alpha)^{(\mathcal{D}+1)/\mathcal{D}}. \tag{4.7}$$

So that the computational cost increase factor $m^{\mathcal{D}+1}n$ is equal to

$$\left(\frac{1}{\alpha^2}\right)^{(\mathcal{D}+1)/\mathcal{D}} = \begin{cases} \alpha^{-3}, & \mathcal{D} = 2 \\ \alpha^{-8/3}, & \mathcal{D} = 3 \end{cases}. \tag{4.8}$$

The penalty in 3D is slightly less since the length scaling leads to faster (cubic vs. quadratic) growth of a data bin volume and hence better signal. Of course, it does not change the fact that 3D simulations are usually much more computationally expensive to start with.

Thus, reducing the Reynolds number ($\alpha < 1$) while avoiding further deterioration of the already weak signal (simulation with $U^* = 0.025$) would lead to a cubic increase in computational time. Although of some interest, those studies were computationally prohibitive at the time of the research discussed here.

4.3 Faster flows — strong temperature and density variations

On the other hand, increasing the flow velocity leads to shorter computation times and better signal-to-noise ratio. In the system with no-slip boundary conditions for the obstacle, the patterns change dramatically when the

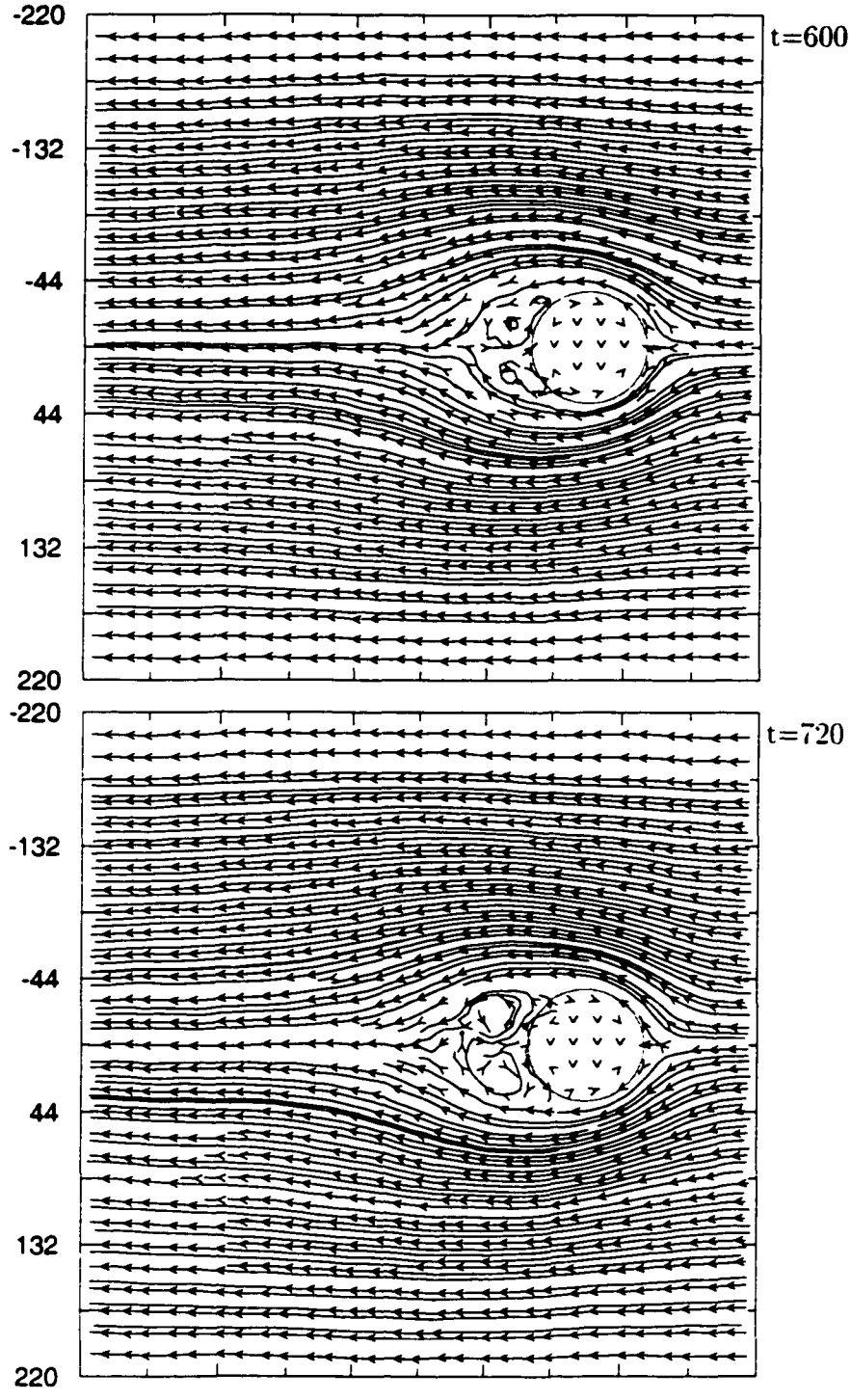

Figure 4.8: Swirl formation (U=0.2, Re=23, Ma=0.14, no-slip)

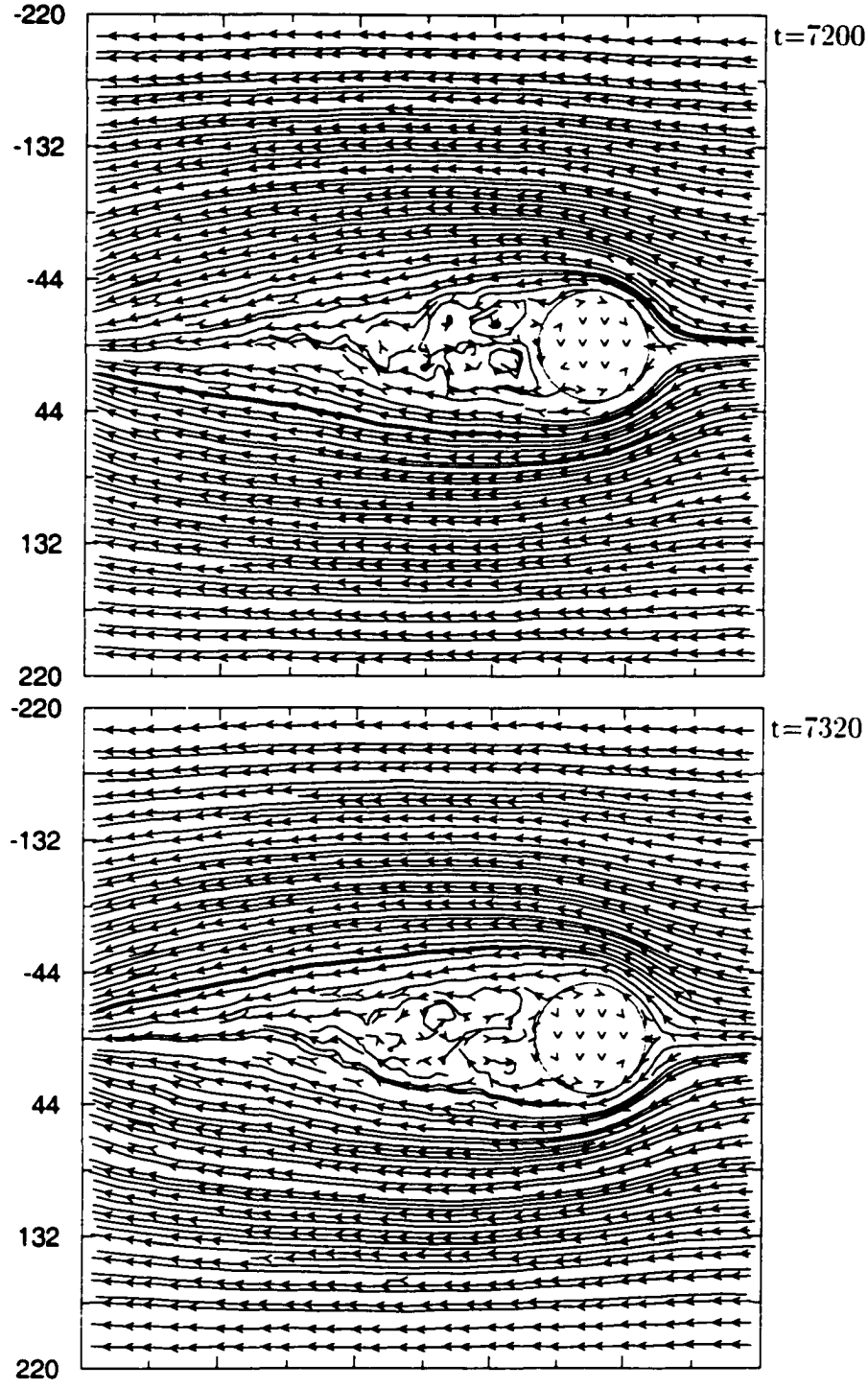

Figure 4.9: Swirl dynamics (U=0.2, Re=23, Ma=0.14, no-slip)

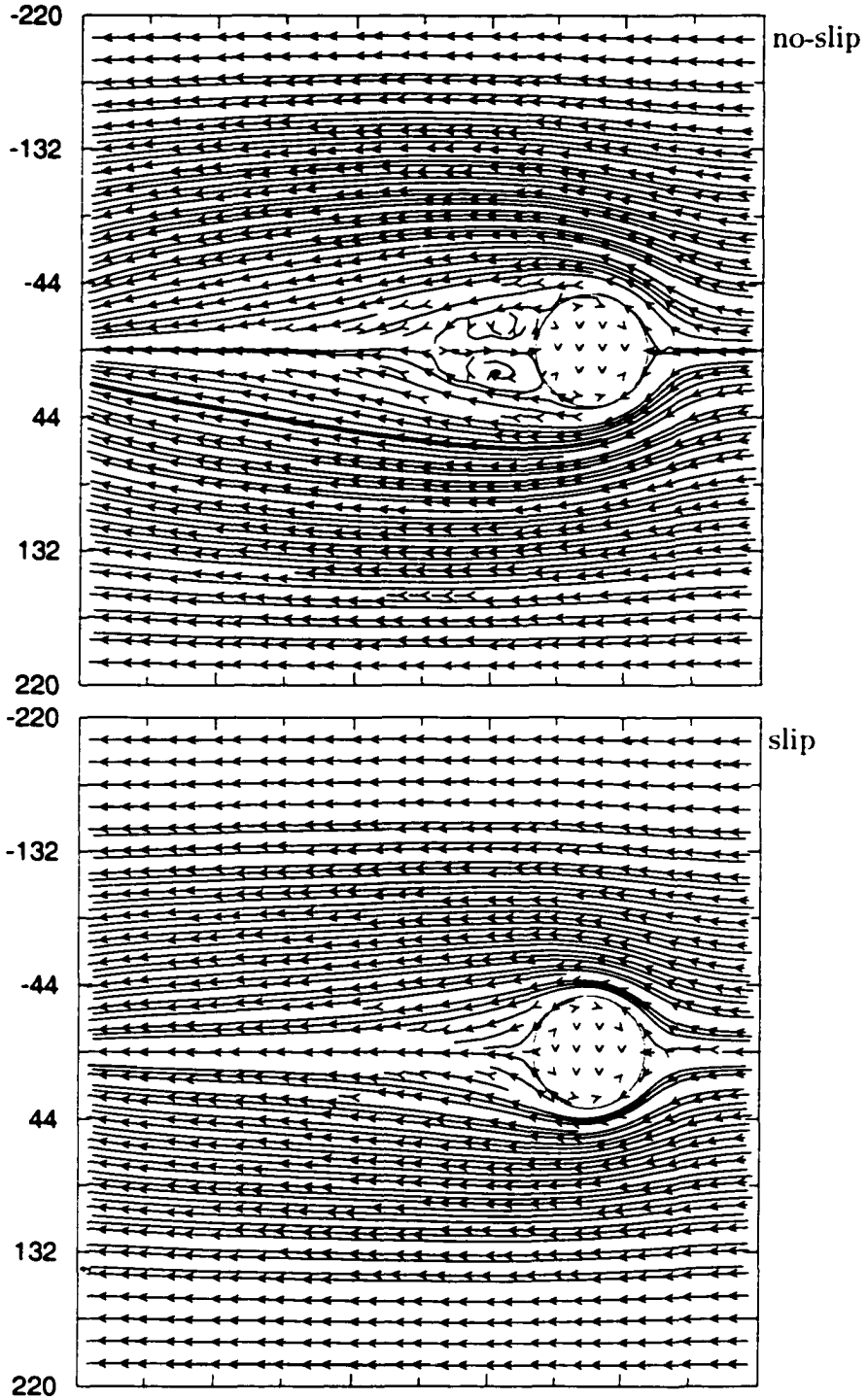

Figure 4.10: Long time averages (τ_E=8640). U=0.2, Re=23, Ma=0.14

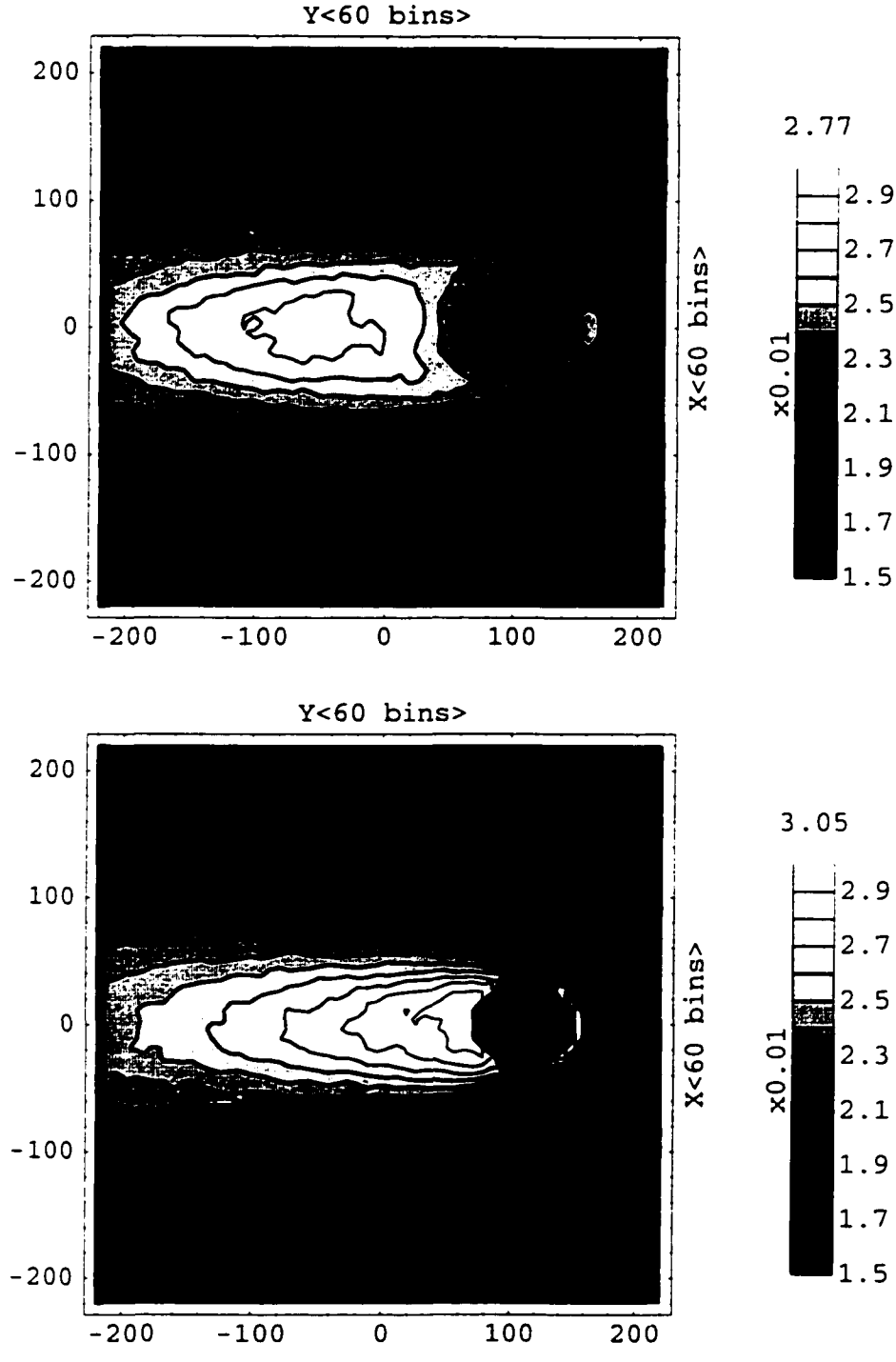

Figure 4.11: Temperature (background: 0.02). U=0.2 (no-slip, slip).

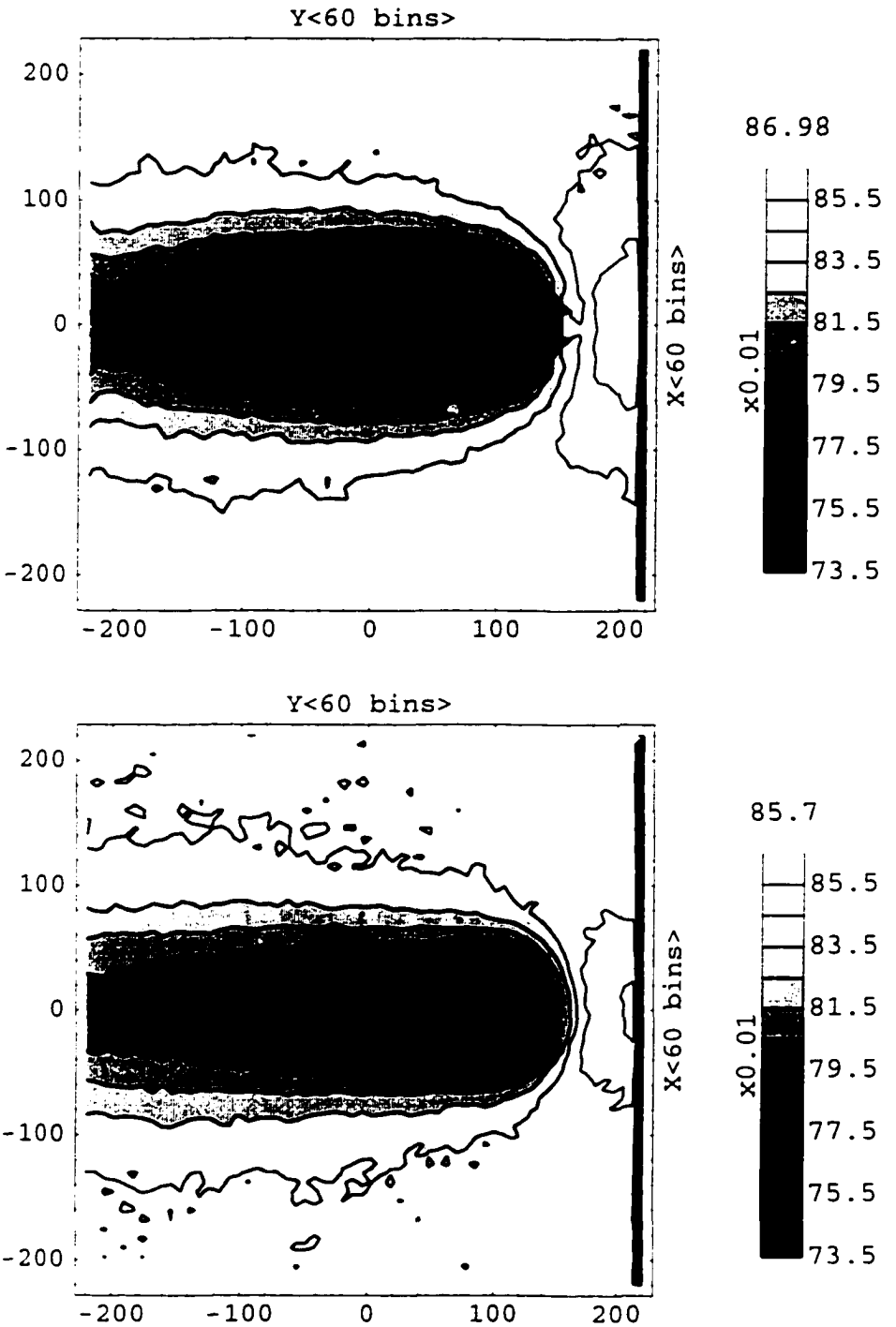

Figure 4.12: Number density (τ_E=8640). U=0.2 (no-slip, slip).

simulations are performed with $U = 0.2$. At the end of the first pass past the obstacle, two symmetric eddy-like structures start forming behind the obstacle. The formation is complete by the end of the second pass (Fig. 4.8) after which the symmetry is broken and the shedding of the structures is accompanied by the formation of new ones (Fig. 4.9). In this simulation the structures persist significantly farther down the stream (about 2 obstacle diameters), and often they disintegrate into smaller but still observable components which eventually dissipate. It is very interesting that, although the short time average frames lack the symmetry due to the continuing process of shedding and forming new structures, the long time averages clearly show the presence of two symmetric counter-rotating swirls (Fig. 4.10). The drastic change of patterns is most probably related to the fact that at this flow speed the system temperature and density change rather strongly as shown in Figs. 4.11 and 4.12 (τ_E=8640). In the context of the present work the term temperature (kT) is defined in terms of peculiar velocities C (e.g. see [CC70]) as

$$\frac{i}{2}kT = \frac{1}{2}m\overline{C^2}. \tag{4.9}$$

where $i = 2$ is the number of degrees of freedom for disks.

It is important to notice that the patterns in the case of slip boundary conditions remained essentially unchanged at $U = 0.2$. The very short lived and small structure (about 1/5 of the obstacle diameter) was observed only once through the whole run (about 24 obstacle sizes). The fact that in the "slip case" the maximum temperature difference is higher might indicate that

the no-slip conditions at the obstacle boundary play very important role in triggering the process of swirl generation discussed here.

When the flow speed is increased to $U^* = 0.4$, the situation becomes more dramatic. In a partially successful attempt to prevent the temperature increase of the whole domain, the length of the simulation box sides was doubled and consequently the number of particles went up four times ($N \approx 6.4 \cdot 10^4$).

The initial development stages for the "no-slip" case are similar to the ones before, but the size of the swirls is larger (Fig. 4.13). At the end of the initial formation, the elongated counter rotating structures reach about 2 obstacle diameters in the flow direction and 3/4 obstacle diameter in the transverse direction. After that the process of swirl generation and dissipation takes over (Fig. 4.14). It is important to notice that, although the simulation domain could accommodate more swirls down the stream, the region of their existence appear to be limited by the high temperature (low density) region (Figs. 4.16, 4.17, 4.18, τ_E=4320). In spite of all the dynamic activity in the region and the rather large size of the structures involved, the long time average picture still shows two well defined symmetric counter rotating swirls (Fig. 4.15).

Finally, the developments in the "slip" case show close resemblance to the patterns described by Rapaport [Rap87] (see also [RC86]). By the end of the second pass of the flow past an obstacle, two well defined symmetric counter rotating swirls are formed. At this point one of the swirls starts decreasing in size and by the end of the "third pass" it becomes rather difficult to

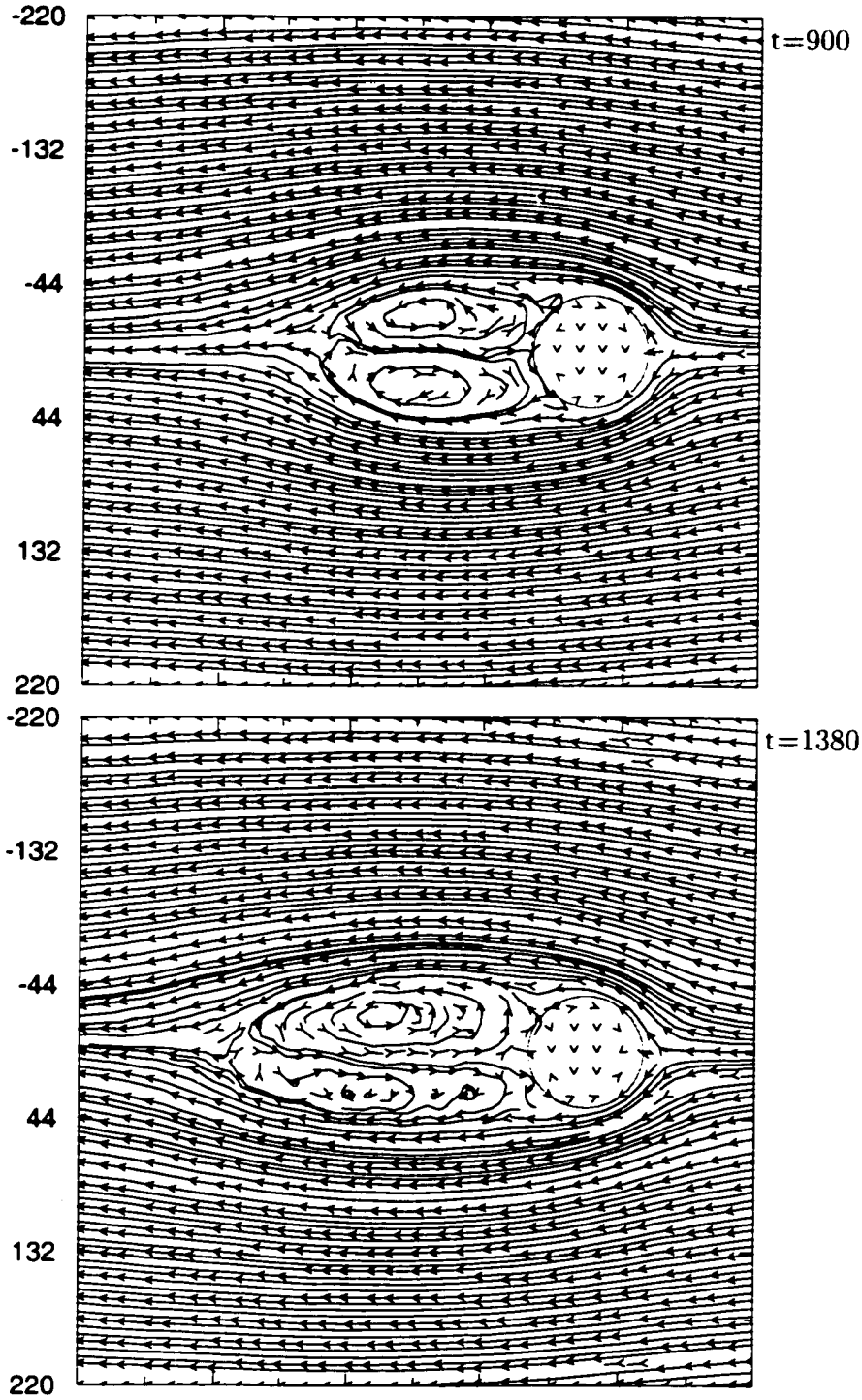

Figure 4.13: Swirl formation (τ_E=60, U=0.4, Re=46, Ma=0.28, no-slip)

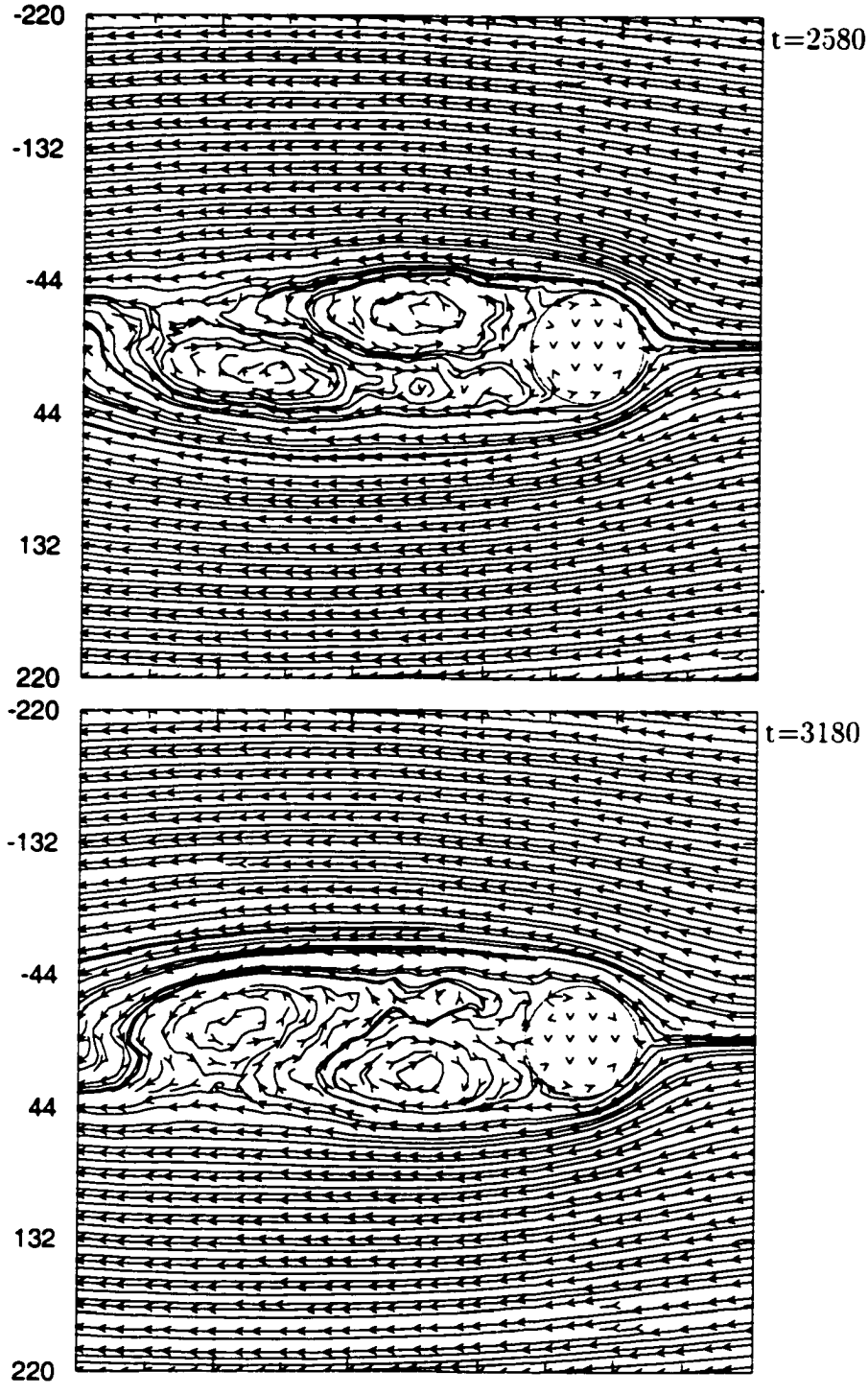

Figure 4.14: Swirl dynamics (τ_E=60, U=0.4, Re=46, Ma=0.28, no-slip)

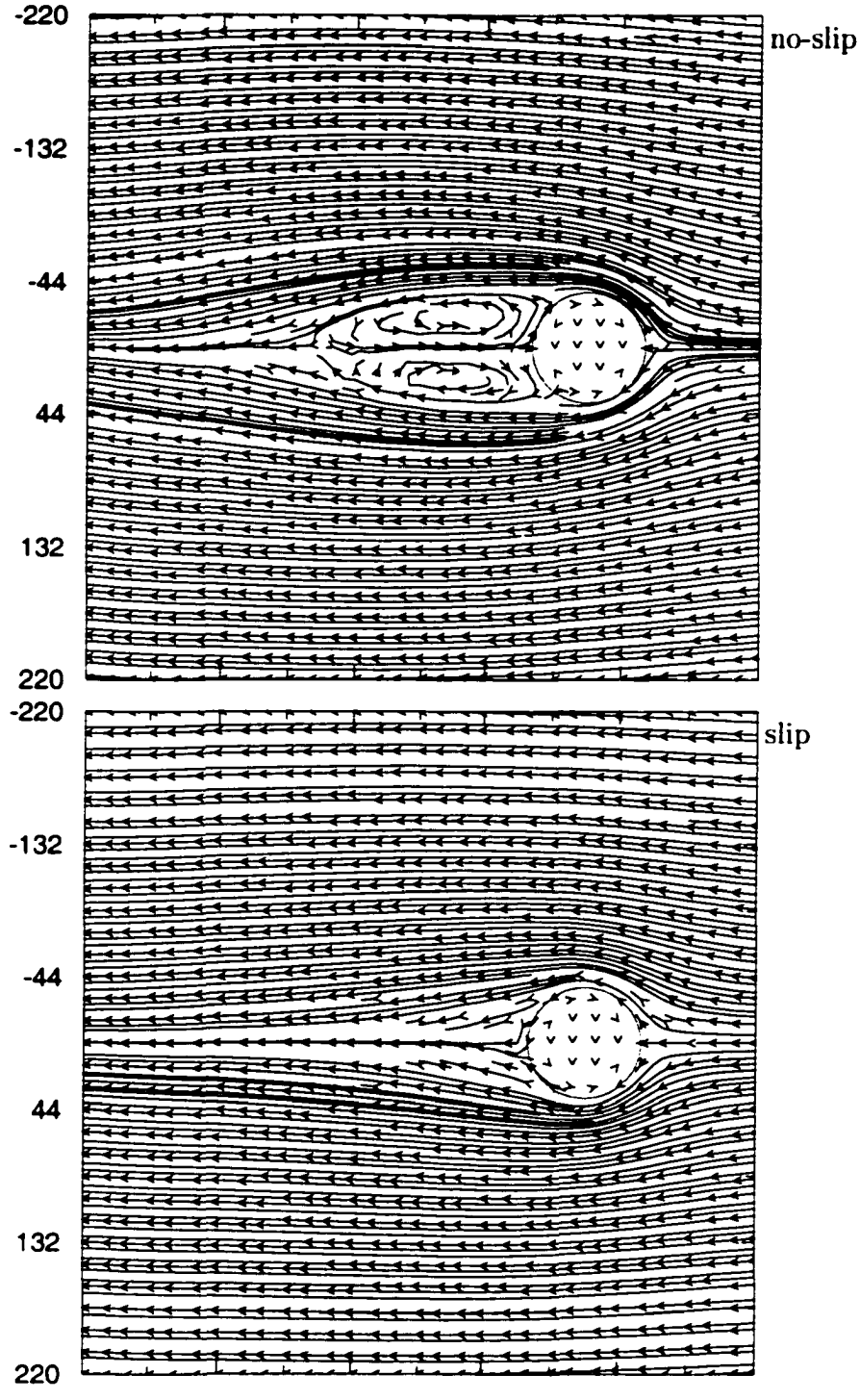

Figure 4.15: Long time averages (τ_E=4320). U=0.4, Re=46, Ma=0.28

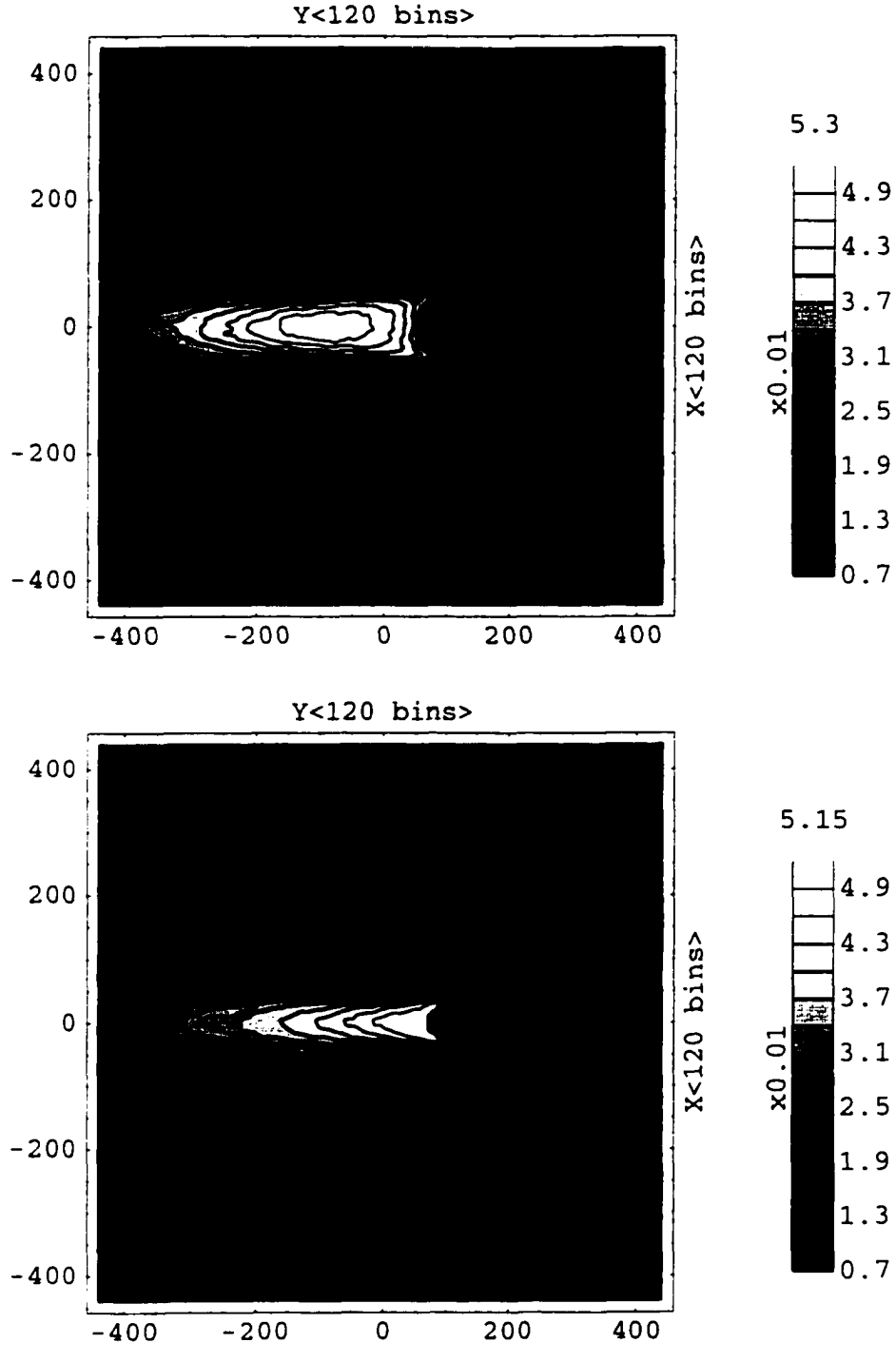

Figure 4.16: Temperature (background: 0.02). U=0.4 (no-slip, slip).

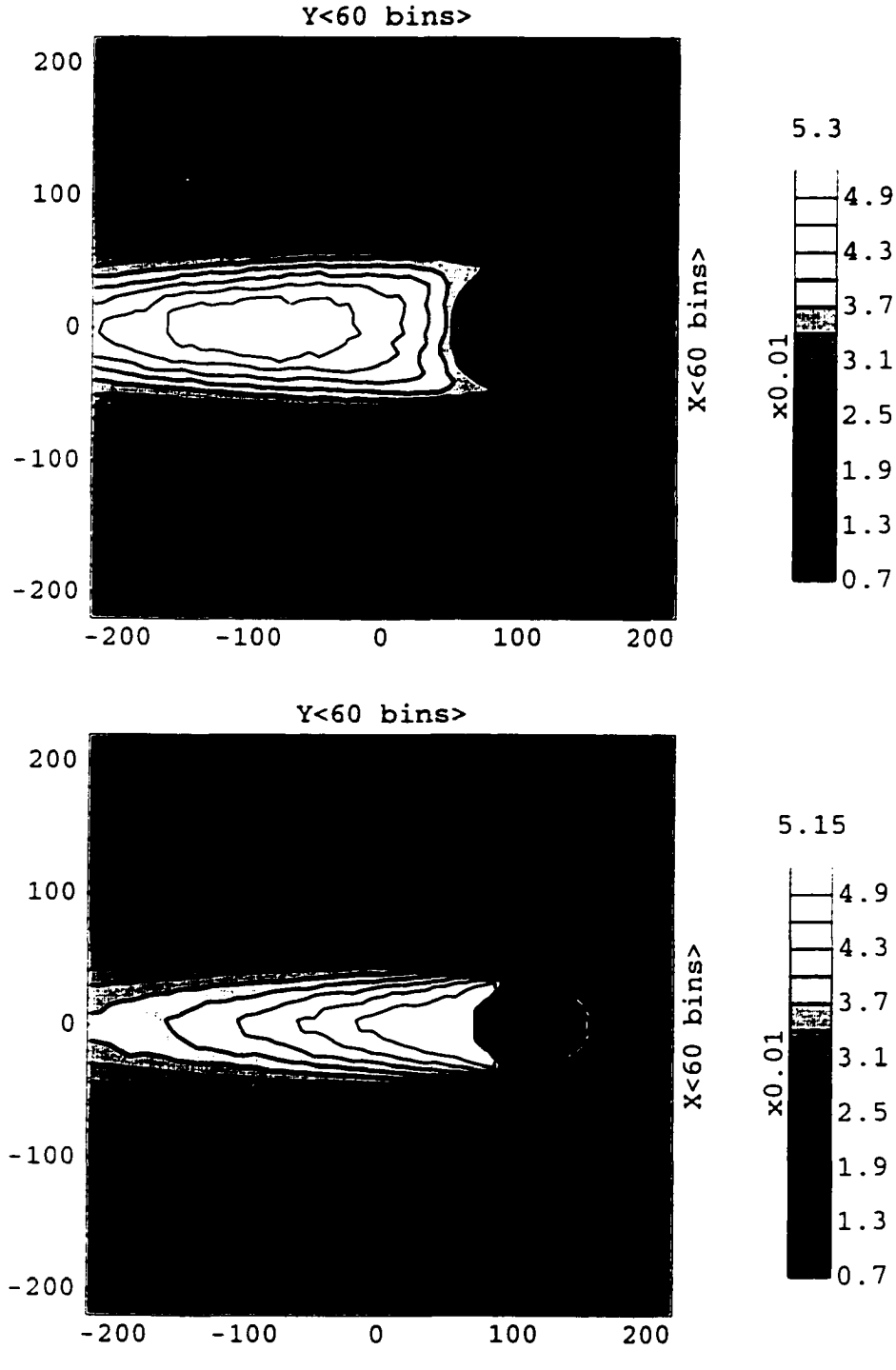

Figure 4.17: Temperature (background: 0.02). U=0.4 (no-slip. slip).

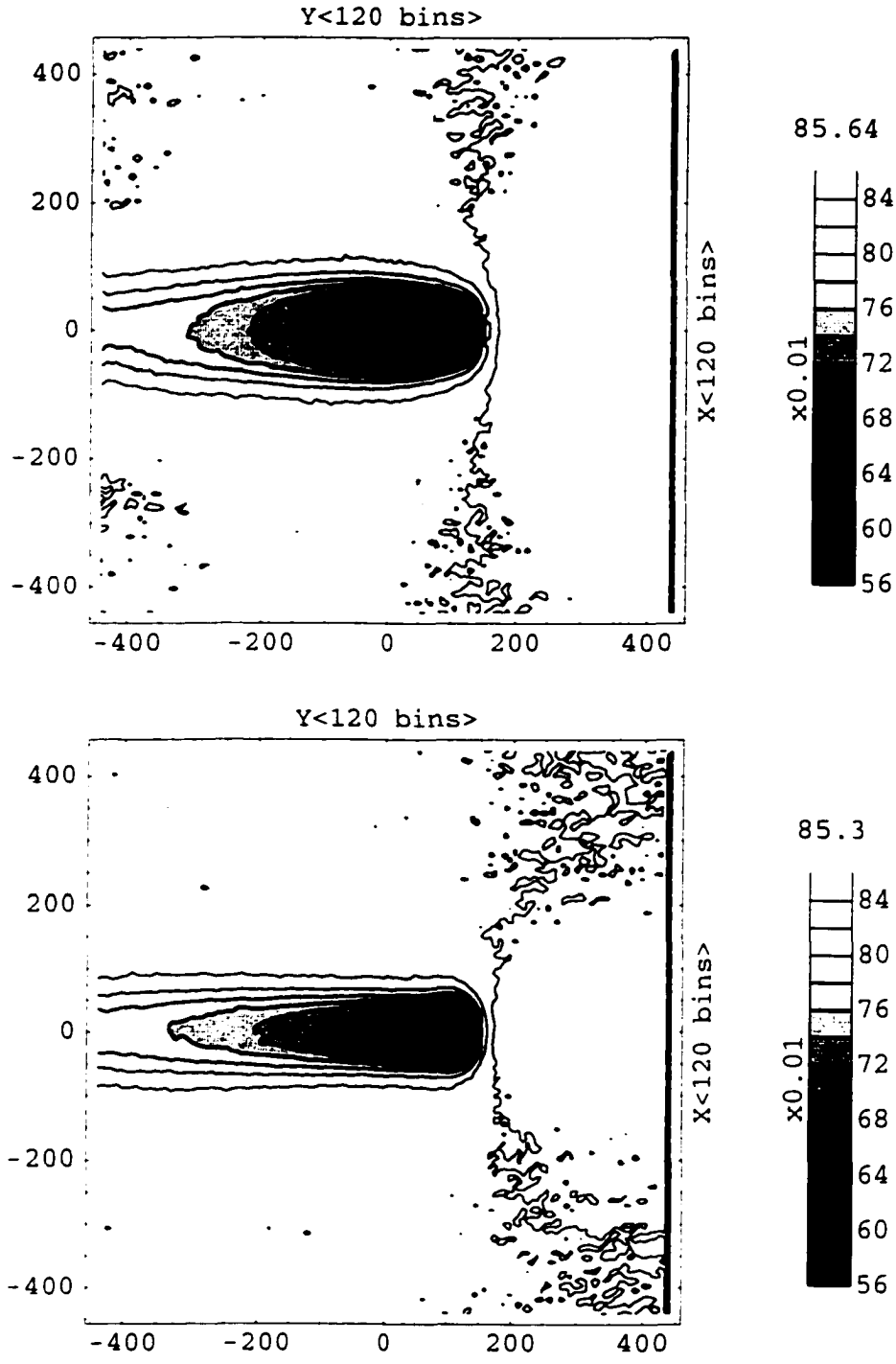

Figure 4.18: Number density (τ_E=4320). U=0.4 (no-slip. slip).

identify the corresponding pattern (Fig. 4.20). Small and short-lived eddy-like structures (Fig. 4.21 top) can still be identified for about four next flow passes, but eventually the "waving tail" pattern prevails (Fig. 4.21 btm). Any other structures, which sometimes can be seen right behind the obstacle, are too small and irregular to analyze, although it seems that the "waving tail" itself is a manifestation of shedding and dissipation of such formations.

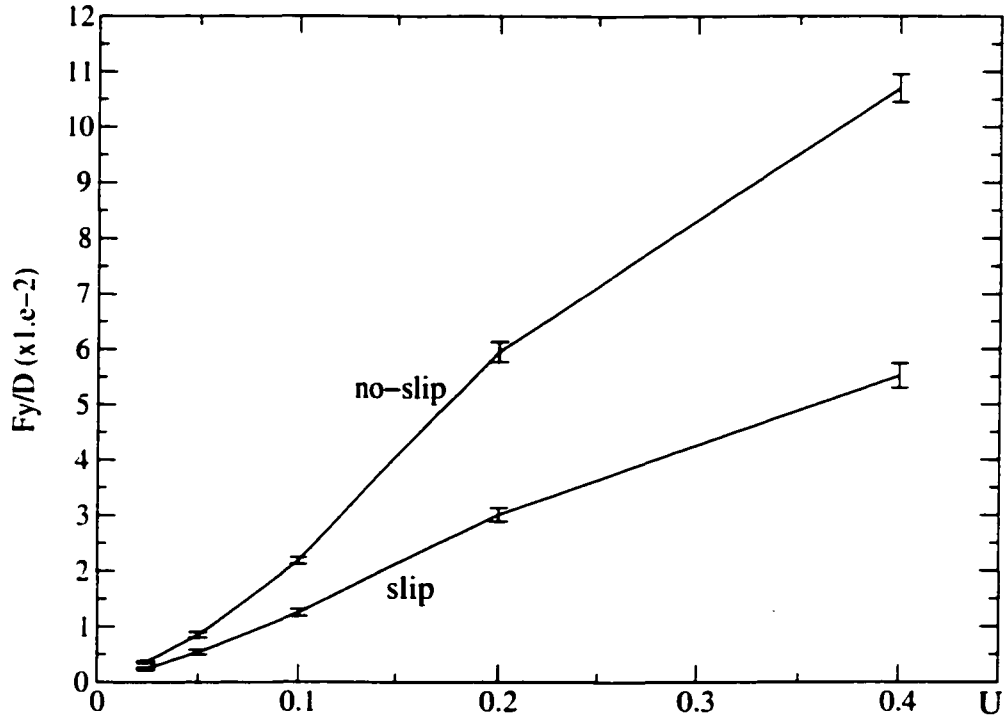

Figure 4.19: Drag force

The drag force (divided by diameter D) on the obstacle is shown in Fig. 4.19 (error bars represent standard deviations). Since there are very few data points, it is rather difficult to identify the flow pattern transition in the drag force dependence. The significant changes in the force dependence

which seem to occur in the velocity range 0.1-0.2, should be treated with caution since in the simulations with U=0.4, periodic boundaries are much farther apart as compared to the simulations of slower flows.

Taking into account the presence of the regions with large temperature differences, as well as the presence of eddy-like structures in the low velocity "no-slip" cases, the interpretation of the flow patterns suggested in [RC86] and [Rap87] does not appear to be correct. Nevertheless, more definitive information could be obtained from the simulations of the flows with velocities low enough to prevent dramatic changes of temperature. One of the ways to keep the Reynolds number constant while reducing the flow velocity is to increase the obstacle size and thus the total size of the system to accommodate the obstacle. Therefore, it is important to choose a reasonably high velocity which is still low enough to avoid drastic changes in temperature. At the same time, the reference simulation (among the ones discussed above) should be chosen to have the lowest Reynolds number corresponding to the appearance of patterns of interest. Since the formation of relatively large, prominent and symmetric swirls was observed already in the "no-slip" case with $U = 0.2$ and obstacle diameter $D = 74$ (the reference case) the simulation with $U = 0.05 = (1/4)\ 0.2$ and $D = 296 = 4 \cdot 74$ can present an interesting study for the comparison. The parameters of this system are $N \approx 2.5 \cdot 10^6$ (number of particles), $S_x \times S_y = 1760 \times 1760$ (simulation domain size), $B_{nx} \times B_{ny} = 60 \times 60$ bins (data grid size). The obstacle was placed at 440 length units from the domain center in the upstream direction.

This configuration is computationally rather expensive. In the reference

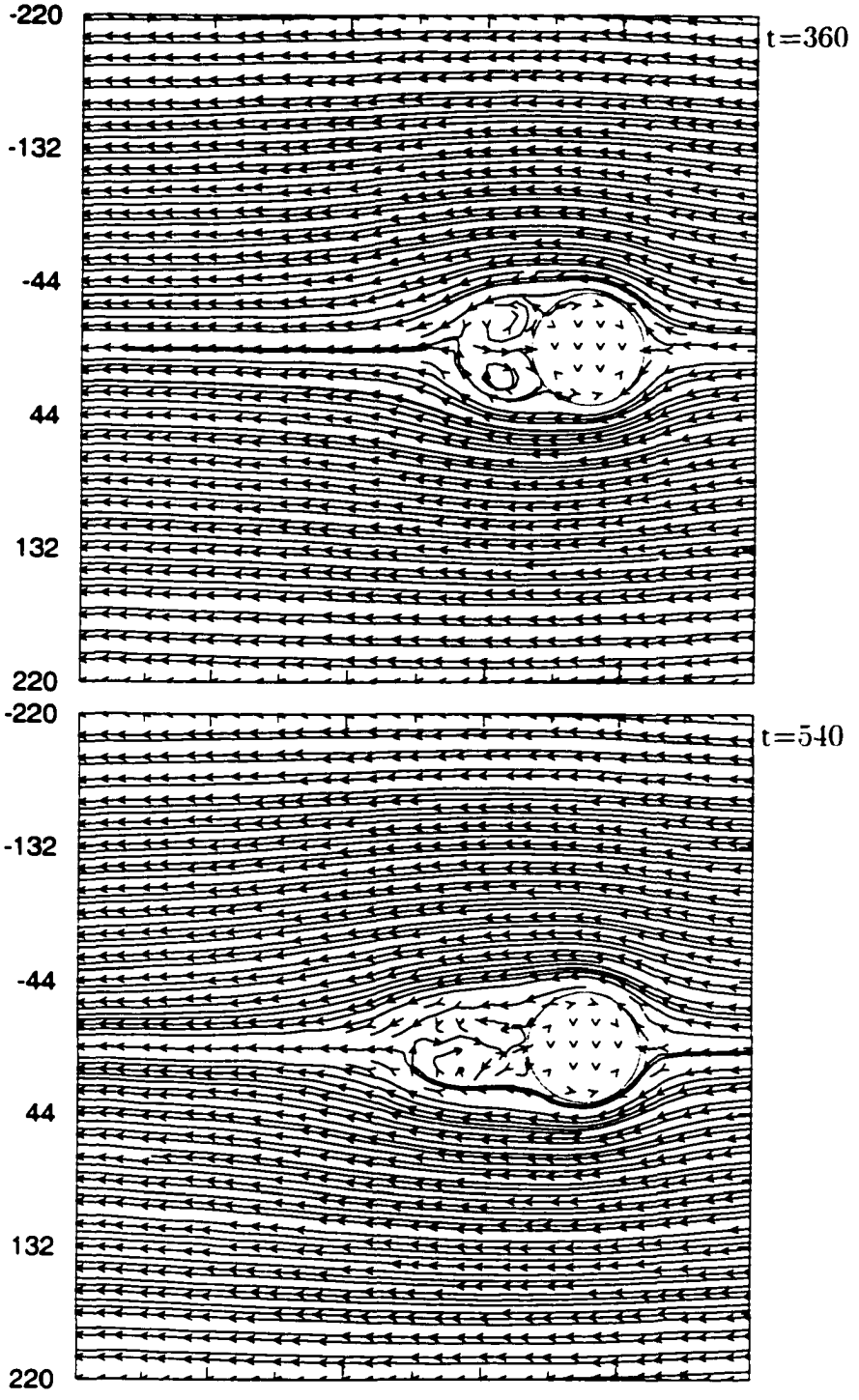

Figure 4.20: Transient swirls (τ_E=60, U=0.4, Re=46, Ma=0.28, slip)

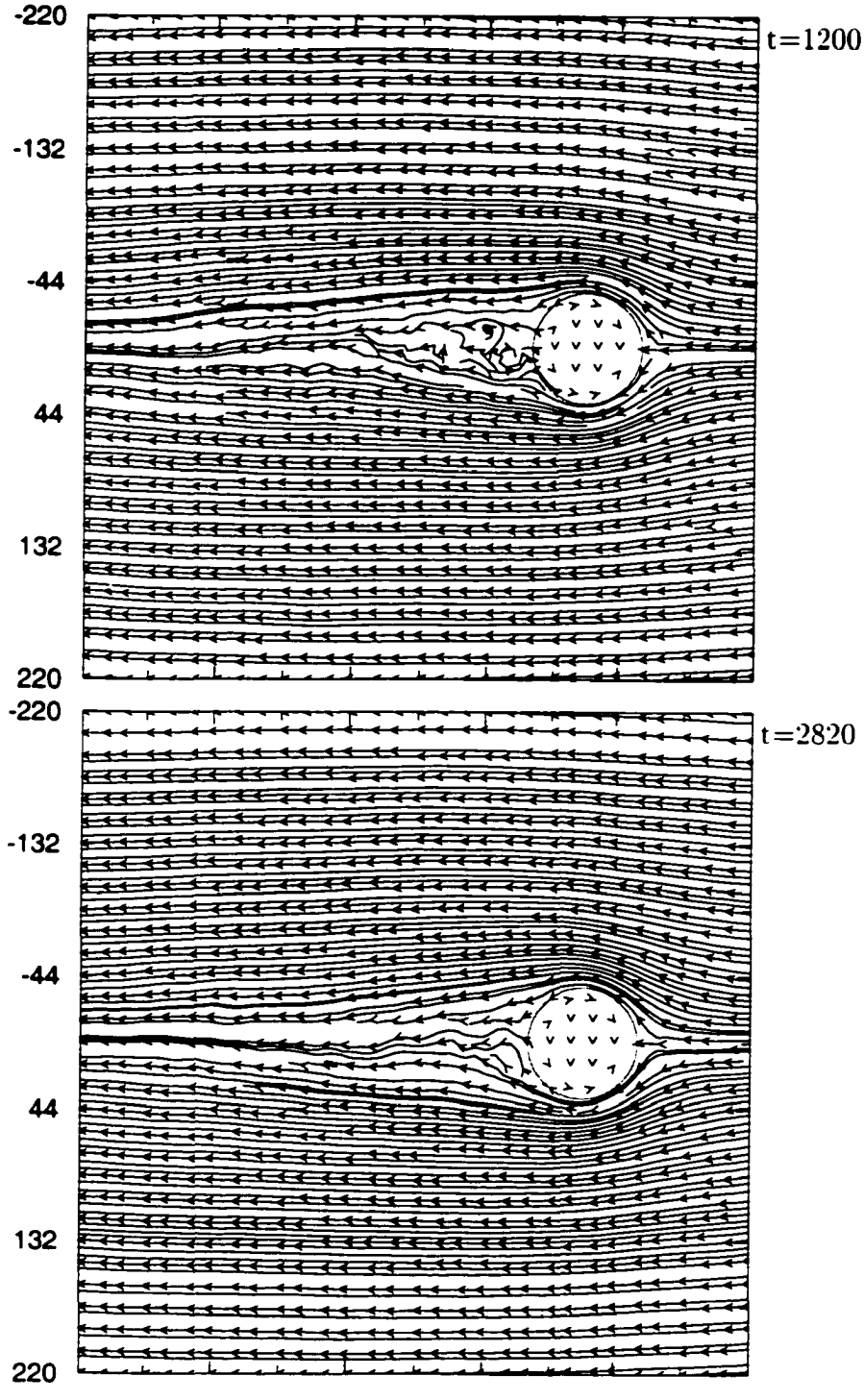

Figure 4.21: "Waving tail". (τ_E=60, U=0.4, Re=46, Ma=0.28, slip)

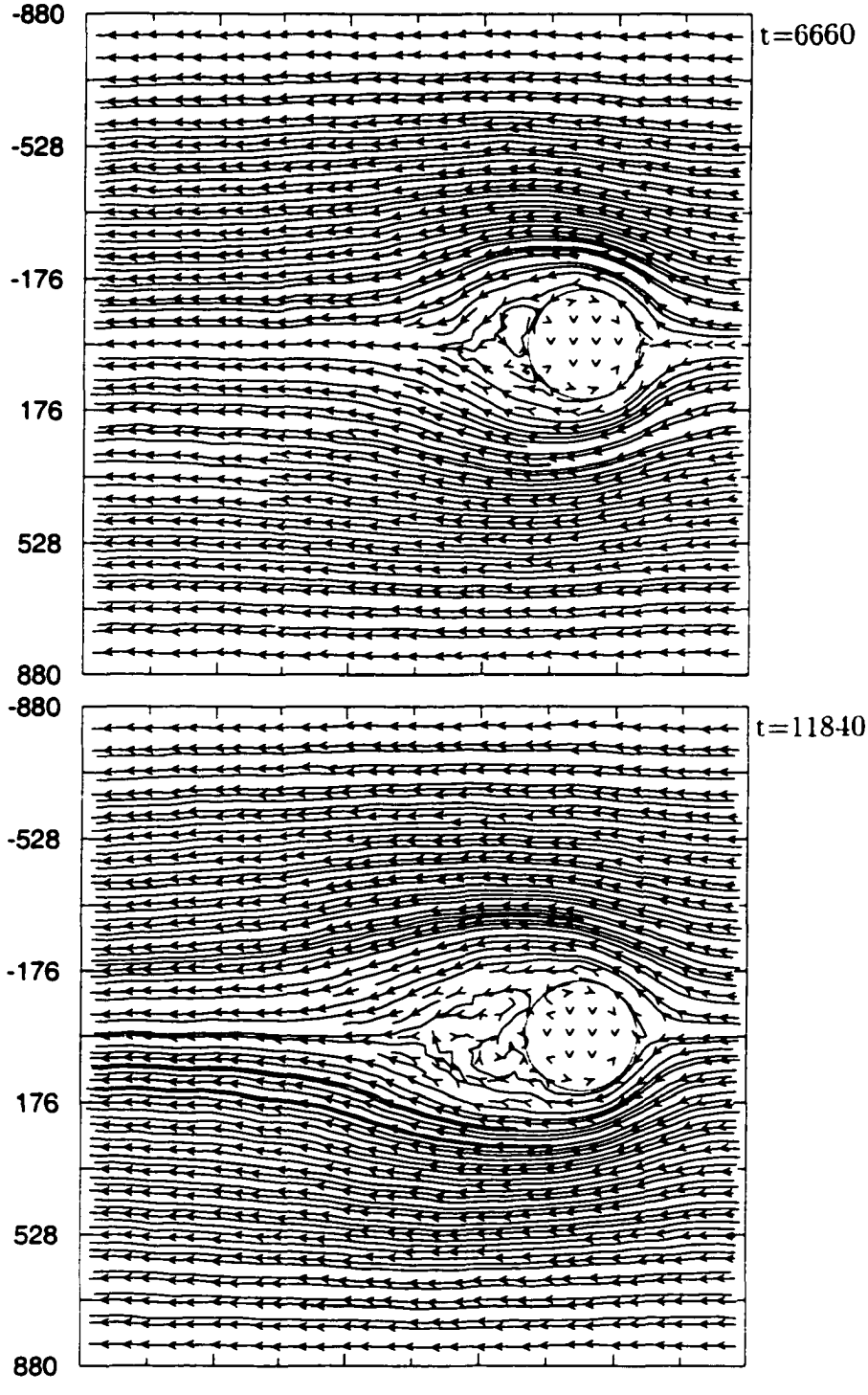

Figure 4.22: U=0.02/4. D=74 · 4. Re=23. Ma=0.036. no-slip (τ_E=740)

case the development of the symmetric swirls is completed by the end of the second pass of the flow past an obstacle. Thus, the comparison case can be limited to the same number of passes which partially alleviates the computational burden. The most important result of this simulation is that while the product $U \cdot D$ is the same as in the reference case, no symmetric swirl formation could be seen. The eddy-like structures, which start appearing by the end of the first "pass" (Fig. 4.22), do not look similar to the well defined and persistent swirls of the reference case. The simulation is too short to make any definitive statements about the similarity of the current patterns to the ones observed in the "no-slip" cases with $U = 0.025, 0.05, 0.1$ and D=74, but among all the cases discussed here they seem to be the only ones suitable for analogy. Both the temperature and the density fields do not exhibit any significant features. The average of the velocity field over the continuation of the whole run does not reveal any eddy-like structures as well.

To complete the current discussion, it is worthwhile to briefly describe another simulation which should have the Reynolds number 50% higher than the case just discussed. The parameters are $N \approx 1.4 \cdot 10^6$ (number of particles), $S_x \times S_y = 1320 \times 1320$ (simulation domain size), $B_{nx} \times B_{ny} = 60 \times 60$ bins (data grid size). The obstacle D=222, placed at $y = 330$ upstream: $U = 0.1$. The computation was performed to simulate about 12 passes of the flow past the obstacle.

The temperature variations were relatively small and the maximum was only about 10% above the base level. Nevertheless, for the no-slip boundary

conditions this increase was enough to cause the swirl patterns similar to the cases with $U = 0.2, 0.4$ and $D = 74$. For the slip boundary conditions, no deviation from the "waving tail" pattern was noticed.

Chapter 5

Impact of inelasticity

The next logical step towards studying systems of inelastic particles is to simply "turn-on" the inelasticity by setting the restitution coefficient less than 1. Due to the loss of particle kinetic energy in collisions, the temperature of the system drops rapidly (see e.g. [LHMZ98]) compared to the time scale of a "macroscopic" motion of the flow past the obstacle, and the flow becomes "very fast" ($U/\sqrt{kT}$ is large).

To prevent "inelastic collapse" described in [MY94] and [MY96], Luding and McNamara [LM98] proposed a time cutoff (TC) model. In this model an elastic collision is performed if a particle experiences a collision and the next collision for this particle occurs sooner than a given cutoff time τ_{TC}. Since these elastic collisions store energy, they are analogous to enduring elastic contacts. This analogy can be used to connect the value of τ_{TC} and the elastic properties of the particles ([LM98]), but in the present work the model is simply used for the purpose of avoiding inelastic collapse.

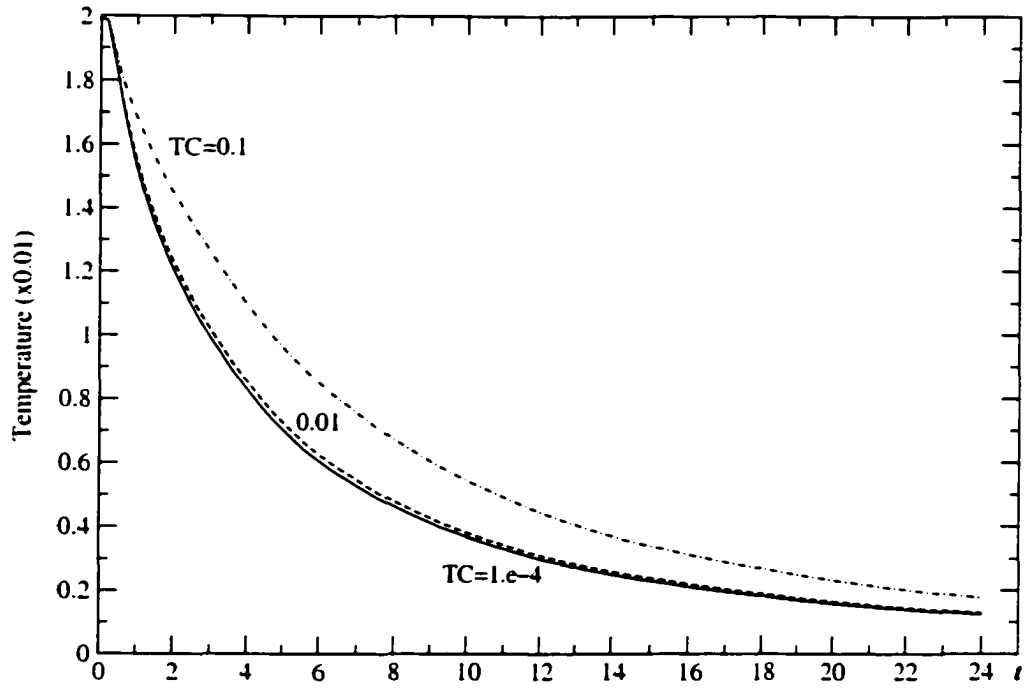

Figure 5.1: Cooling down for TC model ($\tau_E = 0.2, r = 0.87$).

A simulation starts with a relatively small value of τ_{TC}. When the collision time scale in the simulations becomes very small for a given precision, which is usually manifested through partial overlap of particles (i.e., would give a negative next event time), the "offensive" particle is removed from the system. The value of τ_{TC} is increased by some specified amount and the simulation continues. This allows one to avoid a trial and error routine for determining the value of τ_{TC} that removes the computational "singularity". Since the "destruction" of "offensive" particles happens rarely (unless τ_{TC} is poorly chosen to be very small), the density of the system re-

mains essentially unaffected. To illustrate the time scales involved, Fig. 5.1 shows three temperature vs. time (cooling down) curves measured for a small ($S_x \times S_y = 128 \times 128$, $\eta = 0.65$) system with periodic boundary conditions.

Many aspects of interaction of the fast granular flows with an obstacle have already been addressed in [BP98] (estimated average area fraction 0.4) and [RBSS02] (area fraction 0.018). The present work focuses on the initial development of the shock and the comparison of the "dune" described in experiments of [ABK01] with the one of the simulation.

The issue of boundary conditions at the obstacle needs to be re-examined. In case of granular materials the thermal (in granular sense) excitation is very often provided through some kind of vibration or shaking. Thus, the constant temperature boundary would imply some sort of excitation produced at the simulated boundary. One can easily foresee the drastic effect of such "hot" obstacle in the midst of a "cold" flow.

For example, as a short demonstration of what happens during the initial stages of the flow development, Fig. 5.2 shows two velocity field frames. In this case the original test system is the one described above (Fig. 4.5) with $U = 0.025$ and no-slip (thermal) boundary conditions (BC). After the inelasticity was "turned on" ($r = 0.97$), the system starts cooling down, and by the time $t \approx 48$, the thermal energy is reduced by a factor of 7. During that time the flow passes less than 2% of the obstacle diameter. The next "standard" ($\tau_E \approx 1/3$ of an obstacle) frame already shows a pattern very different from the ones in the elastic case. The role of an obstacle as a thermal energy source is now much more prominent, and the overall pattern simply

reflects the fact that the particles close to the obstacle are moving out to the "colder" regions. This leads to the creation of a low density region around the obstacle (Fig. 5.2 at the bottom). Although there are not enough particles in this material depleted region for the proper statistical measurement of the velocity field, they are the ones that keep transferring thermal energy from an obstacle to the outer regions. The presence of similar very low-density regions was previously observed in the non-uniformly excited granular media (e.g. see [DLK95], [GZBN97] and [EP97]). The further progress of the shock is largely distorted because the size of the simulation domain is not sufficient anymore to accommodate the major flow developments. Nevertheless, the low-density region around the obstacle persists through the whole run (about 12 obstacle sizes). Two other simulations, which differed from the one above only in the obstacle boundary conditions, have also shown large distortions caused by the limitations of the relatively small simulation domain.

To reduce the effect of the region boundaries on the shock development, the linear size of the domain of the next simulation was increased 3 times ($N \approx 1.4 \cdot 10^6$). To exclude the effects of thermal excitation at the obstacle boundary while providing the "no-slip" boundary condition, the sign of the tangential to the boundary velocity component of the particles was changed at random (uniform deviate) upon the collision with the obstacle. The "slip" case was implemented by using regular "billiard-like" collision rule for the particles colliding with the obstacle. The restitution coefficient for this type of collisions was chosen to be equal to the one for particle-particle collisions, i.e., $r = 0.97$. To provide the opportunity to observe the change of patterns

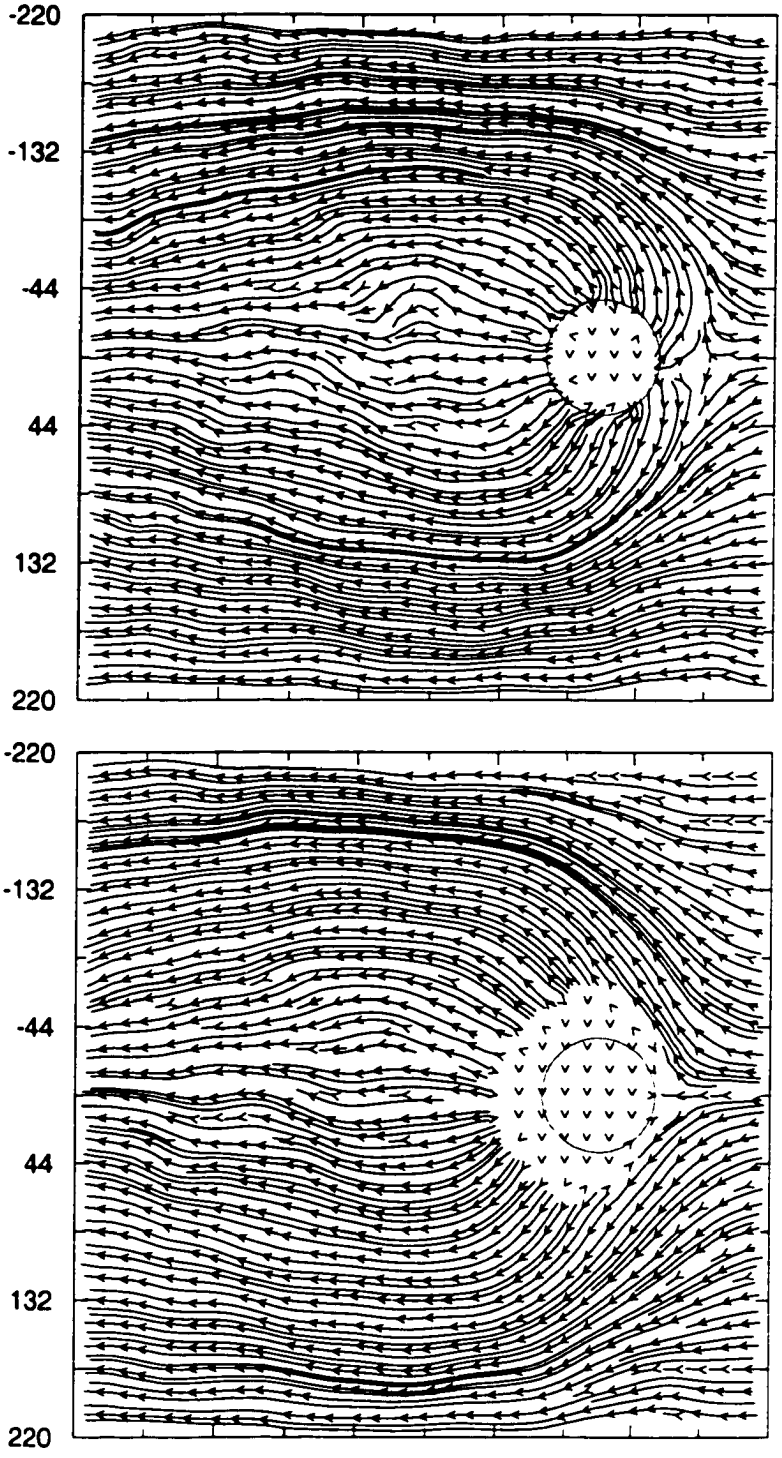

Figure 5.2: Velocity (t=1008, 1968; τ_E=960). U=0.025, D=74 (thermal BC).

from the low speed "elastic" case to inelastic behavior. the flow speed was chosen to be equal 0.1. As it was shown before. that was the highest speed at which the temperature and density changes were relatively small. The resolution and other parameters of the simulation were kept the same as in the previous case.

As a first stage the flow was simulated for about 8 obstacle sizes using the value $r = 1$. i.e. elastic collisions. During this stage the familiar patterns with eddy-like structures had time to develop in a "no-slip" case ("thermal" variety for now) and a "waving tail" pattern in a "slip" case. It is rather important to note that although the most interesting features of the flow were captured correctly in the smaller simulations discussed above. the boundaries were not far enough apart to avoid an influence on the flow. Fig. 5.3 shows part of the region in the vicinity of the obstacle. The size and the location of this "window" is equivalent to the simulation domain in the smaller simulations (440×440). It is clear that the streamlines near the window edge are rather far from straight lines in both "no-slip" and "slip" cases. The periodic boundary if placed at this location would affect the flow to some extent.

Introduction of inelasticity into the system leads to a very fast "cool down". and the previous patterns characteristic to the flow of elastic particles become very clear since the fluctuations and the noise level in the system go down. The part of the system which has these patterns is behind the obstacle and relatively isolated from the shock forming in front and to the sides of the obstacle. The beginning of the process of dissipation and disintegration of these patterns is shown in Fig. 5.4 and Fig. 5.5 for "no-slip" and "slip"

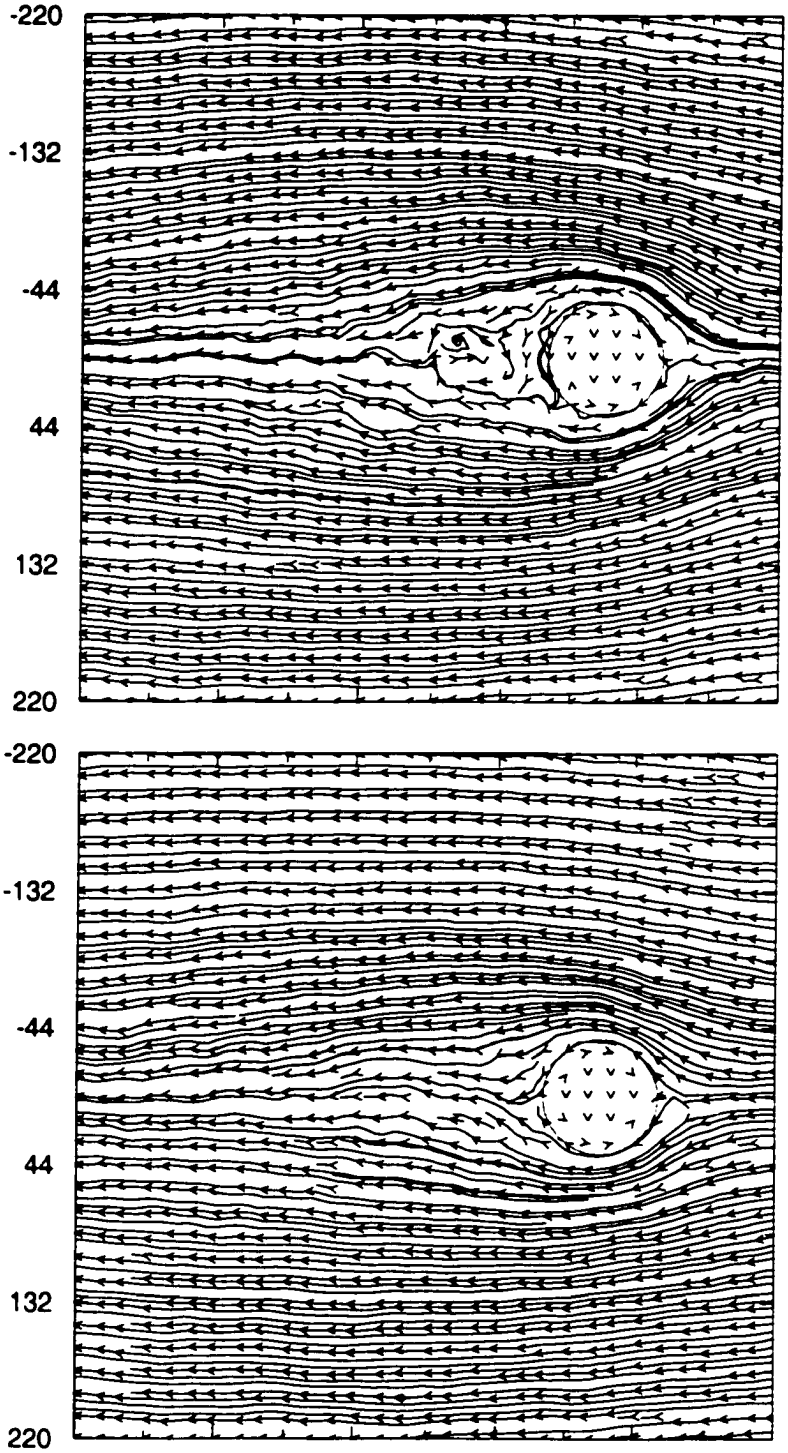

Figure 5.3: Velocity (t=5760: τ_E=240). U=0.1, D=74 (no-slip, slip).

cases correspondingly.

The simulation was performed long enough to allow the stream to pass approximately 1.7 times through the whole domain. Nevertheless, because of the back-flow present in the original stream composed of elastic particles, some of the material behind the obstacle had small velocity and was still present in the system at the end of the run.

At the beginning of the second pass past the obstacle it is clear that the flow is separating into two regions: an internal one inside the wake behind the obstacle, and an external region where all the major development of the shock takes place.

By the end of the first pass, the density peak can be clearly seen in front of the obstacle. It continues to grow wider and higher and by the end of the third pass practically reaches its maximum height. The process of spreading out this high density region continues until the stream finishes its first pass through the system. After that the density field remains essentially unchanged although the final state stabilizes only about two obstacle sizes before the end of the run.

Figs. 5.6, 5.7 and 5.8 (number density, temperature and velocity correspondingly) show that there are no major differences between pictures of the interaction with the obstacle under the "no-slip" and "slip" boundary conditions. Nevertheless, it is clear that instead of one temperature peak right in front of the "slip" obstacle, there are two symmetric peaks about $\pi/2$ apart shifted sidewise along the obstacle boundary in the "no-slip" case. The height of those two peaks is nearly 80% more than the height of the "slip" peak.

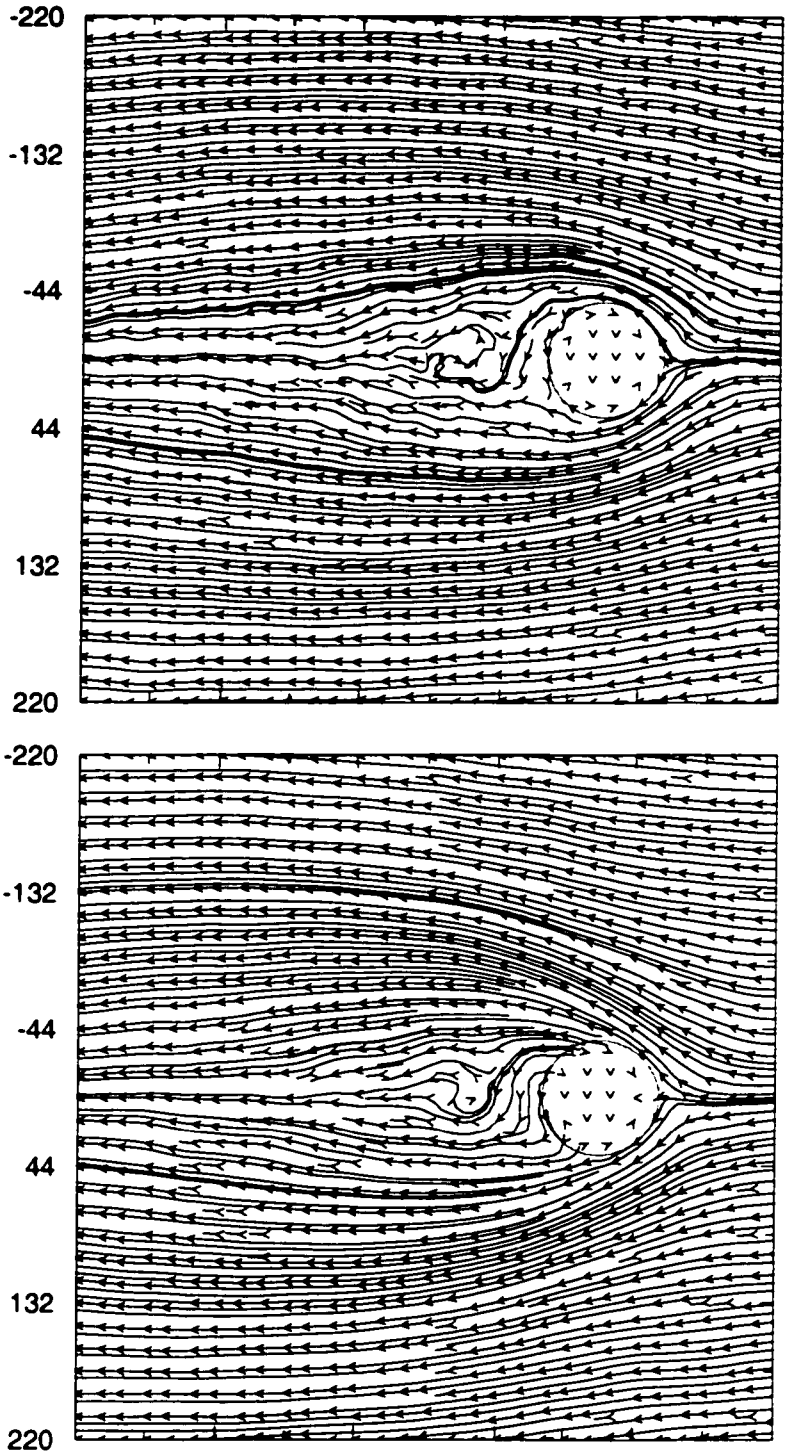

Figure 5.4: Velocity (t=288, 1008; τ_E=240). U=0.1 (no-slip).

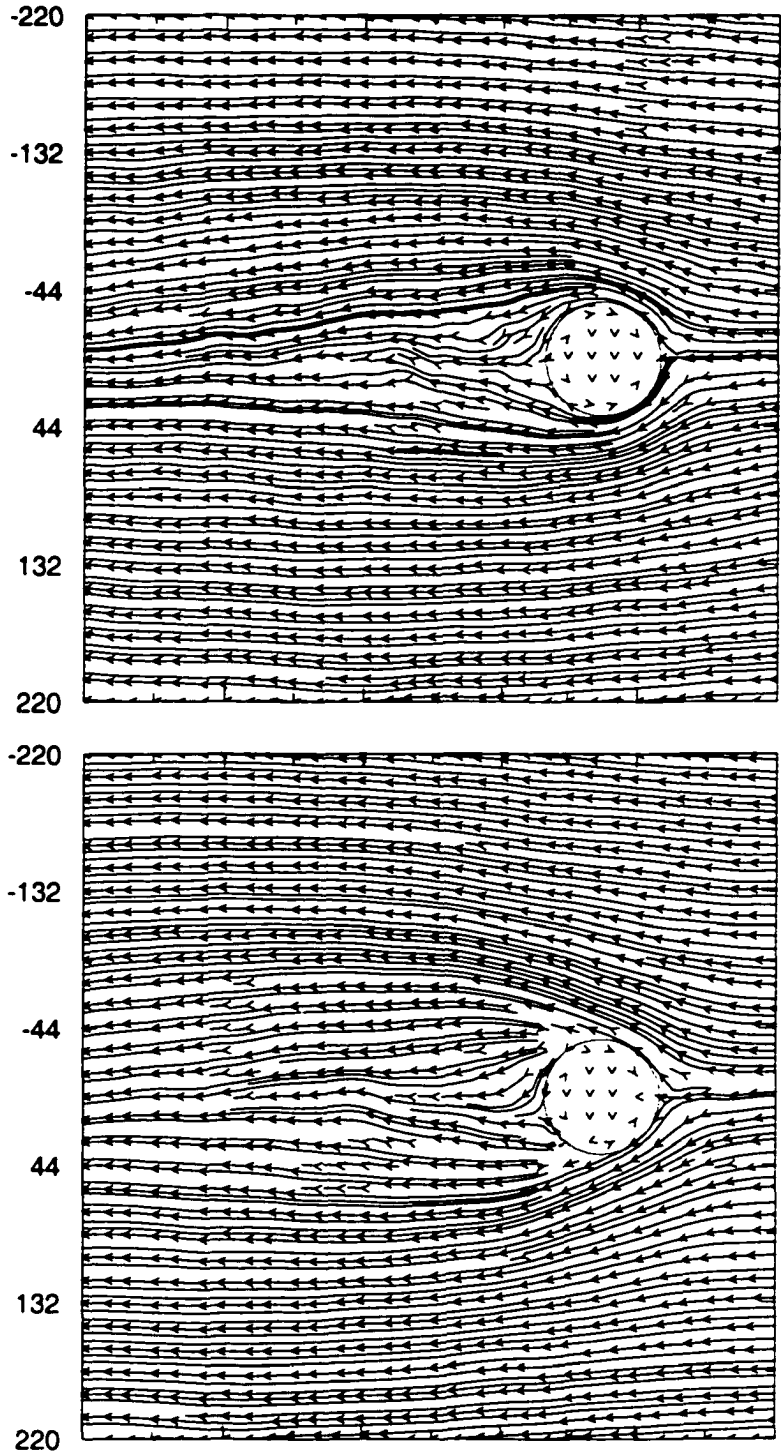

Figure 5.5: Velocity (t=288, 1008; τ_E=240). U=0.1 (slip).

It was already emphasized that the meaning of the temperature here is just the measure of velocity fluctuations around the mean value. Since the mean value itself might not be well defined when such high gradients are involved, the interpretation of the peaks should be done with extra caution. One can also note that the high temperature crest seen at about 2 obstacle sizes in the upstream direction is larger in size and extends slightly farther from the obstacle in the case of "no-slip" BC compared to the "slip" counterpart. It is rather interesting that while there are small temperature peaks on the inner side of the wake behind the "no-slip" obstacle, they are nearly completely absent in the "slip" case.

The plot of the velocity field (Fig. 5.8) shows that within the high density region the motion occurs along the lines which were seen in [ABK01] as nearly straight. Taking into account the difficulty of accurate experimental measurement of individual particle trajectories the agreement is good.

Another important feature observed in [ABK01] was a static "dune" right in front of the obstacle. The plot of the absolute value of velocity in the vicinity of the obstacle (Fig. 5.9) shows that in both "no-slip" and "slip" cases the velocity of the material right in front of the obstacle approaches zero. But in the "slip" case this region is very small and comparable to the size of a data bin. However in the "no-slip" case the static "dune" can easily be identified. This indicates that the friction between the particles and the obstacle (simulated here through no-slip BC) is very important for the formation of the "dune" structure. On the other hand, the absence of particle-particle friction did not prevent the "dune" buildup. It is possible

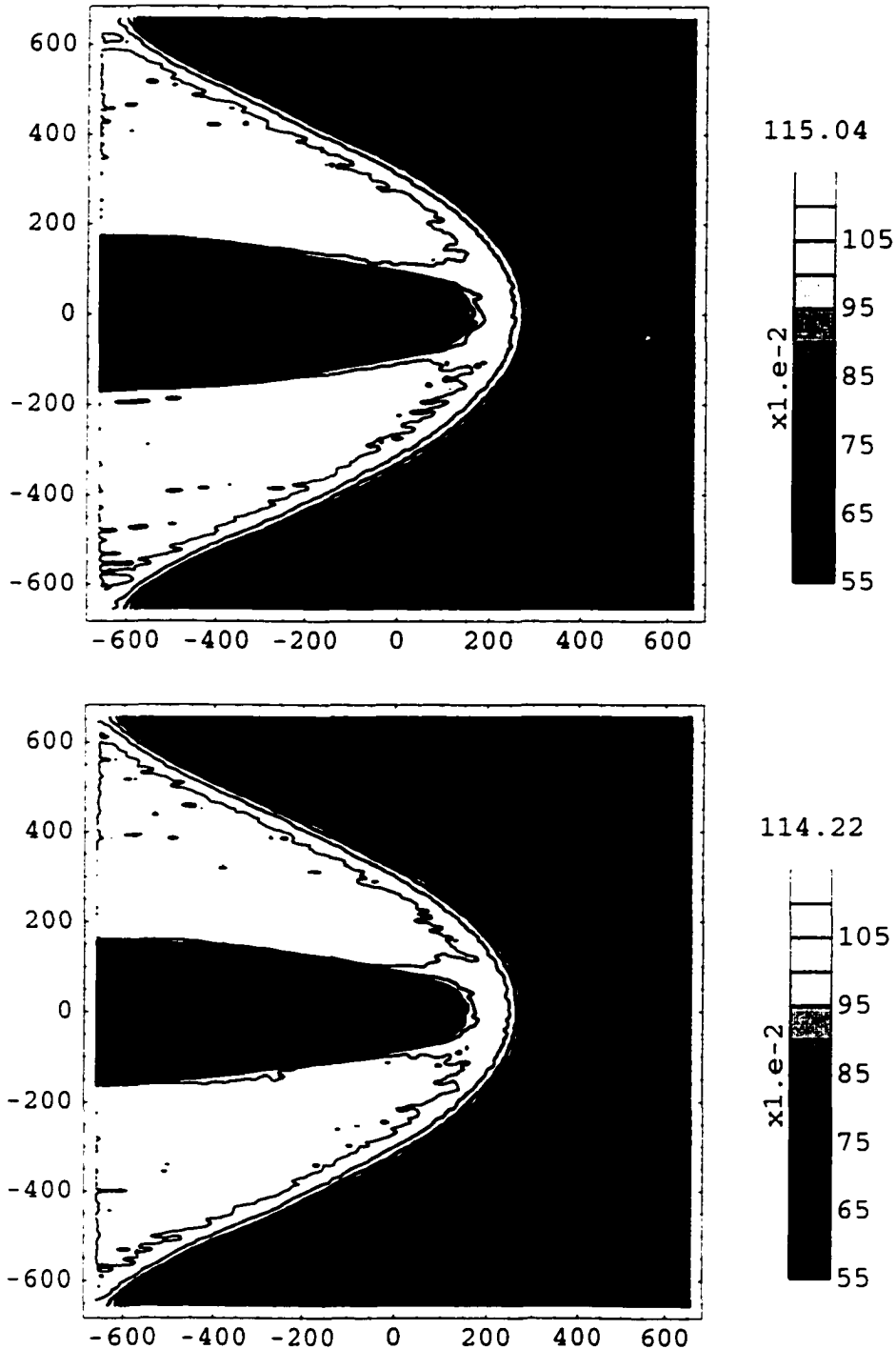

Figure 5.6: Number density (τ_E=1920). U=0.1 (no-slip, slip).

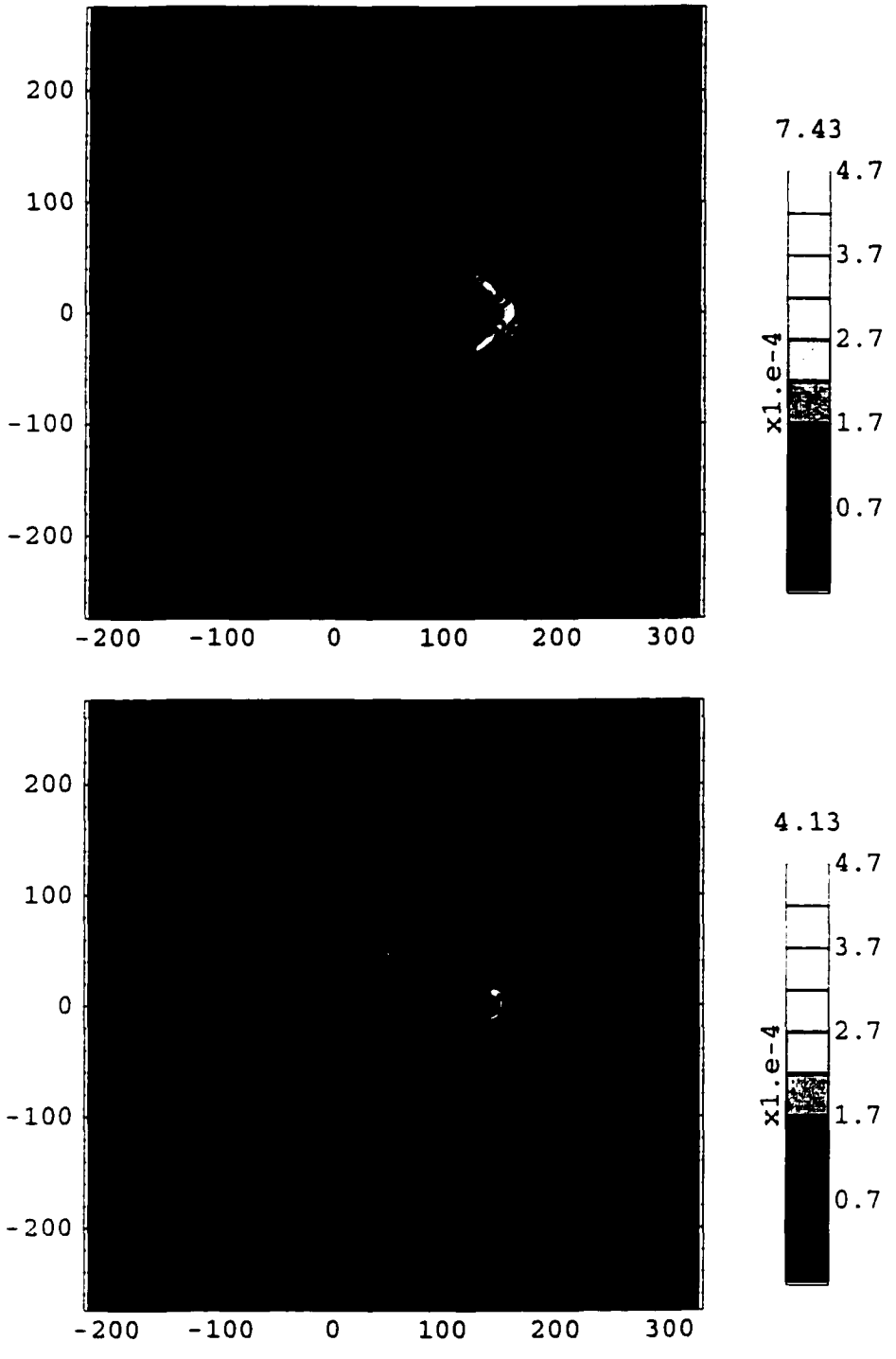

Figure 5.7: Temperature (τ_E=1920). U=0.1 (no-slip. slip).

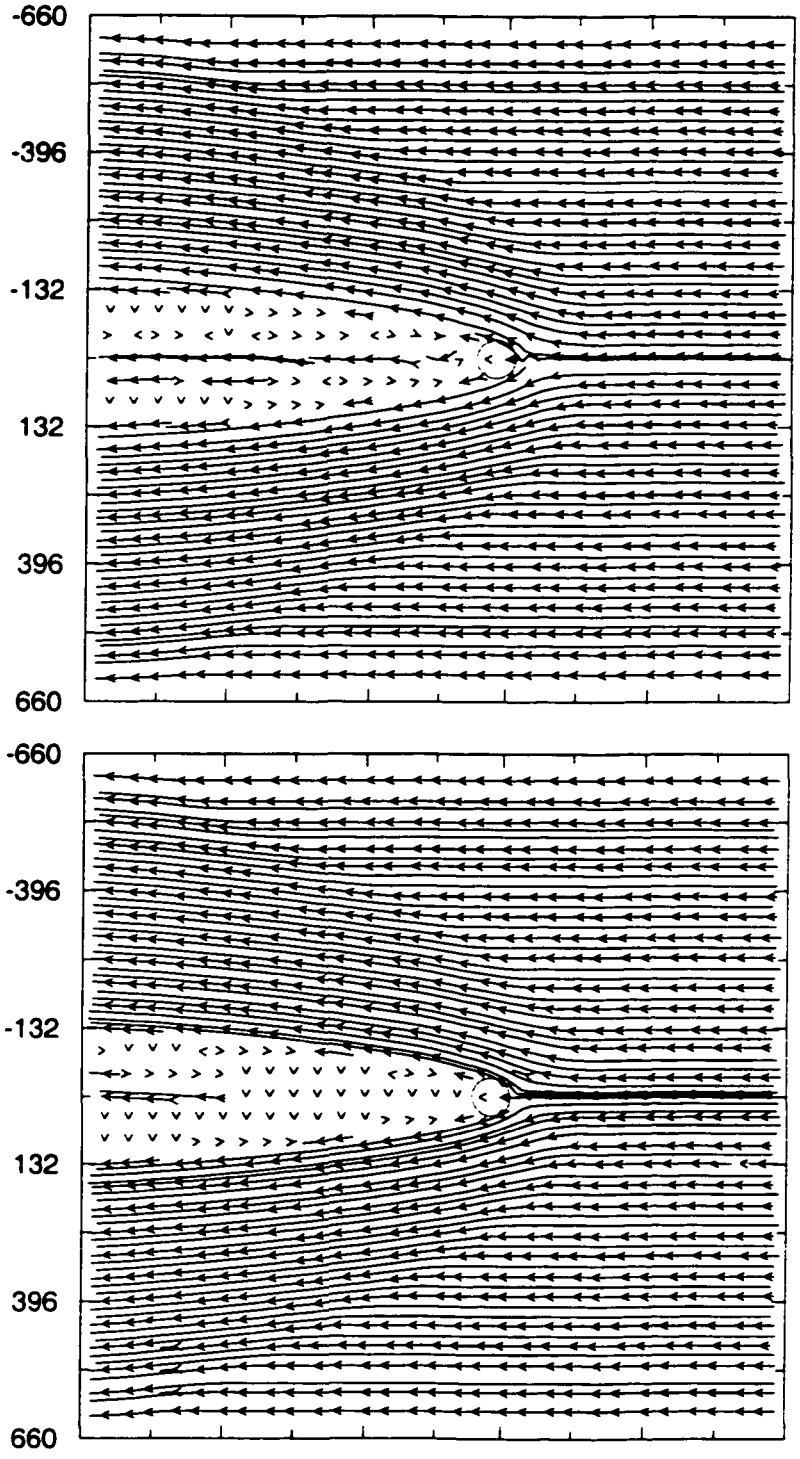

Figure 5.8: Velocity (τ_E=1920). U=0.1 (no-slip. slip).

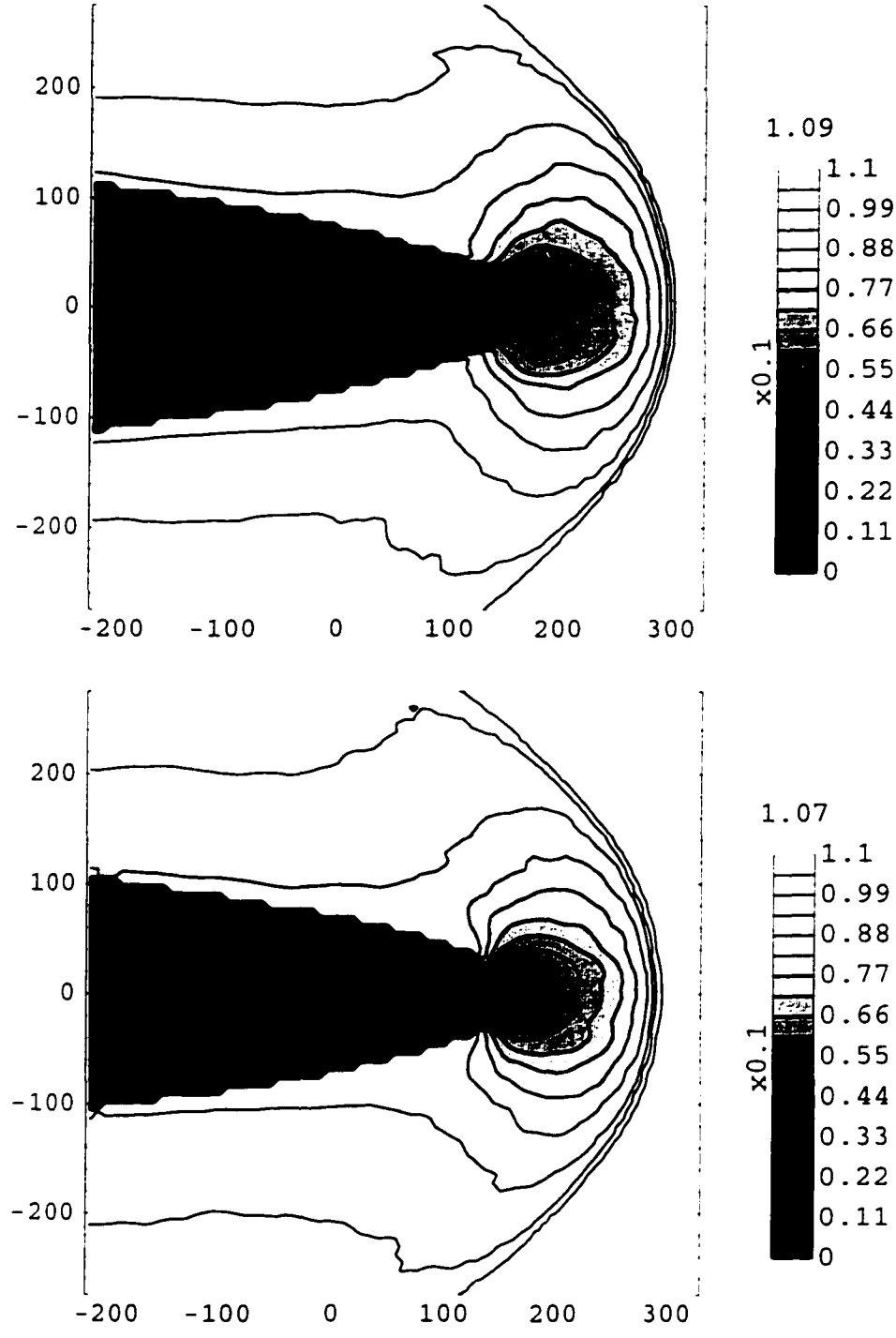

Figure 5.9: Absolute value of velocity (τ_E=1920). U=0.1 (no-slip, slip).

that the particles of the dune are caged inside by the outer layers sliding away with or without friction.

Finally, another rather intriguing observation is that since the density of the incoming stream is rather high, the material in the region of the shock is compressed to the volume fraction of about 0.86 and higher (Fig. 5.6). Taking into account that 2D volume fractions for random close packing are 0.82-0.89 (e.g. see [Ber83]), where the binary collision model breaks down, the ability of the TC model to provide useful results is quite remarkable.

Chapter 6

Flow in the presence of a thermostat

It is rather clear that without some kind of energy source providing velocity fluctuations in the flowing granular media, the separation of time scales typical in the thermodynamics of gases is lost. The two time scales involved here are "micro" for velocity fluctuations and "macro" for the speed of fluid "portions" each consisting of a relatively large number of particles. The changes in the fluctuational (thermal) component of the system energy are of course present in classical thermodynamic systems as well. The fact that the loss of fluctuational energy occurs in granular media so fast (compared to the typical time scale of the media motion) is only part of the picture. If it were the only difference from the typical gases or liquids there would still exist a possibility to insulate the system thermally to prevent the loss of thermal energy. In case of granular materials the dissipation of energy occurs

during the collisions of particles composing the media. The loss of energy occurs not at the system boundary but in the bulk. Thus, the concept of insulation is hardly applicable. As an example one can consider two systems. One composed of several hundred particles and the other of several million particles (of course the particular numbers are not relevant). Provided the systems had the same granular temperature, both systems will "cool down" during the same time period all other conditions being equal.

Just as in typical gases and liquids, the loss or excess of thermal energy can be compensated by using a thermostat. The experimental (practical) implementation of a thermostat does not necessarily have to be mimicked by a computer simulation and vice versa. For example, it is hardly practical to directly simulate a classical heat bath having large volume and high heat capacity. A thermal wall ([TTKB98]) can be used for the same purpose in a simulation. On the other hand, some methods in computer simulations involve direct artificial changes of particle parameters in order to sample the canonical distribution of energies (e.g. see [AT87]). Obviously such manipulation of micro-scale parameters is not practical (or possible) way to achieve a constant temperature in experimental systems.

Nevertheless, in the case of granular media, individual grains are not as difficult to access as particles of regular gases. Therefore, it seems to be important to use some kind of a thermostat model that would be motivated by the real "micro" scale dynamics of the media, namely collisions of the grains between themselves and with boundaries. The stochastic methods ([AT87] and references therein) allow one to identify collisions or events. In this model

random particles are selected at certain time intervals, and their velocities are sampled from the Maxwell-Boltzmann distribution corresponding to a temperature of the thermostat. As a motivation, one might consider a system consisting of a single layer of 3D spheres placed between two parallel horizontal plates which vibrate in vertical direction. The spacing between the plates should be small enough to avoid the substantial deviations from a single layer arrangement but large enough to avoid jamming of particles. In principal, the parameters of the plate vibrations such as amplitude, frequency, phase can be chosen independently. Combined with certain roughness of the plates the proper choice of vibration parameters should allow production of random fluctuations in the system (see also [BO02]). Of course, the specifics of this "motivation" might differ rather strongly from the artificial computer simulation thermostat used in the present work.

To illustrate some other possibilities of thermostat models used for granular media, it is worthwhile to mention the approaches of random accelerations and random kicks. The first one provides random accelerations for particles between collisions, and its motivational analogy is a system of disks on the air table. In the second method, the white noise component is added to velocities of particles. More information about the use of these models can be found in [BSSS99] and [MSS01].

To limit the intrusive effect of the thermostat itself on the flow development, the appropriate time and space average of the local velocity field should be added to the random component sampled from the Maxwell-Boltzmann distribution. This approach is used in the present work since the flow pat-

terns, occurring because of interaction of the granular stream and an obstacle, are of great importance for this investigation. Thus, the thermostat model can be completely described by specifying the parameters: thermostat temperature (T_{th}), thermostat action time (taken per particle, τ_{th}) and parameters of time and space averaging of the velocity field (τ^v_{th}, l^v_{th}). In 2D, the parameter l^v_{th} is actually two lengths specifying an averaging bin. It is important to note that parameters τ^v_{th}, l^v_{th} are not necessarily equal to the corresponding parameters of the averages used for normal data output of the simulation. In an extreme case when the simulation data sampling is done for the purpose of observing long time averages of some dynamic process, the need for a separate time parameter is clear. Similar arguments are valid for the parameters of space averages. This illustrates the fact that although very close in the calculational aspect, these two averaging procedures serve rather different purposes, and in the present work two completely independent coarsening grids were used.

In the event driven simulations a new feature is associated with a certain source of events which provides an opportunity to take the appropriate actions. For example, in [BSSS99] the particle collision events are used for this purpose. Nevertheless, in the present work for the sake of avoiding the coupling (however unimportant it might seem to be) between thermostat events and regular particle events, a separate source of thermostat events was introduced and the timing of its action was done independently.

In granular media each collision between grains dissipates substantial amount of energy, and hence the loss of energy in granular systems is as-

sociated with "micro" rather than "macro" time scales. Thus, maintaining the system at the temperature equal to the temperature of the thermostat requires very high frequency of thermostating events. Basically, on average for each regular collision between grains there would need to be at least two (one for each grain of the collision pair) events, compensating for the loss of energy because of the collision. This approach (e.g. used in [BSSS99]) has an advantage that both the system and the thermostat have the same temperature which is usually the goal of thermostating in classical thermodynamic systems. But such extreme activity of the thermostat in case of granular media does not leave much room for development of the dynamics of the flow itself.

The very first simulations along the line of the "equal temperature" thermostat have shown that, in spite of the attempt to reduce the thermostat effect by taking into account local velocity field, the flow patterns reduce to the very basic ones. This basic flow, showing the material passing about the obstacle without density or temperature changes, is still a rather successful use of thermostating since such a frequent manipulation of "micro" parameters of the media could lead to much more bizarre behavior. The examples of the situation when the artificial nature of the model manifests itself in very strange flow patterns will be seen later.

In spite of the relative success of these simulations, the question of the possibility of existence of flow patterns similar to the ones observed for the systems of elastic particles still remained open. Under such frequent interference caused by thermostat events, the media portions (fluid particles) simply

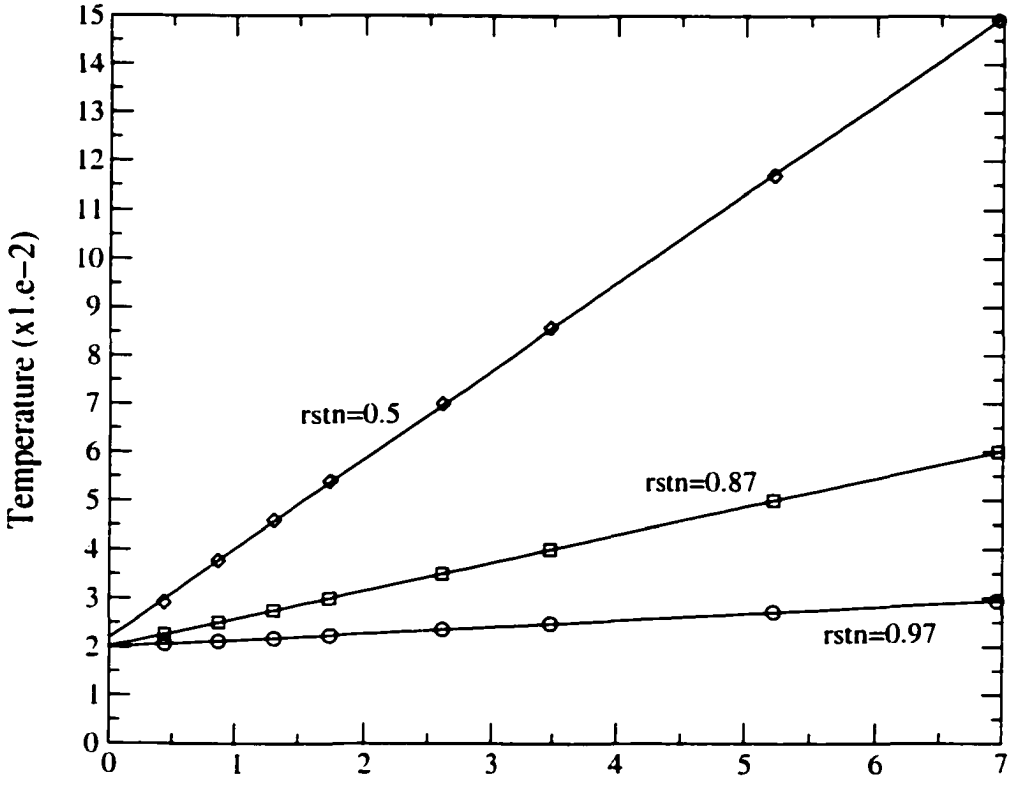

Figure 6.1: Thermostat temperature vs. action time

did not have a chance to develop their own "history".

As an attempt to obtain more information on this subject, the frequency of the thermostat events was reduced. Basically, depending on the chosen thermostat action time τ_{th}, instead of compensating the energy loss due to a single collision, each thermostat event could now provide energy to compensate the loss occurred in several collisions. Naturally, the portion of energy being transfered into the system during each of the thermostat events has to be larger, and consequently the temperature associated with the thermostat itself should be higher than the target temperature of the media. The tem-

perature time dependence of freely cooling granular media was given before (Fig. 5.1). The dependence of the thermostat temperature on the thermostat action time τ_{th} is given in Fig. 6.1. These results were obtained for a system with periodic boundary conditions, volume fraction $\eta = 0.65$, size $S_x \times S_y = 128 \times 128$. The mean free time in the system is 0.43 which gives the relevant time scale of the problem. It is important to understand that although Fig. 5.1 and Fig. 6.1 are related, they describe different situations. While Fig. 5.1 shows the amount of energy of fluctuations left in the freely cooling system at certain time points, Fig. 6.1 gives the *rate* of energy input necessary to keep the system at the target temperature. If the average time interval of energy injections (action time) is increased n times, then the amount of energy in each injection should also be increased n times to keep the rate unchanged. The linear dependence shown in Fig. 6.1 simply reflects this fact.

The important question is how many regular collisions per particle the system should be allowed to retain so that the flow dynamics would not be completely suppressed by the thermostat. This would determine the appropriate choice of the thermostat action time.

The flow behavior in simulations also depends on the choice of the thermostat averaging parameters for the velocity field. If the grid is too coarse then the patterns will be averaged out, and if the grid is too fine then the noise will dominate over the signal. From the simulations of systems with elastic particles, the appropriate maximum size of the space grid bin has already been established, i.e., about 1/10th of the obstacle diameter. The

action time and the averaging time parameters of the thermostat were to be determined by additional simulations. To make the comparison easier, the test system was essentially the same as already familiar system (size $S_x \times S_y = 440 \times 440$, C=0.05, no-slip thermal BC at the obstacle) composed of elastic particles. To isolate the effect of inelasticity from the thermostat influence, the restitution coefficient was still kept to be 1. Thus, the temperature of the thermostat was 0.02, i.e., equal to the temperature of the media. The time parameters in question were taken from the sets: τ_{th}=0.87, 1.74, 3.48, 6.96; τ_{th}^v=30, 60, 120, 240.

Every combination of these values corresponds to a separate simulation. Without going into details of all the runs performed, it is important to say that the familiar eddy-like patterns were not observed in any of the simulations with τ_{th}=0.87, i.e., high frequency of thermostat events (one thermostat event for two particle-particle collisions).

Another, although different in nature, cutoff parameter was found to be τ_{th}^v=30. The level of the noise, distorting the velocity average values and thus being put back into the system through thermostat events (positive feedback), was too high to allow pattern analysis. Of course, this cutoff is rather artificial in nature since the simulation of larger systems would allow better space averages (keeping the same space resolution in terms of the obstacle size), and thus the value of τ_{th}^v can and should be reduced.

In simulations with τ_{th}=1.74, the situation was just a little better - several very small and short lived eddy-like structures could be identified in the run with τ_{th}^v=60. For similar runs with higher values of τ_{th}^v (longer time averages),

the patterns were not observed.

Finally, the combination τ_{th}=3.48 and τ_{th}^{v}=60 resulted in the simulation with very clear eddy-like structures (Fig. 6.2) which were generated and dissipated through the whole run (about 12 obstacle sizes). In general the patterns were very similar to the ones observed in a corresponding elastic case without the thermostat. The major difference was rather large amount of "inertia" in the eddy-like structures due to the time averages involved. Because of this the structures appeared not as often and dissipated not as fast as in the reference case.

This effect was less prominent in the simulations with τ_{th}=6.96 and τ_{th}^{v}=60, but using such a low frequency energy input would lead to unnecessary large difference in the thermostat temperature and the temperature of the media. It is important to note that even though the frequency of the thermostat events is rather low for τ_{th}=6.96, the effect of τ_{th}^{v} can not be neglected. For example, in all the simulations with τ_{th}^{v}=240 ($\approx$ 1/6 obstacle size) the eddy-like patterns were not able to develop. On the other hand, it was obvious that, compared to the simulations with τ_{th}=0.87 and 1.74 units of time, with larger values of τ_{th} there was much more activity in the streamline patterns behind the obstacle.

A rather interesting case of τ_{th}^{v}=120 (τ_{th}=3.48, 6.96) deserves to be mentioned separately. The relatively small parameter τ_{th}^{v}=120 ($\approx$ 1/12 obstacle size) is between τ_{th}^{v}=60 (rather clear pattern structures could be observed through the whole simulation) and τ_{th}^{v}=240 (suppressed formation of structures). After the flow passes about 3 obstacle sizes, the flow becomes globally

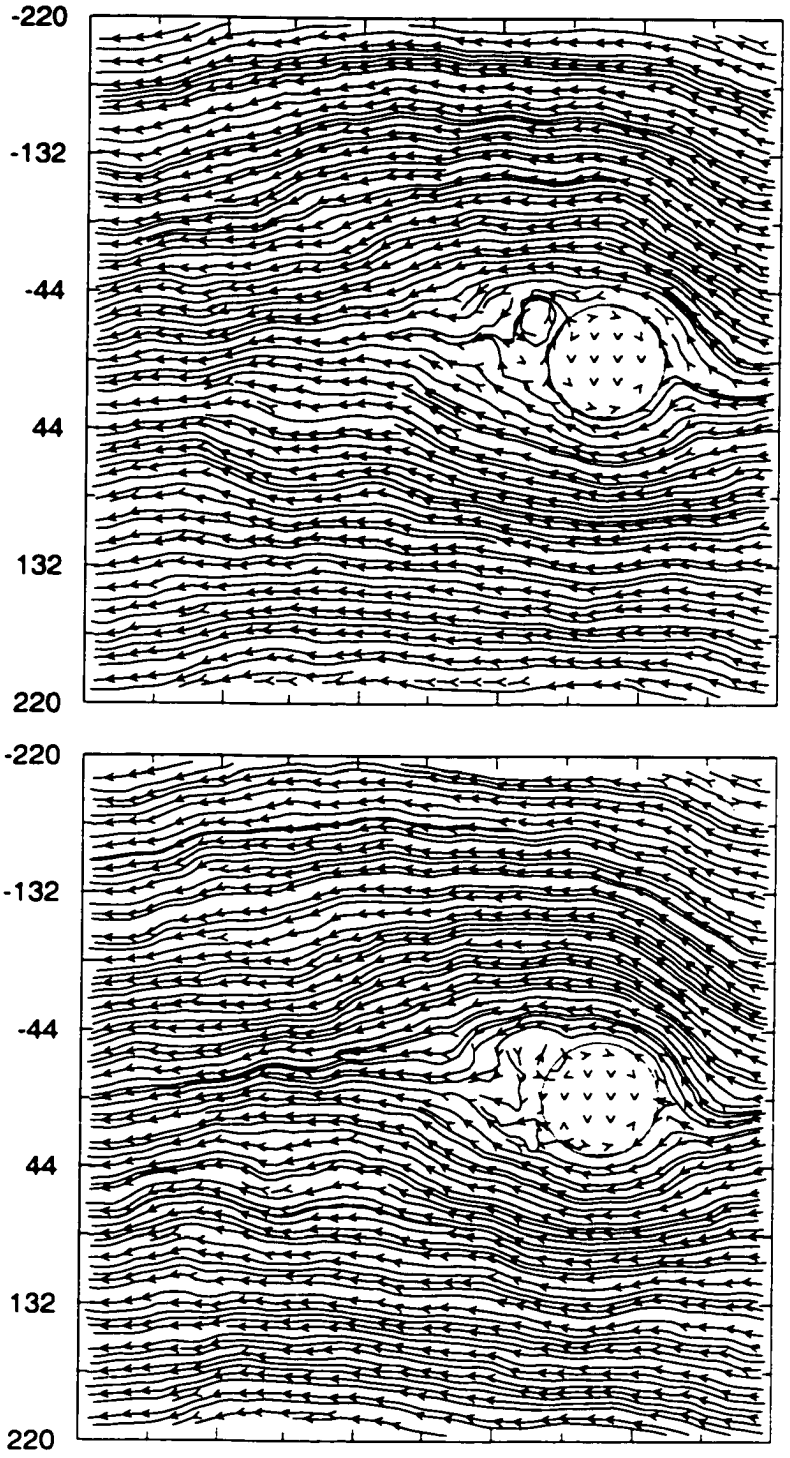

Figure 6.2: Velocity (t=6720, 7200; τ_E=480; τ_{th}=3.48). U=0.05 (no-slip).

distorted and the symmetry across the central line along the flow is lost. This loss of symmetry becomes more and more prominent through the whole run (Fig. 6.3). The most probable reason for that is the "locking" of the flow into whatever deviation occurs first because the time intervals for the averages involved are not too long to completely suppress the deviations and not short enough to allow the complete eddy-like structure formations. This is, of course, a purely artificial effect (albeit not obvious or easy to predict), and it demonstrates the limitations of the particular thermostat model.

This effect is present to some extent in all the simulations using the thermostat with time averages involved. Thus, all further simulations of systems composed of inelastic particles were done without periodic boundary conditions in the direction perpendicular to the flow. Instead, to provide additional stability for the simulations, two "elastic walls" were place at the simulation domain boundaries.

One interesting "flavor" of the stochastic model for the thermostat deserves a special attention. It was used in [BSSS99] and is motivated by the fact that single thermostat events would cause the momentum in the system to perform a random walk. To combat this, the authors were using "dual" events in which one particle would be assigned a velocity (fluctuation part, of course) sampling the desired distribution and the other particle would get the opposite velocity vector. This model seems to be even more artificial than a "single event" one since it is hard to imagine the experimental excitation mechanism which would provide such pairing of velocities. But the potential ability of this approach to reduce the thermostat effect on the patterns of

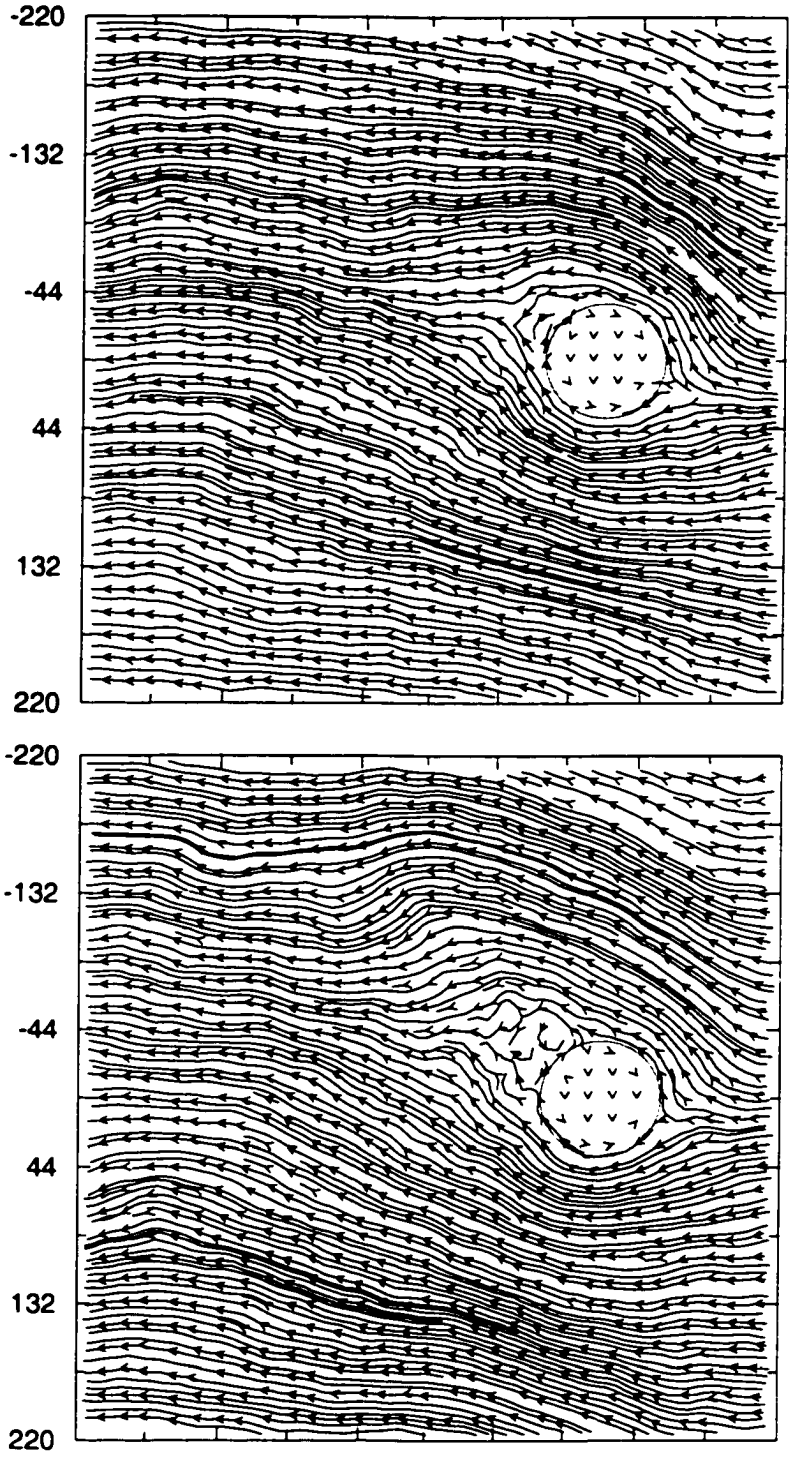

Figure 6.3: Velocity (t=17280: τ_E=480: τ_{th}''=120). τ_{th}=3.48 (top). 6.96

the flow justified a separate investigation.

In the present work the property of interest was not the total momentum of the system but local momentum. Thus, particles of each pair for "dual" thermostat events were chosen locally. The most important thermostat parameters were τ_{th}=6.96 and τ_{th}^{v}=60 which, taking into account the "dual" nature of the model, should be equivalent to the (τ_{th}=3.48; τ_{th}^{v}=60) reference case. The results turned out to be rather interesting. The model definitely provided some additional stability against distortions caused by using time averages. At the same time the eddy-like structures became smaller and could be observed less often compared to the reference case. This might indicate that the "damage" done to local coherence in the media is harder to "repair" in case of "dual" events separated by the time interval $2\tau_{th}$ compared to single events separated by τ_{th}.

Thus, further investigations of systems composed of inelastic particles were chosen to be based on the "single event" thermostat model. The target systems were chosen to be similar to the ones considered before, i.e., size $S_x \times S_y = 440 \times 440$, volume fraction $\eta = 0.65$, target temperature of the media T=0.02, obstacle diameter D=74, placed at 110 units of length from the center up the stream. The "stabilizing" elastic walls were placed at the sides of the simulation domain. The parameters of the thermostat were τ_{th}=3.48, τ_{th}^{v}=60, space average grid 60 × 60 (the same resolution as used for data sampling). The simulations were performed for 0.97 and 0.87 values of the restitution coefficient. The corresponding values of the thermostat temperature were 0.0248 and 0.04 (see Fig. 6.1).

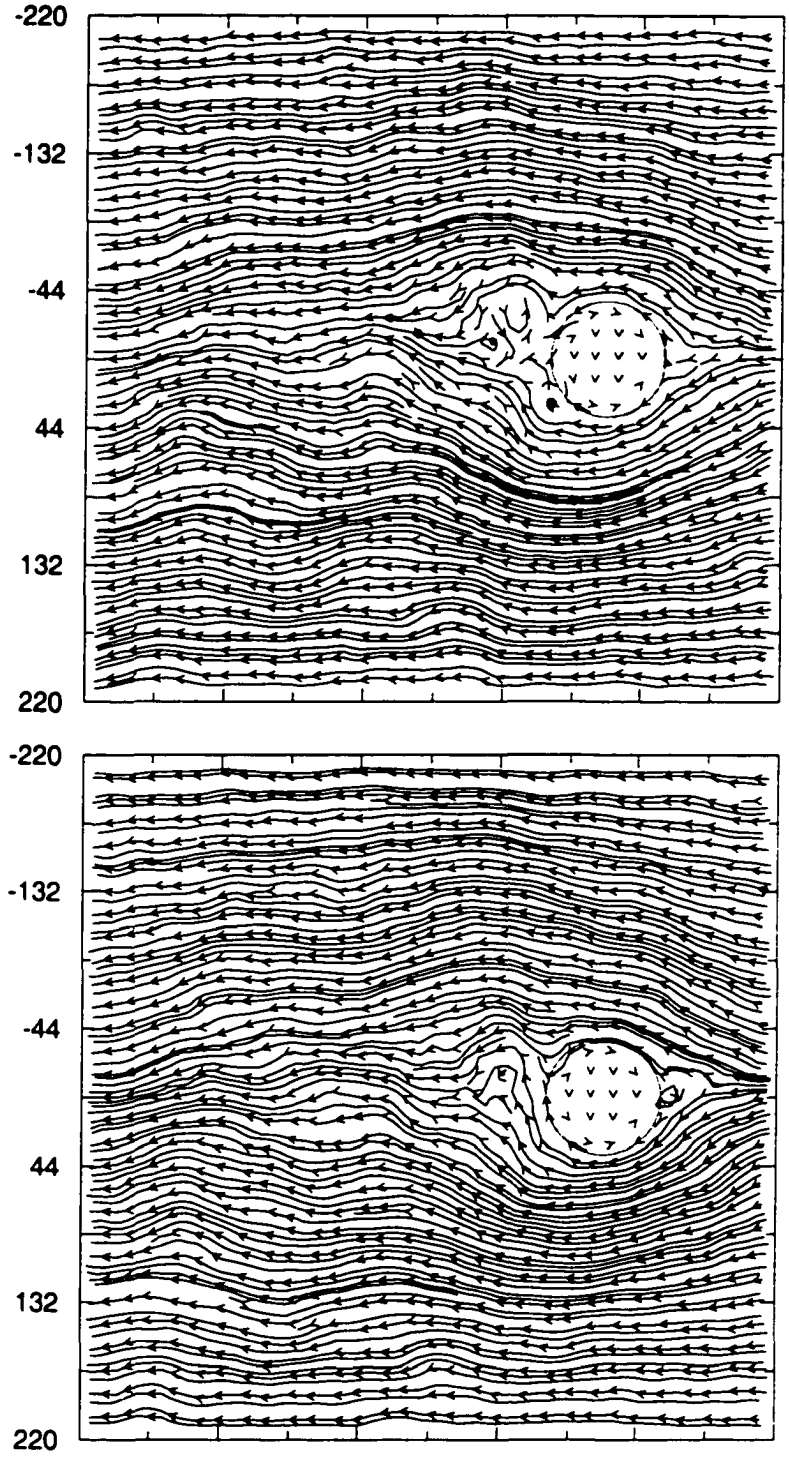

Figure 6.4: Velocity (t=8640, 9120; τ_E=480). U=0.05, r=0.97 (no-slip)

Two sequential frames of the simulation with r=0.97, U=0.05 and no-slip (thermal) boundary conditions at the obstacle are shown in Fig. 6.4. These plots, as well as the rest of the frames in the simulation, showed that at this level of inelasticity the flow patterns are still rather similar to the ones observed in the flow of elastic particles. Unfortunately, the effect of the thermostat, with the temperature about 25% higher than the one of the media and the artificial distortions caused by velocity averages, did not allow more detailed comparison between "elastic" and "inelastic" flows. Nevertheless, the generation and dissipation of eddy-like structures is clearly present in the flow of smooth inelastic particles past an obstacle with no-slip boundary conditions.

The "slip" counterpart of the above simulation (particle-obstacle restitution 0.97) also exhibits the feature characteristic to the previously discussed flow of elastic particles past an obstacle with slip BC, namely the waving tail pattern (Fig. 6.5). Even though the velocity deviations here are much less than in the corresponding "no-slip" case, the cumulative effect of the thermostat becomes very prominent after the flow passes about 10 obstacle sizes. In spite of these large scale distortions, the similarity with the corresponding "elastic" case is rather obvious.

When the flow speed is reduced to U=0.025 (τ_{th}^{v}=120), the analysis of the results becomes even more difficult since the noise level goes up and is immediately picked up by the positive feedback of the thermostat averages. Any patterns appearing in the wake behind the obstacle should be carefully compared to the behavior of the flow relatively far from the obstacle. The

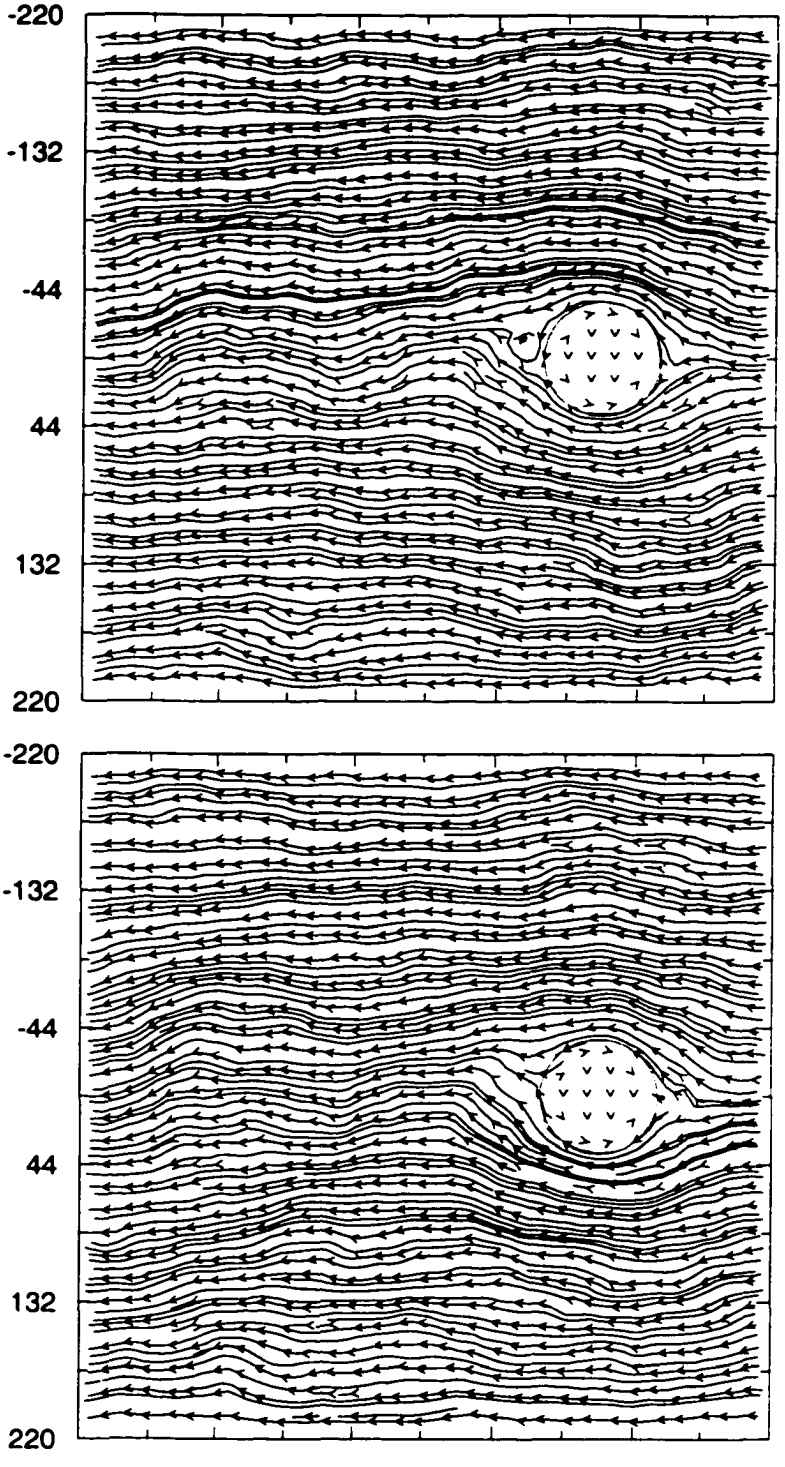

Figure 6.5: Velocity (t=3360, 3840; τ_E=480). U=0.05, r=0.97 (slip)

patterns, which are similar in size and activity to the ones far from the obstacle, should be treated as noise artifacts. It is not clear that even this criteria is sufficient. For example, in a "slip" case a rather large and clear eddy-like structure appeared in the flow after the flow passed about 15 obstacle sizes (Fig. 6.6). Although similar structures appeared and then dissipated one more time before the end of the simulation (23 obstacle sizes), the cautious decision was made to treat this as simulation artifacts caused by noise and thermostat. Nevertheless, only larger simulations with much better quality of averages can truly address this issue.

In spite of these difficulties, the results of the simulation for the "no-slip" case clearly show the generation and dissipation of eddy-like structures. Several frames even confirmed the hypothesis suggested during the analysis of the "elastic" case, i.e., that for the lower flow speeds the structures are being born closer to the sides of the obstacle (Fig. 6.7). Normally, the "inertia" and "amplification" effects of the thermostat are extremely undesirable in all the simulations discussed here. Nevertheless, in this particular case they allowed to see the development of the structures at the very early (farther to the sides) stages.

The last group of simulations to be mentioned here is the runs with r=0.87 (all other parameters kept the same). The fact that the temperature difference between the thermostat and the media is very large (100%) introduces additional complications. Compared to the corresponding simulations with r=0.97, the velocity field shows more noise and distortions. Although video frame sequences still provide useful source of information, separate plots are

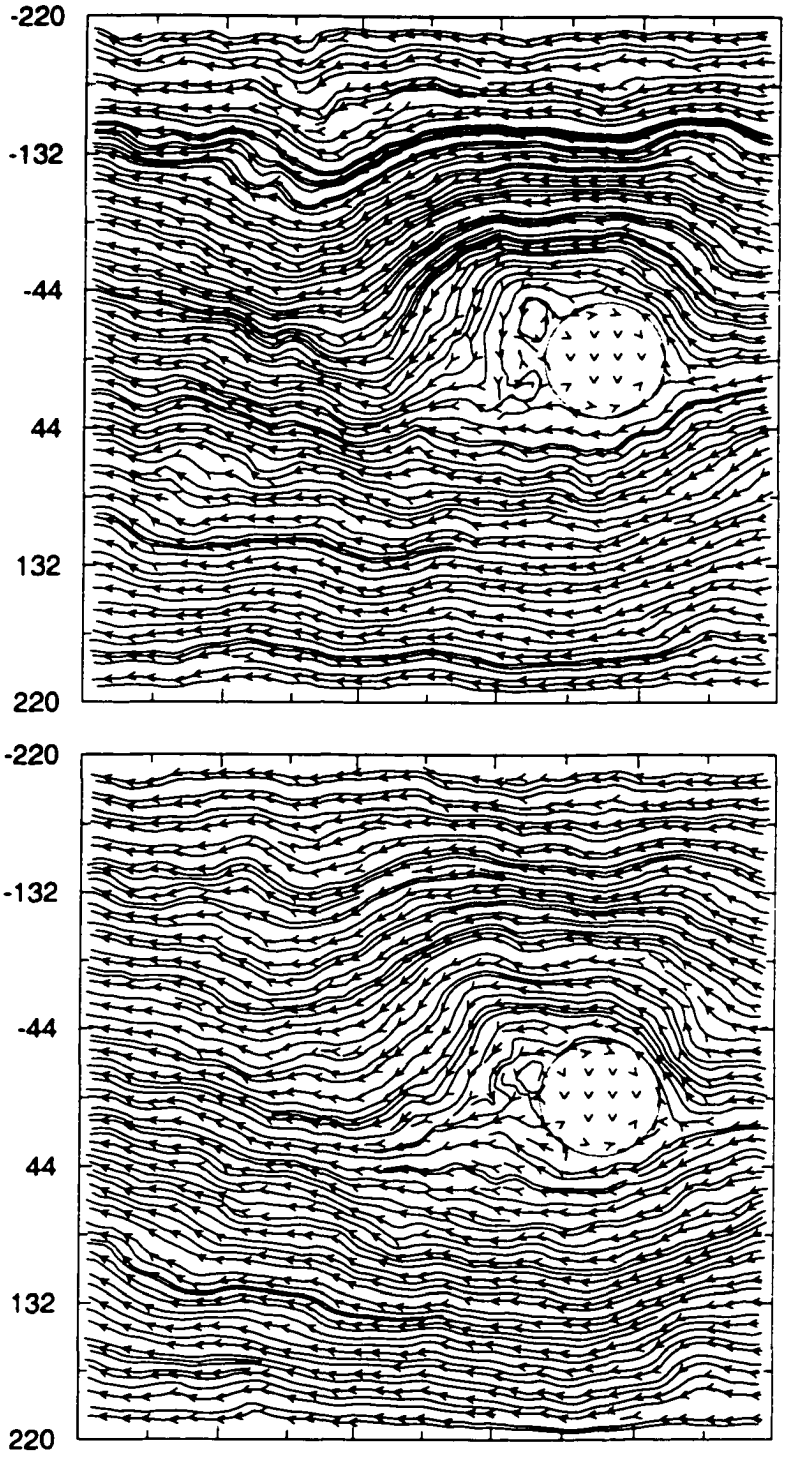

Figure 6.6: Velocity (t=44160, 45120; τ_E=960). U=0.025, r=0.97 (slip)

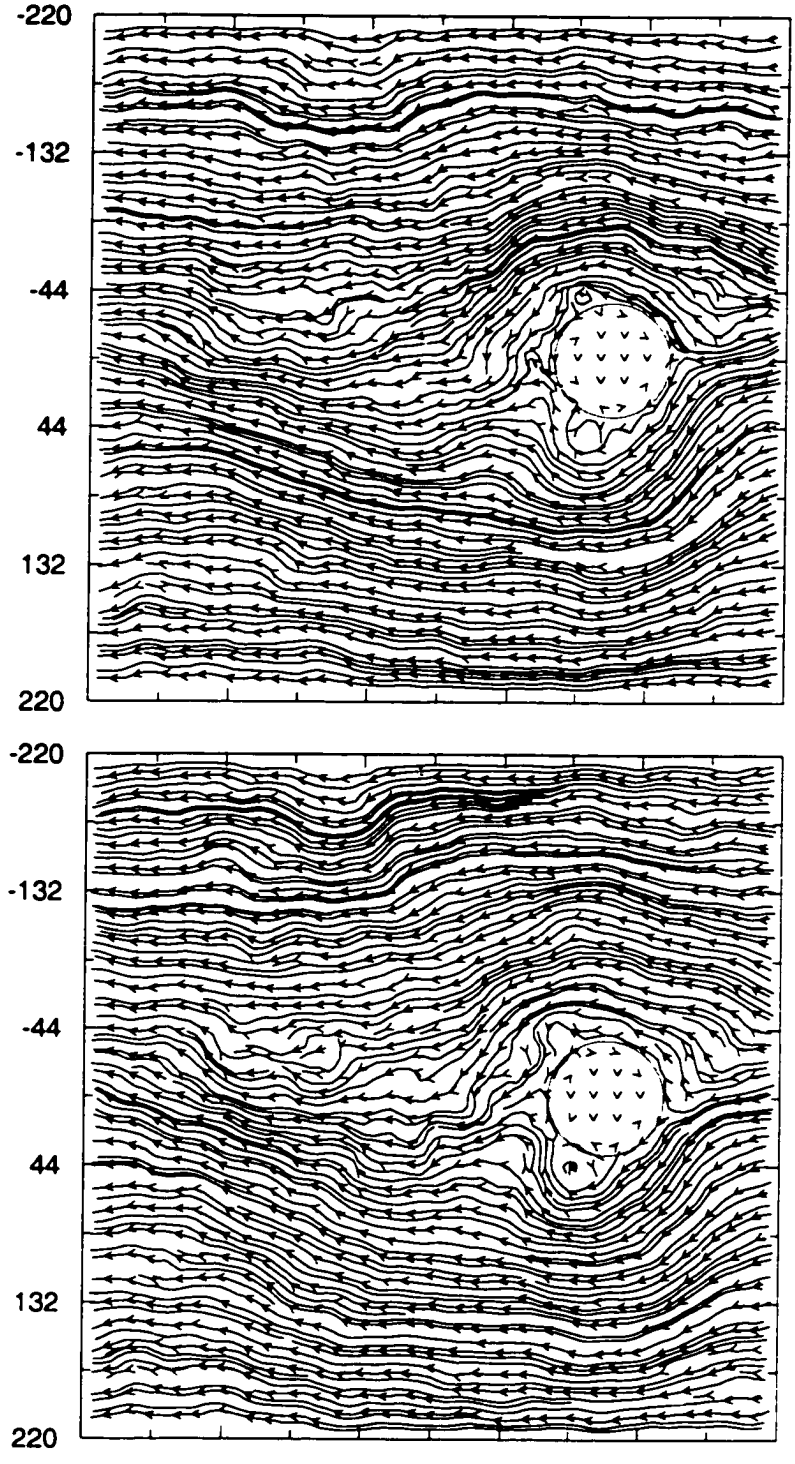

Figure 6.7: Velocity (t=27840, 28800; τ_E=960). U=0.025, r=0.97 (no-slip)

not informative enough anymore.

The most important observation to be made is that in spite of a rather high level of inelasticity in the system, the eddy-like structures are still present in the patterns of the flow in the "no-slip" case. The same can be said about the "waiving tail" pattern in the "slip" case. Taking into account a very high level of interference from rather frequent and energetic "shakes" from the thermostat, such a persistence of characteristic features is very important.

Nevertheless, it is also very important that in simulations with U=0.05 the process of generating eddy-like structures is significantly less active compared to the r=0.97 counterpart. The same observation can *not* be made for the simulation with U=0.025. It has the lowest signal quality among the simulations discussed here, and it is practically impossible to tell apart appearing eddy-like structures from noise artifacts. The only reason this simulation is mentioned here at all is that the corresponding "slip" case does not exhibit nearly as much activity in this aspect. So the very presence of the eddy-like structures in the patterns is rather clear.

The question of why the generation of eddy-like structures seems to be less active when the level of inelasticity is high remains to be answered. To separate the effect of increased inelasticity from the one of the more active thermostat, one would probably need to run much larger simulations (to increase the quality of averages) or to try some other thermostat models. The recent fast strides in computational power availability will hopefully provide an opportunity to properly address this issue.

Summary and Conclusions

The major steps done in the present work can be summarized as follows:

- Designed and implemented the object oriented software model for event driven (ED) simulations of hard sphere systems. The model provides a standardized, well defined interface for adding new features into the system, as well as easier code maintenance due to high level of data and functionality encapsulation.

- Created and implemented a new algorithm for simulating inlet and outlet boundary conditions in uniform streams of particles. Both low and high volume fraction cases can be successfully simulated.

- Performed ED simulations of high volume fraction (vf=0.65) streams composed of a large number ($10^5 - 2.5 \times 10^6$) of identical smooth disks. The behavior of the flow in the presence of a fixed cylindrical obstacle was studied for Re estimated to be in the range from 3 and up to 46. Both slip and no-slip boundary conditions at the obstacle were investigated.

- Studied the changes in the flow occurring as a result of introducing the inelasticity into the system by setting the restitution coefficient to r=0.97.

- Developed an adaptation of a well-known stochastic thermostat model for providing fluctuation energy into the granular material system. Using ED simulations, studied the interaction of the granular flow (r=0.97 and r=0.87) with an obstacle in the presence of the thermostat, and compared the behavior of the systems to the one observed in their elastic counterpart cases.

The simulations of elastic systems performed here have consistently shown that, contrary to some previous reports ([RC86] and [Rap87]), for both "no-slip" and "slip" boundary conditions at the obstacle the flow substantially differed from Stokes flow even at the velocities as small as 0.025. The swirl patterns (velocity ≥ 0.2), previously reported in the literature and analyzed within the framework of a well known change of flow patterns as the Reynolds number becomes larger, are most probably caused by large temperature and density gradients. In spite of relatively low Reynolds numbers used in some of the present simulations, the clear patterns of two counter rotating swirls or the Von Karman vortex street were never observed here in the systems with constant temperature. Instead, the simulations showed the presence of eddy-like structures behind the obstacle, formed and shed irregularly and asymmetrically during the run. The simulations with even lower Reynolds numbers were computationally prohibitive at the time of the research dis-

cussed here.

Introduction of inelasticity into the system leads to a very fast "cool down", and the previous patterns characteristic to the flow of elastic particles become clear since the fluctuations and the noise level in the system go down. There is a satisfactory agreement between the simulation based on the hard sphere TC model [LM98] and the experiment ("Dynamic Sand Dunes", [ABK01]). It was shown that the friction-like ("no-slip") interaction between particles and the obstacle was essential for the formation of the "static dune" while the absence of friction between the particles themselves did not prevent the "dune" from forming. It is also clear that instead of one temperature peak right in front of the "slip" obstacle, there are two symmetric peaks about $\pi/2$ apart shifted sidewise along the obstacle boundary in the "no-slip" case. The height of those two peaks is nearly 80% more than the height of the "slip" peak.

Since in case of granular materials, the dissipation of energy occurs during the collisions of particles composing the media, the concept of thermal insulation is hardly applicable. In simulations of regular liquids and gases on molecular level, the effects of a thermostat on flow patterns are usually a relatively minor issue, since the loss of thermal energy is slow and occurs either due to numerical precision limitations or heat conducting boundaries. The situation is completely different in granular systems, and a special adaptation of the thermostat notion was needed. Such an adaptation was developed on the basis of a well-known stochastic thermostat model (e.g. see [AT87] and references therein), and the appropriate regime, allowing to avoid the

suppression of flow patterns, was established. It was shown that the granular medium driven in this particular way can exhibit flow patterns similar to the ones observed in non-driven systems of elastic particles, i.e., eddy-like structures and a "waving tail" in "no-slip" and "slip" boundary conditions at the obstacle cases correspondingly. In spite of all the effort to reduce the disruptive effect of the thermostat on the flow development, the noise levels were generally substantially higher as compared to the corresponding elastic cases. There is a weak evidence that the generation of eddy-like structures is less active when the level of inelasticity is high (r=0.87 compared to r=0.97), but to separate the effect of increased inelasticity from the one of the more active thermostat, one would probably need to run much larger simulations (to increase the quality of averages) or to try some other thermostat models.

Bibliography

[ABK01] Y. Amarouchene, J. F. Boudet, and H. Kellay. Dynamic sand dunes. *Phys. Rev. Lett.*, 86(19):4286–4289, May 2001.

[APBS99] R. Albert, M. A. Pfeifer, A.-L. Barabási, and P. Schiffer. Slow drag in a granular medium. *Phys. Rev. Lett.*, 82(1):205–208, January 1999.

[AT87] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 1987.

[Bat67] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[Ber83] James G. Berryman. Random close packing of hard spheres and disks. *Phys. Rev. A*, 27(2):1053–1061, February 1983.

[Bir94] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press, 1994.

[BO02] G. W. Baxter and J. S. Olafsen. Granular gas velocity fluctuations for thermalization by a mechanically fluidized bed. *Division of Fluid Dynamics 55th APS Annual Meeting, Session FF - Granular Flows IV*, November 2002.

[BP93] Volkhard Buchholtz and Thorsten Pöschel. A vectorized algorithm for molecular dynamics of short range interacting particles. *Intern. J. Mod. Phys. C*, 4:1049, 1993.

[BP98] Volkhard Buchholtz and Thorsten Pöschel. Interaction of a granular stream with an obstacle. *Granular Matter*, 1:33–41, 1998.

[BP99] Nikolai V. Brilliantov and Thorsten Pöschel. Rolling as a continuing collision. *Europ. Phys. J. B*, 12:299, 1999.

[BSHP96] Nikolai V. Brilliantov, Frank Spahn, Jan-Martin Hertzsch, and Thorsten Pöschel. Model for collisions in granular gases. *Phys. Rev. E*, 53(5):5382–5392, May 1996.

[BSSS99] C. Bizon, M. D. Shattuck, J. B. Swift, and Harry L. Swinney. Transport coefficients for granular media from molecular dynamics simulations. *Phys. Rev. E*, 60(4):4340–4351, October 1999.

[CC70] S. Chapman and T. G. Cowling. *The Mathematical Theory of Non-uniform Gases*. Cambridge University Press, London, 1970.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[dG99] P. G. de Gennes. Granular matter: a tentative view. *Rev. Mod. Phys.*, 71(2):S374–S382, March 1999.

[DLK95] Yunson Du, Hao Li, and Leo P. Kadanoff. Breakdown of hydrodynamics in a one-dimensional system of inelastic particles. *Phys. Rev. Lett.*, 74(8):1268–1271, February 1995.

[EP97] Sergei E. Esipov and Thorsten Pöschel. The granular phase diagram. *J. Stat. Phys.*, 86:1385, 1997.

[FL02] Yichuan Fang and William W. Liou. Computations of the flow and heat transfer in microdevices using DSMC with implicit boundary conditions. *J. Heat Transf.*, 124:338–345, April 2002.

[FLCA94] Samuel F. Foerster, Michel Y. Louge, Hongder Chang, and Khedidja Allia. Measurements of the collision properties of small spheres. *Phys. Fluids*, 6(3):1108–1115, March 1994.

[GSB+98] D. Goldman, M. D. Shattuck, C. Bizon, W. D. McCormick, J. B. Swift, and Harry L. Swinney. Absence of inelastic collapse in a realistic three ball model. *Phys. Rev. E*, 57(4):4831–4833, April 1998.

[GZBN97] E. L. Grossman, Tong Zhou, and E. Ben-Naim. Towards granular hydrodynamics in two dimensions. *Phys. Rev. E*, 55(4):4200-4206, April 1997.

[Iso99] Masaharu Isobe. Simple and efficient algorithm for large scale molecular dynamics simulation in hard disk system. *International Journal of Modern Physics C*, 10(7):1281-1293, December 1999.

[JNB96] Heinrich M. Jaeger, Sidney R. Nagel, and Robert P. Behringer. Granular solids, liquids, and gases. *Rev. Mod. Phys.*, 68(4):1259-1273, October 1996.

[JR85] J. T. Jenkins and M. W. Richman. Kinetic theory for plane flows of a dense gas of identical, rough, inelastic, circular disks. *Phys. Fluids*, 28(12):3485-3494, December 1985.

[KH99] Anna Karion and Melany L. Hunt. Energy dissipation in sheared granular flows. *J. Heat Transfer*, 121(4):984-991, November 1999.

[Knu68] D. E. Knuth. *Fundamental Algorithms*, volume 1 of The Art of Computer Programming. Addison-Wesley, 1968.

[Knu73] D. E. Knuth. *Sorting and Searching*, volume 3 of The Art of Computer Programming. Addison Wesley, 1973.

[Kra96] Alan T. Krantz. Analysis of an efficient algorithm for the hard-sphere problem. *ACM Transactions on Modeling and Computer Simulation*, 6(3):185-209, July 1996.

[LF00] W. W. Liou and Y. C. Fang. Implicit boundary conditions for direct simulation Monte Carlo method in MEMS flow predictions. *Computer Modeling in Engineering & Sciences*, 1(4):119-128, 2000.

[LGIK01] A. Lamura, G. Gompper, T. Ihle, and D. M. Kroll. Multi-particle collision dynamics: Flow around a circular and a square cylinder. *Europhys. Lett.*, 56(3):319-325, November 2001.

[LHMZ98] S. Luding, M. Huthmann, S. McNamara, and A. Zippelius. Homogeneous cooling of rough, dissipative particles: Theory and simulations. *Phys. Rev. E*, 58(3):3416-3425, September 1998.

[LM98] Stefan Luding and Sean McNamara. How to handle the inelastic collapse of a dissipative hard-sphere gas with the TC model. *Granular Matter*, 1(3), 1998.

[Lub91] Boris D. Lubachevsky. How to simulate billiards and similar systems. *J. Comput. Phys.*, 94:255–283, 1991.

[Lud95] S. Luding. Granular materials under vibration: Simulations of rotating spheres. *Phys. Rev. E*, 52(4):4442–4457, October 1995.

[Lud98] S. Luding. Collisions & contacts between two particles. *Physics of Dry Granular Media*, 1998.

[Mar97] M. Marin. Billiards and related systems on the bulk-synchronous parallel model. *Proceedings of the 11th Workshop on Parallel and Distributed Simulation (PADS '97)*, 1997.

[MDK+00] Daniel M. Mueth, Georges F. Debregeas, Greg S. Karczmar, Peter J. Eng, Sidney R. Nagel, and Heinrich M. Jaeger. Signatures of granular microstructure in dense shear flows. *Nature*, 406:385–389, July 2000.

[MJS00] Hernán A. Makse, David L. Johnson, and Lawrence M. Schwartz. Packing of compressible granular materials. *Phys. Rev. Lett.*, 84(18):4160–4163, May 2000.

[MK02] Hernán A. Makse and Jorge Kurchan. Testing the thermodynamic approach to granular matter with a numerical model of a decisive experiment. *NATURE*, 415:614–617, February 2002.

[MSS01] Sung Joon Moon, M. D. Shattuck, and J. B. Swift. Velocity distributions and correlations in homogeneously heated granular media. *Phys. Rev. E*, 64:031303–1–031303–10, August 2001.

[MY94] Sean McNamara and W. R. Young. Inelastic collapse in two dimensions. *Phys. Rev. E*, 50(1):R28–R31, July 1994.

[MY96] Sean McNamara and W. R. Young. Dynamics of a freely evolving, two-dimensional granular medium. *Phys. Rev. E*, 53(5):5089–5100, May 1996.

[OU98] J. S. Olafsen and J. S. Urbach. Clustering, order, and collapse in a driven granular monolayer. *Phys. Rev. Lett.*, 81(20):4369–4372, November 1998.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[Rap87] D. C. Rapaport. Microscale hydrodynamics: Discrete-particle simulation of evolving flow patterns. *Phys. Rev. A*, 36(7):3288–3299, October 1987.

[Rap95] D.C. Rapaport. *The art of molecular dynamics simulation*. Cambridge University Press, 1995.

[RBSS02] Erin C. Rericha, Chris Bizon, Mark D. Shattuck, and Harry L. Swinney. Shocks in supersonic sand. *Phys. Rev. Lett.*, 88(1):014302-1–014302-4, January 2002. cond-mat/0104474 (2001).

[RC86] D. C. Rapaport and E. Clementi. Eddy formation in obstructed fluid flow: A molecular-dynamics study. *Phys. Rev. Lett.*, 57(6):695–698, August 1986.

[Sch96] Herbert Schildt. *Schildt's Expert C++*. Osborne McGraw-Hill, April 1996.

[TKS97] Gabriel I. Tardos, M. Irfan Khan, and David G. Schaeffer. Forces on a slowly rotating, rough cylinder in a couette device containing a dry, frictional powder. *Phys. Fluids*, 10(2):335–341, February 1997.

[TM00] Gabriel I. Tardos and Sean McNamara. A fluid mechanistic approach to slow (coulomb) and intermediate powder flows. 2000. Prepared for AIChE 2000 Annual Meeting, Los Angeles, CA, Nov. 12-17.

[TTKB98] Riina Tehver, Flavio Toigo, Joel Koplik, and Jayanth R. Banavar. Thermal walls in computer simulations. *Phys. Rev. E*, 57(1):R17–R20, January 1998.