From microscopic simulations to macroscopic material behavior

Stefan Luding

ICA1, Universität Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart

New address: DelftChemTech, TU Delft, Julianalaan 136, 2628 BL Delft, The Netherlands

e-mail: `lui@ica1.uni-stuttgart.de` *and* `s.luding@tnw.tudelft.nl` – *homepage:* `www.ica1.uni-stuttgart.de/~lui`

Abstract

One challenge in computational physics is to bridge the gap between microscopic, "atomistic" sizes and the macroscopic length scale of experimental observation. First, an efficient algorithm for hard sphere molecular dynamics is presented, allowing for many-particle simulations of, e.g., granular systems. In the next step, a "micro-macro" transition is introduced which enables continuum quantities like the stress tensor to be accessed. The approach is used for dense and dissipative gases but can also be applied to more complicated systems like membranes.

Key words: granular matter, molecular dynamics, micro-macro transition, equation of state, stress tensor

PACS: 46.10.+z, 51.30.+i, 51.10.+y, 64.70.Dv, 45.70.-n

1. Introduction

A straightforward approach towards the understanding of macroscopic material behavior by just modeling and simulating all atoms in a macroscopic system is not possible due to the huge number of degrees of freedom. Therefore, one can reduce the size of the examined system so that a microscopic simulation of atoms is possible. However, the possible length of such a "probe" is in general too small in order to regard it as macroscopic. Therefore, methods and tools to perform a so-called micro-macro transition [1–4] are discussed in this study. In a first step, microscopic simulations of a small sample lead to macroscopic laws needed to describe the material within the framework of a macroscopic theory, in a second step.

For granular materials, as an example, the grain properties are inserted into a discrete particle molecular dynamics (MD) and lead to the collective behavior of the dissipative many-particle system. From the particle simulation, one can extract, e.g., the pressure of the system as a function of density. This equation of state allows a macroscopic description of the material, which can be viewed as a compressible, non-Newtonian fluid [5]. Here we focus on the monodisperse hard sphere model which exhibits a disorder-order transition at a certain density. For low densities, the system resembles a dilute gas, for intermediate densities one has a disordered fluid, and for the highest densities, one obtains an ordered solid (evidenced by a crystal structure). The elastic hard spheres can be generalized by including inelasticity so that one ends up with a dissipative system for which, however, the micro-macro approach still can be applied.

From the algorithmic point of view, the model system is examined by an event-driven molecular dynamics simulation. One can compute the stress tensor and thus the pressure by summation over the momentum transfer per unit time and volume. In the usual time-driven molecular dynamics, the procedure is similar, only that forces have to be measured. The pressure can be derived for a dynamic system by means of kinetic-theory arguments [2], and for a quasi-static system by means of an average over particles [6] or by a virtual displacement method [1].

Examples are presented in the following, where the above-described methods can be applied and where large-scale computation was used. Standard simulations involve 10^4 - 10^5 particles due to the efficiency of the event-driven algorithm and can be extended to much larger numbers by using supercomputers.

2. Model and Algorithm

In this section, the hard sphere model is introduced together with the event-driven algorithm. The generalized model takes into account a finite contact duration and, besides this physcial parameter, saves computing time.

2.1. *Hard Sphere Model*

The particles are assumed to be perfectly rigid and follow an undisturbed motion until a collision occurs as described below. Due to the rigidity of the interaction, the collisions occur instantaneously, so that an event-driven simulation method [7, 8] can be used.

A change in velocity – and thus a change in energy – can occur only at a collision. The standard interaction model for instantaneous collisions of particles with radius a, mass $m = (4/3)\pi\rho a^3$, and material density ρ is used in the following. (Using the mass of a sphere is an arbitrary choice.) The post-collisional velocities $\boldsymbol{v}'$ of two collision partners in their center of mass reference frame are given, in terms of the pre-collisional velocities $\boldsymbol{v}$, by

$$\boldsymbol{v}'_{1,2} = \boldsymbol{v}_{1,2} \mp (1+r)\boldsymbol{v}_n/2 \,, \qquad (1)$$

with $\boldsymbol{v}_n \equiv [(\boldsymbol{v}_1 - \boldsymbol{v}_2)\cdot\hat{\boldsymbol{n}}]\,\hat{\boldsymbol{n}}$, the normal component of the relative velocity $\boldsymbol{v}_1 - \boldsymbol{v}_2$, parallel to $\hat{\boldsymbol{n}}$, the unit vector pointing along the line connecting the centers of the colliding particles, and the reduced mass $m_{12} = m_1 m_2/(m_1+m_2)$. If two particles collide, their velocities are changed according to Eq. (1), with the change of the translational energy at a collision $\Delta E = -m_{12}(1-r^2)v_n^2/2$, with dissipation for restitution coefficients $r < 1$.

2.2. *Event-Driven Algorithm*

Since we are interested in the behavior of granular particles, possibly evolving over several decades in time, we use an event-driven (ED) method which discretizes the sequence of events with a variable time step adapted to the problem. This is different from classical MD simulations, where the time step is usually fixed.

In the ED simulations, the particles follow an undisturbed translational motion until an event occurs. An event is either the collision of two particles or the collision of one particle with a boundary of a cell (in the linked-cell structure) [9]. The cells have no effect on the particle motion here; they were solely introduced to accelerate the search for future collision partners in the algorithm.

Simple ED algorithms update the whole system after each event, a method which is straightforward, but inefficient for large numbers of particles. In Ref. [7] an ED algorithm was introduced which updates only those two particles which were involved in the last collision. For this a double buffering data structure is implemented, which contains the 'old' status and the 'new' status, each consisting of: time of event, positions, velocities, and event partners. When a collision occurs, the 'old' and 'new' status of the participating particles are exchanged. Thus, the former 'new' status becomes the actual 'old' one, while the former 'old' status becomes the 'new' one and is then free for the calculation and storage of possible future events. This seemingly complicated exchange of information is carried out extremely simply and fast by only exchanging the pointers to the 'new' and 'old' status respectively. Note that the 'old' status of particle i has to be kept in memory, in order to update the time of the next contact, t_{ij}, of particle i with any other object j if the latter, independently, changed its status due to a collision with yet another particle. During the simulation such updates may be neccessary several times so that the predicted 'new' status has to be modified.

The minimum of all t_{ij} is stored in the 'new' status of particle i, together with the corresponding partner j. Depending on the implementation, positions and velocities after the collision can also be calculated. This would be a waste of computer time, since before the time t_{ij}, the predicted partners i and j might be involved in several collisions with other particles, so that we apply a delayed update scheme [7]. The minimum times of event, i.e. the times which indicate the next event for a certain particle, are stored in an ordered heap tree, such that the next event is found at the top of the heap with a computational effort of $O(1)$; changing the position of one particle in the tree from the top to a new position needs $O(\log N)$ operations. The search for possible collision partners is accelerated by the use of a standard linked-cell data structure and consumes $O(1)$ of numerical resources. In total, this results in a numerical effort of $O(N \log N)$ for N particles. For a detailed description of the algorithm see Ref. [7]. Using all these algorithmic tricks, we are able to simulate about 10^5 particles within reasonable time on a low-end PC [10], where the particle number is rather limited by memory than by CPU power.

2.3. *The Generalized TC Model*

In the ED method the contact duration is implicitly zero, matching well the corresponding assumption of instantaneous contacts used for the kinetic theory [11]. Due to this artificial simplification (which disregards the fact that a real contact takes always finite time) ED algorithms run into problems when the time between events $t_{\rm n}$ gets too small: In dense systems with strong dissipation, $t_{\rm n}$ may tend towards zero. As a consequence the so-called "inelastic collapse" can occur, i.e. the divergence of the number of events per unit time. The problem of the inelastic collapse [12] can be avoided using restitution coefficients dependent on the time elapsed since the last event [8]. For the contact that occurs at time t_{ij} between particles i and j, one uses $r = 1$ if at least one of the partners involved had a collision with another particle later than $t_{ij} - t_{\rm c}$. The time $t_{\rm c}$ can be seen as a typical duration of a contact. The effect of $t_{\rm c}$ on the simulation results is negligible for large r and small $t_{\rm c}$, for a more detailed discussion see [8, 10].

3. The Stress in Particle Simulations

The simplest contribution to the stress tensor is caused by the momentum transport due to the particle motion. This is the standard stress in an ideal gas, where the atoms (mass points) move with a certain fluctuation velocity $\boldsymbol{v}$. The kinetic energy $E = \sum_{i=1}^{N} m v_i^2/2$ due to the fluctuation velocity v_i can be used to define the temperature of the gas $k_B T = 2E/(DN)$, with the dimension D and the particle number N. Given a number density $n = N/V$, the stress in the ideal gas is then isotropic and thus quantified by the pressure $p = n k_B T$.

The additional contribution to the stress is due to collisions and contacts and will be derived from the principle of virtual displacement of soft interaction potentials and then modified for hard sphere systems.

3.1. *Stress from a Virtual Displacement*

From the centers of mass $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ of two particles, we define the so-called branch vector $\boldsymbol{l} = \boldsymbol{r}_1 - \boldsymbol{r}_2$, with the reference distance $l = |\boldsymbol{l}| = 2a_0$ and the corresponding unit vector $\hat{\boldsymbol{n}} = \boldsymbol{l}/l$.

The deformation in the normal direction, relative to the reference configuration, is defined as $\boldsymbol{\Delta} = \boldsymbol{l} - 2a_0\hat{\boldsymbol{n}}$. A virtual change of the deformation is then

$$\delta\boldsymbol{\Delta} = \boldsymbol{\Delta}' - \boldsymbol{\Delta} \approx \delta\boldsymbol{l} = \boldsymbol{\varepsilon} \cdot \boldsymbol{l} \ , \qquad (2)$$

where the prime denotes the deformation after the virtual displacement described by the tensor $\boldsymbol{\varepsilon}$.

The corresponding potential energy density due to the displacement of one pair of particles is $u = k\boldsymbol{\Delta}^2/(2V)$, expanded to second order in $\boldsymbol{\Delta}$, leading to the virtual change

$$\delta u = \frac{k}{V}\left(\boldsymbol{\Delta} \cdot \delta\boldsymbol{\Delta} + \frac{1}{2}(\delta\Delta)^2\right) \approx \frac{k}{V}\boldsymbol{\Delta} \cdot \delta\boldsymbol{l}^n \ , \qquad (3)$$

where k is the spring stiffness (the prefactor of the quadratic term in the series expansion of the interaction potential), V is the averaging volume, and $\delta \boldsymbol{l}^n = \hat{\boldsymbol{n}}(\hat{\boldsymbol{n}} \cdot \boldsymbol{\varepsilon} \cdot \boldsymbol{l})$ is the normal component of $\delta \boldsymbol{l}$. (Note that δu depends only on the normal component of $\delta \boldsymbol{\Delta}$ due to the scalar product with $\boldsymbol{\Delta}$, which is parallel to $\hat{\boldsymbol{n}}$.)

From the potential energy density, we obtain the stress from a virtual deformation by differentiation with respect to the deformation tensor components

$$\boldsymbol{\sigma} = \frac{\partial u}{\partial \boldsymbol{\varepsilon}} = \frac{k}{V} \boldsymbol{\Delta} \otimes \boldsymbol{l} = \frac{1}{V} \boldsymbol{f} \otimes \boldsymbol{l} \; , \tag{4}$$

where $\boldsymbol{f} = k\boldsymbol{\Delta}$ is the force acting at the contact, and the dyadic product $\otimes$ of two vectors leads to a tensor of rank two.

3.2. *Stress in Hard Sphere Systems*

Combining the ideal and the collisional contributions to the stress tensor [13], one has

$$\boldsymbol{\sigma} = \frac{1}{V} \left[\sum_i m_i \boldsymbol{v}_i \otimes \boldsymbol{v}_i - \frac{1}{\Delta t} \sum_n \sum_j \boldsymbol{p}_j \otimes \boldsymbol{l}_j \right] , \tag{5}$$

where $\boldsymbol{p}_j$ and $\boldsymbol{l}_j$ are the momentum change and the center-contact vector of particle j at collision n, respectively. The sum in the left term runs over all particles i, the first sum in the right term runs over all collisions n occurring in the time interval Δt, and the second sum in the right term concerns the collision partners of collision n [8]. The force in Eq. (4) is thus replaced by the momentum exchange per unit time $\boldsymbol{f} = \boldsymbol{p}_j / \Delta t$.

3.3. *The Equation of State*

The mean pressure $p = (\sigma_1 + \sigma_2)/D$, with the eigenvalues σ_1 and σ_2 of the stress tensor, for $D = 2$ can now be obtained from simulations for different volume fractions [5, 13]. The dimensionless reduced pressure $P = pV/E - 1$ contains only the collisional contribution and the two-dimensional simulations agree nicely with the theoretical prediction $P_0 = 2\nu g_{2a}(\nu)$ [13], with the pair-correlation function $g_{2a}(\nu) = (1 - 7\nu/16)/(1-\nu)^2$, and the volume fraction $\nu = N\pi a^2/V$, see Fig. 1. A small correction to P_0, based on a fit to the numerical data, is $P_4 = P_0(1 - a_g \nu^4)$ with $a_g = 0.1$. The dimensionless pressure P is related to the collision rate $t_n^{-1} = 4\nu g_{2a}(\nu)\sqrt{T/(\pi a^2 m)} = 2P\sqrt{T/(\pi a^2 m)}$ and the temperature. For a system with homogeneous temperature, the collision rate is thus proportional to the dimensionless pressure $t_n^{-1} \propto P$.

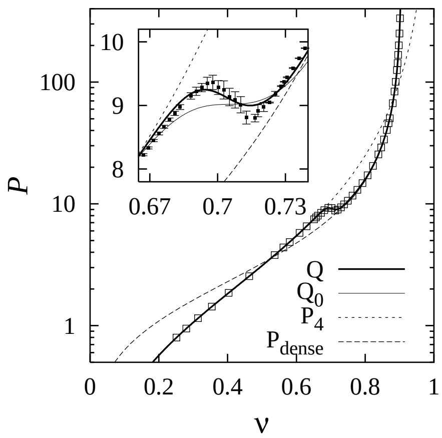

Fig. 1. The dashed lines are P_0 and $P_{\rm dense}$ as a function of the volume fraction ν, and the symbols are simulation data, with standard deviations as given by the error bars in the inset. The thick solid line is Q, the corrected global equation of state from Eq. (6) with the fit parameters $a_g = 0.1$, $a_d = 0.340$, $a_p = 1.09$, $\nu_c = 0.701$, $\nu_{\max} = 0.9069$, and $m_0 = 0.00928$. The thin solid line is Q_0 without corrections, i.e. $a_g = 0$, $a_d = 0$, and $m_0 = 0.0015$ and $\nu_c = 0.7$, so that $Q_0 = P_0 + m(\nu)[P_{\rm fv} - P_0]$.

When plotting P against ν with a logarithmic vertical axis, in Fig. 1, the simulation results can not be distinguished from P_0 for $\nu < 0.65$. Crystallization is evidenced at the point of the liquid-solid transition $\nu_c \approx 0.7$, and the data clearly deviate from P_0. The pressure is strongly reduced due to increase of free volume caused by order. The data diverge at the maximum packing fraction $\nu_{\max}^{\rm mono} = \pi/(2\sqrt{3})$ for a perfect triangular array. For high densities, one can compute from free-volume models, the reduced pressure $P_{\rm fv} = (\sqrt{\nu_{\max}/\nu} - 1)^{-1}$, with the maximum volume fraction $\nu_{\max}$ [13]. Slightly different functional forms do not lead to much better agreement [5]. Based on the numerical data, we propose the corrected high density pressure $P_{\rm dense} = [1 + a_d(\nu_{\max} - \nu)^{a_p}]\, P_{\rm fv}$, where the term in brackets $[\ldots]$ is a fit function with $a_d = 0.340$ and $a_p = 1.09$.

To our knowledge, no theory exists, which combines the disordered and the ordered regime. Therefore, we propose a global equation of state

$$Q = P_4 + m(\nu)[P_{\text{dense}} - P_4] \ , \tag{6}$$

with an empirical merging function $m(\nu) = [1 + \exp(-(\nu - \nu_c)/m_0)]^{-1}$ which selects P_4 for $\nu \ll \nu_c$ and P_{dense} for $\nu \gg \nu_c$ with the width of the transition m_0. In Fig. 1, the fit parameters $\nu_c \approx 0.70$ and $m_0 \approx 0.009$ lead to qualitative and quantitative agreement between Q (thick line) and the simulation results (symbols). However, a simpler version Q_0 (thin line) without numerical corrections also leads to reasonable agreement when $m_0 = 0.015$ is used. In the transition region, the function Q_0 has no negative slope but is continuous and differentiable, so that it allows for an easy and compact numerical integration of P. We selected the parameters for Q_0 as a compromise between the quality of the fit on the one hand and the treatability of the function on the other hand.

As an application of the global equation of state, the density profile of a dense granular gas in the gravitational field has been computed for monodisperse [13] and bidisperse situations [5]. In the latter case, however, segregation was observed and the mixture theory could not be applied.

4. Large-Scale Computational Examples

In this section, the ED algorithm is used to simulate a freely cooling dissipative gas [10] and a membrane consisting of hard spheres connected by strings. The large number of particles is necessary in order to allow for either good statistics or high resolution of patterns and structures.

4.1. *Cluster Growth*

In the following, a two-dimensional system of length $= l/d = 500$ with $N = 79524$ dissipative particles of diameter d is examined [8, 10], whose volume fraction is here $\nu = 0.25$. Initially the particles are arranged on a square lattice with random velocities drawn from an interval with constant probability for each coordinate. The mean total velocity, i.e. the random momentum due to the fluctuations, is eliminated in order to have a system with its center of mass at rest. The system is allowed to evolve for some time, until the arbitrary initial condition is forgotten, i.e. the density is homogeneous, and the velocity distribution is a Gaussian in each coordinate. Then dissipation is switched on and the evolution of the system is reported for $r = 0.8$. In order to avoid the inelastic collapse [10] artefact of the inelastic hard sphere model, the TC model is applied, which reduces dissipation if the time between collisions drops below a value of $t_c = 10^{-5}$ s.

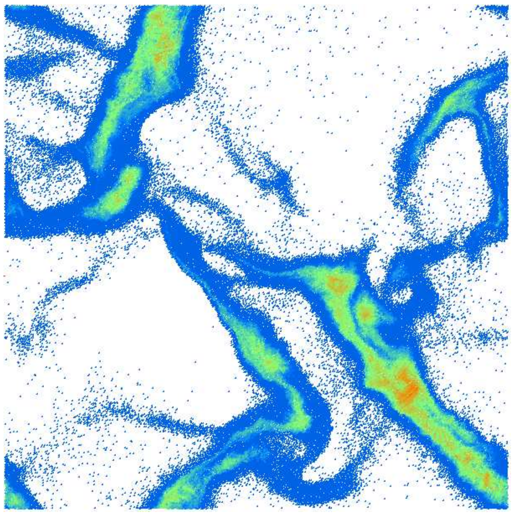

Fig. 2. Collision frequency of individual particles from a simulation with about 5200 collisions per particle. The colors indicate large (red), medium (green), and small (blue) collision rates.

For the value of r used here, the system becomes inhomogeneous quite rapidly [10]. Clusters, and thus also dilute regions, build up and have the tendency to grow. Since the system is finite, their extension will reach system size at a finite time. Thus we distinguish between three regimes of system evolution: (i) the initially (almost) homogeneous state, (ii) the cluster growth regime, and (iii) the system size dependent final stage where the clusters have reached system size. We note that a cluster does not behave like a solid body, but has internal motion and can eventually break into pieces after some time.

In Fig. 2, a snapshot is presented and the collision rate is color-coded. The collision rate and the pressure are higher inside the clusters than at their surface. Note that most of the computational effort is spent in predicting collisions and to compute the velocities after the collisions. Therefore, the regions with the largest collision frequencies require the major part of the computational resources.

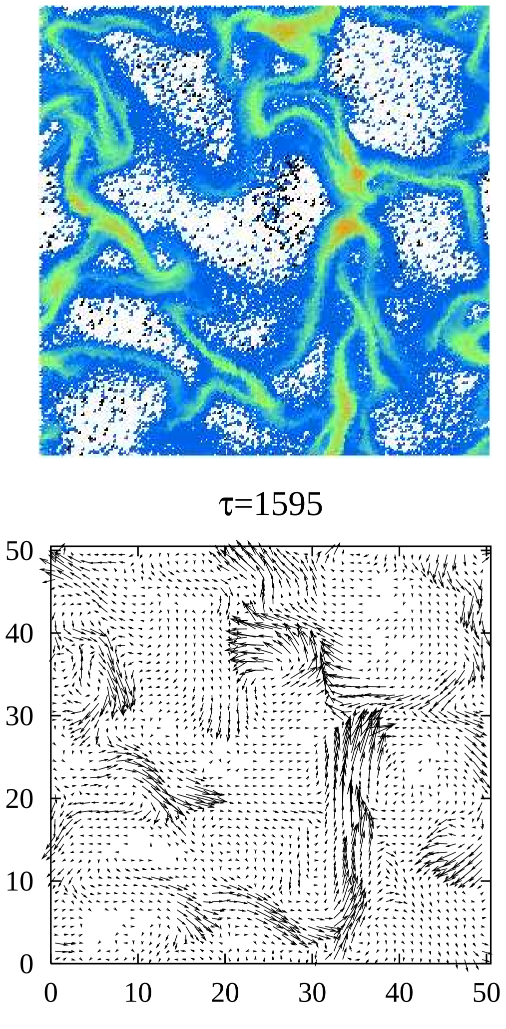

Fig. 3. Snapshot (top) of a system with $N = 99856$ particles, $r = 0.9$, and $\nu = 0.25$ together with the flux field (bottom) on a 50×50 grid. The dimensionless time τ roughly corresponds to the number of collisions per particle.

In Figs. 3 and 4, the macroscopic fields for a snapshot of another simulation are presented. Note that the clusters are not solid objects, but have internal (shear) motion. The density field is related to the snapshot, whereas the temperature and the pressure fields show patterns somewhat different from the density field.

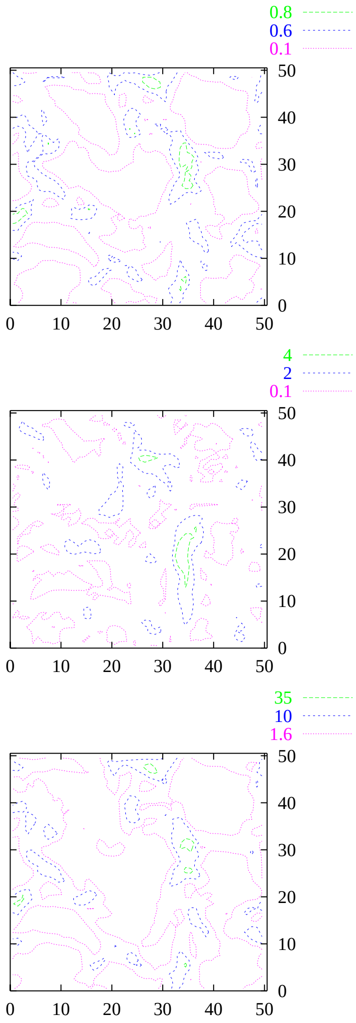

Fig. 4. Contour-plots of the density, temperature, and pressure fields on a 50×50 grid. The values of the iso-lines are given in arbitrary units, for the latter two plots.

4.2. *Membranes*

The event-driven simulations can simply be extended to account for attractive interactions, too [in preparation]. In addition to the repulsive hard sphere interaction, a string can be attached to the centers of connected particles. If the string is stretched to its maximum length, the velocities of the particles are exchanged in the normal direction. In Fig. 5, about $2.5\,10^4$ particles are arranged on a hexagonal lattice with random velocities, and neighbors are connected by strings of length $s/(2a) = 1.1$. After the simulation is started, the "membrane" evolves towards a state with spontaneous curvature and fluctuating collision rate, see Fig. 5.

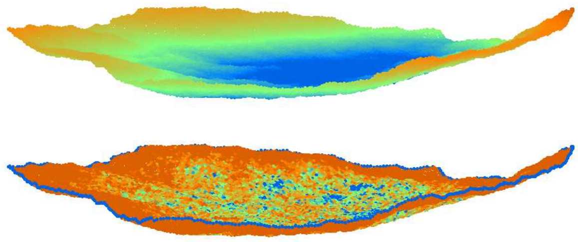

Fig. 5. Snapshot of a membrane simulated with the event-driven method. (Top) The vertical position is color-coded. (Bottom) The collision rate is color-coded.

5. Summary and Conclusion

As an example, simulations of freely cooling, two-dimensional systems were presented. The system is initially in a homogeneous cooling state, and the deviations from this theoretically well-understood situation occur due to fluctuations and dissipation. When dissipation is strong enough, density variations build up and lead to clusters of particles. The deviation from the homogeneous cooling state goes ahead with the growth of these formations.

Acknowledgements

We acknowledge the support of the the Deutsche Forschungsgemeinschaft (DFG).

References

[1] P. A. Vermeer, S. Diebels, W. Ehlers, H. J. Herrmann, S. Luding, and E. Ramm, editors. *Continuous and Discontinuous Modelling of Cohesive Frictional Materials*, Berlin, 2001. Springer. Lecture Notes in Physics 568.

[2] T. Pöschel and S. Luding. *Granular Gases*. Springer, Berlin, 2001. Lecture Notes in Physics 564.

[3] J. G. Kirkwood and Buff, J. Chem. Phys. 17:338, 1949.

[4] J. Christoffersen, M. M. Mehrabadi and S. Nemat-Nasser, *A micromechanical description of granular material behavior*, J. Appl. Mech. 48(2):339, 1981.

[5] S. Luding and O. Strauß. The equation of state of polydisperse granular gases. In T. Pöschel and S. Luding, editors, *Granular Gases*, pages 389–409, Berlin, 2001. Springer.

[6] M. Lätzel, S. Luding, and H. J. Herrmann. Macroscopic material properties from quasi-static, microscopic simulations of a two-dimensional shear-cell. *Granular Matter*, 2(3):123–135, 2000. cond-mat/0003180.

[7] B. D. Lubachevsky. How to simulate billiards and similar systems. *J. of Comp. Phys.*, 94(2):255, 1991.

[8] S. Luding and S. McNamara. How to handle the inelastic collapse of a dissipative hard-sphere gas with the TC model. *Granular Matter*, 1(3):113–128, 1998. cond-mat/9810009.

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

[10] S. Luding and H. J. Herrmann. Cluster growth in freely cooling granular media. *Chaos*, 9(3):673–681, 1999.

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

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

[13] S. Luding. Global equation of state of two-dimensional hard sphere systems. *Phys. Rev. E*, 63:042201–1–4, 2001.