GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

MAX-PLANCK-GESELLSCHAFT

Bachelorarbeit am Max-Planck-Institut Göttingen
Abt. Dynamik komplexer Fluide

Hydrodynamics of driven granular matter: Leidenfrost effect and convection rolls

Hydrodynamische Betrachtunng getriebener granularer Systeme: Leidenfrost Effekt und Konvektionsrollen

Joscha Tabet
joscha.tabet@stud.uni-goettingen.de

Erstgutachter:	Dr. Marco G. Mazza MPI für Dynamik und Selbstorganisation Göttingen Dynamik komplexer Fluide
Zweitgutachterin:	Prof. Annette Zippelius Georg-August-Universität Göttingen Institut für Theoretische Physik
Datum:	15. November 2016

Abstract

Two-dimensional driven granular matter exhibits a number of nonequilibrium steady states such as “undulations”, “Leidenfrost effect” and “convection rolls” that have been found both in experiments and molecular dynamics simulations. We study this system by solving the granular Navier-Stokes equations and employing direct numerical simulations. The Leidenfrost effect is reproduced and for the first time, convection rolls are observed in a hydrodynamic model of granular matter. Classifiers are developed for distinguishing the nonequilibrium steady states in the continuum representation of fields. The data is arranged according to two dimensionless control parameters and combined into a phase diagram. Lastly, the comparison to previous work on the subject is drawn.

Contents

1 Introduction

Granular matter is a term used to describe materials made of macroscopic particles, for which thermal motion is not important. The grain can be from microns to kilometers in size. The most important physical feature that determines granular dynamics is the dissipation of energy into smaller length scales. Many comparisons can be drawn to regular fluids, as they share some behavioral patterns. Granular matter can develop convection rolls resembling Rayleigh-Bénard convection rolls [1], undulations resembling Faraday waves [2], the granular counterpart of the Leidenfrost effect [3], and a wealth of other phenomena corresponding to fluid behavior. This motivates a key question in current granular matter research. To what extent can granular dynamics be described by a hydrodynamic-like model?

This Bachelor's thesis investigates the above question by employing direct numerical simulations (DNS) to explicitly solve the granular Navier-Stokes equations. Our goal is to see if the known phenomena can be reproduced and to map the phase space. We simulate vertically shaken granular matter in a container and develop a framework in MATLAB that automatically extracts information from the output data. We combine this information with observation to extract descriptors for the nonequilibrium steady states and develop a phase space diagram.

2 Model

2.1 Coefficient of Restitution

Microscopically, granular matter is made up of inelastically colliding particles. In the present work we assume that they are monodisperse spheres. If the particles deform very little during collision, which is the case for most materials, the collision time is very short and, consequently, the number of triple- and higher order collisions is vanishingly small. Thus, the dynamics of the system are exhaustively described by a sequence of pairwise collisions.

The elastic collision of two identical particles with initial velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ results in the final velocities $\boldsymbol{v}_1'$ and $\boldsymbol{v}_2'$ (in the following the final state will always be denoted by a prime), which are easily found from conservation of momentum

$$\boldsymbol{v}_1' = \boldsymbol{v}_1 - (\boldsymbol{v}_{12} \cdot \boldsymbol{e})\, \boldsymbol{e}, \qquad \boldsymbol{v}_2' = \boldsymbol{v}_2 + (\boldsymbol{v}_{12} \cdot \boldsymbol{e})\, \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{e} \equiv (\boldsymbol{r}_1 - \boldsymbol{r}_2) / |\boldsymbol{r}_1 - \boldsymbol{r}_2|$ is the unit vector joining the two centers of mass. For dissipatively colliding particles we define the coefficient of restitution $\varepsilon = \boldsymbol{v}_{12}' \cdot \boldsymbol{e} / \boldsymbol{v}_{12} \cdot \boldsymbol{e}$ as the ratio of the relative velocity before $\boldsymbol{v}_{12} \equiv \boldsymbol{v}_1 - \boldsymbol{v}_2$ and after the collision. Conservation of momentum leads to

$$\boldsymbol{v}_1' = \boldsymbol{v}_1 - \frac{1}{2}(1+\varepsilon)\,(\boldsymbol{v}_{12} \cdot \boldsymbol{e})\, \boldsymbol{e}, \qquad \boldsymbol{v}_2' = \boldsymbol{v}_2 + \frac{1}{2}(1+\varepsilon)\,(\boldsymbol{v}_{12} \cdot \boldsymbol{e})\, \boldsymbol{e}. \tag{2}$$

The coefficient of restitution ε characterizes the dissipative properties of the collision,

as during each collision the involved particles lose a fraction $\frac{1}{4}m_p(1-\varepsilon^2)(\boldsymbol{v}_{12}\cdot\boldsymbol{e})^2$ of their kinetic energy [4]. The coefficient of restitution may be obtained from the elastic and dissipative interaction forces and, experimentally, turns out to be a function of the impact velocity [5]

$$\varepsilon = 1 - C_1\left|\boldsymbol{v}_{12}\cdot\boldsymbol{e}\right|^{1/5} + C_2\left|\boldsymbol{v}_{12}\cdot\boldsymbol{e}\right|^{2/5} \mp ....$$

In the remainder of this thesis we take the simplifying assumption for it to be a constant material property.

2.2 Granular Temperature

The kinetic energy of a granular medium can be decomposed into directed movement and fluctuation. The former results in a convective flux and mass transport. The latter is called "thermal" (granular) velocity $\boldsymbol{u}$ and is studied in terms of the granular temperature

$$T = \frac{1}{2}(\langle u_1^2\rangle + \langle u_2^2\rangle) \tag{3}$$

where $\langle\,\rangle$ denotes the ensemble average and u_i is the fluctuating velocity in direction i which has a time average of $\langle u_i\rangle_t = 0$.

2.3 Continuum Description

In this thesis we deal with driven granular matter contained in a rectangular box. Instead of considering the position and velocity of individual particles, one may examine hydrodynamic fields. Namely, the velocity $\boldsymbol{v}(\boldsymbol{r},t)$, the pressure $p(\boldsymbol{r},t)$, the granular temperature $T(\boldsymbol{r},t)$ and the density $\rho(\boldsymbol{r},t)$ (all particles are defined to have material density 1). To obtain the time evolution of these fields we solve the hydrodynamic equations for granular matter.

The dominance of hydrodynamics in molecular fluids relies on the fact that the asymptotic state is uniform and stationary. In undriven granular systems stationarity is not achieved due to the energy dissipation. Still, Brey *et al.* show that a universal solution is approached asymptotically for a wide class of spatially homogeneous initial conditions, which is only indirectly time dependent through the temperature field [6]. As a generalization of the Chapman-Enskog method for deriving hydrodynamics from the Boltzmann equation, granular hydrodynamics can be constructed by perturbative expansion of this state in powers of the gradients of the hydrodynamic fields [6, 7]. The reference state can be far from the Maxwellian and may change on time scales of similar to the relaxation time of small density perturbations, in the case of strong dissipation. The long term domination of hydrodynamics is still assured by the spatial uniformity of the anymptotic state [6, 8]. Additional assumptions of this model are rapid granular flows, i.e. binary particle collisions [9], molecular chaos [8, 10, 11] and scale separation. The latter implies that the characteristic length scale is much larger than the mean free

path and the characteristic time scale is much larger than the time between collisions. The final condition breaks down in systems with high dissipation $(1-\varepsilon^2)$, even at low densities [12]. At high densities the molecular chaos assumption breaks down. It follows that the constitutive relations, which are required for the closure of hydrodynamics, can no longer be derived from first principle. Nonetheless, Meerson *et al.* showed that this domain can still be described by the granular Navier-Stokes equations [10]. They propose that this is because they consider driven systems, similar to those in this thesis, which have a vanishing mean flow and as a consequence, the viscosity terms are zero. In 2D with external forces $\boldsymbol{a}$ the Navier-Stokes granular hydrodynamics equations take on the form

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{4}$$

$$\frac{\mathrm{D}}{\mathrm{D}t}\boldsymbol{v} + \tfrac{1}{\rho m_p}\nabla \cdot \left\{ p\mathbb{1} - \eta\left[\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T - (\nabla \cdot \boldsymbol{v})\mathbb{1}\right]\right\} = \boldsymbol{a} \tag{5}$$

$$\frac{\mathrm{D}}{\mathrm{D}t}T - \tfrac{1}{\rho}\eta\left\{\left[\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T - (\nabla \cdot \boldsymbol{v})\mathbb{1}\right] : \nabla \boldsymbol{v}\right\} + \tfrac{1}{\rho} p \nabla \cdot \boldsymbol{v} + \tfrac{1}{\rho}\nabla \cdot \boldsymbol{q} = -\xi T \tag{6}$$

where m_p is the mass of a particle, ξ is the cooling rate, η is the viscosity, $\mathrm{D}/\mathrm{D}t \equiv \partial_t + (\boldsymbol{v}\cdot\nabla)$ is the material derivative and $\mathbb{1}$ is the unit tensor. The hydrostatic pressure p and the heat flux $\boldsymbol{q}$ are given by

$$p = \rho T\left[1 + (1+\varepsilon)\phi g_2(\rho)\right], \tag{7}$$

$$\boldsymbol{q} = -\kappa \nabla T - \mu \nabla \rho. \tag{8}$$

$g_2(\rho)$ is the Carnahan-Starling pair correlation function [13]

$$g_2(\rho) = \frac{1 - \frac{7}{16}\rho}{(1-\rho^2)}. \tag{9}$$

Equation 4 is the continuity equation of mass, Eq. 5 expresses the conservation of momentum and Eq. 6 gives the energy balance. Compared to the Navier-Stokes equations for molecular fluids there are differences in the transport coefficients, corrections to the pressure p and an additional term in the expression of the heat flux depending on the density gradient $-\mu\nabla\rho$, which we term "pycusthermal". The most important way in which the granular hydrodynamic equations differ from the Navier-Stokes equations is the cooling rate $0 \leq \xi \leq 1$ which is factored into the temperature on the right hand side of Eq. 6. It makes the system dissipative.

2.4 Knudsen Number

The Knudsen number $\mathcal{K} = \ell/R$ is the ratio of a particle's mean free path ℓ and a representative macroscopic length scale R. It is critical to evaluating our results, because it indicates the validity of the continuum approach. Hydrodynamics are applicable for $\mathcal{K} < \mathcal{K}_c = 0.1$ [14]. In some conditions this constraint holds for granular systems.

2.5 Driving

Generally, a granular fluid constantly loses energy due to dissipative collisions. According to Haff's law, the temperature follows the decrease [15]

$$T(t) = \frac{T_0}{(1 + t\tau^{-1})^{5/3}}, \tag{10}$$

where the characteristic cooling time τ can be calculated from the material properties of the system.

Because we are interested in steady states, we need to supply a source of energy injection into the system to balance the continuous energy loss. A very common and practical way to achieve this is by driving the system via moving external walls. In this way the dissipation of energy is counteracted by an external force field that behaves according to

$$\boldsymbol{a} = \boldsymbol{e}_y \Gamma \cdot \sin(2\pi f t), \tag{11}$$

with constant frequency f, thereby shaking the container.

From a computational point of view it is wise to gradually introduce the driving force into the system, so as to avoid numerical instabilities. At all times before a predetermined transient time t_{trans} all external forces are multiplied by

$$\gamma(t \leq t_{\text{trans}}) = 3\left(\frac{t}{t_{\text{trans}}}\right)^2 - 2\left(\frac{t}{t_{\text{trans}}}\right)^3 \tag{12}$$

which is the lowest order strictly increasing polynomial with vanishing derivatives at start and end point.

2.6 Model Parameters

The core microscopic parameters of granular matter are the particle diameter σ and the coefficient of restitution ε. The particle diameter cancels out in the nondimensionalization of most parameters. For the purposes of our simulation it is in fact only meaningful for the scaling of time and for the shaking strength S (see below). We consider a two-dimensional system with the y-direction conventionally chosen as the vertical direction along which gravity and driving take place. The nondimensionalized width and height of the container are L_x and L_y, measured in units of σ.

Driven granular matter may adopt a number of different phases. They are determined by the shaking of the container and the number of particle layers inside. The filling fraction ϕ is a primary input parameter to our simulation. We obtain the more meaningful number of particle layers F in the following way:

$$F = \frac{\phi L_y}{\eta_h} \tag{13}$$

where η_h is the maximum packing density in 2D, i.e., hexagonal packing with $\eta_h =$

$\pi/\sqrt{12} \approx 0.907$. This definition is not universal. Eshuis *et al.* left out the correction factor η_h^{-1} [1]. Using these parameters, the Knudsen number simplifies to [16]

$$\mathcal{K} = \frac{\ell}{R} = \frac{\pi}{8\phi}\frac{1}{L_y} = \frac{\pi}{8\eta_h F}. \tag{14}$$

Along with F and ϕ, the presence of particles in the container may also be described by the aspect ratio L_x/F. Theoretical results can most easily be obtained by assuming an infinite aspect ratio [17].

The container is shaken with frequency f and amplitude a. As they increase the behavior of the granular medium changes from that of a crystalline solid to a fluid-like motion. The universal shaking parameter is the shaking strength [2]

$$S_{\text{general}} = \frac{(a2\pi f)^2}{gl} \tag{15}$$

where l is the typical displacement of the particles and g is the gravitational acceleration $9.81\,\mathrm{m/s^2}$. At low fluidization the particles stay ordered so that $l = a$. Thus, the shaking parameter reduces to

$$\Gamma = \frac{a(2\pi f)^2}{g}, \tag{16}$$

called the shaking acceleration.

At high fluidization the displacement of particles is on the scale $l = \sigma$ and we obtain

$$S = \frac{(a2\pi f)^2}{g\sigma}. \tag{17}$$

For intermediate fluidization there is a competition of length scales and the choice of appropriate shaking parameter is not clear. The transition has been described in terms of a critical fluidization threshold [18] or as a gradual change [2]. With the way our DNS is set up we can adjust Γ, but not the amplitude a directly. As Eshuis *et al.* [2] did, we therefore assume Γ to be the primary shaking parameter.

2.7 Computational Model

Our goal is to solve equations (4)–(8) numerically for the density ρ, the momentum $\rho\boldsymbol{v}$ and the energy density ρE. Therefore, we discretize our system into finite volumes $\Delta x \Delta y$ with a resolution of $N_x \times N_y$. Integrating the granular hydrodynamic equations and applying them to these cells is very advantageous in our case. Unlike in a differential implementation, conservation laws are strictly maintained and stability of the solver is increased. Crucially, between time steps, we don't save the progression of every field and variable across space, but only the cell-averages. As per Gauss' theorem, the volume integrals are substituted by surface integrals over the cell's boundaries. These integrals are then calculated via Gaussian quadrature with 2 nodes in each dimension. This requires a higher resolution of the variables. The more precise values are interpolated

with a seventh order WENO method [19, 20]. The flux is approximated with the assistance of a multi-stage numerical scheme (MUSTA) [21]. At this stage we have reduced the time evolution to a Riemann problem. Figure 1 shows a flow chart of the iterative multi-stage scheme we use to solve it. Details can be found in Hummel's doctoral thesis [16]. The simulation uses a dynamic time step

$$\Delta t = C_{\mathrm{CFL}} \frac{\Delta x}{w_{\max}} \tag{18}$$

that adapts to the maximum propagation speed of a convective wave $w_{\max}$ (Courant-Friedrichs-Lewy condition) [22]. We typically use an initial CFL-number of $C_{\mathrm{CFL}} = 0.2$. When our simulations become unstable (specifically, when high density gradients develop), this number is halved, iteratively, but typically it is then necessary to decrease it by multiple orders of magnitude, making further simulation unfeasible.

To aid stability and validity of our solutions we have additional solvers available. They are, however, computationally expensive. In section 6.2 we examine their practicality. Briefly, time integration can be carried out explicitly with a second order arbitrary high order derivative (ADER) scheme [23] and for the diffusive flux we may use the explicit dGRP-flux. In addition, an implicit solver is available. Explicit methods work locally and calculate the next time step as a function of the current one. Their solutions only fulfill the underlying equations approximately. In contrast, implicit methods work backwards in that they adjust solutions that necessarily satisfy the governing equations to the previous time step. Implicit schemes have a much higher accuracy for a given time step Δt, but they take more time because they couple the entire calculation domain. The difference an implicit solver makes is demonstrated in section 6.2.

The principal parameters that our DNS allows us to adjust are shown in table 1. The length multiplier, number and size of the finite volumes produce the container volume $L_x \times L_y = LN_x\Delta x \times LN_y\Delta y$. The filling fraction ϕ is a rescaling of F according to equation 13 and the initial convective velocities of the finite volumes are normally distributed in random directions with expected absolute value v_{init}.

Number of finite volumes	N_xN_y
Size of finite volumes	$\Delta x\Delta y$
Length multiplier	L
Filling fraction	ϕ
Shaking acceleration	Γ
Gravity	g
Shaking frequency	f
Coefficient of restitution	ε
Friction coefficient	M
Initial velocity	v_{init}
Transient time	t_0

Table 1: Principal input parameters to our DNS

The boundaries of the container are treated in the same way as the rest of the

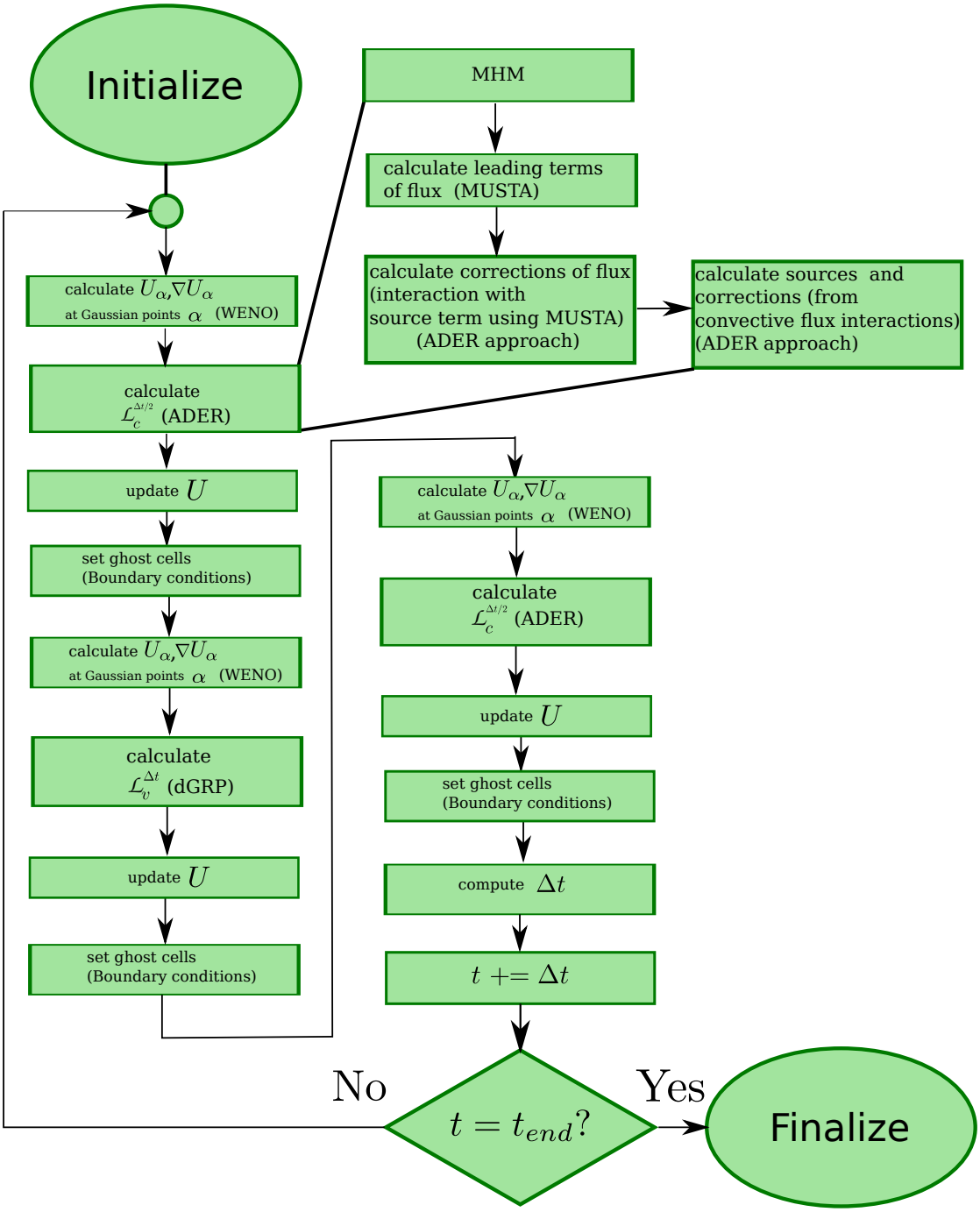


Figure 1: Flow chart of the structure of the algorithm for the direct numerical simulations [16].

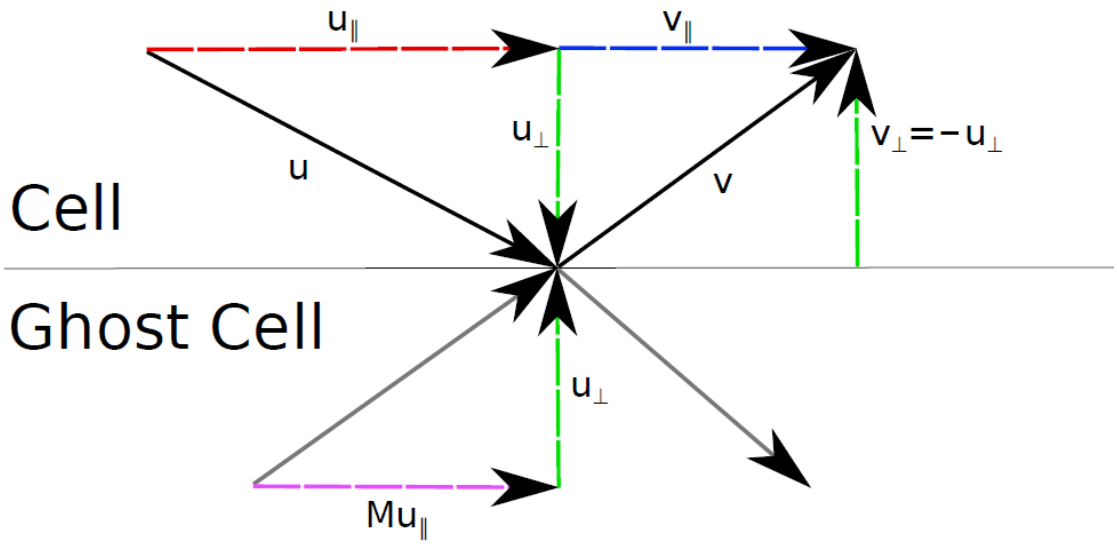


Figure 2: Friction at the walls

system. This is achieved by adding 4 ghost cells in every dimension, on each side. This number is needed because the WENO method requires 4 cells to perform interpolation. Analogously to image charges in electrodynamics, image fields are assigned to the ghost cells that solve the specified boundary conditions, e.g. mirrored fields for hard walls. If friction is activated, the ghost cell's velocity parallel to the wall gets multiplied by the friction coefficient M, as illustrated in figure 2. As $\boldsymbol{u}$ is transformed into $\boldsymbol{v}$, the perpendicular component (green) remains the same (changing sign), but the parallel component (red) changes in magnitude depending on M.

2.8 Physical Interpretation

All length scales are measured in terms of the particle diameter σ. A change of length scale requires a rescaling of time with

$$t_{\text{ref}} = \frac{L\sigma}{v_{\text{ref}}}. \tag{19}$$

In our simulations the length multiplier is $L = 2$, the reference velocity is $v_{\text{ref}} = v_{\text{init}} = 0.1\,\text{m/s}$ and the particle diameter is $\sigma = 1.9 \times 10^{-4}\,\text{m}$.

3 Nonequilibrium Steady States

Driven granular matter has been shown to exhibit many nonequilibrium steady states. Faraday heaping, oscillations, surface waves and different standing wave patterns in experiments and numerics have been documented [24, 25]. These require 3-dimensional systems and high shaking amplitudes, but some of the phases extend to 2D or quasi-2D setups. Namely, undulations, convection rolls and the granular Leidenfrost effect which may all develop from a bouncing bed.

3.1 Bouncing Bed

At $\Gamma \leq 1$ the granular system remains solid. There is almost no difference from the undriven state as the particle bed rests on the moving bottom of the container. If the box acceleration Γ is greater than both gravity and friction with walls and particle bed combined, a bouncing bed develops. The critical value Γ_{BB} depends on the materials used for the particles and the walls, but it is bounded by $\Gamma_{BB} < 2$ [2]. Figure 3 shows snapshots of a bouncing bed in an experiment, at different times.

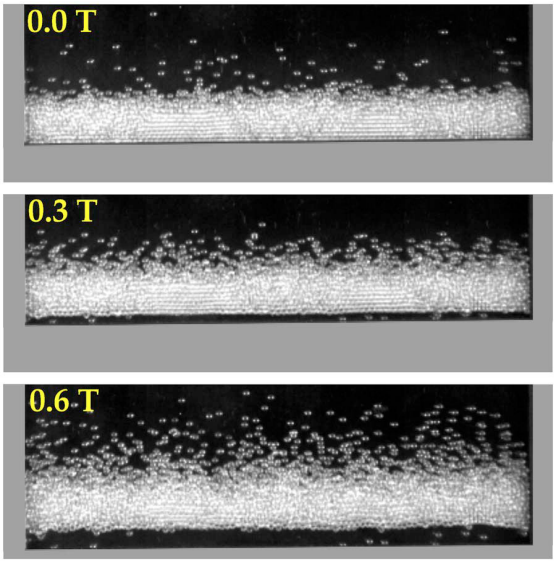


Figure 3: Bouncing bed for $F = 8.1$ layers of $\sigma = 1\,\text{mm}$ glass beads at amplitude $a = 4\,\text{mm}$ and frequency $f = 12\,\text{Hz}$ ($\Gamma = 2.3$). A snapshot is shown for three different phases within one shaking period T [2].

Starting from the bouncing bed and increasing Γ results in different phases, depending on the number of particle layers. Eshuis *et al.* observed a granular gas for $F \leq 3$, convection rolls at $3 < F \leq 6$ and undulations at $F > 6$ [2].

3.2 Undulations

Undulations, as seen in Figure 4, are transversal standing waves that form at mild fluidization. The standing wave property implies $L_x = n\lambda/2$ where n is an integer number. λ is connected to the elastic properties of the particles and n depends on the shaking (Γ and f).

As Γ increases, the first undulation mode $n = 1$ is formed from a bouncing bed in the following way: Whenever the bottom layer of the particle bed collides with the container, it dilates. A larger number of particle layers F leads to a higher pressure, resulting in increased dilation. Increasing the shaking acceleration Γ and consequently the kinetic energy in the system has the same effect. Once a certain threshold is reached, the horizontal expansion makes the particle bed buckle into a waveform.

From a different point of view, each collision with the bottom of the container causes compaction waves in the particle bed at velocity v, which can be assumed constant.

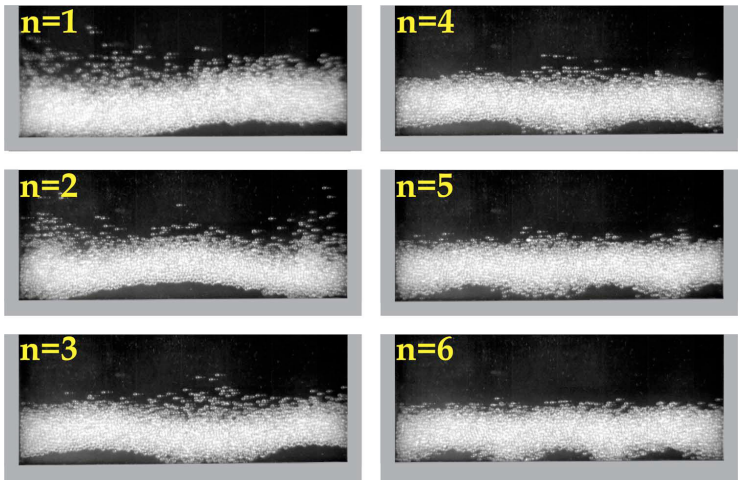


Figure 4: Undulation modes $n = 1, ..., 6$ for $F = 9.4$ layers, $\sigma = 1\,\text{mm}$ and $a = 2\,\text{mm}$, at shaking frequencies $f = 29.0, 32.6, 38.2, 39.3, 46.1, 50.2\,\text{Hz}$ [2].

They follow the arches and meet at the crests, causing them to fall down. In turn, the previously lower parts rise above the rest. Since this process repeats and becomes periodic after the next shaking period, undulations are also known as "$f/2$-waves". Experiments have shown that there is either a minimum or a maximum at the walls [2, 26].

Sano's model predicts the angle between particle bed and x-axis $\theta(s)$ at position s to follow the pendulum equation [26] with

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}s^2} = -\alpha \sin\theta. \tag{20}$$

Although it can be solved analytically, the small angle approximation, i.e. the harmonic oscillator, is a valid approximation for most purposes. The previously mentioned boundary conditions $\theta(0) = \theta(L_x) = 0$ result in

$$\theta(s) = \theta_{\max} \sin\left(\frac{n\pi s}{L}\right). \tag{21}$$

In the small angle approximation, the measured length along the particle bed s simplifies to the Cartesian coordinate x. Integrating over L_x yields the shape of the undulations

$$y(x) = \theta_{\max}\frac{L}{n\pi}\left[1 - \cos\left(\frac{n\pi x}{L}\right)\right]. \tag{22}$$

To get a better understanding of the different undulation modes, we take a more thorough look at shock waves. As previously discussed, they are essential to the formation of undulations. We can relate their velocity v to the geometry, mode and shaking Γ by considering the distance $\lambda/2$ traveled (across an arch) per shaking period. Consequently,

$$v = \frac{\lambda/2}{T} = \frac{L/n}{1/f} = \frac{Lf}{n} \tag{23}$$

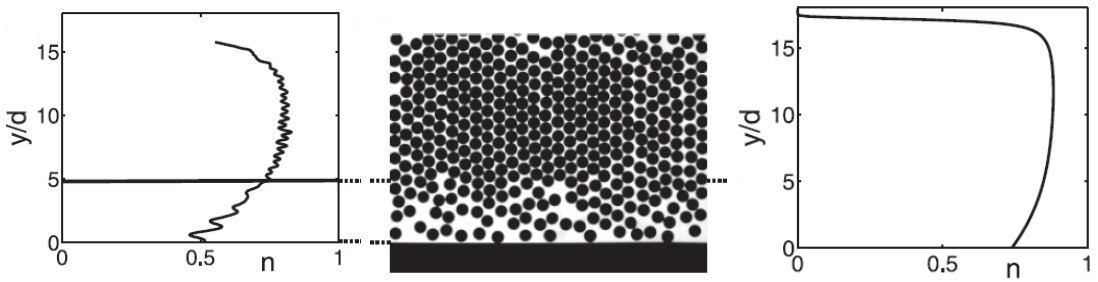
and the frequency required for a particular mode is

$$f_n = \frac{nv(n)}{L}. \tag{24}$$

We see that n increases as Γ rises. The indicated n-dependence of v has experimentally been shown to decrease according to $v(n) = a_v - b_v(n-1)$ with coefficients $a_v, b_v > 0$, at least for $n = 1, ..., 6$ [2]. In the setup of Eshuis *et al.* measurements suggested $v(n) = 2\,\mathrm{m/s} - (n-1) \cdot 0.2\,\mathrm{m/s}$ [2].

3.3 Granular Leidenfrost Effect

Originally, the Leidenfrost effect was observed in liquids that are laid on a hot surface. The liquid directly above the surface is vaporized and serves as a cushion for droplets that bounce on top of them. This marks a density inversion and the coexistence of the liquid- and the gas-state. Analogously, in a granular system a crystalline cluster of particles may float or bounce on top of a dilute (gaseous) layer as shown in figure 5. The gaseous cushion greatly diminishes the heat contact of the cluster, allowing it to remain in a solid- or liquid-like state. The cluster is stationary in time while the dilute layer oscillates along with the bottom of the container [3].



(a) Experimental density profile.

(b) High-speed camera snapshot of the experiment.

(c) Theoretical profile for $S = 80$.

Figure 5: Glass beads, vertically vibrated above a critical shaking acceleration, form a crystalline cluster that is elevated and supported by a vapor-like layer of fast particles underneath. Density profiles in experiment and theory at $F = 16$, $\Gamma = 51.5$, $f = 80\,\mathrm{Hz}$ by Eshuis *et al.* [3]. The height-origin is set at the maximal positive displacement of the bottom. The theoretical profile was determined by equations 25–27.

The granular Leidenfrost state arises at a shaking acceleration larger than that for undulations, provided that the number of particle layers satisfies $F \gtrsim 8$ [3]. Due to the aforementioned conflict of length scales, the crucial parameter for the onset of the granular Leidenfrost effect is not yet clear. Eshuis *et al.* have found conflicting results. In 2005 [3] they found the governing shaking parameter to be S. They worked with a particle diameter of $\sigma = 4\,\mathrm{mm}$, an aspect ratio on the order of $L_x/F \approx 1$ and a quasi 2D container, as $z \approx \sigma$. In a different paper in 2007 their experiments were in better agreement with Γ as the shaking parameter [2]. They attribute the difference to the change in setup. They increased Γ gradually, passing through undulations. Their experiments had a high aspect ratio $L_x/F \approx 10$ and a container-depth in z-direction of

$z = 5\,\sigma$. The particle diameter $\sigma = 1\,\mathrm{mm}$ was smaller by a factor of four.

Interestingly, the granular Leidenfrost effect may occur at shaking accelerations below the critical point, if the undulations become unstable. Especially the modes with odd n are energetically less favorable and result in intermediate Leidenfrost regions. A description of these regimes requires the parameters f and a rather than the usual Γ or S, as they only appear at specific frequencies and their total area in F-Γ-phase space depends on the amplitude. They occupy a smaller region as a increases and for $a \geq 4\,\sigma$ they disappear completely.

The granular Leidenfrost effect has been successfully modeled by a 1D hydrodynamic continuum description [3, 10]. This model solves the standard force balance of pressure p, particle mass m_p and number density n

$$\frac{\mathrm{d}p}{\mathrm{d}y} = -m_p g n \tag{25}$$

and an approximation of the energy balance between the heat flux through the vibrated bed and the dissipation due to the inelastic particle collisions

$$\frac{\mathrm{d}}{\mathrm{d}y}\left\{\kappa \frac{\mathrm{d}T}{\mathrm{d}y}\right\} = \frac{\mu}{\psi \ell} n (1-\varepsilon^2) T^(3/2) \tag{26}$$

with (granular) thermal conductivity $\kappa \propto T^(1/2)$ and constants μ, ψ. This is done using the equation of state

$$p = nT\frac{\eta_h + n}{\eta_h - n} \tag{27}$$

and three boundary conditions. Constant granular temperature $T_{\mathrm{bot}} = \text{const.}$ at the bottom of the container, zero energy flux at the top of the system $\lim_{y\to\infty}[\kappa(y)\mathrm{d}T/\mathrm{d}y] = 0$ and the conservation of particles $\int_0^\infty n(y)\mathrm{d}y = F\eta_h\sigma$.

3.4 Convection Rolls

At intermediate-to-strong fluidization granular matter develops circular convective fluxes called convection rolls. Figure 6 shows convection rolls in experiment, molecular dynamics (MD) simulation and theory.

Different types of convection rolls have been described. They can form from a bouncing bed and continue the bouncing motion, but in case of a large number of layers F the convection rolls develop from a Leidenfrost state. These convection roll clusters maintain the stationary position of their predecessors.

The greatest distinction is made between boundary driven convection at low fluidization and nearly homogeneous density, and buoyancy driven convection in the presence of large density gradients. At high shaking acceleration Γ and great enough aspect ratio L_x/F the Leidenfrost state's inverted density profile becomes unstable due to buoyancy and convection rolls develop [1]. This has been modeled numerically by Paolotti *et al.* [28] and there is agreement that S best describes this regime [2, 28]. Boundary driven

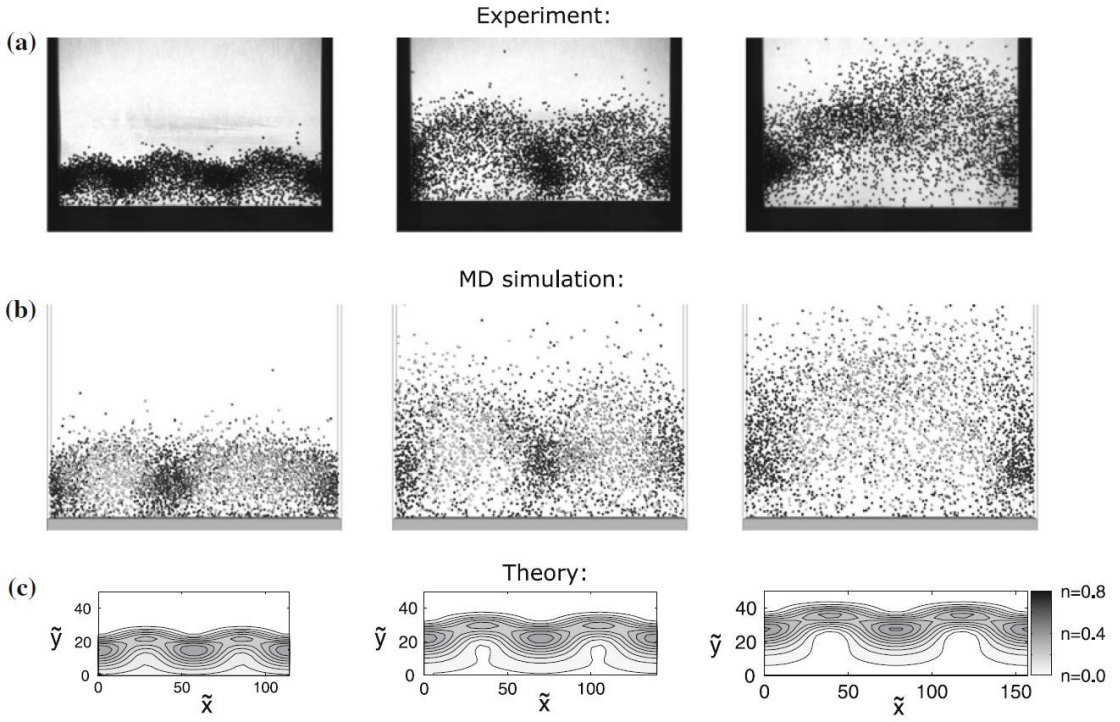


Figure 6: Convection patterns in experiment, MD simulation, and theory for $F = 6.2$ particle layers in a container of length $L/\sigma = 101$ at three consecutive shaking strengths: $S = 58$, $S = 130$, and $S = 202$ [27].

convection rolls develop at $\Gamma \approx 15$ [2] and will be our main focus. The convection rolls that form from a bouncing bed or from the Leidenfrost state develop in a similar way: Due to statistical fluctuations some parts of the cluster will develop a higher granular temperature, causing them to move upwards. The hole they leave will be filled by neighboring particles. This motion initiates the convection rolls [1]. The downward motion is most likely to be performed by cluster regions with low granular temperature. Therefore the first convection rolls develop near the dissipative walls. The onset of convection can be seen in figure 7.

The larger the number of layers F, the stronger the shaking needs to be in order to cause convection rolls. This is because the dissipation rate, which increases with F, must be compensated by the shaking-energy input. Eshuis *et al.* found the critical shaking strength for convection S_c to grow roughly exponentially as $S_c \propto \exp(0.2F)$ [1].

The number of rolls that form depends on the number of layers F and the shaking strength S. As F grows, the increasing energy dissipation makes the clusters, i.e. convection rolls bigger. Conversely, as S grows, the increasing energy input reduces the convection rolls in size. Depending on their size, a different number of convection rolls will fit into the container. However, due to the downward motion at the sidewalls the number will always be even. The change happens stepwise in increments of two rolls.

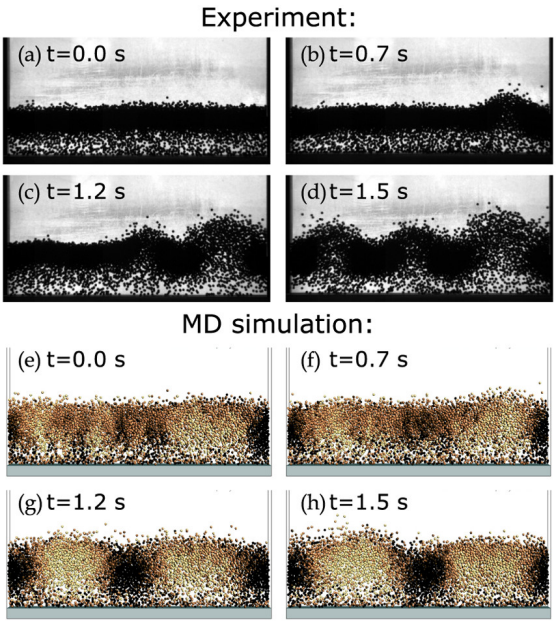


Figure 7: Onset of convection from the steady Leidenfrost state in experiment and in MD simulations for $F = 11.1$ layers of 1 mm steel beads in a container of length $L/d = 101$ shaken at an amplitude of $a = 3\,\mathrm{mm}$. The frequency was linearly increased from $f = 42\,\mathrm{Hz}$ to $f = 48\,\mathrm{Hz}$ and convection took place at $f = 45\,\mathrm{Hz}$. The light particles are moving upward and the dark ones downward [1].

4 Results

In our hydrodynamic DNS we were able to reproduce the Leidenfrost effect as well as convection rolls, but not undulations. We created an automated script that is able to analyze the data provided by our simulation output. We developed classifiers to assign data sets and their corresponding (Γ, F)-points in phase space to the the bouncing bed, Leidenfrost effect and convection rolls. This gives rise to a phase space diagram.

The observations detailed in this chapter can be followed via the attached videos. Included are both a representative example of the Leidenfrost effect and convection rolls.

4.1 Phase Space

We found that the occurrence of the granular Leidenfrost effect and convection rolls does not exclusively depend on the primary parameters F and Γ. We always maintained $\varepsilon = 0.95$, but many configurations of f, t_{trans}, L, $N_{x,y}$ and $\Delta(x, y)$ did not produce any effects in the later developed F-Γ-range. The parameter configuration we settled on can be seen in table 2 and is very close to the phase space region where we first discovered the effects. The influence of the secondary parameters is briefly dealt with in section 4.4.

4.2 Granular Leidenfrost Effect

Each of our simulations starts in a homogeneous state at $t = 0$. The particles have initial velocities $v_{\mathrm{init}} = 0.1$ with randomized direction. As time passes and the exterior forces (gravity and driving) are turned on, the density ρ condenses into a cluster near

Filling fraction	ϕ	6 to 16
Shaking acceleration	Γ	6 to 36
Number of finite volumes	$N_x \times N_y$	256×64
Size of finite volumes	$\Delta x \times \Delta y$	0.9×0.9
Length multiplier	L	2
Gravity	g	1
Shaking frequency	f	50 Hz
Coefficient of restitution	ε	0.95
Friction coefficient	M	0.9
Initial velocity	v_{init}	0.1
Transient time	t_{trans}	0.25 s

Table 2: Principal input parameters to our DNS and their respective values

the bottom. By around $t = 0.14\,\text{s}$ the cluster reaches its final mass, leaving the top half of the container with less than 5% density. Depending on Γ and F the system is now receptive to the Leidenfrost phase.

During the transient time each system moves on a trajectory $(F, \gamma(t)\Gamma)$ through F-Γ-phase space. We find that, as a critical point is crossed, the granular Leidenfrost effect develops. It takes only around 1–2 periods to stabilize, even if $t < t_{\text{trans}}$ and the exterior forces still change.

Barring convection rolls, the system becomes stationary at around $t = 0.38\,\text{s}$. Nonetheless, simulation instability required measurements to be taken earlier than that (detailed in section 6.4). Figure 8 shows the stationary state of two systems with the same number of particles, but different shaking accelerations.

(a) $\Gamma = 6$ (b) $\Gamma = 28$

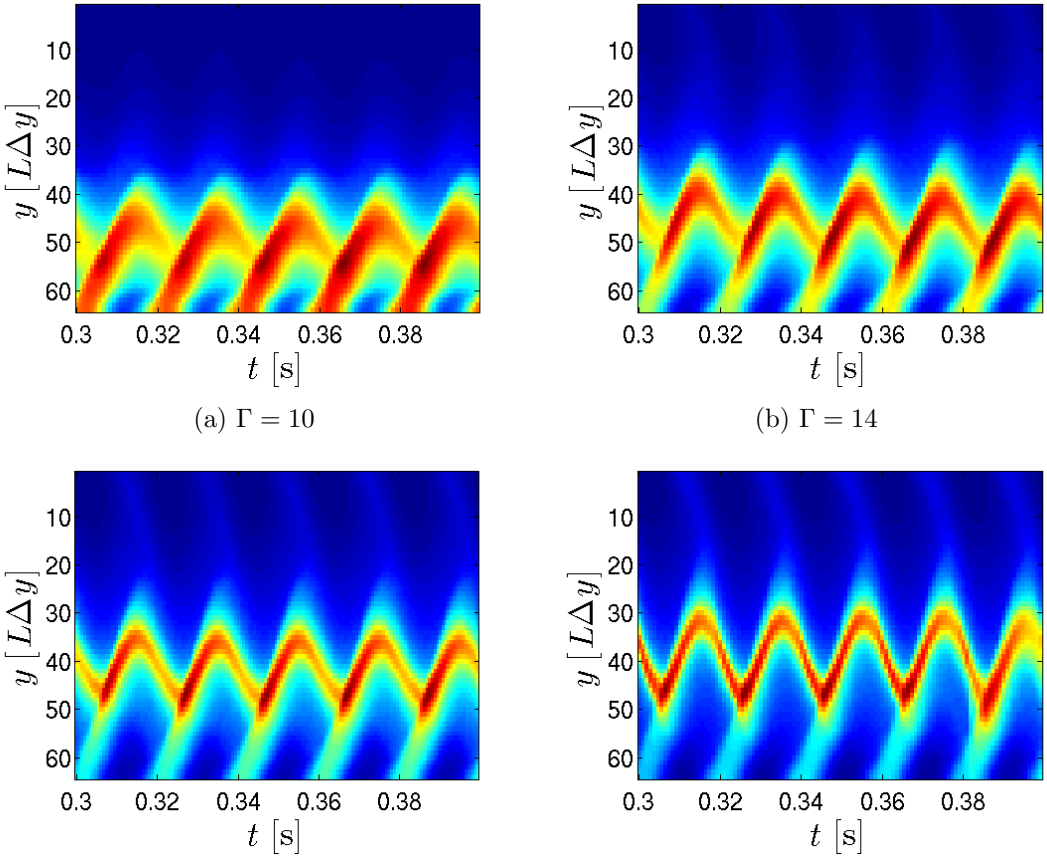
Figure 8: Density (black) $\rho(x, y, t)$ for $F = 17.8$ at $t = 0.60\,\text{s}$. 21% into a period. This and the following density distributions have been interpolated cubically.

There is a discernible visual difference between Fig. 8a and Fig. 8b, and given these extreme cases of $\Gamma = 6$ and $\Gamma = 28$ one may use the positions of the clusters at this point in time as a classifier for the Leidenfrost phase (figure 8b) and for the bouncing bed (figure 8a). This approach, however, breaks down just as quickly as the concepts of "clusters" and corresponding "positions".

The main issue in classifying the Leidenfrost effect is the smooth phase transition we encounter in F-Γ-phase space. Still, a time-inclusive investigation did gives us a clear distinction to the bouncing bed.

For a detailed analysis it is important to exclude the area around the side walls. As we worked with significant friction, the outer parts of the container exhibit boundary effects and behave very differently from the inner parts, i.e. faster, chaotic movement

and very low density (gas-like). In the numeric analysis of the Leidenfrost effect we chose to cut off 73 of the 256 cells on each side in the x-direction. Averaging over the remaining length in x we obtain $\bar{\rho}(y,t)$. Figure 9 shows examples of $\bar{\rho}$ for 5 periods, starting at $t = 0.3\,\mathrm{s}$. In our measurements and calculations the time before $t = 0.3\,\mathrm{s}$ is also cut off, but later times are maintained if possible (see sections 6.1 and 6.2).



(a) $\Gamma = 10$ (b) $\Gamma = 14$

(c) $\Gamma = 18$ (d) $\Gamma = 22$

Figure 9: $\bar{\rho}(y,t)$ in the reference frame of the container for $F = 14.0$ and $0.3\,\mathrm{s} \leq t < 0.4\,\mathrm{s}$. The density is color coded with red representing high density.

The graphs in figure 9 show two overlapping modes (most clearly visible in figures 9c and d). During the first quarter of a period some fraction of the particles rises from the bottom and some fraction falls from the crest. We may call these the bouncing bed mode and the Leidenfrost mode. Thus, we have reduced the classification problem to the identification of the dominant mode. Defining the quotient of the modes' densities

$$\kappa \equiv \frac{\langle \max_y \bar{\rho}(y > y_{\max}, t) \rangle_t}{\langle \max_y \bar{\rho}(y < y_{\max}, t) \rangle_t} \tag{28}$$

where the density maximum in y-direction (measured from the bottom) $y_{\max}$ is determined by

$$\langle \bar{\rho}(y = y_{\max}, t) \rangle_t = \max_y \langle \bar{\rho}(y,t) \rangle_t \tag{29}$$

we are left with the condition $\kappa > 0.5$ for the Leidenfrost phase. It produces the phase diagram in figure 10. Evidently it has deficiencies at low shaking acceleration. They are easily explained when looking at the corresponding $\bar{\rho}(y,t)$-diagrams in figure 11. In these cases $y_{\max}$ is so low that no part of the cluster will drop below it and only one mode emerges. The other breakdown of the classifier occurs when a system needs an unusually long time to become stationary and the simulations cut off too early (see section 6.4).

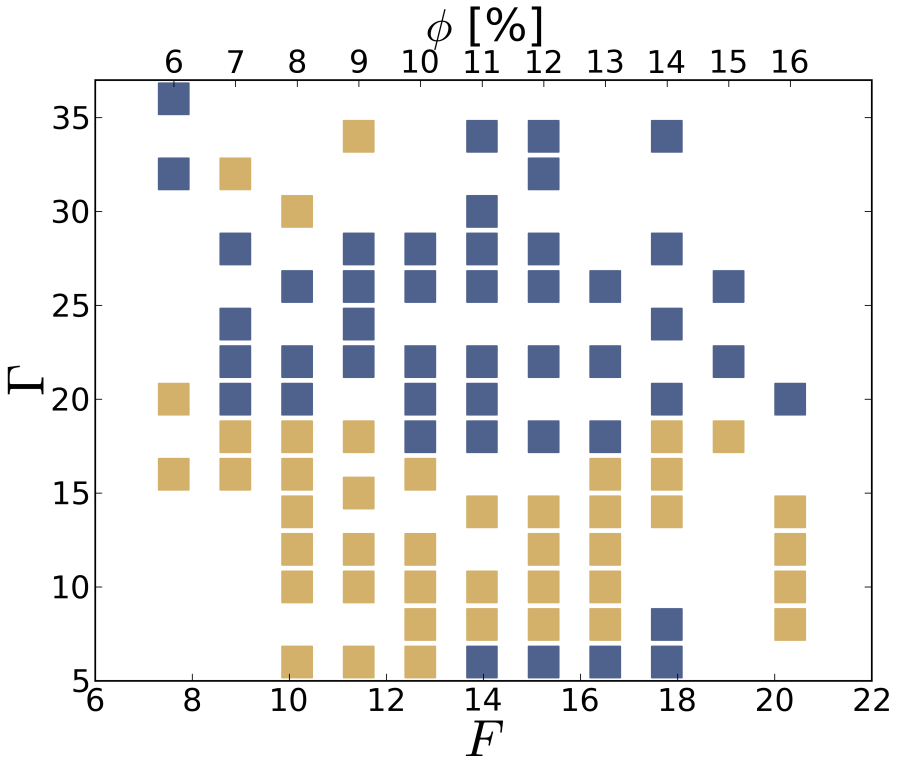
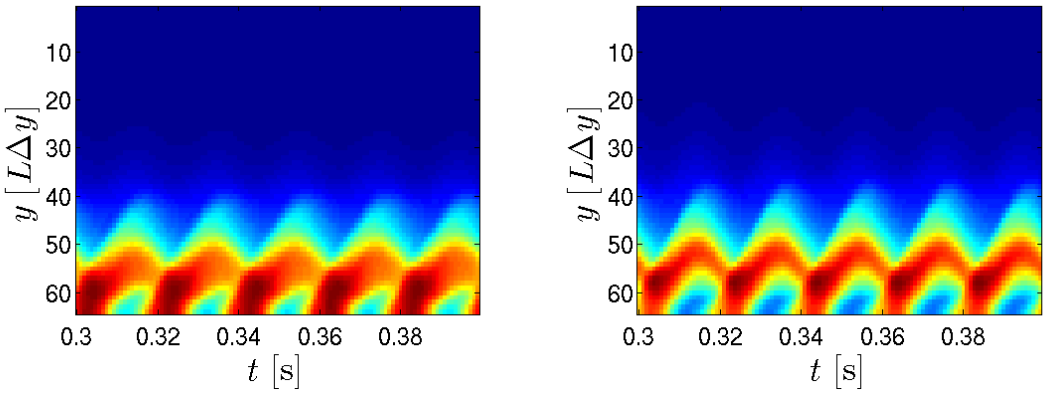


Figure 10: Leidenfrost effect (blue) according to $\kappa > 0.5$ in the F-Γ-phase diagram.



(a) $\Gamma = 6$ (b) $\Gamma = 8$

Figure 11: $\bar{\rho}(y,t)$ of a bouncing bed for $F = 17.8$ and $0.3\,\mathrm{s} \leq t < 0.4\,\mathrm{s}$.

Both deficiencies are remedied by one adjustment. Figures 9 and 11 show that the two modes only form as the densest part of the cluster peels away from the bottom with

increasing Γ. We may quantify this with $\beta y_{\max}$ where

$$\beta \equiv \frac{\langle \bar{\rho}(y = y_{\max}, t) \rangle_t}{\langle \bar{\rho}(y = 0, t) \rangle_t}. \tag{30}$$

In Leidenfrost states the cluster is always centered around $y = 34$. The condition

$$\kappa \beta y_{\max} > 34 \tag{31}$$

(in units of particle diameter σ) for the Leidenfrost phase is in very good agreement with our observations. The resulting phase space diagram can be seen in figure 12.

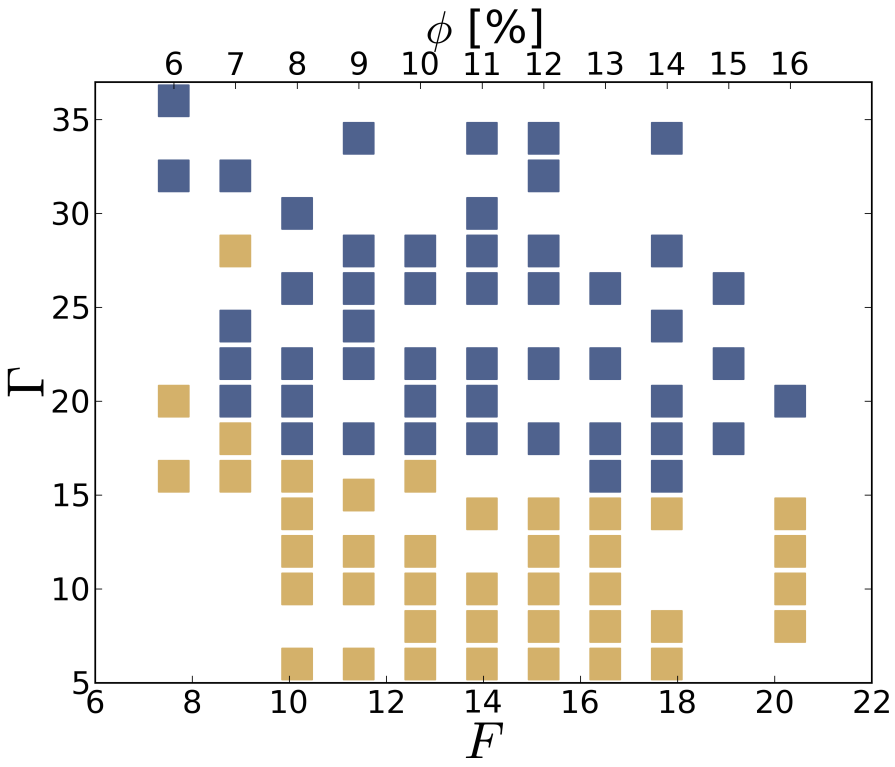


Figure 12: Leidenfrost effect (blue) in the F-Γ-phase diagram.

4.3 Convection Rolls

In many dilute systems (small F) we observe arches that form within the first 0.5 s. The field map for momentum $\rho \boldsymbol{u}(x, y, t)$ reveals that they are caused by convection rolls. Figure 13 shows a density map as well as the rolls and their sense of rotation.

In general, the convection rolls that we observe can be categorized into two different phases. Rolls that form from a bouncing bed and move up and down and rolls that form within the Leidenfrost regime. The latter continue to exhibit the characteristic traits of a cluster floating atop a gaseous layer.

Convection rolls take much more time to develop than the granular Leidenfrost effect. The process starts near the walls. At low shaking acceleration Γ two arches form next to the left and right boundaries (figure 14) within the first 0.25 seconds. The higher Γ, the more space separates them from the walls. Subsequently, new arches form next to

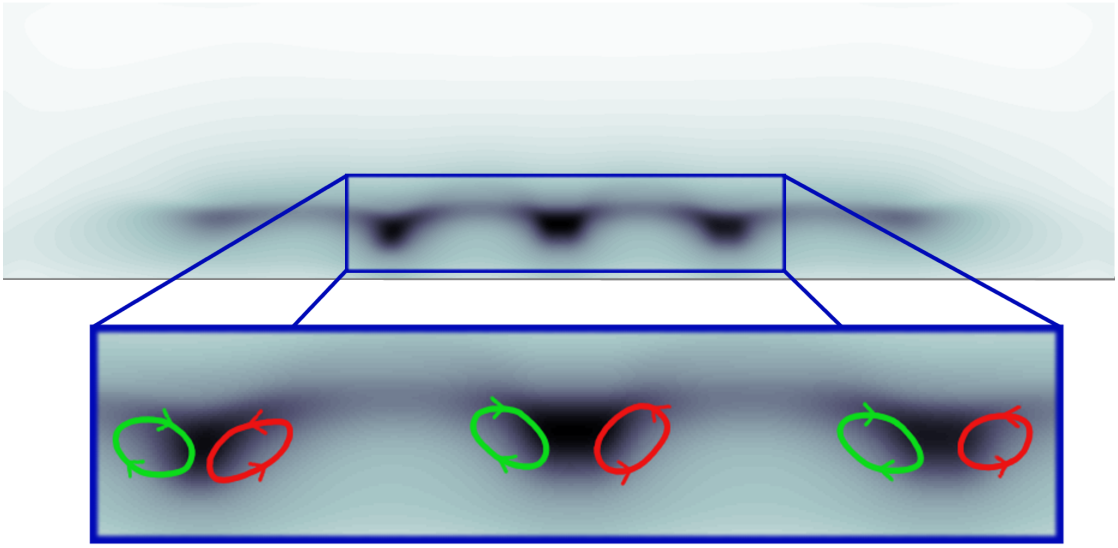
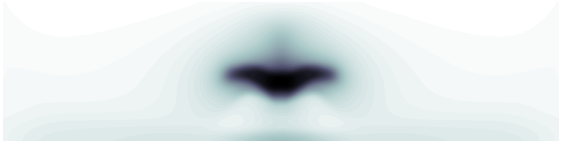


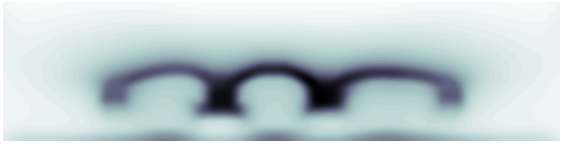
Figure 13: Convection rolls at $t = 0.50\,\mathrm{s}$ ($F = 11.43$, $\Gamma = 24$). Density $\rho(x, y)$ in black and momentum $\rho\boldsymbol{u}$ in green/red.

Figure 14: Onset of convection rolls at $t = 0.26\,\mathrm{s}$ ($F = 12.7$, $\Gamma = 18$). Only the outer arches have formed by this time.

the existing ones. That procedure continues inward until the whole cluster is buckled. At every meeting point of two arches there is an area with high density in which two convection rolls with opposing senses of rotation are located. This state is shown for different parameter configurations in figure 15. The pictures cover the whole range of convection roll patterns we observed. In most cases they are axially symmetrical, $\rho(L_x/2 + x, y, t) = \rho(L_x/2 - x, y, t)$. With increasing shaking acceleration Γ the walls' area of influence expands, leading to a contraction of the cluster. Therefore, figure 15b and 15d show identical cluster lengths. Figure 15a and similar configurations show that in some extreme cases there is only enough space left for one pair of convection rolls in the center. Starting at $t \approx 0.5\,\mathrm{s}$ we see the convection rolls slowly moving away from the



(a) $F = 8.9$, $\Gamma = 36$



(b) $F = 12.7$, $\Gamma = 26$

(c) $F = 15.2$, $\Gamma = 14$

(d) $F = 15.2$, $\Gamma = 26$

Figure 15: Density at $t = 0.50\,\mathrm{s}$ showing the developed convection roll phase.

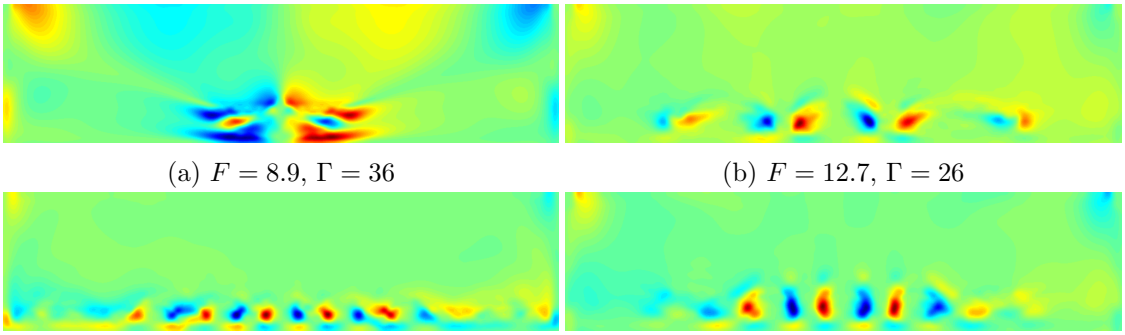
center with a convective velocity of about $v_{\text{rolls}} = (86 \pm 5)\,\sigma/\text{s}$. The movement continues until the rolls are around twice their initial distance apart, at which point the arches becomes unstable and collapse. Like in the onset of convection rolls, a new pair of rolls forms at the crests of the arches. They descend until they are in-line with the rest. Rolls that move beyond the ends of the cluster, as dictated by Γ, simply dissolve.

This whole process is not, yet, periodic. It takes around one pair of rolls to dissipate until a stationary state is reached (see section 6.4). The filigree delicateness of the convection roll pattern makes for high computational costs. Consequently, the long term observation of its behavior is limited. In exceptional cases of prolonged simulation time ($\approx 5\,\text{s}$) there seems to be a turning point at around 3.3 s where the roll movement reverses inward. Presumably, it will reverse periodically thereafter.

To describe the convection of this phase state we need to look at the mass transport (momentum $\rho\boldsymbol{u}$) in addition to the density. The circular nature of the convection rolls suggests looking at the curl of the momentum

$$\omega \equiv (\nabla \times \rho\boldsymbol{u})_z \tag{32}$$

where $\boldsymbol{u}$ is the convective velocity in the laboratory frame of reference. We call this the momentum vorticity in analogy to the vorticity $\boldsymbol{\Omega} \equiv \nabla \times \boldsymbol{u}$ in fluid dynamics.



(a) $F = 8.9$, $\Gamma = 36$ (b) $F = 12.7$, $\Gamma = 26$

(c) $F = 15.2$, $\Gamma = 14$ (d) $F = 15.2$, $\Gamma = 26$

Figure 16: Momentum vorticity distributions corresponding to figure 15 at $t = 0.50\,\text{s}$. ω is normalized for each image individually. Negative vorticity is represented in blue, positive values are red and green equates to vanishing vorticity.

Figure 16 shows the momentum vorticity distributions corresponding to the frames shown in figure 15. The convection rolls come in pairs and are distinctly visible. The momentum vorticity in the rest of the container vanishes for the most part. Still, strong vortexes do occur near the boundaries during most parts of each period. Examples are shown in figures 16a and 17.

To extract a classifier for the convection rolls, we therefore cut off the area near the top (10 of the 64 cells) and side (40 of the 256 cells) walls. Starting at $t = 0.50\,\text{s}$, 21% into each period we count the number of rolls. This is accomplished by finding the y-coordinate of the global momentum vorticity maximum at this point in time and examining the vorticity along this line for local extremal values. Each extremum that has an absolute value exceeding the predetermined critical value $\omega_{\text{crit}} = 0.8$ counts as a

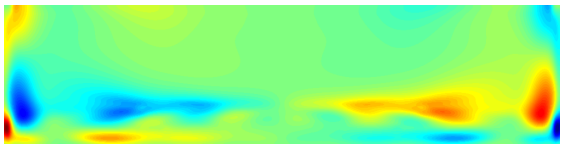


Figure 17: Normalized momentum vorticity ω at $t = 0.50\,\mathrm{s}$ for $F = 15.2$ and $\Gamma = 26$, exhibiting significant boundary effects.

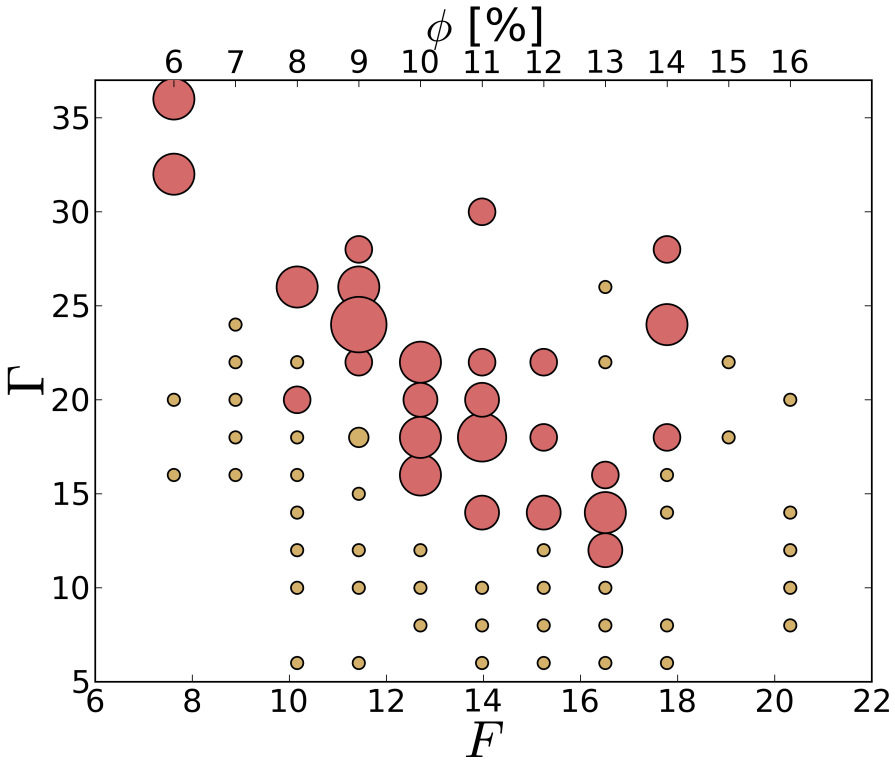


Figure 18: Phase space diagram for the number of convection rolls, counted automatically. The size of the red dots represents the count, small beige dots signal the absence of convection rolls.

convection roll. We found this to be the best classifier for identifying the convection roll phase algorithmically. Figure 18 shows the count across the F-Γ-phase diagram.

4.4 Additional Parameters

We mainly explored the F-Γ-phase diagram. Still, it is interesting to form an idea of how the observed phenomena change with regards to other parameters. As previously mentioned, there are a plethora of additional parameters available. The possible combinations of parameters scaling exponentially with their number makes a naive mapping of the higher dimensional phase space a very laborious task. Therefore, we merely looked at a few examples of easy-to-categorize variations. Specifically, we varied the friction, the length of the container in x-direction L_x and its associated aspect ratio L_x/F. The following simulations were performed at $F = 11.43$ and $\Gamma = 24$ where, in the normal setup, we both observed the Leidenfrost effect and convection rolls. Figure 19 was produced using the standard setup and serves as a reference.

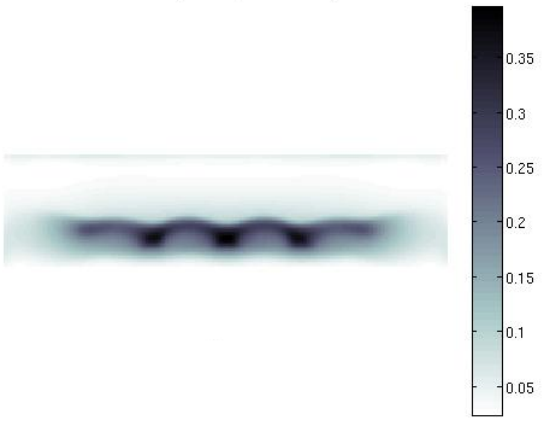
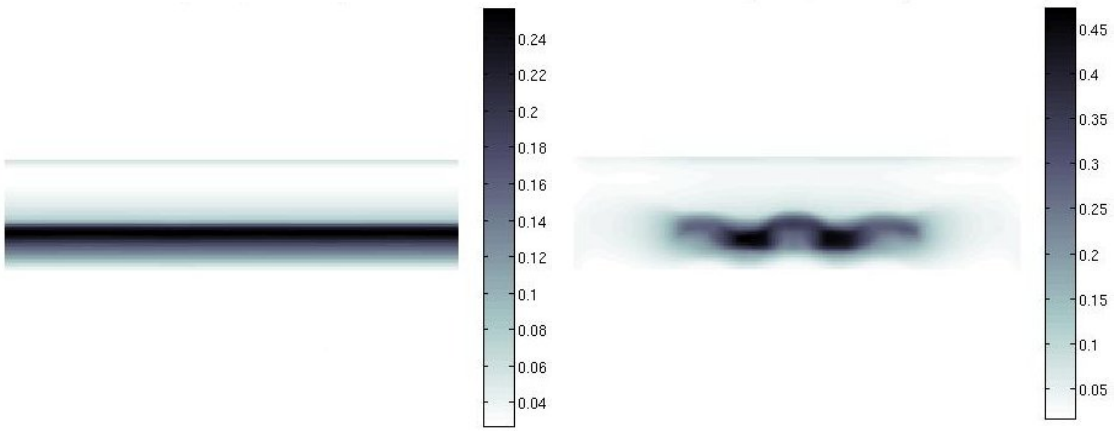


Figure 19: Density at $t = 0.61\,\mathrm{s}$ in the standard setup ($F = 11.43$, $\Gamma = 24$).

Deactivating the friction made the density $\rho(x, y, t)$ one-dimensional $\rho(y, t)$. Naturally, no convection rolls occurred, confirming that they are caused by the friction of the walls. Figure 20a gives a visual representation. The momentum vorticity did not homogenize completely, but it diminished by three orders of magnitude. Increasing the friction also had a visible impact on the convection rolls. In a comparable way to increasing the shaking acceleration Γ, it resulted in a larger gap between the cluster and the walls and higher local densities. The example in figure 20b shows that, as a consequence, the number of arches changed from 4 to 3. In summary, varying the friction produced similar result to varying the shaking acceleration Γ with regards to convection rolls (above a necessary threshold), but did not influence the occurrence of the Leidenfrost effect.



(a) No friction

(b) High friction

Figure 20: Density at $t = 0.61\,\mathrm{s}$ in simulations with varied friction ($F = 11.43$, $\Gamma = 24$).

Variation of the container length L_x has a significant impact on the formation as well as the final state of convection rolls. Since convection rolls are formed at the boundaries and develop inward, the time required for full formation scales with L_x. After the same amount of time a simulation with doubled container length $L_x = 922$ (figure 21, $N_x = 512$) shows much less progress than its $L_x = 461$ counterpart (figure 19).

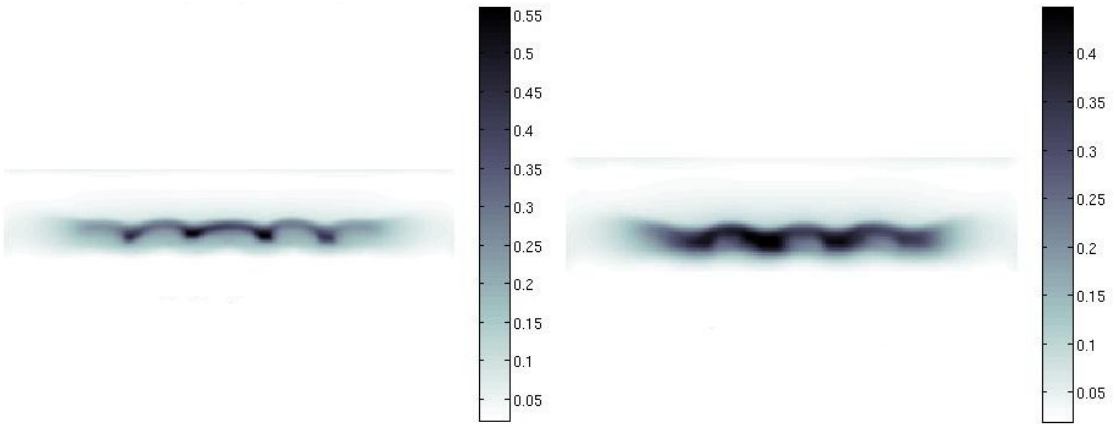
Depending on L_x the cluster also has varying area available. Figure 22a shows the

Figure 21: Density at $t = 0.61\,\mathrm{s}$. L_x increased by a factor of 2 ($F = 11.43$, $\Gamma = 24$).

effect scaling L_x by a factor of 1.25 has ($L_x = 576$, $N_x = 320$). With the increased space, the cluster developed an additional convection roll pair, maintaining the same arch-length as in the normal setup of about 70. Thus, increasing the container length by 1/4 increases the number of arches by the same amount. This is not exact, because the idle space near the boundaries needs to be subtracted, but it affirms the point. Inferring from the obvious analogy to a standing wave, we expect there to be certain container lengths (multiples of the wavelength) that lead to the most stable convection rolls. This is supported by our observations. Using $\Delta x = 1$ instead of $\Delta x = 0.9$ and, thus, scaling L_x by 10/9 we get a more aberrant shape (Figure 22b) that gets even more irregular with time (Figure 23b). The stretched systems in figures 23c and d also show very different development to the reference arrangement in 23a. They maintain axial symmetry, but the number of convection roll pairs is changed in both directions. Stretching by 1.25 led to a reduction from 3 to 2 pairs and stretching by 2 led to an increase to 8. We may only hypothesize about the reasons. While friction at the wall is necessary for the formation of convection rolls, it seems to inhibit their development near the boundaries. Thus, mainly wide systems lend themselves to the study of unperturbed convection rolls. The aspect ratio must, however, be maintained, as for even larger L_x we did not observe any convection rolls.

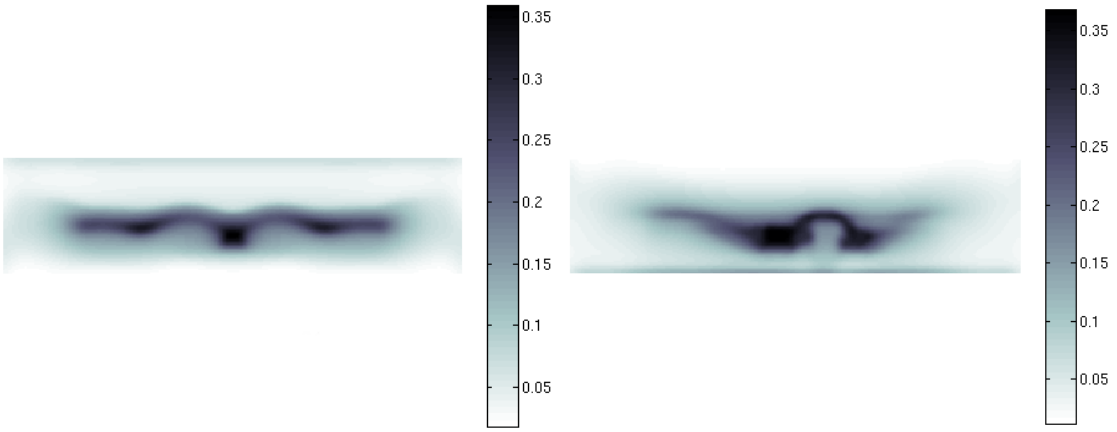
5 Analysis

To place this thesis in relation to previously published works we need to consider which parameters are relevant. The coefficient of restitution ε plays an important role in every model of the Leidenfrost effect [2, 3, 10] and convection rolls [1, 2, 29]. In all studies that

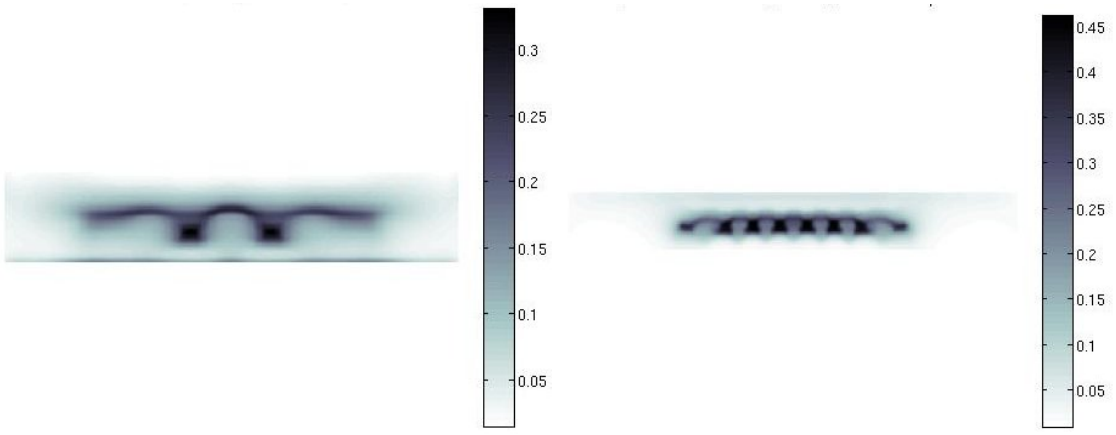


(a) 1.25 (b) 10/9

Figure 22: Density at $t = 0.61\,\mathrm{s}$ for varying container lengths L_x. The respective scaling factors are noted.



(a) 1 (reference) (b) 10/9 ($\Delta x = 1$)



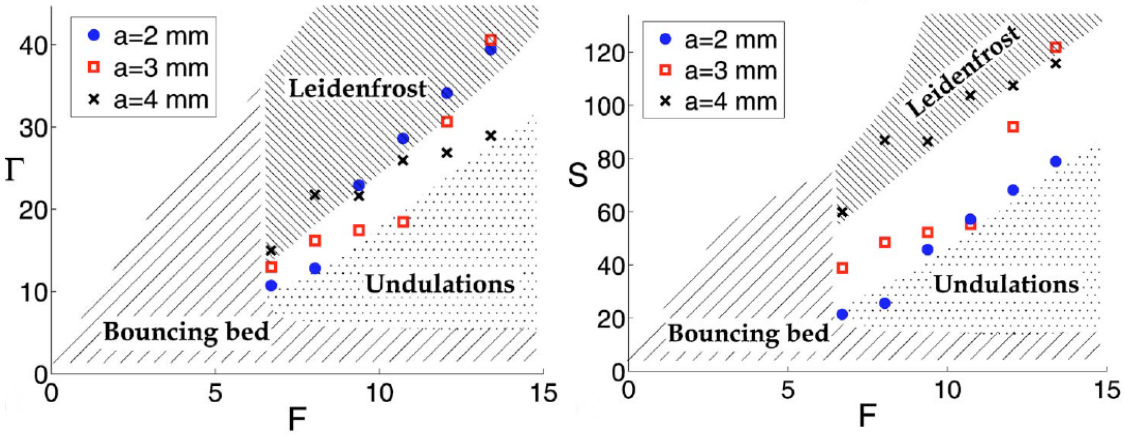
(c) 1.25 (d) 2

Figure 23: Density at $t = 1.44\,\mathrm{s}$ (a,b,c) or $t = 1.44\,\mathrm{s}$ (d) for varying container lengths L_x.

we are aware of it is within the range 0.9–0.95. For comparison glass has a coefficient of $\varepsilon_{\mathrm{glass}} = 0.905$ and the value for steel is $\varepsilon_{\mathrm{steel}} = 0.957$. The temperature dependency is always neglected, as even the difference between the extremes 0.9 and 0.95 is a factor

of less than 2 in the energy dissipation rate $(1-\varepsilon^2)$. For our DNS we chose a fixed $\varepsilon = 0.95$. Apart from F, Γ and S, Eshuis *et al.* concluded that that the aspect ratio L_x/F may play an important role in the emergence of nonequilibrium steady states [2]. We use a high aspect ratio of 23–60. This is an order of magnitude larger than early work ($L_x/F \approx 1$) [3, 10] and four times more than in newer research [2].

As mentioned in section 3.3, there is disagreement on the fundamental shaking parameter. This has been investigated by performing experiments with varying amplitudes and looking if the phase transition points coincide better in the F-Γ-space or the F-S-space [1, 2, 30]. In spite of simple hydrodynamic models such as the one presented in equations 25–27 being controlled by S, the experimental evidence suggest Γ as the better choice for control parameter [2]. Still, the data shows an amplitude-dependence (figure 24a), but in F-S-space there is even less alignment (figure 24b). This is an important consideration for the comparison since we used an amplitude higher by one order of magnitude than Eshuis *et al.* [1–3] and Wildman *et al.* [29]. We did not, however, control a independently[1]. The shortcomings of the model in equations 25–27 might be due to the much smaller respect it pays to the particle character of granular matter compared to ours.



(a) Γ-F plane (b) S-F plane

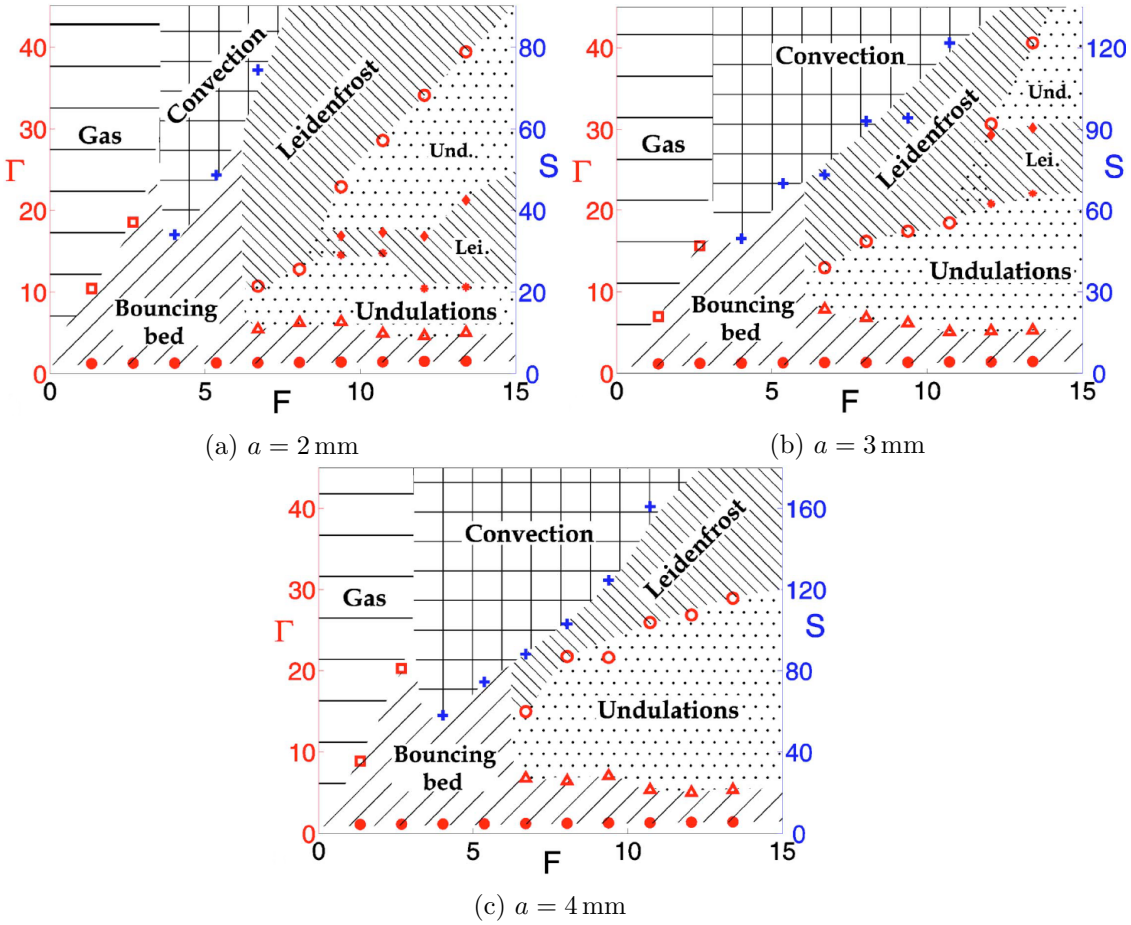
Figure 24: The transition from undulations to the granular Leidenfrost effect for increasing frequency f and fixed amplitude $a = 2.0, 3.0, 4.0\,\mathrm{mm}$. In Eshuis *et al.*'s experiments the Leidenfrost state always originates from the undulation phase [2].

Figure 25 shows the phase diagram that Eshuis *et al.* mapped in their studies. Comparing to our phase diagram in Fig. 12 we see a qualitative agreement in the Leidenfrost regime forming the upper half of the phase-diagram area. The onset value in our experiments does however show much less F-dependence. This can be explained by the much higher amplitude. With the increase of a in figure 25a to 25c the Leidenfrost regime extends further down replacing the undulation phase and possibly tending toward the same F-independent phase transition we encounter.

Eshuis *et al.* report a minimum number of particle layers $F \gtrsim 8$ to be required for the Leidenfrost phase to form [2, 3]. We observe the same property in our simulations.

[1]The adjustment of Γ and F leaves one degree of freedom between a/σ and g/σ.

The critical value cannot be determined exactly. This relates to the smoothness of the phase transitions. In experiments and MD simulations the transitions are already fairly gradual. In the past, the particle nature of the system had to be used explicitly to separate the gaseous phase (bottom layer) and the crystalline phase (top layer), for example using a pair correlation function to identify the periodicity of a crystalline layer [3]. In our hydrodynamic model all phase transitions are even smoother and such tools are not available. As Γ increases and the phase changes gradually from a bouncing bed to the Leidenfrost state, we have to introduce a classifier to distinguish the two because none have been necessary, previously. Quantitatively, the critical shaking acceleration of $\Gamma_{\text{crit}} \approx 19$ we found is in good agreement with the diagrams in Fig 25. Our findings also reinforce the dependency of the Leidenfrost effect on Γ rather than S, which is more in accord with Eshuis *et al.*'s later work which used smaller particles σ and higher aspect ratio L_x/F [2] compared to [3]. A trend which we continued.



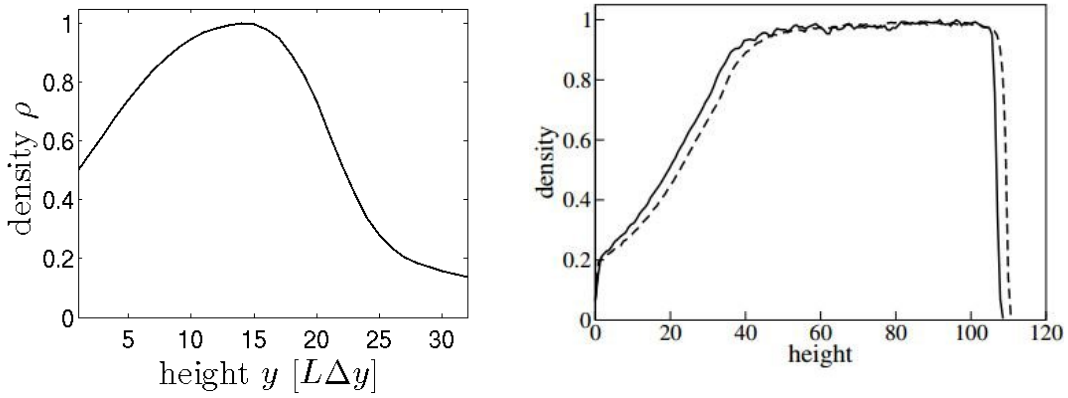
(a) $a = 2\,\text{mm}$ (b) $a = 3\,\text{mm}$

(c) $a = 4\,\text{mm}$

Figure 25: Diagram of the phases Eshuis *et al.* explore in [2] at three fixed values of the shaking amplitude a.

The density profiles our systems show in the Leidenfrost phase look similar to ones produced in MD simulations by Meerson *et al.* [10], as illustrated in figure 26. The main difference is in the density decline at the top of the cluster. We observe a gradual slope rather than an abrupt drop in density. The same steep transition is seen in the theoretical density profile by Eshuis *et al.* (figure 5c) [3]. Another notable disparity

between the images in figure 26 is the inflection in the density profile below the cluster in 26b. It appears neither in our simulations nor in those by Eshuis *et al.* [3]. Judging from these pictures, our density profile is in the better agreement with the experiment in figure 5a/b than both the theoretical model of equations 25 to 27 and the MD simulations by [10].



(a) Density average over time in the whole system at $F = 17.8$, $\Gamma = 20$.

(b) Two density profiles of MD simulations by Meerson *et al.* in the Leidenfrost regime [10].

Figure 26: Leidenfrost density profile comparison. The graphs on the right are cropped, ensuring that the height-axis intercept is at the highest point that the container reaches. The upper container volume is also cut off.

We have not yet discussed where the Leidenfrost phase gives way to convection rolls. Figure 27 combines our two phase space diagrams. The convective regime has moved to the lower right compared to the diagram by Eshuis *et al.*, extending below the Leidenfrost phase and allowing bouncing convection rolls to form (red). Up to now this has only been observed in systems below the supposed critical number of layers $F < 8$. Additionally, the convection rolls appear at $\Gamma \geq 12$ which is lower than what has been previously found. Looking at the diagrams in figure 25, the onset of convection rolls happens for lower Γ as a increases. We conclude that the reason for the different observations is the high driving amplitude used in our simulations.

The blue marked area at low F in our phase diagram is not found in figure 25. In actuality, though, it is convective. The patterns we see (figure 28) are simply not expressed strongly enough to be captured by our classifying algorithm.

Extensive research has been done on the convection of granular matter experimentally, numerically and theoretically [2]. These studies almost exclusively deal with mild fluidization ($\Gamma \leq 10$) and consequently nearly constant density in the system. Their convection is accepted to be boundary driven [2].

The convection roll pattern we observe closely resembles the ones produced by Eshuis and Paolotti [2, 28], which they classify as buoyancy driven. They show the same density gradients and arches (compare figure 13 to 6 or 7).

Paolotti *et al.* obtained their results numerically with elastic (dissipation-less) walls. Therefore, the convection rolls in their systems cannot be boundary driven. As detailed in section 4.4, in our systems dissipative walls are required for convection rolls to form.

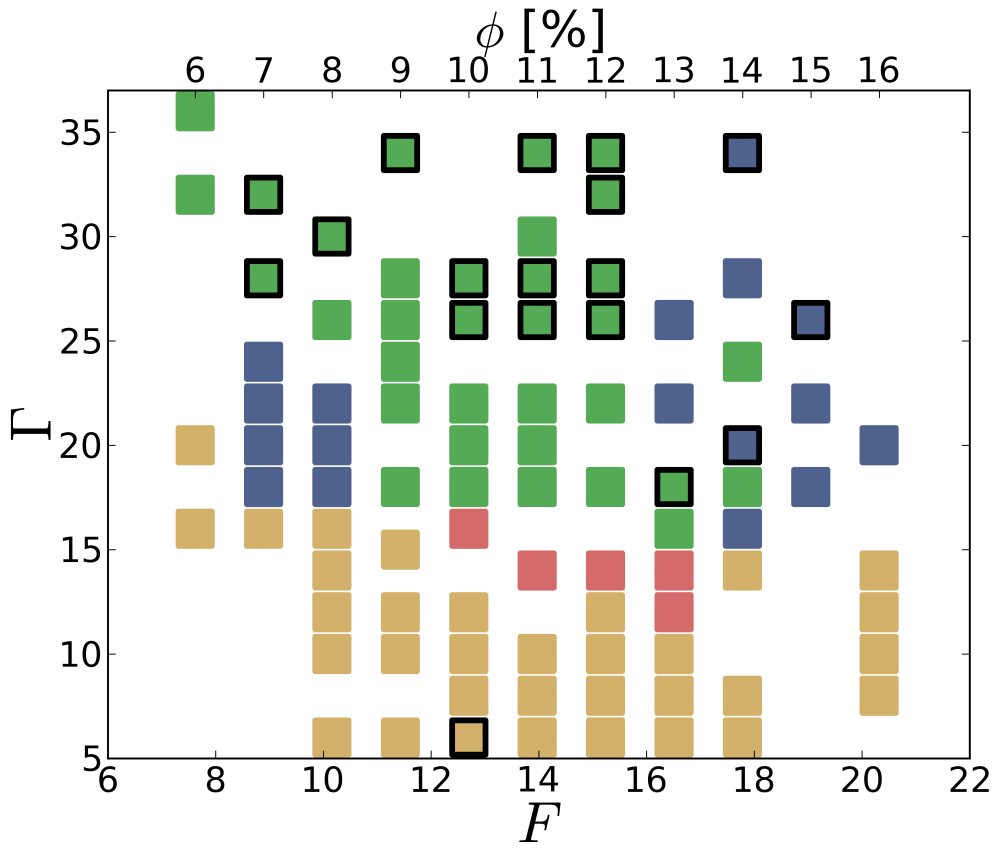
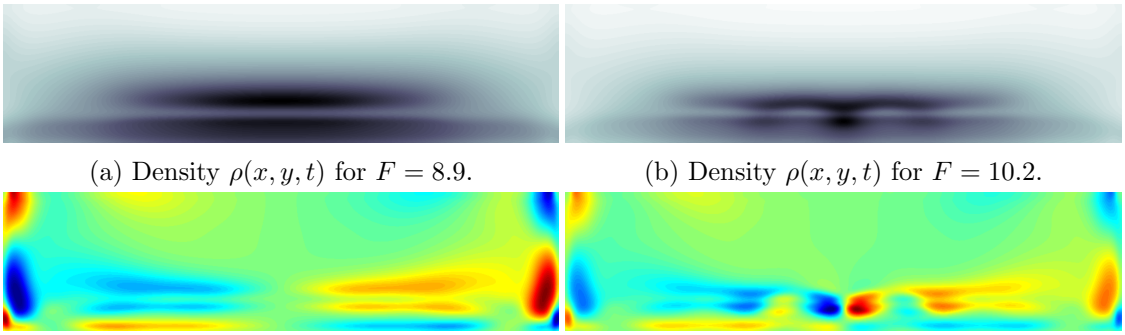


Figure 27: Complete phase space diagram. The bouncing bed is colored beige, the Leidenfrost phase is blue and convection rolls are marked green when they emerge from the Leidenfrost phase. Convection rolls that develop from a bouncing bed are shown in red. Black outlines mark points with insufficient data for automatic detection of convection rolls. Their affiliation was determined visually from $\rho(x, y, t)$.



(a) Density $\rho(x, y, t)$ for $F = 8.9$.

(b) Density $\rho(x, y, t)$ for $F = 10.2$.

(c) Momentum vorticity $\omega(x, y, t)$ for $F = 8.9$.

(d) Momentum vorticity $\omega(x, y, t)$ for $F = 10.2$.

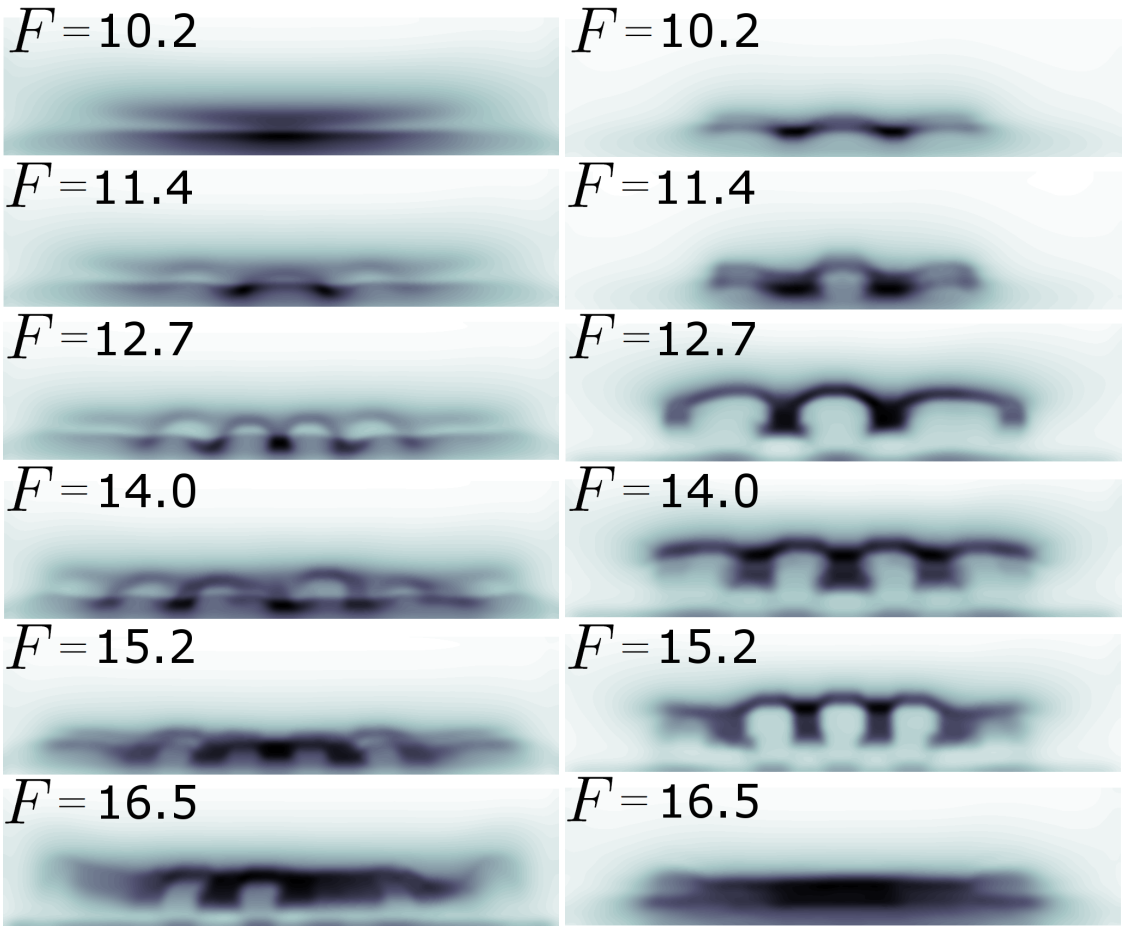
Figure 28: Fringe cases of convection at $\Gamma = 20$.

Since our high amplitude is less likely to induce small scale motion than a low amplitude and our shaking acceleration Γ did not exceed 36, we may classify our systems as mildly fluidized and the convection rolls to be boundary driven. The former point is further facilitated by the fact that we did not find granular gas phases in the systems that we explored with $\Gamma < 8$.

Normally, convection rolls hug the walls of the container [2, 28]. In our simulations this is not the case because of the relatively high friction the walls exhibit. Still, like

Eshuis *et al.*, and unlike Paolotti *et al.*, we always encounter downward motion at the walls, confirming that this is a consequence of dissipative walls. In section 4.4 we demonstrated that the dissipative properties of the wall play a significant role in the formation of nonequilibrium steady states. This is to be expected, but has not been addressed in previous research. We saw that, while friction at the wall is necessary for the formation of our convection rolls, it seems to inhibit their development near the boundaries. Thus, mainly systems with a high aspect ratio lend themselves to the study of unperturbed convection rolls. We therefore agree with Eshuis *et al.* that the aspect ratio is an important control parameter [2], at least regarding convection rolls.

Eshuis *et al.* report that more dissipation, i.e. more particle layers F or higher shaking acceleration Γ, leads to bigger clusters, i.e. rolls. The result is that less convection rolls fit inside the system [2]. In our results we do see the number of convection rolls decreasing as Γ grows. But increasing the number of particle layers F as the opposite effect. Figure 29 shows this with some examples of convection roll patterns we observe in our simulations.



(a) $\Gamma = 18$ (b) $\Gamma = 26$

Figure 29: Increasing number of layers F at fixed shaking acceleration Γ.

We observe the Leidenfrost phase arising from a bouncing bed. This is contrary to some findings by Sano and Eshuis [2, 26]. Typically granular matter passes through an intermediate state of undulations. Its absence in our phase diagram may have to do

with the high aspect ratio we employ. But since a standing wave pattern is generally not expected to become unstable through a mere quadrupling of its available length, we deem this possibility unlikely. Eshuis *et al.* only found undulations experimentally in their 3D setup with $L_z = 5\,\sigma$ [2] not in their $L_z \approx 1\,\sigma$-systems [3] and conjectured that 3 dimensions may be necessary for undulations to form. In their MD simulations Sano did, however, find undulation patterns [26]. The most likely reason for the lack of undulations in our systems is the hydrodynamic approach. The corrections we use for the Navier-Stokes equations to capture the particle nature of granular matter seem to break down at this point. The key step in the development of undulations appears to be that whenever the bottom layer of the particle bed collides with the container, it dilates [26]. This dilation seems not to be captured by our formulation. An adjustment of equation of state (Eq. 7), possibly in the form of a different pair correlation function might lead to undulations, but that investigation is beyond the scope of this thesis.

Lastly, the bouncing bed we observe is not exactly the same that is observed in experiments and MD simulations, i.e. one single cluster moving uniformly [2]. In our simulations the cluster does not fully bounce up and down. Some of the energy transferred by the bottom is absorbed by the inner collisions of the cluster. This leaves a fraction of the density remaining on the container floor.

6 Discussion and Conclusions

6.1 Data Gathering

As the discovery of convection rolls and the Leidenfrost effect within our model took a lot of time, terabytes of data with varying usability were accumulated. Variation of the secondary parameters such as frequency, box dimensions and transient time were not included in this work barring the ones discussed in section 4.4. For the most part they did not lead to interesting observations. In parallel to our understanding of the phase space the method of data storage improved. Initially, every N-th step the state of the three main fields was saved, because the DNS was originally built for observing the cooling process of granular matter which happens on an exponential time scale (equation 10). This constant lapse N led to inhomogeneous time steps in the output data, containing unnecessary, but lacking important information. In the updated method a constant number of steps per period was extracted. Although this number evolved between simulations it provided a much more consistent picture. Still, a lot of the produced data has the former structure. A secondary MATLAB script (*analyzeEXP.m*) had to be created replicating the capabilities of the main one (*analyze1.m*) to accommodate the old structure and extract most of the relevant information. This may lead to discrepancies between data sets and the simulations they represent.

We did not track the initial CFL number (equation 18). Since it regulates the time steps, it might affect early development and ultimately have longer lasting effects. It likely does not have any notable effect, but the influence is hard to assess.

6.2 Issues of Computational Complexity

The simulations are computationally very expensive. The best way to cut down on the complexity is to decrease the resolution. We did this repeatedly to end up with the current compromise. The used resolution appears to very much suffice in the case of the Leidenfrost phase. As discussed in section 4.4, the same results could probably be reproduced in a 1D system. In contrast, the structures formed by convection are very delicate. One convection roll takes up about 0.5% of the container. With a resolution of 256×64 this leaves a roll diameter of about 10 computational cells, even lower in extreme cases. The simulations still produce adequate results, but for smaller, more densely distributed rolls, such as in figure 29a, the resolution limits the simulation.

The simulations lose stability as F and Γ grow. Thus, for many parameters the simulations become unfeasibly slow as the forces get turned on or even crash after some time. It typically starts to happen around $F > 12$ or $\Gamma > 32$. This is most problematic in the long term observation of convection rolls. Figure 30 illustrates the issue. The histogram shows that many simulations could not be run for the desired duration of more than 2 seconds. To make the analysis more balanced we decided to cut off all data after 2.5 s.

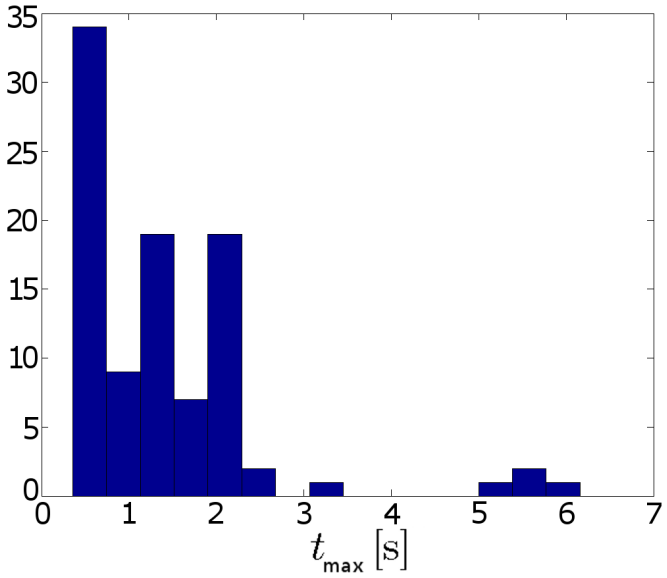
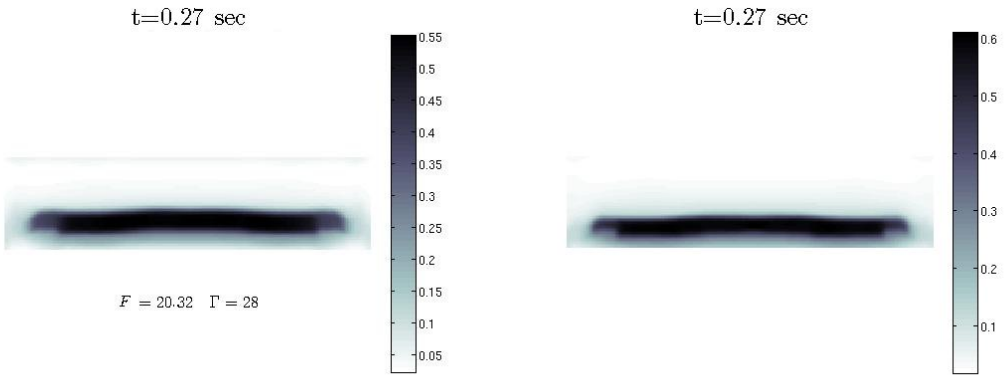


Figure 30: Histogram of the final time reached in each of our simulations.

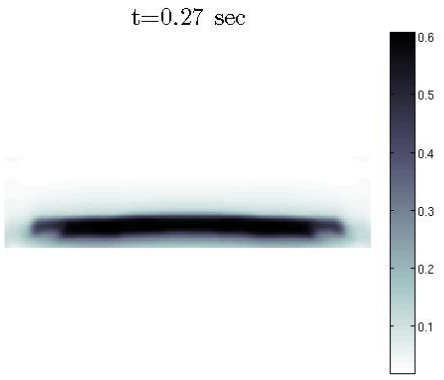
After trying multiple solving schemes, such as ADER, dGRP-flux and an implicit solver it became apparent that they increased the time required for computation, but had little influence on the results. Figure 31 shows the the schemes and a reference in the top left (31a) for comparison.

Using the ADER-scheme led to slightly earlier development of the convection rolls. The implicit solver did the same and, additionally, produced more refined structures. We observed no change from applying the dGRP-flux. Each of the solving schemes about doubles the computation time per time step. Ultimately, we deemed the trade-off not worthwhile and decided against the employment of any of the three.

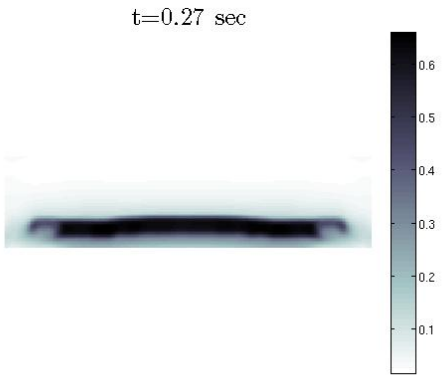


(a) Standard solver

(b) ADER scheme



(c) dGRP-flux



(d) Implicit solver

Figure 31: Snapshots of four simulations with the same parameters using different solvers. The pictures were all taken at $t = 0.27\,\mathrm{s}$ (the transient time is $0.25\,\mathrm{s}$).

6.3 Hydrodynamic Model

Our model proved adequate in reproducing the Leidenfrost effect and convection rolls. This is reflected in the Knudsen number which stayed in the range from 0.021 ($F = 20.32$) to 0.057 ($F = 7.62$) and satisfied the imposed condition $\mathcal{K} < 0.1$.

Our simulations are two-dimensional, do not account for air and cannot handle high local densities or shock waves. Therefore, many phenomena that have previously been observed in experiments, such as oscillations, stripes, spirals [25], Faraday heaping and surface waves [24], could not be reproduced.

Possible adjustments include the implementation of a tangential coefficient of restitution. This may increase computational effort substantially. Its importance wholly depends on the granular material that is studied. The effects of such an adjustment are likely better studied within discrete systems. Another potential adjustment was discussed in section 5 and pertains to the pressure term in equation 7. Corrections that lead to a dilation of particle layers being launched against the boundaries might allow for the reproduction of the undulation phase within our model.

6.4 Stationarity

In our observations the Leidenfrost phase becomes stationary at around 0.38 s, but for some parameter configurations the simulation time diverged quickly. Thus some data sets did not have much time depth. Since usually the systems are very close to the stationary state from 0.28 s to 0.38 s, we start taking measurements at 0.30 s, treating all systems equally. Figure 32 shows how little change happens during this time.

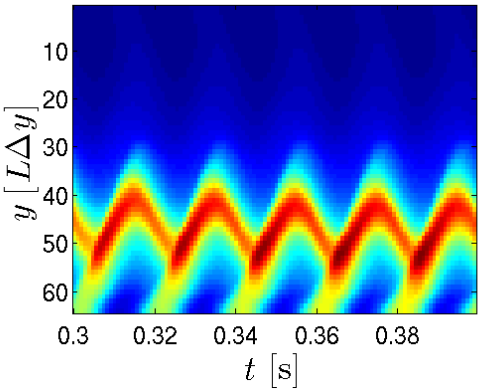


Figure 32: $\bar{\rho}(y,t)$ ($F =$, $\Gamma = 16$).

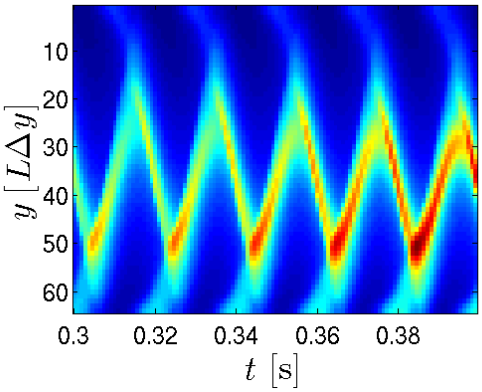
This approach only became problematic in dealing with very high Γ and low F. Systems in this area of phase space took more time before arriving in a stationary state. Specifically, the configurations shown in figure 33 became an issue because they did not conform to the phase separating condition $\kappa > 0.5$ for the Leidenfrost effect. As figure 33 alludes to, this is explicitly due to the deficient data and not to a failure of the classifying condition.

The issue of stationarity is more prominent in the consideration of convection rolls. The temperature curves show quick convergence toward a periodic state. An example is presented in figure 34. The observations in 4.3, however, prove this to be an inadequate basis for judgment. The convection rolls develop an outward movement, forming new ones as the available space allows for. The process starts after the 0.5 s that the energy trajectories suggest.

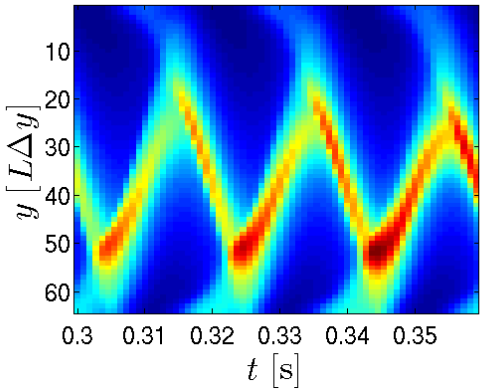
Taking the volume average of the squared momentum vorticity $\langle\omega^2\rangle_{x,y}$ (excluding the boundary area as previously), one arrives at a usable measure of stationarity. It indicates that most systems reach a steady state between $t = 0.6$ s and 1 s, which agrees with our visual observations. Figure 35 shows some examplesof the temporal evolution of $\langle\omega^2\rangle_{x,y}$. As pointed out in section 6.2 (figure 30), not all data sets extended far enough into this time interval. This derogates the accuracy of our results pertaining to convection rolls in part.

6.5 Conclusions

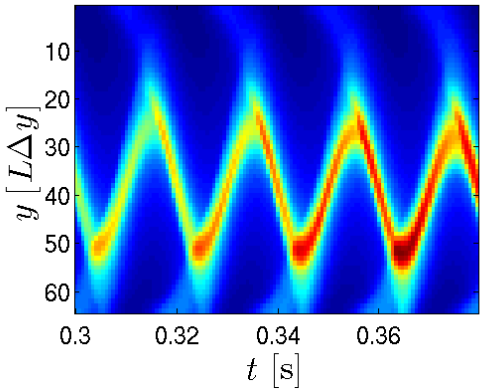
We have reproduced convection rolls and the granular Leidenfrost effect in a hydrodynamic DNS. In this continuum model the phase transitions are less clear than in the



(a) $F = 10.2$, $\Gamma = 30$



(b) $F = 8.9$, $\Gamma = 32$



(c) $F = 11.41$:, $\Gamma = 34$

Figure 33: All systems that did not reach stationary states within simulated time. The data stops before $t = 0.42\,\mathrm{s}$. The plots show $\bar{\rho}(y, t)$ for all fully recorded periods starting at $t = 0.30\,\mathrm{s}$.

study of particles. Therefore, it was necessary to determine classifiers. We automated the analysis of the DNS-output in MATLAB, assigning phase states and extracting additional information such as density profiles. This gave rise to a phase diagram. Taking into account the high amplitude we deal with, it is in good agreement with previous research. The most remarkable discrepancies are the lack of undulations and the increase in convection rolls with with the number of particle layers F in our simulations. All changes and transitions becoming more gradual is also of note. We confirmed that driven granular systems are not exhaustively described by the coefficient of restitution ε, the shaking acceleration Γ and the number of particle layers F. The aspect ratio L_x/F and the amplitude a are independently important. The detailed descriptions we achieved for the Leidenfrost effect and convection rolls match experiments very well and we contributed to the quantitative understanding of the nonequilibrium phase states.

7 Outlook

The process is now automated, so an extension of the phase diagram is just a matter of computational time. Tracking only the Leidenfrost phase is especially easy since we now

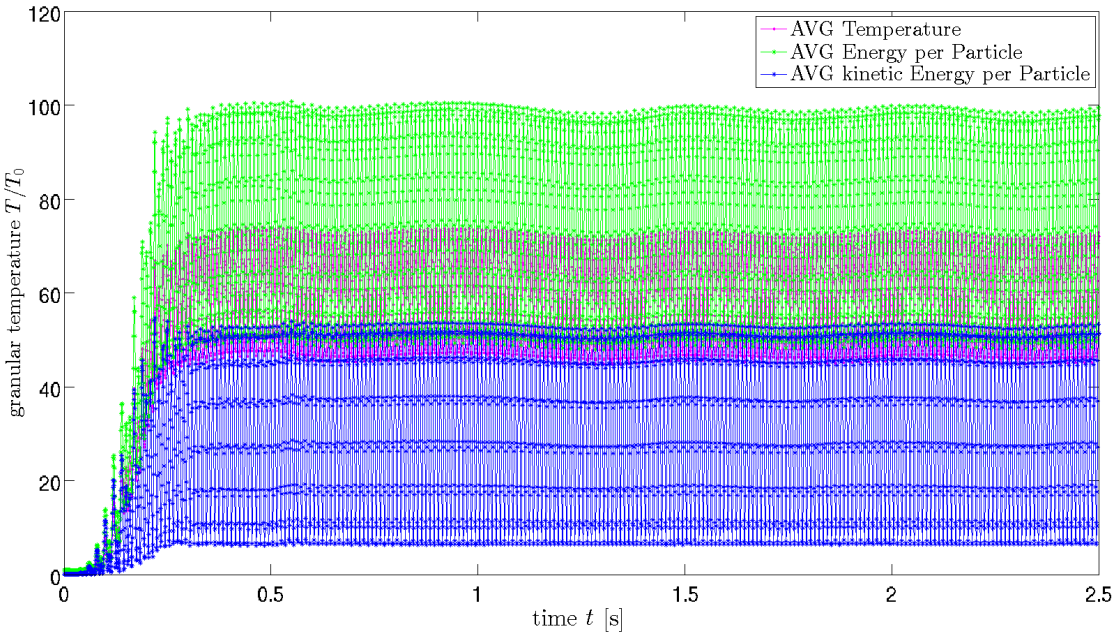


Figure 34: Average temperature, average energy per particle and average kinetic energy per particle as a function of time ($F = 11.43$, $\Gamma = 24$).

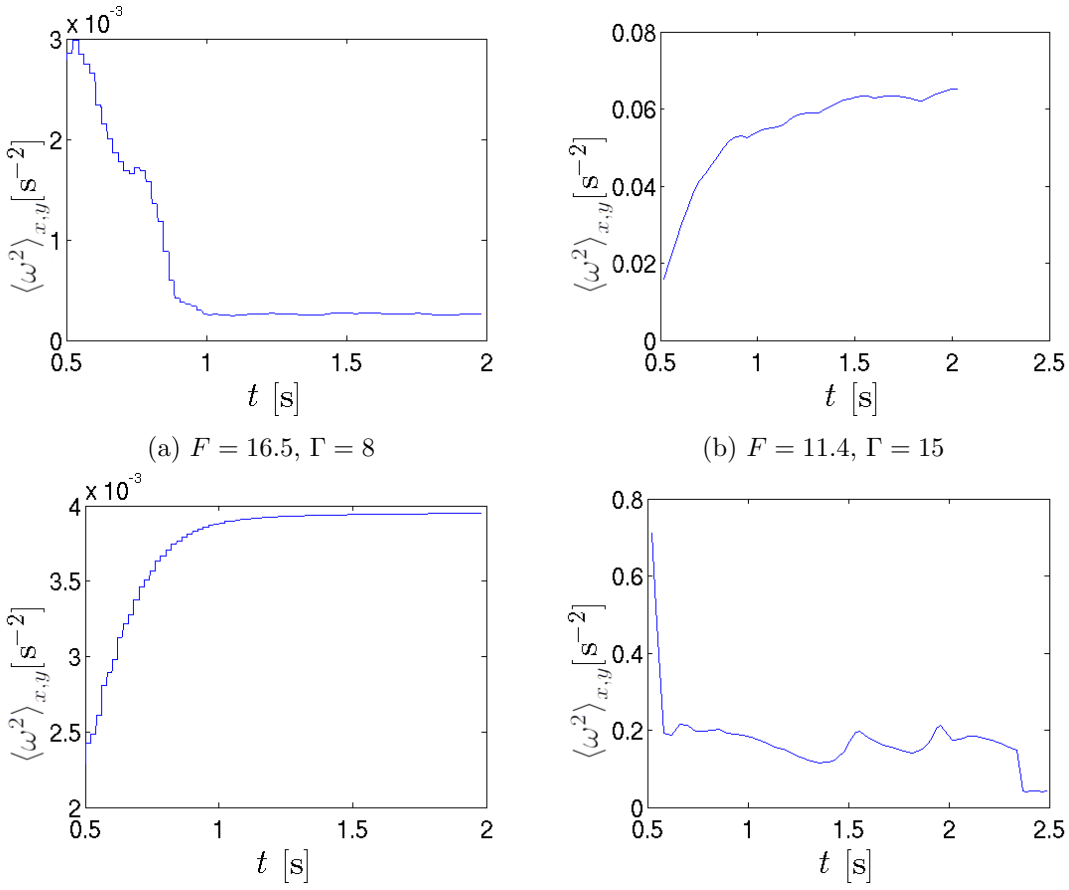
know that we may turn off friction and reduce L_x which significantly boosts performance. The most extensively studied part of phase space, corresponding to the transitions between, bouncing bed, Leidenfrost effect and convection rolls has been mapped in this thesis, but the amplitude-dependence that became apparent inspires a mapping in this third phase space direction. In its current setup the DNS is mainly bounded in F. High densities are hard to compute and will eventually lead to a breakdown of the hydrodynamic model. On the other hand, an increase in Γ beyond what is shown in the phase diagram of Fig. 27 is less problematic.

The impact that the aspect ratio L_x/F has on convection roll patterns has also not been studied rigorously in current research. This DNS provides a possible platform.

Our results also raise the question: What does the spatial transition in the Leidenfrost phase look like, exactly? In what way does the top layer exhibit crystalline characteristics and the bottom one gaseous properties? These questions have been addressed for particle systems [3], but not in a hydrodynamic context.

In sections 4.3 and 5 we noted that mainly systems with a high aspect ratio lend themselves to the study of unperturbed convection rolls. In case of additional computational investigation, a good compromise needs to be found, since observed systems with doubled $N_x = 512$ all crashed before $t = 1.2\,\mathrm{s}$.

Further, the transition to a granular gas has not been discussed in this thesis. Mathias Hummel's work [16] focuses on this, but mostly considers systems without exterior forces. An exploration in this direction is certainly easy for the DNS to deal with.



(a) $F = 16.5$, $\Gamma = 8$

(b) $F = 11.4$, $\Gamma = 15$

(c) $F = 8.9$, $\Gamma = 18$

(d) $F = 11.4$, $\Gamma = 24$

Figure 35: Volume-averaged squared momentum vorticity over time. Typically, between 0.7 and 1 s a stable value is reached. The top and sides of the container are excluded from the average and the graph is coarse grained to use the maximal value each shaking period.

A Matlab Code

The accompanying CD contains the MATLAB code (including *analyzeEXP.m*) and videos of the Leidenfrost effect and convection rolls from our simulations.

analyze1.m:

```
addpath('/home/jtabet/plots/mfunctions');

doCR = 1;
doL = 1;
doQ = 1;
doCount = 1;
doNrolls = 1;
pics = 1;

cutofftime = 2.5;

cstartingtime = 0.5;
lstartingtime = 0.3;
cstep = 1;%0 =^= spp
```

```
lstep = 1;

%analysis boundaries VORTICITY
sbw = 40/256;%cut on each side
tbw = 10/64;%cut top
%analysis boundaries DENSITY
sbd = 73/256;%cut on each side

tic

dataallocator;


for fld = 1:length(listdata)

folderName = char(listdata(fld));

include_params;

initialization;

%Convection rolls:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if doCR==1 && dC(fld)==1
    step = cstart;
    time(step) = 0;
    while( step <= length(listing) )

        loadgrids;

        %Vorticity: bottom at line 1

        momentumY_plane = density_plane(:,:,1).*(momentumY_plane./
            density_plane(:,:,1))...
        + FORCE_MULTIPLAYER*cos(2*3.14159*50*time(step))/(2*3.14159*50);

        cau = curl(momentumX_plane' , momentumY_plane' );

        val=0;
        siz=0;
        for i = 1:round(Ny-tbw*Ny)
            for k = round(sbw*Nx):round((1-sbw)*Nx)
                val = val + cau(i,k)^2;
                siz = siz + 1;
            end
        end
        wsq = [wsq val/siz];

        [FX, FY] = gradient(cau(1:Ny-tbw*Ny,round(sbw*Nx):round((1-sbw)*Nx)
            ));
        gw = [gw mean(FX.^2+FY.^2)];

```

```
65          if mod(step,spp)==countrolltimer && doCount==1
66              %idex: y-coord of max in column
67              [wmax, idex] = max(cau(1:Ny/2,:));
68              %idex2: x-coord of global max; idex(idex2): y-coord of global
                    max
69              [max2, idex2] = max(wmax);
70              if (max2>=maxvort)
71                  maxvort=max2;
72              end
73
74              nrolls=0;
75              dw=0;
76              for i = round(sbw*Nx):round(Nx*(1-sbw))
77                  if ( abs(cau(idex(idex2),i)) >= abs(cau(idex(idex2),i-1)) )
78                      dw=1;
79                  elseif( dw==1 && ( abs(cau(idex(idex2),i-1)) >= 0.8 ) )
80                      nrolls=nrolls+1;
81                      dw=-1;
82                  else
83                      dw=-1;
84                  end
85              end
86              arolls = [arolls nrolls];
87          end
88
89          step = step + cstep;
90      end
91      ylabel('$\omega^2$ ','Fontsize',20);
92      xlabel('$t$ [s]','Fontsize',20);
93  end
94  %Leidenfrost:
95  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96  if doL==1 && dL(fld)==1
97      step = lstart;
98      time(step) = 0;
99      while( step <= length(listing) )
100
101         loadgrids;
102
103         %Density: bottom at line 1
104
105         if (Ny==64)
106             %old orientation:
107             curdenp = FILLING_FRACTION*mean(density_plane(round(sbd*Nx):
                    round((1-sbd)*Nx),:,1));
108         else
109             curdenp=NaN(1,64);%flip for old orientation
110             k=1;
111             for i = 1:64
112                 while (k<i/64*Ny)
113                     k=k+1;
114                 end
```

```
115                 %old orientation:
116                 curdenp(i)=FILLING_FRACTION*mean(density_plane(round(sbd*Nx
                        ):round((1-sbd)*Nx),k,1));
117             end
118         end
119
120         denp((step-lstart)/lstep+1,:) = curdenp;
121
122         [ymax((step-lstart)/lstep+1), ind((step-lstart)/lstep+1)] = max(
                curdenp);
123
124         step = step + lstep;
125     end
126 end
127 %Quiver image output:
128 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
129 if doQ==1
130     if ((length(listing))>spp*25)
131         step = spp*25;
132     else
133         step = floor((length(listing)-1)/spp)*spp;
134     end
135
136     loadgrids;
137
138     momentumY_plane = density_plane(:,:,1).*(momentumY_plane./density_plane
            (:,:,1)) + FORCE_MULTIPLAYER*cos(2*3.14159*50*time(step))
            /(2*3.14159*50);
139
140     fileName_quiv = sprintf('%2.0f-%2.0fq.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
141
142     [ixqu, iyqu] = meshgrid(ix(1:Nx/16:Nx),iy(1:round(Ny/11):Ny));%32x13
143
144     figure(1);
145     set(1,'Color',[1 1 1]);
146
147     u = interp2(ixm,iym, momentumX_plane', ixqu, iyqu,'spline');
148     v = interp2(ixm,iym, momentumY_plane', ixqu, iyqu,'spline');
149
150     colormap('default');
151     cbar = colorbar('east');
152
153     %u=u/sqrt(mean(mean(u.^2)))*sqrt(mean(mean(v.^2)));
154
155     quiver(ixqu, iyqu, u, v, 0.6, 'Color', 'black');
156     title(sprintf('t = %3.2f sec', time(step)), 'FontSize', 20);
157     axis equal;
158     axis off;
159     print(strcat('/home/jtabet/plots/autopics/',fileName_quiv),'-dpng')
160 end
161 %count rolls:
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if doNrolls && dC(fld)==1
    if ((length(listing))>spp*35)%.21
        step = spp*35;%+round(0.21*spp);
    else
        step = floor((length(listing)-1)/spp-0.22)*spp+round(0.21*spp);
    end

    loadgrids;

    %Vorticity: bottom at line 1

    momentumY_plane = density_plane(:,:,1).*(momentumY_plane./density_plane
        (:,:,1)) + FORCE_MULTIPLAYER*cos(2*3.14159*50*time(step))
        /(2*3.14159*50);

    cau = curl(momentumX_plane' , momentumY_plane');

    cau = cau(1:Ny/2,round(sbw*Nx)-1:round(Nx*(1-sbw)));
    [wmax, idex] = max(cau(:,:));%idex: y-coord of max in column
    [max2, idex2] = max(wmax);%idex2: x-coord of global max; idex(idex2): y
        -coord of global max
    nrolls=0;
    dw=0;
    rollcoords=[];
    for i = 2:length(cau)
        if ( abs(cau(idex(idex2),i)) >= abs(cau(idex(idex2),i-1)) )
            dw=1;
        elseif( dw==1 && ( abs(cau(idex(idex2),i-1)) >= 0.8 ) )%max2/2.7
            nrolls=nrolls+1;
            rollcoords=[rollcoords i];%x-coords; y-coord=idex(idex2);
            dw=-1;
        else
            dw=-1;
        end
    end
    if (Nx~=256)
        rollcoords = rollcoords*256/Nx;
    end
    rolldiff = diff(rollcoords);
end
%Output:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if doCR==1 && dC(fld)==1 && pics==1
    %wsq:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    fileName_wsq = sprintf('%02.0f-%02.0fwsq.png', FILLING_FRACTION*100,
        FORCE_MULTIPLAYER);
    figure(5);
    periodend = fix(length(listing)/spp)*spp;
    goback = min([fix((periodend-cstart)/spp), 4]);
    plot( time(periodend-goback*spp:periodend), wsq(periodend-goback*spp-
```

```
        cstart:periodend-cstart) )
210     %title(sprintf('t = %3.2f...%3.2f sec', finaltime-0.1,finaltime),'
            FontSize',20);
211     set(5,'Color',[1 1 1]);
212     ylabel('$\omega^{2}$ [s$^{-2}$]','Fontsize',28);
213     xlabel('$t$ [s]','Fontsize',28);
214     %print(strcat('/home/jtabet/plots/wsqpics/',fileName_wsq),'-dpng')
215     set(gca,'FontSize', 20)
216     prnt = getframe(gcf);
217     imwrite (prnt.cdata,strcat('/home/jtabet/plots/wsqpics/',fileName_wsq))
218     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
219     fileName_gw = sprintf('%02.0f-%02.0fgw.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
220     figure(5);
221     periodend = fix(length(listing)/spp)*spp;
222     goback = min([fix((periodend-cstart)/spp), 4]);
223     plot( time(periodend-goback*spp:periodend), gw(periodend-goback*spp-
            cstart:periodend-cstart) )
224     set(5,'Color',[1 1 1]);
225     ylabel('$\omega^{2}$ [s$^{-2}$]','Fontsize',28);
226     xlabel('$t$ [s]','Fontsize',28);
227     set(gca,'FontSize', 20)
228     prnt = getframe(gcf);
229     imwrite (prnt.cdata,strcat('/home/jtabet/plots/wsqpics/',fileName_gw))
230     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
231     figure(6);
232     fileName_wm = sprintf('%02.0f-%02.0fwm.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
233     wm = NaN(1,length(wsq)-spp);
234     for i = 1:length(wsq)-spp
235         wm(i) = max(wsq(i:i+spp));
236     end
237     plot( time(cstart:cstep:(cstart+(length(wm)-1)*cstep)), wm )
238     set(6,'Color',[1 1 1]);
239     ylabel('$\omega^{2}$ [s$^{-2}$]','Fontsize',28);
240     xlabel('$t$ [s]','Fontsize',28);
241     set(gca,'FontSize', 20)
242     prnt = getframe(gcf);
243     imwrite (prnt.cdata,strcat('/home/jtabet/plots/wsqpics/',fileName_wm))
244     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
245
246 end
247
248 result_output;
249
250 fclose('all');
251 end
252 timeneeded = toc;
253 toc
254 %Doublings:
255 for i = 1:length(listdata)
256     for k = 1:i
```

```
257             if i~=k && p_phi(i)==p_phi(k) && p_Gamma(i)==p_Gamma(k)
258                 fprintf(1,'Doubling at %4.2f, %2.0f\t%.0f and %.0f\n',...
259                 p_phi(i), p_Gamma(i),k,i);
260             end
261         end
262 end
263 %Durations:
264 if size(finaltv(~isnan(finaltv)))==size(finaltv)
265     figure(4);
266     set(4,'Color',[1 1 1]);
267     hist(finaltv,15);
268     fileName_t = sprintf('DurationsHistogram.png');
269     print(strcat('/home/jtabet/plots/',fileName_t),'-dpng')
270 end
271
272 %ExpData:
273 if (strcmp(folderName,char(listdata(end))) && timeneeded > 50 )
274     analyzeEXP;
275 end
```

result_output.m:

```
1  %density profiles:
2  if doL==1 && dL(fld)==1
3      ind = ind(~isnan(ind));%y location of max
4      ymax = ymax(~isnan(ymax));%magnitude of max
5      if spp ~= 0
6          den0avg(fld,:) = mean(denp((1+floor(spp/4)):spp:end,:));
7      else
8          den0avg(fld,:) = mean(denp(steppicker/lstep-lstart+1,:));
9      end
10     if spp ~= 0
11         denp = denp(1:fix(length(denp)/spp)*spp/lstep,:);
12         ymax = ymax(1:fix(length(denp)/spp)*spp/lstep);
13         ymdot = diff(ind)./diff(time(lstart:lstep:(lstart-1+lstep*length(
               ind))));
14     else
15         [~, finalstep] = min(abs(time-(fix(finaltime/T)*T)));
16         denp = denp(1:(finalstep-lstart)/lstep,:);
17         ymax = ymax(1:(finalstep-lstart)/lstep);
18         ymdot = diff(ind)./diff(time(lstart:lstep:(lstart-1+lstep*length(
               ind))));
19     end
20     denavg(fld,:) = mean(denp);
21
22
23     if spp ~=0
24         dentemp = zeros(spp,64);
25         labdenp = zeros(1,64);
26         labdenpex = zeros(1,64);
27         dentempex = zeros(spp,64);
28         amp = round(FORCE_MULTIPLAYER/1.8*0.785);
29         for i = 1:spp
```

```
30              dentemppre = [zeros(1,50) mean(denp(i:spp:end,:)) zeros(1,50)];
31              for k = 2:64
32                  dentemp(i,k) = dentemppre(k+50+amp-round(amp*sin(2*pi*i/spp
                        )));
33                  dentempex(i,k) = dentemppre(k+1+50+amp-round(amp*sin(2*pi*i
                        /spp)));
34              end
35              dentemp(i,1) = dentemppre(1+50+amp-round(amp*sin(2*pi*i/spp)));
36              dentempex(i,1) = sum(dentemppre(1:1+50+amp-round(amp*sin(2*pi*i
                    /spp))));
37
38              labdenp = labdenp + dentemp(i,:);
39              labdenpex = labdenpex + dentemp(i,:);
40          end
41          labdenp = labdenp/max(labdenp);
42          den0avg(fld,:) = labdenp;
43      end
44
45      %classifiers
46      pofx = NaN(1,ceil(max(ymdot))-floor(min(ymdot))+1);
47      for i = 1:length(pofx)
48          pofx(i) = length(ymdot( (ymdot>i-1+floor(min(ymdot))) & (ymdot<i+
                floor(min(ymdot))) ));
49      end
50      l_mom3 = -sum(([floor(min(ymdot)):ceil(max(ymdot))].^3) .* pofx/length(
            ymdot))/var(ymdot);%-3rd moment
51      %2
52      l_1persistence = length(ind(ind==1))/length(ind);%BB for sure > 0.05
            maybe 0.01
53      %3
54      l_1prominence = denavg(fld,1)/max(denavg(fld,:));%BB > 0.5
55      l_hover = max(denavg(fld,:))/denavg(fld,1);
56      %4
57      l_locounter = length(ymdot(ymdot<-7000))/length(ymdot);%BB >0.04
58      %5
59      l_loco2 = length(ymdot(ymdot<-1.2*max(ymdot)))/length(ymdot);%BB > 0.1?
60      %6
61      l_losquarer = sum((ymdot(ymdot<-max(ymdot))/max(ymdot)).^2)/length(
            ymdot);
62      %7
63      l_lomeaner = max(-mean(ymdot(ymdot<-max(ymdot)))/max(ymdot),0);
64      %8
65      [~, locofmax]=max(denavg(fld,:));
66      %9
67      l_belowmax = length(ind(ind<locofmax))/length(ind);
68      %10 11
69      if locofmax <= 3 %complete BB
70          l_botprom = 1;
71          l_botprom2 = 1;
72          l_topprom = 0;
73      elseif l_belowmax <= 0.05 %complete Leid
74          l_botprom = 0;
```

```
75          l_botprom2 = 0;
76          l_topprom = 1;
77      else
78          tymax = NaN(1,size(denp,1));
79          tind = NaN(1,size(denp,1));
80          bymax = NaN(1,size(denp,1));
81          bind = NaN(1,size(denp,1));
82          for i = 1:size(denp,1)
83              [tymax(i), tind(i)] = max(denp(i,locofmax:64));
84              [bymax(i), bind(i)] = max(denp(i,1:locofmax));
85          end
86          l_botprom = mean(bymax(tind~=bind)./(tymax(tind~=bind)+bymax(tind~=
                bind)));
87          l_botprom2 = mean(bymax(tind~=bind))/(mean(tymax(tind~=bind))+mean(
                bymax(tind~=bind)));
88          l_topprom = mean(tymax(tind~=bind)./(tymax(tind~=bind)+bymax(tind~=
                bind)));
89      end
90
91  end
92  %Output:
93  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94  cd('/home/jtabet/plots/autoanalysis/');
95  if doL+doCR>=1
96      [ixnew, iynew] = meshgrid(1:0.25:Nx,1:0.25:Ny);
97  end
98
99  %%%%%% main output
100 if doL==1 && dL(fld)==1
101     if doCR==0 || dC(fld)==0
102         mean_wsq=-1;
103         mean_gw=-1;
104     else
105         mean_wsq=mean(wsq);
106         mean_gw=mean(gw);
107     end
108
109     %F phi; Gamma; leidfactors....
110     lfFile = fopen('leid.dat','a');%     1       2       3       4       5       6
                7       8       9       10      11      wsq     gw
111     fprintf(lfFile,'%4.2f\t%3.2f\t%2.0f\t%6.0f\t%.3f\t%.3f\t%.3f\t%.3f\t
            %4.3f\t%4.3f\t%.0f\t%.3f\t%.3f\t%.3f\t%.3f\t%5f\n',...
112         F,FILLING_FRACTION,FORCE_MULTIPLAYER,l_mom3,l_1persistence,
                l_1prominence,l_locounter,l_loco2,l_losquarer,...
113         l_lomeaner,locofmax,l_belowmax,l_botprom,l_botprom2,mean_wsq,
                mean_gw);
114     fclose(lfFile);
115     density = interp2(ixm,iym, density_plane(:,:,1)',ixnew,iynew,'spline');
116 end
117 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118
119 %%%%%% cr output
```

```
120 if doCR==1 && dC(fld)==1
121     if (finaltime<=cstartingtime)
122         nrolls = -1;
123     end
124     if mean(wsq) >= 0.0055
125         c1 = 1;
126     else
127         c1 = 0;
128     end
129     if doCount==0
130         maxvort = 0;
131     end
132     %F phi Gamma S mean(wsq) var(wsq) wmax^2 nrolls rolldiff finaltime
133     crFile = fopen('cr.dat','a');%Gamma  S      MWsqVWsqMaxW  1      #rl@6
        roldi finalt
134     fprintf(crFile,'%4.2f\t%3.2f\t%2.0f\t%4.1f\t%f\t%f\t%.2f\t%1.0f\t%.0f\t
        %.2f\t%.3f\n',F,FILLING_FRACTION,FORCE_MULTIPLAYER,S,...
135         mean(wsq(~isnan(wsq))),var(wsq(~isnan(wsq))),maxvort*maxvort,c1,
            nrolls,max(mean(rolldiff),0),finaltime);
136     fclose(crFile);
137     %phi Gamma #rolls(#T)...
138     if doCount==1
139         arFile = fopen('cr-arolls.dat','a');
140         fprintf(arFile,'\n%.2f\t%2.0f',FILLING_FRACTION,FORCE_MULTIPLAYER);
141         fprintf(arFile,'\t%.0f',arolls);
142         fclose(arFile);
143     end
144 end
145
146 %%%%%% PD output
147 if doCR==1 && doL==1
148     if (finaltime<=cstartingtime)
149         nrolls = -1;
150     end
151     if dC(fld)==0
152         nrolls = -1;
153     end
154     if dL(fld)==0
155         l_hover = -1;
156         l_topprom = -1;
157         locofmax = -1;
158     end
159     crFile = fopen('pd.dat','a');%Gamma  hov   top   loc   nrolls
160     fprintf(crFile,'%4.2f\t%3.2f\t%2.0f\t%.3f\t%.3f\t%.0f\t%.0f\n',F,
        FILLING_FRACTION,FORCE_MULTIPLAYER,...
161         l_hover,l_topprom,locofmax,nrolls);
162     fclose(crFile);
163 end
164
165
166 %%%%%% image output
167 cd('/home/jtabet/plots/autopics/');
```

```
168 if doCR==1 && dC(fld)==1 && pics==1
169     cau = DX*LENGTH_MULTIPLIER*curl(momentumX_plane' , momentumY_plane');
170     cau = interp2(ixm,iym, cau,ixnew,iynew,'spline');
171     fileName_vort = sprintf('%02.0f-%02.0fw.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
172     imwrite((flipud(cau)-min(min(cau)))/max(max((flipud(cau)-min(min(cau)))
            ))*64,def,fileName_vort);
173 end
174 if doL==1 && dL(fld)==1 && pics==1
175     %Density:
176     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
177     fileName_dens = sprintf('%02.0f-%02.0fr.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
178     imwrite(flipud(density)/max(max(density))*64,invbone,fileName_dens);%
            make like rest
179     %avgprofile:
180     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
181     figure(7);
182     fileName_pro = sprintf('%02.0f-%02.0fp.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
183     plot(denavg(fld,:),'Color',[0.1 0.4 0.8],'LineWidth',2)
184     view(90,-90)
185     axis([1 64 0 1.1*max(denavg(fld,:)) -1 1])
186     xlabel('$y$ [$L \Delta y$]','Fontsize',28);
187     ylabel('$\rho$','Fontsize',28);
188     set(7,'Color',[1 1 1]);
189     set(gca,'FontSize', 20)
190     prnt = getframe(gcf);
191     imwrite (prnt.cdata,strcat('/home/jtabet/plots/leidpics/',fileName_pro)
            )
192     %0avgprofile:
193     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
194     figure(9);
195     fileName_0pro = sprintf('%02.0f-%02.0fp0.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
196     plot(labdenp(1:32),'Color',[0 0 0],'LineWidth',2)
197     axis([1 32 0 1.05])
198     xlabel('height $y$ [$L \Delta y$]','Fontsize',28);
199     ylabel('density $\rho$','Fontsize',28);
200     set(gca,'FontSize', 20)
201     set(9,'Color',[1 1 1]);
202     prnt = getframe(gcf);
203     imwrite (prnt.cdata,strcat('/home/jtabet/plots/leidpics/',fileName_0pro
            ))
204     %0avgexprofile:
205     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
206     figure(10);
207     fileName_0proex = sprintf('%02.0f-%02.0fp0ex.png', FILLING_FRACTION
            *100, FORCE_MULTIPLAYER);
208     plot(labdenpex(:),'Color',[0.1 0.4 0.8],'LineWidth',2)
209     view(90,-90)
210     axis([1 64 0 1.2*max(den0avg(fld,:)) -1 1])
```

```
211     xlabel('$y$ [$L \Delta y$]','Fontsize',28);
212     ylabel('$\rho$','Fontsize',28);
213     set(gca,'FontSize', 20)
214     set(10,'Color',[1 1 1]);
215     prnt = getframe(gcf);
216     imwrite (prnt.cdata,strcat('/home/jtabet/plots/leidpics/',
            fileName_0proex))
217     %denp(t):
218     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
219     fileName_leid = sprintf('%02.0f-%02.0fl.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
220     figure(2);
221     if spp == 0
222         [~, enddenp] = min(abs(time-(lstartingtime + 5 * T)));
223         spp = (enddenp - lstart)/5;
224     end
225     if length(listing)>=lstart+5*spp
226         imagesc( time([lstart lstart-1+lstep*(round(5*spp))]),[1 64],
                flipud(denp(1:(round(5*spp)),:)'))
227     else
228         imagesc( time([lstart lstart-1+lstep*(fix( (length(listing)-lstart)
                /spp)*round(spp) )]),[1 64],flipud( denp(1:(fix( (length(
                listing)-lstart) /spp)*round(spp) ),:)' ) )
229     end
230     set(2,'Color',[1 1 1]);
231     set(gca,'FontSize', 20)
232     ylabel('$y$ [$L \Delta y$]','Fontsize',28);
233     xlabel('$t$ [s]','Fontsize',28);
234     prnt = getframe(gcf);
235     imwrite (prnt.cdata,strcat('/home/jtabet/plots/leidpics/',fileName_leid
            ))
236     %Histogram:
237     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
238     fileName_j = sprintf('%02.0f-%02.0flj.png', FILLING_FRACTION*100,
            FORCE_MULTIPLAYER);
239     figure(3);
240     [his, cen] = hist(ymdot,30);
241     bar(cen,his/trapz(cen,his))
242     set(3,'Color',[1 1 1]);
243     ylabel('count','Fontsize',28);
244     xlabel('$v_{max}$ [$\sigma /$s]','Fontsize',28);
245     set(gca,'FontSize', 20)
246     print(strcat('/home/jtabet/plots/leidpics/',fileName_j),'-dpng')
247
248 end
249 cd('/home/jtabet/plots/');
```

initialization.m:

```
1 MAXIMAL_PACKING = 0.9069;
2
3 bonecm = colormap('bone');
4 invbone = flipud(bonecm);
```

```
def = colormap('jet');

F = FILLING_FRACTION * LENGTH_MULTIPLIER * (N_Y-8) * DY / MAXIMAL_PACKING;
Gamma = FORCE_MULTIPLAYER;
S = FORCE_MULTIPLAYER * FORCE_MULTIPLAYER * 9.81/(4 * PI^2 *
    DIAMETER_PARTICLE * SHACKING_FREQUENZ_Y^2);
T = 1/SHACKING_FREQUENZ_Y;

fprintf(1,'%4.2f, %2.0f\tdataset %2.0f/%2.0f\t', FILLING_FRACTION, Gamma,
    fld,length(listdata));

listing = dir (strcat((folderName),'/density_t*.dat'));
listingName = [{listing(:).name};{listing(:).date}];
[d1,d2] = sort(datenum(listingName(2,:)'));
listing = listingName(:,d2);

tt = NaN(11,1);
tt(1) = 0;
schr = 1;

foo = char(listing(1,1));
strnum = (foo(strfind(foo,'_t')+2 :1: strfind(foo,'.dat')-1));
ti = eval(strnum)*TREF;
if (ti~=0)
    listing = [[cellstr('density_t0.0000000e+00.dat'); cellstr('01-Aug-2015
        00:00:10')] listing];
    ti = 0;
end

for loopn = 1:10
        while (ti < loopn * 0.02 && schr<=length(listing))
                foo = char(listing(1,schr));
                strnum = (foo(strfind(foo,'_t')+2 :1: strfind(foo,'.dat')
                    -1));
                ti = eval(strnum)*TREF;%real time
                schr = schr + 1;
        end
        tt(loopn+1) = schr - 2; %steps (schr) in current (loopn+1) period
end
spp = round(sum(tt)/55);%steps per period
diffvec = diff(tt);
accuracy = 0;
for loopn = 1:10
    if abs(diffvec(loopn)-spp) < 2
        accuracy = accuracy + 0.1;
        end
end

finalstring = char(listing(1,length(listing)));
finalstrnum = (finalstring(strfind(finalstring,'_t')+2 :1: strfind(
    finalstring,'.dat')-1));
finaltime = eval(finalstrnum)*TREF;
```

```
52  finaltv(fld) = finaltime;
53
54
55  p_phi(fld) = FILLING_FRACTION;
56  p_Gamma(fld) = Gamma;
57  p_Data(fld) = finaltime;
58
59  if ( ( sum(diffvec(1:5)) <= sum(diffvec(6:end)) && abs(accuracy-1)>=0.01 &&
        diffvec(1) ~= diffvec(end) ) || spp < 8 )
60      fprintf(1,'exp------\tt_max_=_%3.2f\n',finaltime);
61      p_cspp(fld) = 0;
62      exFile = fopen('/home/jtabet/plots/autoanalysis/exponential-data.dat','
            a');
63      fprintf(exFile,'%s\n',folderName);
64      fclose(exFile);
65      continue
66  else
67      p_cspp(fld) = 1;
68  end
69
70  fprintf(1,'spp_=_%.0f\t', spp);
71
72  if (finaltime >= cutofftime+0.2/spp)
73      step = ceil(cutofftime/finaltime*length(listing)-spp);
74      cur_t = 0;
75      while ( cur_t < cutofftime+0.2/spp )
76          step = step + 1;
77          foo = char(listing(1,step));
78          strnum = (foo(strfind(foo,'_t')+2 :1: strfind(foo,'.dat')-1));
79          cur_t = eval(strnum)*TREF;
80      end
81      listing = listing(:,1:step-1);
82  end
83
84  time = NaN(1,length(listing));
85  avgEnergyPerParticle = NaN(1,length(listing));
86  avgKinEnergyPerParticle = NaN(1,length(listing));
87  minDensity = NaN(1,length(listing));
88  avgDensity = NaN(1,length(listing));
89  maxDensity = NaN(1,length(listing));
90  maxTemperature = NaN(1,length(listing));
91  avgTemperature = NaN(1,length(listing));
92  minTemperature = NaN(1,length(listing));
93  energy = NaN(1,length(listing));
94
95  rounds=1;
96  maxvort=0;
97  if spp == 10
98      countrolltimer = 2;
99  end
100 if spp == 12
101     countrolltimer = 2;
```

```
end
if spp == 22
    countrolltimer = 5;
end
if spp == 30
    countrolltimer = 7;
end

Nx = N_X-8;
Ny = N_Y-8;
ix = 1:1:Nx;
iy = 1:1:Ny;
%%%%
cstart = round(1 + cstartingtime*50*spp);
lstart = round(1 + lstartingtime*50*spp);
if cstep==0
    cstep=spp;
end
if lstep==0
    lstep=spp;
end
if doCount == 1
    arolls=[];
end
wsq=[];
gw=[];
ymax = NaN(1,length(listing)-lstart);
ind = NaN(1,(length(listing)-lstart)/lstep);
[ixm, iym] = meshgrid(ix,iy);
denp = NaN((length(listing)-lstart)/lstep,64);

if min([cstep lstep])==1
    step = 1;
    while (step <= length(listing))
        foo = char(listing(1,step));
        strnum = (foo(strfind(foo,'_t')+2 :1: strfind(foo,'.dat')-1));
        time(step) = eval(strnum)*TREF;
        step = step + 1;
    end
end

if (min([cstart lstart])+3*spp > length(listing))
    fprintf(1,'t_max_=_%.2f_NOT_ENOUGH_DATA\n', finaltime);
    insFile = fopen('/home/jtabet/plots/autoanalysis/insufficient-data.dat'
        ,'a');
    fprintf(insFile,'%s\n',folderName);
    fclose(insFile);
    continue
else
    fprintf(1,'t_max_=_%.2f,_%4.0f_steps\n', finaltime,length(listing));
end

```

```
153 if finaltime<lstartingtime+3*T
154     dL(fld)=0;
155 end
156 if finaltime<cstartingtime+3*T
157     dC(fld)=0;
158 end
```

loadgrids.m:

```
1  foo = char(listing(1,step));
2  strnum = (foo(strfind(foo,'_t')+2 :1: strfind(foo,'.dat')-1));
3
4  fileName_density = sprintf('%s/density_t%s.dat',folderName , strnum);
5  fileName_densityEnergy = sprintf('%s/densityEnergy_t%s.dat', folderName,
       strnum);
6  fileName_momentumX = sprintf('%s/momentumX_t%s.dat', folderName, strnum);
7  fileName_momentumY = sprintf('%s/momentumY_t%s.dat', folderName, strnum);
8  fileName_momentumZ = sprintf('%s/momentumZ_t%s.dat', folderName, strnum);
9
10 fh_density = fopen(fileName_density,'rb');
11
12 if (fh_density==-1)
13     continue;
14 end
15
16 fh_densityEnergy = fopen(fileName_densityEnergy,'rb');
17 if (fh_densityEnergy == -1)
18     fclose(fh_density);
19     continue;
20 end
21
22 fh_momentumX = fopen(fileName_momentumX,'rb');
23 if (fh_momentumX == -1)
24     fclose(fh_density);
25     fclose(fh_densityEnergy);
26     continue;
27 end
28
29 fh_momentumY = fopen(fileName_momentumY,'rb');
30 if (fh_momentumY == -1)
31     fclose(fh_density);
32     fclose(fh_densityEnergy);
33     fclose(fh_momentumX);
34     continue;
35 end
36
37 fh_momentumZ = fopen(fileName_momentumZ,'rb');
38
39 if (fh_momentumZ == -1)
40     fclose(fh_density);
41     fclose(fh_densityEnergy);
42     fclose(fh_momentumX);
43     fclose(fh_momentumY);
```

```
44      continue;
45  end
46
47  if (isnan(time(step)) || time(step)==0)
48      time(step) = eval(strnum)*TREF;
49  end
50
51  ary_density = fread(fh_density, N_X * N_Y * N_Z , FLOATTYPE);
52  fclose(fh_density);
53
54  ary_momentumX = fread(fh_momentumX, N_X * N_Y * N_Z , FLOATTYPE);
55  fclose(fh_momentumX);
56
57  ary_momentumY = fread(fh_momentumY, N_X * N_Y * N_Z , FLOATTYPE);
58  fclose(fh_momentumY);
59
60  ary_momentumZ = fread(fh_momentumZ, N_X * N_Y * N_Z , FLOATTYPE);
61  fclose(fh_momentumZ);
62
63  ary_densityEnergy = fread(fh_densityEnergy, N_X * N_Y * N_Z , FLOATTYPE);
64  fclose(fh_densityEnergy);
65
66  ary_velocity2 = (ary_momentumX .* ary_momentumX + ary_momentumY .*
        ary_momentumY + ary_momentumZ .* ary_momentumZ)./(ary_density .*
        ary_density);
67
68  ary_Temperature = 1.0/DIMENSION*(2.0 * ary_densityEnergy./ ary_density -
        ary_velocity2);
69
70  if (length(ary_density) ~= (length(ary_momentumX) + length(ary_momentumY) +
         length(ary_momentumZ) + length(ary_densityEnergy))/4 );
71      continue;
72  end
73
74  if (~isempty(ary_Temperature(isnan(ary_Temperature(:)))))
75      continue;
76  end
77
78  energy(step) = mean(ary_densityEnergy./ary_density);%not used
79  %
80  pressure_plane = NaN(N_X,N_Y);
81
82  density_plane = reshape(ary_density,[N_X,N_Y,N_Z]);
83
84  momentumX_plane = reshape(ary_momentumX,[N_X,N_Y,N_Z]);
85
86  velX_plane = momentumX_plane./density_plane;
87
88  momentumY_plane = reshape(ary_momentumY,[N_X,N_Y,N_Z]);
89
90  velY_plane = momentumY_plane./density_plane;
91
```

```
92  momentumZ_plane = reshape(ary_momentumZ,[N_X,N_Y,N_Z]);
93
94  velZ_plane = momentumZ_plane./density_plane;
95
96  densityEnergy_plane = reshape(ary_densityEnergy,[N_X,N_Y,N_Z]);
97
98  temperature_plane = reshape(ary_Temperature,[N_X,N_Y,N_Z]);
99  %
100 density_plane = density_plane(5:1:N_X-4,5:1:N_Y-4,1);%bottom at line 1
101
102 momentumX_plane = momentumX_plane(5:1:N_X-4,5:1:N_Y-4,1);
103 velX_plane = velX_plane(5:1:N_X-4,5:1:N_Y-4,1);
104 globMomX = mean(momentumX_plane(:));
105
106 momentumY_plane = momentumY_plane(5:1:N_X-4,5:1:N_Y-4,1);
107 velY_plane = velY_plane(5:1:N_X-4,5:1:N_Y-4,1);
108 globMomY = mean(momentumY_plane(:));
109
110 momentumZ_plane = momentumZ_plane(5:1:N_X-4,5:1:N_Y-4,1);
111 globMomZ = mean(momentumZ_plane(:));
112
113 densityEnergy_plane = densityEnergy_plane(5:1:N_X-4,5:1:N_Y-4,1);
114
115 temperature_plane = temperature_plane(5:1:N_X-4,5:1:N_Y-4,1);
116 %
117 avgEnergyPerParticle(step) = mean( densityEnergy_plane(:)./density_plane(:)
      );
118 avgKinEnergyPerParticle(step) = mean( (2.0*densityEnergy_plane(:)./
    density_plane(:) ) - DIMENSION * temperature_plane(:) );
119 minDensity(step) = min(density_plane(:));
120 avgDensity(step) = mean(density_plane(:));
121 maxDensity(step) = max(density_plane(:));
122 maxTemperature(step) = max(temperature_plane(:));
123 avgTemperature(step) = mean( temperature_plane(:) );
124 minTemperature(step) = min( temperature_plane(:) );
```

dataallocator.m:

```
1  addpath('/home/jtabet/plots/mfunctions');
2  masterfolder = '/scratch.local/data/joscha_tabet/ClusterData/';
3  cd(masterfolder);
4  a=subdir;
5  b=[];
6  listdata=[];
7  for i = 1:length(a)
8      c=char(a(i));
9      c=c(46:end);
10     if (c(1)=='M')
11         b = [b strcat(masterfolder,(cellstr(c)))];
12     end
13 end
14
15 for i = 1:length(b)
```

```
16      subd=subdir(char(b(i)));
17      for k = 1:length(subd)
18          subsub=char(subd(k));
19          subsub=subsub(52:end);
20          listdata=[listdata strcat(b(i),'/',subsub)];
21      end
22  end
23
24  finaltv = NaN(1,length(listdata));
25
26  cd('/home/jtabet/plots/');
27
28  p_phi = NaN(1,length(listdata));
29  p_Gamma = NaN(1,length(listdata));
30  P_Data = NaN(1,length(listdata));
31
32  dC = ones(1,length(listdata));
33  dL = ones(1,length(listdata));
34
35  denavg = NaN(length(listdata),64);
36  den0avg = NaN(length(listdata),64);
37
38  set(0,'defaulttextinterpreter','latex')
```

References

[1] P. Eshuis, D. Van Der Meer, M. Alam, H. J. Van Gerner, K. Van Der Weele, and D. Lohse, *Onset of convection in strongly shaken granular matter*, Physical Review Letters **104**, 038001 (2010).

[2] P. Eshuis, K. van der Weele, D. van der Meer, R. Bos, and D. Lohse, *Phase diagram of vertically shaken granular matter*, Physics of Fluids **19**, 123301 (2007).

[3] P. Eshuis, K. Van Der Weele, D. Van Der Meer, and D. Lohse, *Granular Leidenfrost effect: Experiment and theory of floating particle clusters*, Physical Review Letters **95**, 258001 (2005).

[4] T. Pöschel and S. Luding, *Granular gases*, vol. 564, (Springer Science & Business Media 2001).

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

[6] J. J. Brey, F. Moreno, and J. W. Dufty, *Model kinetic equation for low-density granular flow*, Physical Review E **54**, 445 (1996).

[7] J. J. Brey, J. W. Dufty, and A. Santos, *Dissipative dynamics for hard spheres*, Journal of Statistical Physics **87**, 1051 (1997).

[8] J. J. Brey, J. W. Dufty, C. S. Kim, and A. Santos, *Hydrodynamics for granular flow at low density*, Physical Review E **58**, 4638 (1998).

[9] C. Campbell, *Rapid Granular Flows*, Annual Review of Fluid Mechanics **22**, 57 (1990).

[10] B. Meerson, T. Pöschel, and Y. Bromberg, *Close-Packed Floating Clusters: Granular Hydrodynamics Beyond the Freezing Point*, Physical Review Letters **91**, 024301 (2003).

[11] 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**, 3485 (1985).

[12] E. L. Grossman, T. Zhou, and E. Ben-Naim, *Towards granular hydrodynamics in two dimensions*, Physical Review E **55**, 4200 (1997).

[13] N. F. Carnahan, *Equation of State for Nonattracting Rigid Spheres*, The Journal of Chemical Physics **51**, 635 (1969).

[14] H. Struchtrup and P. Taheri, *Macroscopic transport models for rarefied gas flows: a brief review*, IMA Journal of Applied Mathematics **76**, 672 (2011).

[15] P. K. Haff, *Grain flow as a fluid-mechanical phenomenon*, Journal of Fluid Mechanics **134**, 401 (1983).

[16] M. Hummel, *Hydrodynamics of granular gases: clustering, universality and importance of subsonic convective waves*, Ph.D. thesis, Universität Göttingen (2016).

[17] H. Jeffreys, *LXXVI. The stability of a layer of fluid heated below*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 833 (1926).

[18] M. Scheel, D. Geromichalos, and S. Herminghaus, *Wet granular matter under vertical agitation*, Journal of Physics: Condensed Matter **16**, 4213 (2004).

[19] X. X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes* (1994).

[20] G.-S. Jiang and C.-W. Shu, *Efficient Implementation of Weighted ENO Schemes*, Journal of Computational Physics **126**, 202 (1995).

[21] V. A. Titarev and E. F. Toro, *Finite-volume WENO schemes for three-dimensional conservation laws*, Journal of Computational Physics **201**, 238 (2004).

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, vol. 1 (2007).

[23] E. F. Toro and V. A. Titarev, *TVD fluxes for the high-order ADER schemes*, Journal of Scientific Computing **24**, 285 (2005).

[24] C. R. Wassgren, C. E. Brennen, and M. L. Hunt, *Vertical Vibration of a Deep Bed of Granular Material in a Container* (1996).

[25] S. S. Hsiau and S. J. Pan, *Motion State Transitions in a Vibrated Granular Bed*, Powder Technology **96**, 219 (1998).

[26] O. Sano, *Dilatancy, buckling, and undulations on a vertically vibrating granular layer*, Physical Review E **72**, 051302 (2005).

[27] P. Eshuis, K. Van Der Weele, M. Alam, H. J. Van Gerner, M. Van Der Hoef, H. Kuipers, S. Luding, D. Van Der Meer, and D. Lohse, *Buoyancy driven convection in vertically shaken granular matter: Experiment, numerics, and theory*, Granular Matter **15**, 893 (2013).

[28] D. Paolotti, A. Barrat, U. M. B. Marconi, and A. Puglisi, *Thermal convection in monodisperse and bidisperse granular gases: A simulation study*, Physical Review E **69**, 061304 (2004).

[29] R. D. Wildman, J. M. Huntley, and D. J. Parker, *Convection in highly fluidized three-dimensional granular beds*, Physical Review Letters **86**, 3304 (2001).

[30] N. Rivas, S. Luding, and A. R. Thornton, *Low-frequency oscillations in narrow vibrated granular systems*, New Journal of Physics **15** (2013).

Acknowledgments

I would like to thank everyone who helped me in the process of creating this Bachelor thesis. In particular my friends from university and my family. I am very thankful to my advisor Dr. Marco G. Mazza who initiated me into the world of professional academia and who provoked much of my research through his insightful questions. I especially want to thank Mathias Hummel who provided the basis for my work. Literally, in the sense that my results are based on the model and the solvers he implemented, but also in terms of his tireless help, explaining to me the intricacies of his code and much more.

Erklärung

Nach §13(9) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen: Hiermit erkläre ich, dass ich diese Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, also solche kenntlich gemacht habe. Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den 15. November 2016

(Joscha Tabet)