Granular Matter (2012) 14:363–380
DOI 10.1007/s10035-012-0346-z

ORIGINAL PAPER

An implementation of the soft-sphere discrete element method in a high-performance parallel gravity tree-code

Stephen R. Schwartz · Derek C. Richardson · Patrick Michel

Received: 20 January 2012 / Published online: 30 March 2012

Abstract We present our implementation of the soft-sphere discrete element method (SSDEM) in the parallel gravitational N-body code PKDGRAV, a well-tested simulation package that has been used to provide many successful results in the field of planetary science. The implementation of SSDEM allows for the modeling of the different contact forces between particles in granular material, such as various kinds of friction, including rolling and twisting friction, and the normal and tangential deformation of colliding particles. Such modeling is particularly important in regimes for which collisions cannot be treated as instantaneous or as occurring at a single point of contact on the particles' surfaces, as is done in the hard-sphere discrete element method already implemented in the code. We check the validity of our soft-sphere model by reproducing successfully the dynamics of flows in a cylindrical hopper. Other tests will be performed in the future for different dynamical contexts, including the presence of external and self-gravity, as our code also includes interparticle gravitational force computations. This will then allow us to apply our tool with confidence to planetary science studies, such as those aimed at understanding the dynamics of regolith on solid celestial body surfaces, or at designing efficient sampling tools for sample-return space missions.

Keywords Bulk solids · Solar system · DEM · Hopper · SSDEM

1 Introduction

The study of granular materials and their dynamics is of great importance for a wide range of applications in industry, but also in the field of planetary science. Most celestial solid bodies' surfaces are not bare rock, but are instead covered by granular material. This material can take the form of fine regolith as on the Moon, or gravels and pebbles as on the 320-m size near-Earth asteroid Itokawa, which was visited by the Japan Aerospace Exploration Agency (JAXA) space mission Hayabusa in 2005, returning to Earth in 2010 with some samples [1]. Moreover, it has been found that this granular material can flow due to various circumstances, such as landslides in crater walls, or global shaking due to the propagation of seismic waves as a result of small impacts on low-gravity bodies (e.g., Richardson et al. [2]). However, the response of granular materials on these bodies to various kinds of processes, as a function of their material properties and over the changes in surface gravity suffered due to encounters with other bodies, is still not well understood. Such an understanding is important for the interpretation of images of surfaces of planets, satellites, and small bodies sent to us by spacecraft. It is also relevant to the development of efficient sampling designs and anchoring tools for space missions aimed at attaching to, or obtaining a sample from, the surface of such bodies.

Electronic supplementary material The online version of this article (doi:10.1007/s10035-012-0346-z) contains supplementary material, which is available to authorized users.

S. R. Schwartz (✉) · D. C. Richardson
Department of Astronomy, University of Maryland, College Park, MD 20742-2421, USA
e-mail: srs@astro.umd.edu

S. R. Schwartz · P. Michel
Lagrange Laboratory, University of Nice Sophia Antipolis, CNRS, Côte d'Azur Observatory, Observatoire de la Côte d'Azur, B.P. 4229, 06304 Nice Cedex 4, France

In this paper, we present our implementation of the soft-sphere discrete element method (SSDEM) in the parallel gravitational N-body code PKDGRAV. Richardson et al. [3] presented the implementation of the hard-sphere discrete element method (HSDEM) in this code, along with a discussion of the primitives (walls) that can be used to represent various kinds of boundary conditions (crater floors, geometry of sampling tools, experimental conditions, etc.). We refer the reader to the introduction of that paper for a brief overview of the importance of the dynamics of granular materials in planetary science.

Different approaches exist to perform modeling of granular materials [4]. The discrete element method (DEM), which is a numerical method for computing the motion of large numbers of particles of micron-scale size and above, is a commonly used approach. DEM is however relatively computationally intensive, which limits either the length of a simulation or the number of particles. Several DEM codes take advantage of parallel processing capabilities to scale up the number of particles or length of the simulation (e.g., [5,6]). An alternative to treating all particles separately is to average the physics across many particles and thereby treat the material as a continuum. In the case of solid-like granular behavior, the continuum approach usually treats the material as elastic or elasto-plastic and models it with the finite element method or a mesh-free method (e.g., [7]; also see [8–10] for use of analytical and continuum approaches in modeling asteroid shapes). In the case of liquid-like or gas-like granular flow, the continuum approach may treat the material as a fluid and use computational fluid dynamics. However, the homogenization of granular-scale physics is not necessarily appropriate for capturing the discrete nature of the particles and the forces between them (and the forces between them their wall-boundaries) [11]. Therefore, limits of such homogenization must be considered carefully before attempting to use a continuum approach.

Discrete element method numerical codes are typically carried out by way of Hard-sphere (HSDEM) or Soft-Sphere (SSDEM) particle dynamics. They have been used successfully in many granular physics applications. For instance, Hong and McLennan [12] used hard-sphere molecular dynamics to study particles flowing through a hole in a two-dimensional box under the influence of gravity. Huilin et al. [13] used a Eulerian-Lagrangian approach coupled with a discrete hard-sphere model to obtain details of particle collision information in a fluidized bed of granular material. Also, Kosinski and Hoffman [14] compared the standard hard-sphere method including walls to a hard-sphere model with walls that also accounts for particle adhesion. The van der Waals type interaction is presented as a demonstration case.

The hard-sphere discrete element method predicts collisions in advance and treats these collisions between grains as instantaneously occurring at a single point of contact that lies on the particles' surfaces. Collision prediction can prove to be difficult, especially between particles and walls with complex geometries and/or motions. Furthermore, collisions between grains in a granular medium are not in fact instantaneous, but rather involve surface deformation of the grains at contact along with complex contact forces (friction) that are not well accounted for in HSDEM. Therefore, while HSDEM may still be appropriate in dilute/ballistic regimes given its ability to handle larger timesteps, it is not well adapted to dense regimes in which grain deformation and the complexity of frictional forces during contact cannot be neglected. SSDEM is called for in these regimes. However, although SSDEM has the advantage of not requiring collisions to be predicted in advance, it comes at the expense of much smaller integration timesteps, which can limit the integration timescale. On the other hand, because it can be implemented into a code such as ours that is fully and efficiently parallelized, it is currently possible to follow the evolution of millions of particles over a fairly large range of conditions.

The soft-sphere discrete element method is commonly used in the study of granular materials, and has often been applied to industrial problems [5,15]. However, it is only very recently that it has started to be applied to the realm of planetary science (see e.g., Sànchez et al. [16], Tancredi et al. [17], although some groups, such as [17], make use of third-party software not fully under their control). An important characteristic of SSDEM in PKDGRAV, in addition to its parallelization, is that it can cover a very wide range of gravity regimes and boundary conditions, as is required to study the large variety of environments encountered in the Solar System. This is an important asset given that the dynamical behavior of granular material can depend strongly on the local gravity, which can vary greatly from one Solar System object (e.g., small bodies, such as asteroids and comets) to another (e.g., planets). Moreover, the implementation of certain types of frictions (e.g., tangential, static) differs from some of the previous approaches. For instance, in an early implementation of a three-dimensional SSDEM code, Gallas et al. [18] did not use tangential restoring forces and thus did not have a static friction limit ($\mu_s = 0$). The only tangential force included was a viscous damping force. Therefore, the particles did not have memory of previous contacts and the Coulomb frictional coefficient used was instead one of kinetic friction, which puts a limit on the tangential damping force. Our treatment of tangential forces is similar to that of Silbert et al. [19] in that both elastic and plastic tangential deformation terms are supported. In addition, we consider other frictional terms, including rolling and twisting. In all, we outline four parameters that govern the exchange of tangential and rotational momentum between particles in contact, and allow for realistic dissipation of rotational and translational energy under a wide variety of conditions. This ensures the code is general in scope, has the ability to simulate many

different types of material, and that it can be used to explore a large parameter space.

However, before applying the code to realistic situations and over the wide range of size scales important in applications to planetary science, comparison with well-known results and laboratory experiments is required for validation. We first present, in Sect. 2, SSDEM as implemented in PKDGRAV, including all the parameters used and the contact forces that are taken into account. Then we compare, in Sect. 3, cylindrical hopper simulations and experiments in order to check the validity of our numerical model. Conclusions and perspectives are presented in Sect. 4.

2 Method

We have implemented SSDEM in the N-body code PKDGRAV, a parallel gravity code originally designed for collisionless cosmology simulations [20] and adapted for collisional Solar System applications [21,22]. SSDEM permits realistic modeling of the contact forces between particles in a granular material. The soft-sphere collisional model is carried out by allowing particle surfaces to penetrate each other [23]. When an overlap occurs, the particles are subject to forces that depend on the degree of overlap and the relative velocities and spins of the particles, as well as their material properties. Overlaps are detected each timestep by taking advantage of PKDGRAV's hierarchical tree data structure [21] to generate particle neighbor lists in $\mathcal{O}(N \log N)$ time, where N is the number of particles in the simulation.

We use a second-order leapfrog integrator, in which particle positions and velocities are alternately "drifted" and "kicked" according to

$$\begin{aligned} \dot{\mathbf{r}}_{i,n+\frac{1}{2}} &= \dot{\mathbf{r}}_{i,n} + (h/2)\ddot{\mathbf{r}}_{i,n} && \text{“kick”} \\ \mathbf{r}_{i,n+1} &= \mathbf{r}_{i,n} + h\dot{\mathbf{r}}_{i,n+\frac{1}{2}} && \text{“drift”} \\ \dot{\mathbf{r}}_{i,n+1} &= \dot{\mathbf{r}}_{i,n+\frac{1}{2}} + (h/2)\ddot{\mathbf{r}}_{i,n+1} && \text{“kick”}, \end{aligned} \tag{1}$$

where $\mathbf{r}_{i,n}$ is the position of particle i at step n, h is the (constant) timestep that takes the system from step n to step $n+1$, and the derivatives are with respect to time. This integrator has the desirable property that it is symplectic, meaning it exactly solves an approximate Hamiltonian of the system, thereby conserving phase-space volume so that, for example, the energy error remains bounded (for sufficiently small h; see [24] for details). Symplectic methods are ideal for equations of motion of the form $\ddot{x} = F(x)$, of which the simple harmonic oscillation of a spring is a prime example. For this reason, this approach is well-suited in general to SSDEM.

However, most SSDEM simulations include dissipation, in the form of a damped spring (cf. Sect. 2.3), and/or other types of friction (Sects. 2.4 and 2.5). Although this would seem to negate the usefulness of the leapfrog, the relative simplicity of the integrator, coupled with the ease with which it can be parallelized, still make it a good choice for SSDEM applications. One complication is that the damping term is usually an explicit function of velocity, which is out of sync with position during the leapfrog integration step. We get around this in a naïve way by using "predicted" velocities $\dot{\mathbf{r}}_{i,n+1} \approx \dot{\mathbf{r}}_{i,n} + h\ddot{\mathbf{r}}_{i,n}$ and spins $\dot{\boldsymbol{\omega}}_{i,n+1} \approx \dot{\boldsymbol{\omega}}_{i,n} + h\ddot{\boldsymbol{\omega}}_{i,n}$ to solve for $F(x)$. The proper way is to derive the correct Hamiltonian for the modified equations of motion and construct an appropriate leapfrog integration scheme from that (see Quinn et al. [25] for discussion in the context of the shearing-sheet scenario). However, this level of sophistication is not needed here because we take very conservative timesteps (small h) and integration errors are generally subsumed in the imposed damping and/or friction anyway.

Since SSDEM forces are computed only once per timestep, and simultaneously for all particles, the approach benefits tremendously from PKDGRAV's parallelization, with wallclock time dropping nearly linearly with the number of cores (the precise scaling prefactor depends on the details of the networking between cores). As a result, simulations of systems comprised of millions of particles, such as those presented in Sect. 3, can be completed in a matter of a few days on single 12-core 3-GHz nodes.

Our methodology in carrying out particle-particle collisions with SSDEM is based on the work of Cundall and Strack [23], although we have added several more features, such as rolling and twisting friction. In this section, we describe our modeling of normal and tangential deformations, along with different types of friction, in particular the rolling and twisting friction that are often neglected in SSDEM codes. We then explain how we treat the coefficient of restitution in the case of particle collisions and how the timestep necessary to perform the computations is chosen.

When computing forces in PKDGRAV as part of the regular integration step, neighbor searches are performed using the tree code. We then compute the amount of overlap between neighboring particles, given by

$$x = s_p + s_n - |\boldsymbol{\rho}|, \tag{2}$$

where s_p is the particle radius (particles are spheres), s_n is the neighbor particle radius, and $\boldsymbol{\rho} = \mathbf{r}_n - \mathbf{r}_p$ is the relative position between the particle and neighbor centers of mass (COMs), so $|\boldsymbol{\rho}|$ is the scalar distance between the particle COMs. Quantitatively, x represents the extent of particle overlap, but can be interpreted physically as the sum of the particles' deformations along the line that connects their centers due to their mutual contact (see Fig. 1a).

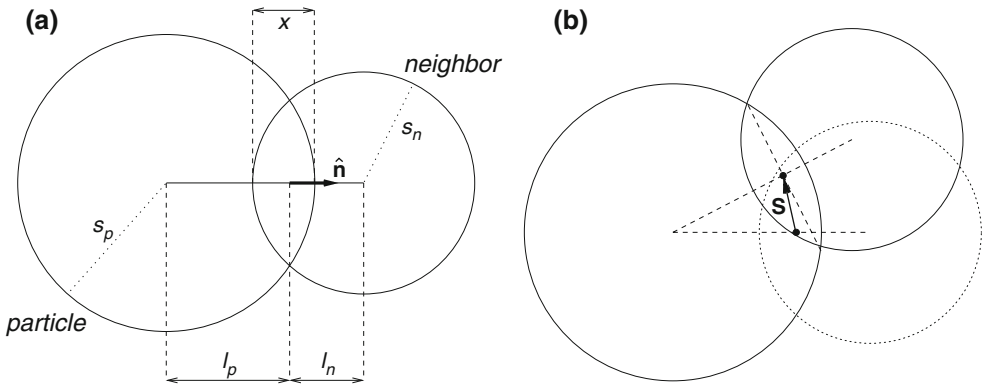

Fig. 1 A (spherical) particle in overlap with one of its neighbors. **a** Labels show quantities needed to compute the reaction force due to overlap, regardless of duration. Here s_p, l_p, s_n, and l_n are the radii and moment arms for the particle and its neighbor, respectively. The particle's moment arm is given by the scalar distance from the particle's center to the contact point (likewise for the neighbor's moment arm). The quantity x is the scalar distance between the surfaces of the two particles along the line that connects their centers and $\hat{\mathbf{n}}$ is a unit vector that gives the direction from the contact point to the neighbor's center. **b** Illustration of the quantity needed for the tangential component of the restoring force, which arises from a persistent contact: **S** is the tangential component of a vector that points from the equilibrium contact point to the current contact point and is generated by integrating all tangential motion that has occurred over the history of the contact. Note that both x and **S** are exaggerated to illustrate the method; x typically does not exceed $0.005(s_n + s_p)$, and $|\mathbf{S}| \ll s_p, s_n$, so that **S** is close to perpendicular to both the initial and current lines that connect the two particles' centers

2.1 Normal deformation

Consider a pair of particles for which x is positive, a repulsive restoring spring force is generated along the normal according to Hooke's law,[1]

$$\mathbf{F}_{N,\text{restoring}} = -k_n x\, \hat{\mathbf{n}}, \tag{3}$$

where $\hat{\mathbf{n}} \equiv \boldsymbol{\rho}/|\boldsymbol{\rho}|$ is a unit vector that gives the direction from the particle's center to the neighbor's center and k_n is the constant for the normal spring, which can be adjusted in order to control the amount of interparticle penetration that is allowed in a given simulation. In choosing a value for k_n, it has been the practice to limit x to $\sim$1 % of the smallest particle radius in the simulation [26]. In order to choose k_n so that we ensure the maximum values of x are close to this limit, we consider two regimes: one where particle kinetic energy dominates and determines the interparticle penetration, and another where the confining pressure on low-energy particles in a dense medium is responsible for the maximum interparticle penetration. In the first regime, if the maximum particle speed during the simulation can be predicted, putting that kinetic energy entirely into a single spring with x equal to the desired maximum value $x_{\max}$ gives a recommended value for k_n of

$$k_n \sim m\left(\frac{v_{\max}}{x_{\max}}\right)^2, \tag{4}$$

where the mass, m, corresponds to the typical mass of these most energetic particles. For example, if typical particles are 10 g in mass with diameters of 1 cm, and maximum speeds do not tend to exceed about 10 m s^{-1}, a k_n value of $\sim$ few $\times 10^8$ kg s^{-2} is suggested. Note we do not need to compute an "exact" k_n, just a conservative value; we do monitor each collision and generate a warning if $x_{\max}$ is greatly exceeded.

In the second regime of low-energy particles under confining pressure due to a global potential (e.g., gravity or spin), we simply have to estimate what this pressure might be and then choose a value of k_n such that the maximum opposing SSDEM normal force (Eq. (3)) will correspond to the desired maximum penetration, $x_{\max}$. For example, consider a box with an open top of height H filled with low-energy, identical particles of radius s ($s \ll H$) and density ρ under the influence of gravity (here ρ refers to the density of a single particle, as distinguished from the bulk density of the collection). In this case, particles near the bottom would be expected to each exhibit typical repulsive forces of $\sim \phi\rho a_g H s^2$, where a_g is the uniform gravitational acceleration and ϕ is the packing efficiency (so the bulk density is $\phi\rho$). Taking $x_{\max} \sim 0.01s$ and $\phi = 65$ %, and balancing with Eq. (3) gives

$$k_n \sim \frac{\phi\rho a_g H s^2}{x_{\max}} \sim 65\rho a_g H s. \tag{5}$$

As an example, if we take $\rho = 4\,\text{g cm}^{-3}$, $s = 1\,\text{cm}$, $H = 1\,\text{m}$, and a_g equal to Earth gravity, g, we find an optimum value for k_n of a few $\times 10^4$ kg s^{-2}.

Fundamentally, a higher k_n results in smaller overlaps, but a larger repulsive force, so the principal disadvantage of raising k_n is that smaller timesteps are needed to resolve the forces (see Sect. 2.7).

In cases where it is important to match the sound speed of real materials, k_n can also be chosen to control the speed

[1] Other functional dependencies on x are easily implemented, such as $x^{3/2}$ (Hertzian), etc. We chose a linear dependence because it is a simple choice that is often used.

of energy propagation through a medium represented by soft-sphere particles. In densely packed material, this speed should be close to $2s\tau_{\text{overlap}}^{-1}$ for monodisperse particles (for polydisperse particles, replace s with an appropriate mean or typical value for particle radius), where τ_{overlap} is the typical duration of collisional overlap (see Sects. 2.6 and 2.7 along with Eq. (36) for the derivation of τ_{overlap}). In practice, some experimentation will be needed to tune the sound propagation speed. For instance, it might be beneficial to adjust k_n to match the sound speeds of real material for impact events in energy regimes where we do not expect significant fracturing, but where nonetheless sound propagation might still be important. However, for many slow-speed granular processes, the computational cost of using a "realistic" value for k_n could be very high (Eq. (39) relates the value of k_n to a recommended timestep), especially for certain materials, and may not result in any worthwhile insight. Indeed, once above a certain k_n value, the outcomes can be largely independent of the specific value. In these cases, "softening" the material while increasing the resolution of the simulation might be more cost-effective than ensuring that a specific value of k_n is being used.

2.2 Tangential deformation

The restoring force in the tangential direction is given by

$$\mathbf{F}_{T,\text{restoring}} = k_t \mathbf{S}, \tag{6}$$

where k_t is the constant for the tangential spring and $\mathbf{S}$ is the tangential displacement from the equilibrium contact point, defined as

$$\mathbf{S} \equiv \int_{\text{overlap}} \mathbf{u}_t(t)\, dt + \mathbf{S}_0, \tag{7}$$

where the integral is over the duration of the static overlap (i.e., the interval over which static friction is acting), $\mathbf{u}_t$ is the relative tangential motion at the contact point (see Eq. (12)), and $\mathbf{S}_0$ is the tangential extension at the start of a static overlap. $\mathbf{S}_0$ is zero when particles first penetrate, but can be nonzero in the event of slipping (see Sect. 2.4).

Essentially, $\mathbf{S}$ is a vector that gives the tangential component of the deformation, and so its negative points from the current contact point to the point of tangential equilibrium (see Fig. 1b). As particles move, not only will the contact point move, but the equilibrium contact point will also change in the reference frame of the system. We account for this motion at every step by transforming $\mathbf{S}$ according to the change in $\hat{\mathbf{n}}$ over the previous step. This is done in two stages: a rotation around the $\hat{\mathbf{n}}$ vector, and a rotation around the vector orthogonal to $\hat{\mathbf{n}}$, around which $\hat{\mathbf{n}}$ has rotated over the previous step. This calculation, which is done for every contact in the system at every step, can be computationally expensive, but is important for lasting contacts.

2.3 Kinetic friction (damping)

Kinetic friction is implemented by damping the springs in the normal and tangential directions according to the widely used "dashpot" model. We start with the total relative velocity, which is given by

$$\mathbf{u} = \mathbf{v}_n - \mathbf{v}_p + l_n(\hat{\mathbf{n}}\times\boldsymbol{\omega}_n) - l_p(\hat{\mathbf{n}}\times\boldsymbol{\omega}_p), \tag{8}$$

where $\mathbf{v}_p$ is the COM velocity of the particle, $\mathbf{v}_n$ is the COM velocity of the neighbor particle, $\boldsymbol{\omega}_p$ is the spin of the particle, $\boldsymbol{\omega}_n$ is the spin of the neighbor particle, and l_p and l_n are lever arms from the particle centers to the effective point of contact, which is taken to be at the center of the circle that corresponds to the intersection of the particles' spherical surfaces. This point lies on the line segment that connects the particles' centers, at a distance

$$l_p = \frac{s_p^2 - s_n^2 + |\boldsymbol{\rho}|^2}{2|\boldsymbol{\rho}|} \tag{9}$$

from the particle in question. The lever arm for the neighbor particle is simply

$$l_n = |\boldsymbol{\rho}| - l_p. \tag{10}$$

The normal and tangential components of $\mathbf{u}$ are given by

$$\mathbf{u}_n = (\mathbf{u}\cdot\hat{\mathbf{n}})\hat{\mathbf{n}}, \tag{11}$$

$$\mathbf{u}_t = \mathbf{u} - \mathbf{u}_n. \tag{12}$$

The tangential unit vector $\hat{\mathbf{t}}$ is then given by $\hat{\mathbf{t}} \equiv \mathbf{u}_t/|\mathbf{u}_t|$ (if $\mathbf{u}_t = 0$, we set $\hat{\mathbf{t}}$ to zero). The normal and tangential components of the damping forces are then given by

$$\mathbf{F}_{N,\text{damping}} = C_n \mathbf{u}_n, \tag{13}$$

$$\mathbf{F}_{T,\text{damping}} = C_t \mathbf{u}_t, \tag{14}$$

where C_n and C_t are the damping coefficients along $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$, respectively. For the Hooke's restitution law, C_n can be related to the familiar normal coefficient of restitution, ε_n, according to

$$C_n = -2\ln\varepsilon_n \sqrt{\frac{k_n\,\mu}{\pi^2 + (\ln\varepsilon_n)^2}} \tag{15}$$

(see Sect. 2.6 for the derivation), where μ is the reduced mass of the colliding pair ($\mu \equiv m_p m_n/(m_p + m_n)$, where m_p and m_n are the masses of the particle in question and its neighbor, respectively). There is no equivalent simple correspondence between C_t and the tangential coefficient of restitution sometimes used in HSDEM implementations (see Sect. 2.6 for further discussion).

Combining Eqs. (3)–(6), (13), and (14), the normal and tangential components of the total SSDEM force are

$$\mathbf{F}_N = -k_n x\hat{\mathbf{n}} + C_n\mathbf{u}_n, \tag{16}$$

$$\mathbf{F}_T = k_t\mathbf{S} + C_t\mathbf{u}_t. \tag{17}$$

By Newton's 3rd law, the neighbor particle feels the same total force in the opposite direction.

2.4 Static friction

Depending on the coefficient of static friction (μ_s) at the contact point, slippage may occur as a result of tangential stress. For real material, this is governed by the molecular arrangements around the point of contact. The coefficient of static friction in common use is a macroscopic approximation that estimates the total amount of tangential force that can be supported by the contact, with the assumption that this threshold of force scales linearly with the normal force at the contact. In our implementation, if this force is exceeded, depending on the value of a parameter, b, that ranges from zero to unity, $\mathbf{S}$ is reset to $b\mathbf{F}_{T,\max}$, where $\mathbf{F}_{T,\max}$ is this threshold tangential force, given by

$$|\mathbf{F}_{T,\max}| = \mu_s|\mathbf{F}_N|. \tag{18}$$

In the event of slipping, this allows us the option to set the tangential strain at the contact point to zero (the default is $b = 0$) or to some fraction ($b > 0$) of its maximum allowed value, $b\mathbf{F}_{T,\max}$. Additionally, if this damping force alone exceeds $\mathbf{F}_{T,\max}$, then $\mathbf{S}$ is reset to zero for any value of b. So using $b = 0$ (default), Eq. (17) now becomes

$$\mathbf{F}_T = \min\left\{\mu_s|\mathbf{F}_N|\hat{\mathbf{S}};\ k_t\mathbf{S} + C_t|\mathbf{u}_t|\hat{\mathbf{t}}\right\}, \tag{19}$$

where $\hat{\mathbf{S}} \equiv \mathbf{S}/|\mathbf{S}|$. When this tangential force is applied as torques to the particle and its neighbor (opposite sign), with lever arms of l_p and l_n, respectively, a change in rotation is induced in both particles. The changes in the rotations of both particles are along the same spin vector ($\hat{\mathbf{n}} \times \hat{\mathbf{t}}$) and of the same sign. To compensate for this gain in angular momentum, the COMs of both particles feel a tangential force equal and opposite to the forces at their respective surfaces. This correction, which serves to conserve angular momentum, is often neglected in SSDEM implementations.

2.5 Rolling and twisting friction

Rolling friction is often ignored in the modeling of granular materials. Particles are rolling if $|\mathbf{u}|$ is zero despite relative rotational motion of the particles. To account for the transformation of rotational energy of rolling particles into frictional energy (i.e., microscopic vibrations/heat), a coefficient of rolling friction (μ_r) is introduced in the code. When this quantity is non-zero, it decreases the relative velocity at the contact point that is due to rotation ($\mathbf{v}_{\text{rot}}$, defined below) by adding a spin vector that points in the opposite direction of this motion. The induced torque on the particle due to rotational friction is given by

$$\mathbf{M}_{\text{roll}} = \mu_r \frac{l_p(\mathbf{F}_N \times \mathbf{v}_{\text{rot}})}{|\mathbf{v}_{\text{rot}}|}, \tag{20}$$

where $\mathbf{v}_{\text{rot}} \equiv l_p(\boldsymbol{\omega}_p \times \hat{\mathbf{n}}) - l_n(\boldsymbol{\omega}_n \times \hat{\mathbf{n}})$.

There has been considerable debate in the materials science community regarding the optimal way to account for rolling friction, including whether or not it should depend on the speed of rotation [27–29]. Here we choose a simple implementation that depends only on the rolling friction coefficient, the normal force, and the sign of the "rolling axis" (which is given by the cross product of $\hat{\mathbf{n}}$ with $\mathbf{v}_{\text{rot}}$).

Twisting friction (dissipation of relative rotation of the particles around $\hat{\mathbf{n}}$, i.e., the normal axis that passes through the contact point and the particles' centers) is another kind of friction that is often neglected in granular material modeling. Similarly to the case of rolling, these relative rotational motions of the particles are coupled to each other and should damp out (the reason being that the contact "point" is in reality more like a contact area, where the components of the particles grind against each other and dissipate energy). To account for this effect, we include a twisting frictional term given by

$$\mathbf{M}_{\text{twist}} = -\mu_t r_c \frac{(\boldsymbol{\omega}_n - \boldsymbol{\omega}_p)\cdot\mathbf{F}_N}{(\boldsymbol{\omega}_n - \boldsymbol{\omega}_p)\cdot\hat{\mathbf{n}}}. \tag{21}$$

where r_c is the scalar distance from the contact point out to the circumference of the circle that describes the intersection of the two particles' surfaces.

There are some issues that arise with our treatment of rolling friction and twisting friction as we have defined them thus far. For example, consider two colliding particles with relative rotation at the contact point. For somewhat high-speed collisions, the normal force between particles will be large, making the damping forces given by Eqs. (20) and (21) accordingly large. In order to have these types of rotational damping take effect only when appropriate, we can set $\mathbf{M}_{\text{roll}}$ and $\mathbf{M}_{\text{twist}}$ to zero initially (by setting μ_r and μ_t to zero). This is done when the duration of the current overlap, t_{overlap}, is less than the characteristic duration of collision, τ_{overlap} (see Sects. 2.6 and 2.7); this means that particles that are actively bouncing do not experience rolling or twisting friction, whereas particles that are in persistent contact experience these frictional forces. It is worthwhile to note that the change in angular momentum of the particle in both the cases of rotational friction and of twisting friction is equal and opposite to the change in angular momentum of the neighbor, so total angular momentum is conserved.

Combining Eqns. (16), (19), (20), (21), and the conservation of angular momentum constraint, the total force on the

particle COM is given in Eq. (22) and the total torque on the particle is given in Eq. (23),

$$\mathbf{F}_p = -k_n x\hat{\mathbf{n}} + C_n\mathbf{u}_n + \min\left\{\mu_s|\mathbf{F}_N|\hat{\mathbf{S}};\ k_t\mathbf{S} + C_t|\mathbf{u}_t|\hat{\mathbf{t}}\right\}. \tag{22}$$

$$\mathbf{M}_p = -l_p\left[\min\left\{\mu_s|\mathbf{F}_N|\hat{\mathbf{S}};\ k_t\mathbf{S} + C_t|\mathbf{u}_t|\hat{\mathbf{t}}\right\} + \mu_r\frac{\mathbf{F}_N\times\mathbf{v}_{\text{rot}}}{|\mathbf{v}_{\text{rot}}|}\right]\times\hat{\mathbf{n}} - \mu_t r_c\frac{(\boldsymbol{\omega}_n-\boldsymbol{\omega}_p)\cdot\mathbf{F}_N}{(\boldsymbol{\omega}_n-\boldsymbol{\omega}_p)\cdot\hat{\mathbf{n}}}. \tag{23}$$

with corresponding expressions for the neighbor particle.

2.6 The coefficient of restitution

It is often useful to parameterize particle collisions with a (normal) coefficient of restitution, ε_n. The coefficient of restitution of a material depends not only on the restitution law used (e.g., Hooke's or Hertzian law), but also on the collisional speeds that we wish to consider. In order to derive a value for the normal damping coefficient C_n that corresponds directly to a given ε_n (assuming appropriate timesteps are used—cf. Sect. 2.7), we start with the definition of ε_n,

$$\varepsilon_n = \frac{|\mathbf{u}_n(t)_{\text{final}}|}{|\mathbf{u}_n(t)_{\text{initial}}|}, \tag{24}$$

and the solution to the second-order differential equation of motion for a spring attached to two non-fixed masses of reduced mass μ that obeys Hooke's law with simple damping,

$$x(t)\hat{\mathbf{n}} = x_{\max}\left[e^{-\alpha_n t}\cos(\omega_{1,n}t+\phi)\right]\hat{\mathbf{n}}, \tag{25}$$

where: $x_{\max}$ is the maximum amplitude, or distention, of the spring, and represents the maximum overlap between the two particles, which, physically, could in turn be taken to represent the maximum deviation (deformation) of the particles from perfect spheres; ϕ is the phase angle; $\alpha_n \equiv \frac{C_n}{2\mu}$; and $\omega_{1,n}$ is the damped harmonic frequency of the oscillating system (along $\hat{\mathbf{n}}$), which is given in terms of the undamped harmonic frequency, $\omega_{0,n}$, and α_n by

$$\omega_{1,n}^2 \equiv \omega_{0,n}^2 - \alpha_n^2, \tag{26}$$

where $\omega_{0,n}^2 \equiv \frac{k_n}{\mu}$. In order to solve for ε_n in these terms, we need to solve for the relative normal velocity, or at least the ratio of relative normal velocities before and after a given collision. To solve for $\mathbf{u}_n(t)$ in Eq. (27), we take the first derivative of x as given in Eq. (25), recalling that x and $x_{\max}$ lie along the normal.

$$\mathbf{u}_n(t) = -x_{\max}e^{-\alpha_n t}\left[\omega_{1,n}\sin(\omega_{1,n}t+\phi) + \alpha_n\cos(\omega_{1,n}t+\phi)\right]\hat{\mathbf{n}}. \tag{27}$$

We will take t just prior to the impact to be zero, which corresponds to a point in the phase where $|\mathbf{u}_n(t)|$ is maximum and $x(t)$ is zero, giving ϕ the value of $\pi/2$. The value of t just after the collision is equal to the time that it takes to complete one collision, which is $\frac{\pi}{\omega_{1,n}}$. In solving for ε_n, the cosine terms are zero at both $t=0$ and $t=\frac{\pi}{\omega_{1,n}}$, the remaining constants cancel, and the sine terms, offset by a half-phase, are equal and opposite, leaving simply

$$\varepsilon_n = \left|e^{\frac{-\alpha_n\pi}{\omega_{1,n}}}\right|. \tag{28}$$

Replacing the terms α_n and $\omega_{1,n}$ with C_n, k_n, and μ, then solving for C_n gives Eq. (15). Our result agrees with the formulation of C_n used in Cleary [26].

The formulation of C_t is somewhat more complex because it involves two different frequencies, $\omega_{1,n}$ and $\omega_{1,t}$, which can be independent of each other. Therefore, the sine and cosine arguments at $t=0$ and at $t=\frac{\pi}{\omega_{1,n}}$ will not be separated unconditionally by a half-phase, but instead by $\pi\frac{\omega_{1,t}}{\omega_{1,n}}$. There is also an additional factor to account for the relative tangential acceleration due to both frictionally induced COM motion and the corresponding rotation of the particle. This acceleration is given as

$$\mathbf{a}_t(\mathbf{S},\mathbf{u}_t,x) = \left(\frac{k_t}{\mu}\mathbf{S} + \frac{C_t}{\mu}\mathbf{u}_t\right)\left\{1 + \frac{5\left[l_p(x)\right]^2}{2s_p^2}\right\}. \tag{29}$$

To simplify things greatly, we will assume that $x \ll s_p$ (which is appropriate in the hard-sphere limit as l_p approaches s_p), and thus $\mathbf{a}_t(\mathbf{S},\mathbf{u}_t,x)$ becomes

$$\mathbf{a}_t(\mathbf{S},\mathbf{u}_t) = \frac{7}{2}\left(\frac{k_t}{\mu}\mathbf{S} + \frac{C_t}{\mu}\mathbf{u}_t\right) \tag{30}$$

$$= \omega_{0,t}^2\mathbf{S} + 2\alpha_t\mathbf{u}_t, \tag{31}$$

where $\omega_{0,t}^2 \equiv \frac{7k_t}{2\mu}$ and $\alpha_t \equiv \frac{7C_t}{4\mu}$. Note that this treatment of $\omega_{0,t}$ and α_t deviates from how we have defined $\omega_{0,n}$ and α_n: although the damped tangential frequency is still likewise defined as $\omega_{1,t}^2 \equiv \omega_{0,t}^2 - \alpha_t^2$, we have absorbed the factor of $\frac{7}{2}$ into the definitions of $\omega_{0,t}$ and α_t. The relative tangential velocity as a function of time, t, can now be expressed as Eq. (32),

$$\mathbf{u}_t(t) = -S_{\max}e^{-\alpha_t t}\left[\omega_{1,t}\sin(\omega_{1,t}t+\phi) + \alpha_t\cos(\omega_{1,t}t+\phi)\right]\hat{\mathbf{t}}_i. \tag{32}$$

$$\mathbf{u}_{t,\text{initial}} \equiv \mathbf{u}_t(0) = -S_{\max}\omega_{1,t}\hat{\mathbf{t}}_i. \tag{33}$$

$$\mathbf{u}_{t,\text{final}} \equiv \mathbf{u}_t\left(\frac{\pi}{\omega_{1,n}}\right) = -S_{\max}e^{\frac{-\alpha_t\pi}{\omega_{1,n}}}\left[\omega_{1,t}\cos\left(\pi\frac{\omega_{1,t}}{\omega_{1,n}}\right) - \alpha_t\sin\left(\pi\frac{\omega_{1,t}}{\omega_{1,n}}\right)\right]\hat{\mathbf{t}}_i. \tag{34}$$

$$\varepsilon_t \equiv \frac{\mathbf{u}_{t,\text{final}}\cdot\hat{\mathbf{t}}_i}{\mathbf{u}_{t,\text{initial}}\cdot\hat{\mathbf{t}}_i} = e^{\frac{-\alpha_t\pi}{\omega_{1,n}}}\left[\cos\left(\pi\frac{\omega_{1,t}}{\omega_{1,n}}\right) - \frac{\alpha_t}{\omega_{1,t}}\sin\left(\pi\frac{\omega_{1,t}}{\omega_{1,n}}\right)\right]. \tag{35}$$

where $S_{\max} \equiv \max(|\mathbf{S}|)$, which represents the amplitude of the oscillation, and $\hat{\mathbf{t}}_i$ is the direction of tangential motion of the neighbor particle at the point of contact at the start of the overlap. We still have t just prior to the impact equal to zero, and t just after the collision equal to $\frac{\pi}{\omega_{1,n}}$. After some simplification of terms we arrive at Eqs. (33) and (34) as expressions for the initial and final tangential velocities, respectively, which, by Eq. (24), gives the coefficient of tangential restitution in Eq. (35).

Buried within the expressions for α_t and $\omega_{1,t}$ we find the dependence of C_n on ε_t. We can discern some of the behavior of ε_t by examining different cases. When $\omega_{1,t} = \omega_{1,n}$, the tangential spring will have completed one half-cycle at the same time that the normal spring will have completed its half-cycle at the end of the collision, thus ε_t will be negative and have a magnitude equal to the exponential term, which represents the decay of the oscillation due to damping. In the case of $\omega_{1,t} \ll \omega_{1,n}$, the collision will have ended before the phase of the tangential oscillation has had time to evolve, and so ε_t will still be positive, and will have a magnitude equal to the exponential term, which will depend on how the value of α_t compares to $\omega_{1,n}$. So we can say that for $q\omega_{1,t} = \omega_{1,n}$, where q is a whole number, the value of ε_t will depend only on the exponential term, and when q is even, ε_t will be positive, and when q is odd, ε_t will be negative. At quarter phases ($\frac{q}{2}$), there is an additional term of $\pm\frac{\alpha_t}{\omega_{1,t}}$ when the magnitude of the cosine term is zero and the sine term is at a maximum.

From these examples, we can begin to understand how the quantities C_t, C_n, k_t, k_n, and μ affect ε_t, although a general analytical solution that gives C_t as a function of ε_n and these other quantities is not as simple as finding the appropriate C_n given the desired value of ε_n (see Eq. (15)). The appropriate way to a solution that gives C_t as a function of ε_t would be to use an iterative method, keeping in mind that the solution found would still be based on a hard-sphere approximation (cf. Eqs. (29) and (30)).

2.7 Timestep considerations

Correctly resolving the oscillation half-period of an isolated two-particle collision requires, at a very minimum, 10–20 timesteps over the course of the collision, and preferably close to around 50 [25]. From Sect. 2.6, and introducing the damping coefficient, $\xi \equiv \frac{\alpha_n}{\omega_{0,n}}$, we have:

$$\tau_{\text{overlap}} = \frac{\pi}{\omega_{1,n}}, \tag{36}$$

$$= \frac{\pi}{\omega_{0,n}\sqrt{1-\xi^2}}, \tag{37}$$

$$= \pi\sqrt{\frac{\mu}{k_n(1-\xi^2)}}. \tag{38}$$

This suggests a good timestep would be

$$h \approx \frac{\pi}{50}\sqrt{\frac{\mu}{k_n(1-\xi^2)}}. \tag{39}$$

Provided that the value of k_n has been chosen appropriately with consideration given to the velocities of particles in the simulation (cf. Sect. 2.1), and that an appropriate timestep is chosen with respect to this value of k_n (Eq. (39)), fast-moving particles will not be missed, and particle overlaps will be fully resolved.

3 Application: Comparison to cylindrical hopper experiments

Experiments of flow from cylindrical hoppers are ideal benchmark tests for numerical simulations, since such flows have been a matter of practical interest for some time (see Nedderman et al. [30] for background summary). Moreover, empirical relations between mutual parameters involved in these systems have been formulated and rigorously tested, and these provide stringent constraints for the validation of granular physics codes. Hence Bertrand et al. [31] suggest the use of the well-explored, fairly simple regime of a particle hopper as a test for collisional codes for which there is much experimental data in the literature. We thus follow this suggestion for our choice of validation experiments. Other tests could be considered, but given our particular interest to explore them in more detail in the context of planetary science applications, we elect to leave them for future dedicated studies (see Sect. 4 for some examples).

In this section, we briefly describe the empirical relations that we aim to reproduce, taken primarily from Nedderman et al. [30], explain our simulation setup, present our results, and then finally compare these results to the empirically derived, analytical relations.

3.1 Empirical findings

Beverloo et al. [32] found a correlation between the aperture size of a cylindrical hopper and the (mass) flow rate to the 2/5 power:

$$W^{2/5} \propto D + Z, \tag{40}$$

where W represents the discharge rate of the hopper in mass per unit time, D is the aperture diameter, and Z is the offset found in the correlation. The 5/2 dependence on the aperture diameter makes intuitive sense as the flux, W, should depend on the product of the aperture area, A, and the velocity normal, v. The area is proportional to D^2, and if we make the assumption that there exists a height from which the particles begin to free-fall above the outlet, and that this height is linearly proportional to D, then we get an additional dependence

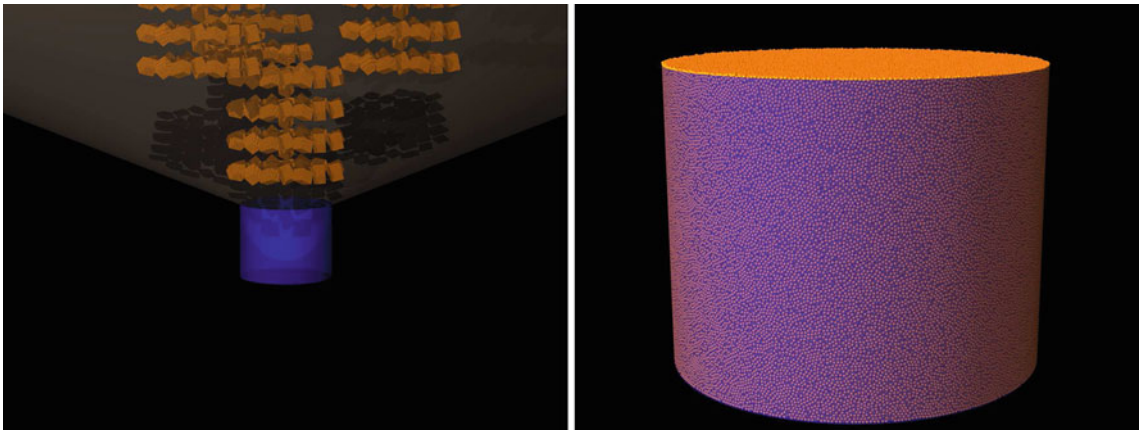

Fig. 2 Using numerical boundary conditions, hoppers are filled by first suspending randomly oriented rectangular arrays of particles in cubic-close-pack configurations over a funnel that deposits the particles (1 cm radius) into the initially empty hopper (*left*). After allowing particles to settle, the funnel, along with any excess particles above the hopper rim, are removed, leaving a hopper full of particles (*right*, zoomed image). This particular example used about 1.5 million particles and a hopper 2.4 m across. See Online Resource 1 for an animation of the hopper-filling process

of $\sqrt{D}$ from the downward speed of the particles as they exit through the outlet. Through dimensional analysis [30], the discharge rate also depends linearly on density and should go as the square-root of the acceleration due to gravity.

We will compare our results to the empirical relation

$$W = C\rho\sqrt{a_g}(D - kd)^{5/2}, \tag{41}$$

where ρ is the particle density, a_g is the acceleration due to gravity, D is the aperture diameter, and d is the particle diameter (for a monodisperse particle system), while C and k are unitless constants.

3.2 Our setup

To construct a hopper of radius R_{hopper}, we defined a cylindrical boundary of that radius and height H sufficiently large relative to the radius (i.e., large enough such that a constant discharge rate should be achieved in most cases), along with a confining bottom disk also of radius R_{hopper} (see the Appendix for an explanation of the types of boundary primitives that we use and how they are integrated into the code). In order to fill the hopper with particles, we placed a large funnel atop its rim such that the narrow end of the funnel had radius R_{hopper}. We then suspended randomly oriented rectangular arrays of monodisperse particles in cubic-close-pack configuration within the funnel portion of the cylinder (Fig. 2a). Next, we turned on gravity to fill the hopper, then removed the funnel and shaved off the particles that were left heaping up over the rim of the cylinder (Fig. 2b). We then used these initial conditions and placed circular holes of varying sizes into the centers of the bottom disk to measure properties such as the discharge rate, the velocity distribution, and the distribution of stresses on the particles within the hopper; discharge profiles for various hopper configurations are shown in Fig. 3 and are discussed further below. In addition to discharging the hopper with varying aperture sizes, we also discharged it using an array of densities and material properties (represented by parameters k_n, k_t, ε_n, C_t, μ_s, μ_r, ρ). We also performed some runs varying a_g and R_{hopper}. We allowed every discharge simulation to run until the hopper was nearly empty, so we were also able to test how discharge rate depends on the height of the particles remaining in the cylinder (there should be no dependence until the hopper is close to empty). In each case, we used 0.8 for the coefficient of restitution of all boundaries (walls), and a static friction coefficient identical to the particle-particle coefficient used in the given simulation.

In simulations we have the benefit of being able to capture directly the instantaneous state of the system throughout, including all of the positions and motions of each particle, and the forces acting on them, and seeing how the state of the system evolves. For example, we have the ability to trace the contact forces and construct a map of the force network, and then see how this evolves in time (cf. Fig. 4). We simulated a total of 61 hopper discharges using a range of physical and material parameters and gained insight on how these parameters affect the rate and quality of the discharge (see Table 1). Some of the more important dependencies are discussed below.

3.3 Results

For our primary task, we set out to explore the 5/2 dependence of the flow rate on $(D - kd)$ to ensure that the code

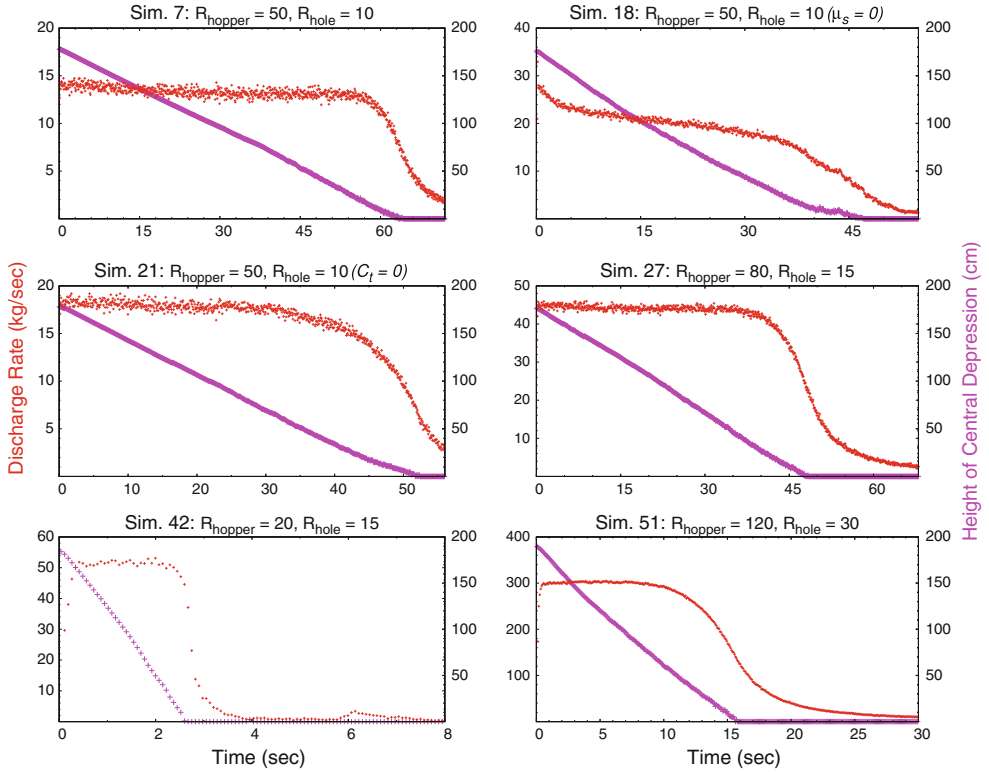

Fig. 3 Discharge rate in kg s^{-1} (*red/small crosses*) and height of particles above the center of the hole (*magenta/larger crosses*) vs. time for six different hopper discharge simulations (see Table 1 for a list of parameters for each simulation). Sims. 7 (*upper-left*), 27 (*middle-right*), 42 (*lower-left*), and 51 (*lower-right*) have the same material parameters and differ only by the widths of the hoppers and the widths of their apertures. The radii of these hoppers are 50, 80, 20, and 120 cm, respectively; their aperture radii are 10, 15, 30, and 15 cm, respectively. Sims. 18 and 21 are identical in both hopper size and aperture size to Sim. 7, but each has a single material parameter that differs: Sim. 18 has $\mu_s = 0$ and Sim. 21 has $C_t = 0$. In both of these two cases, the height of the particles in the hopper was not enough to produce a steady flow rate (although Sim. 21 showed a flow rate that was flatter and more sustained than that of Sim. 18). Sims. 27 and 42 differ only in hopper radius—their aperture radii are identical, which shows that the Beverloo et al. correlation [32] fails as the radius of the aperture approaches the radius of the hopper, as they predict. Note that it takes only a few seconds for essentially all of the particles in Sim. 42 to drain from the hopper. (Color figure online)

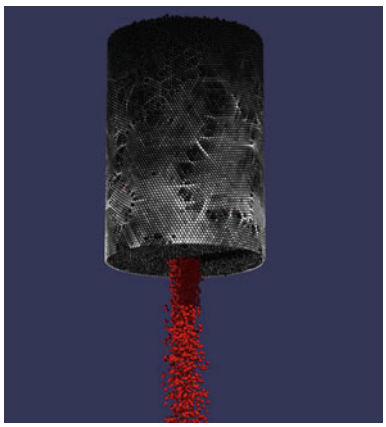

Fig. 4 Snapshot of Sim. 24 (see Table 1 for simulation parameters) showing the network of normal force distribution on particles at the hopper wall after 12.2 s of discharge. Particles under maximum stress are shown in white; darker particles feel less stress; red particles (mostly flowing out the bottom) feel no stress. See Online Resource 2 for an animation of the discharge of Sim. 24 (the duration of the animation reflects the simulated duration of the discharge). (Color figure online)

is able to reproduce this satisfactorily. We also explored the dependence on ρ, a_g, H, the height of the particles in the hopper, R_{hopper}, and the material parameters of the particles. Here we describe just a few of these relations—see Table 1 for estimates of the steady-state flow rate for all 61 simulations. We are able to estimate steady-state flow rates because the flow is independent of the height of the particles remaining inside the hopper. It was demonstrated (e.g., by Janssen [33], Shaxby and Evans [34]) that the walls of the hopper itself bear the majority of the weight of the particles and that the stress near the aperture is largely unaffected by the height of the material in the hopper until it gets below a certain level (e.g., see Fig. 3). Rose and Tanaka [35] argue that the flow rate stays constant even below this level, until the height of the material in the hopper becomes comparable to the size of the aperture, but we find that this occurs earlier.

Figure 5 shows the flow rate as a function of D for different sizes of the hopper drainage aperture. Using least-squares

Table 1 Complete list of simulation parameters and corresponding steady-state discharge rates that were achieved

Sim. no.	Discharge rate (10^4 g s^{-1})	Hopper radius (cm)	Hole radius (cm)	Particle density (g cm^{-3})	Time-step (10^{-6} s)	k_n (kg s^{-2})	k_t (kg s^{-2})	ϵ_n	$C_t/\sqrt{k_n}$ (kg s^{-1})	μ_s	μ_r, μ_t	a_g (g)
1	1.421	20	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
2	1.357	25	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
3	1.335	30	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
4	1.333	35	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
5	1.313	40	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
6	1.306	45	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
7	1.310	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
8	1.312	55	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
9	1.315	60	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
10	1.316	65	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
11	1.314	70	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
12	1.318	75	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
13	1.319	80	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
14	–[a]	50	10	1	12	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
15	1.304[a]	50	10	1	6	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
16	1.314	50	10	1	1.5	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
17	1.313	50	10	1	0.75	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
18	2.064[a]	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.0	0.0	1
19	1.281[a]	50	10	1	3	8×10^4	8×10^4	0.8	0.0176	0.2	0.0	1
20	1.303	50	10	1	1.5	3.2×10^5	9.14×10^4	0.8	0.0176	0.2	0.0	1
21	1.785[a]	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0	0.2	0.0	1
22	1.330[a]	50	10	1	3	8×10^4	2.29×10^4	0.2	0.0176	0.2	0.0	1
23	1.326	50	10	1	3	8×10^4	2.29×10^4	0.5	0.0176	0.2	0.0	1
24	1.274	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0765	0.2	0.0	1
25	1.356	50	10	1	3	8×10^4	2.29×10^4	0.2	0.0765	0.2	0.0	1
26	1.146	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.5	1
27	4.400	80	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
28	10.06	80	20	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
29	18.86	80	25	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
30	3.047	80	30	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
31	4.396	75	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
32	4.400	70	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
33	4.398	65	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
34	4.387	60	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
35	4.385	55	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
36	4.382	50	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
37	4.370[a]	45	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
38	4.352[a]	40	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
39	4.540[a]	35	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
40	4.582[a]	30	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
41	4.828[a]	25	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
42	5.176[a]	20	15	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
43	1.164	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.5	0.0	1
44	1.124	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.8	0.0	1
45	1.342	50	10	4	1.5	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1

Table 1 continued

Sim. no.	Discharge rate (10^4 g s^{-1})	Hopper radius (cm)	Hole radius (cm)	Particle density (g cm^{-3})	Time-step (10^{-6} s)	k_n (kg s^{-2})	k_t (kg s^{-2})	ϵ_n	$C_t/\sqrt{k_n}$ (kg s^{-1})	μ_s	μ_r, μ_t	a_g (g)
46	1.338	50	10	4	0.75	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
47	1.367	50	10	10	0.75	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
48	1.326	50	10	2	1.5	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
49	1.359	50	10	7	0.75	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
50	1.395	50	10	15	0.75	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
51	30.29	120	30	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
52	1.862	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
53	46.10	120	35	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	2
54	18.25	120	25	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
55	1.331	50	10	1	1.5	3.2×10^5	2.29×10^4	0.8	0.0176	0.2	0.0	1
56	1.374	50	10	1	3	8×10^4	2.29×10^3	0.8	0.0176	0.2	0.0	1
57	2.325	50	10	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	3
58	2.767	50	10	1	1.5	3.2×10^5	2.29×10^4	0.8	0.0176	0.2	0.0	4
59	66.13	120	40	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
60	90.27	120	45	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1
61	119.2	120	50	1	3	8×10^4	2.29×10^4	0.8	0.0176	0.2	0.0	1

The last column gives the uniform downward acceleration due to gravity that was used in each simulation, in units of Earth's gravity. (See text for a definition of the other parameters indicated in the table)

[a] A steady-state discharge rate was never achieved although an estimate may be shown

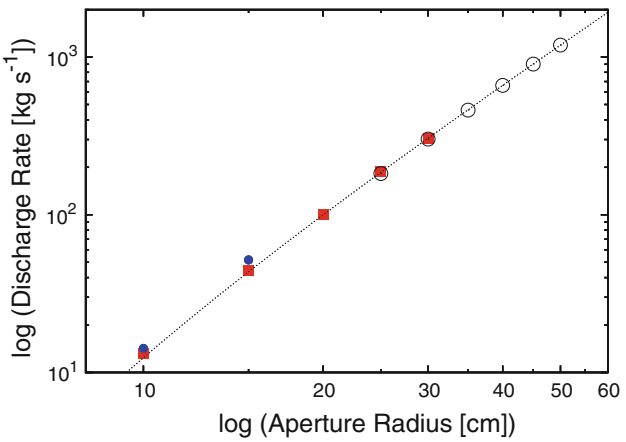

Fig. 5 Discharge rate as a function of hole size (log-log scale). The *filled* (*red*) *squares* represent simulations from hoppers of radius 80 cm (Sims. 13, 27–30), the *open* (*grey*) *circles* represent simulations from hoppers of radius 120 cm (Sims. 51, 53, 54, 59–61), and the *filled* (*blue*) *circles* represent simulations from hoppers of radius 20 cm (Sims. 1, 42). The *dashed line* is a least-squares fit for the material constants C and k in the function represented by Eq. (41). The rate of flow conforms well to the Beverloo et al. correlation ([32], Eq. (40)). Note the apertures for the 20-cm hoppers (*filled/blue circles*) are very large relative to the hoppers, resulting in the observed deviation from the empirical model. (Color figure online)

minimization, we fit a function in the form of Eq. (41) using the flow rates derived from simulations (Sims.) 7, 27–30, 53, and 59–61, solving for the constants C and k (we find 0.697 ± 0.003 and 2.32 ± 0.07, respectively), with ρ, g, d, and all other material constants held fixed (see Table 1 for the values of these parameters). The nine simulations chosen for this fit use hopper radii of 80 cm with aperture radii of 10, 15, 20, 25, and 30 cm and hopper radii of 120 cm with aperture radii of 35, 40, 45, and 50 cm, taking care to ensure that all hoppers were wide enough such that increasing the width further had no effect on flow rate (e.g., in Fig. 6 we compare discharge rates through apertures of 10 cm and 15 cm, varying the sizes of the hoppers to find a range of hopper sizes where the flow rates are independent of the hopper size). The slope derived from the simulated flow rates matches well that of the empirical relation so long as the width of the hopper is large enough relative to the size of the hole (the flow rates are too high from narrow hoppers of 20 cm radius, especially for Sim. 42 where the hole size is 15 cm, as can be inferred from Figs. 3, 5, 6).

The flow rate should be linear with respect to the bulk density of particles ($\phi\rho$, as we define it in Sect. 2.1)—that is, varying the value of $\phi\rho$ should have no effect on the *number* of particles that are discharged over a period of time, as is shown in Fig. 7, which relates the particle flow rate to individual particle density (ρ) and to initial bulk density ($\phi\rho$) for equal-sized particles. We would also expect the number of particles discharged per unit time to be independent of individual particle density ρ, however, since we use the same value for the stiffness parameter k_n in all those simulations,

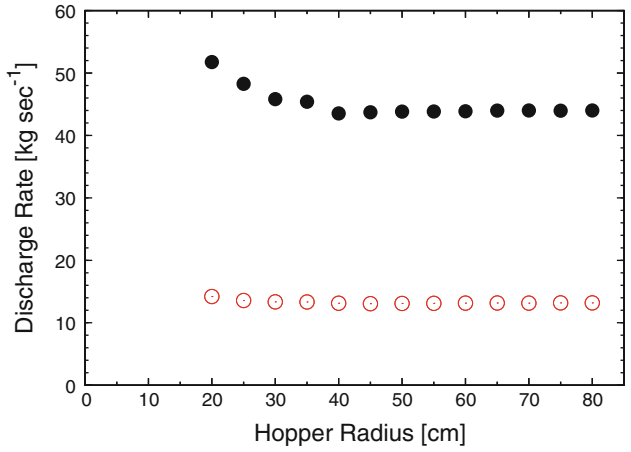

Fig. 6 Discharge rate as a function of hopper radius. The *open* (*red*) *circles* represent simulations that have aperture radii of 10 cm (Sims. 1–13), and the *filled* (*black*) *circles* represent simulations that have aperture radii of 15 cm (Sims. 27, 31–42). It can be seen that the rate of flow is largely independent of the radius of the hopper provided that it is large enough with respect to the radius of the opening at the bottom of the hopper. Note, however, that estimates of steady-state discharge rates can be less reliable for simulations with hole sizes approaching the sizes of the hopper (see Sim. 42 in Fig. 3 for the most extreme case that we simulated). (Color figure online)

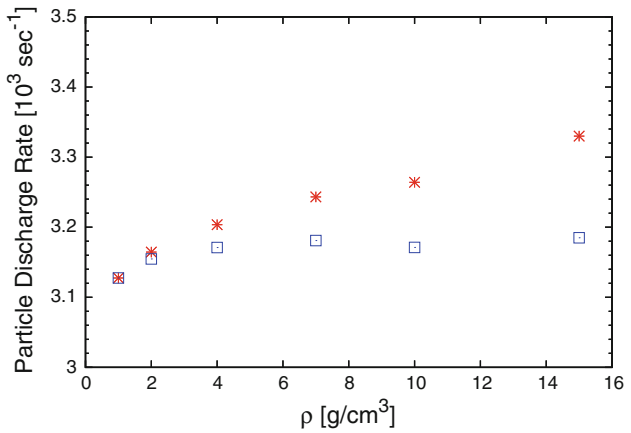

Fig. 7 Discharge rate (in number) vs. particle density. (*Red*) *asterisks* show the discharge rate at different mass densities. (*Blue*) *squares* show this discharge rate divided by the initial porosity of the material inside the hopper at that density. Notice that the slope is close to constant if we consider bulk density (*blue*/*squares*), in agreement with Beverloo et al. [32], whereas the rate increases if we consider the density of individual particles (*red*/*asterisks*) due to increased compaction at higher densities. (Color figure online)

the hoppers that are filled with more dense particles have material that is more compacted than those that are filled with less dense particles. This means that the hoppers that are filled with more dense particles will have an increased (mass) flow rate roughly in proportion to the increase in bulk density of the material inside the hopper, which explains what we see in Fig. 7.

Several simulations were carried out with different k_n and k_t. A greater value of k_n should result in a slightly slower discharge rate for a similar reason that greater particle mass density shows a slightly faster discharge rate: the bulk density $\phi\rho$, increases with either a decrease in k_n or an increase in particle density. However, since the material that we use is already quite stiff, using a larger value of k_n does not significantly decrease the degree of overlap and thus has little effect on $\phi\rho$. In fact, comparing the differences in flow rate between Sims. 7 and 55 in Table 1, we may be seeing slightly faster flow at higher k_n, although the difference is very small and potentially not significant. However, in light of the discussion at the end of Sect. 2.1 about using a "softened" k_n to speed up certain simulations, it will be useful to know the full effects, even subtle ones, that come with using different values of k_n. We might speculate that the higher k_n could be leading to an effective decrease in tangential friction since the time that some particle pairs are in contact is shortened, and that this effect is greater than the opposing effect of having slightly greater packing. In fact, Sims. 7 and 20 have the same 7/2 ratio of k_n/k_t, and may imply a decrease in flow rate at higher stiffness (this ratio is a natural choice: it comes out of Eq. (30) as the stiffness ratio needed to keep normal and tangential oscillation frequencies equal in the hard-sphere limit ignoring the effect of damping on frequency). Increasing k_t alone (Sim. 7 vs. Sim. 19) appears to impede the flow rate. Although these simulations allow us to see some trends, a much more complete study of these parameters would be necessary to draw any firm conclusions.

The flow rate correlates fairly well with the square root of the gravitational acceleration a_g, which agrees with the result given by dimensional analysis [30]. We fit to the equation $W = \beta_1 a_g^{0.5}$ (the dotted/magenta line in Fig. 8), solving for β_1, and to $W = \beta_2 a_g^{\gamma}$ (the dashed/blue line), solving for β_2 and γ. The reduced χ^2 is lower by a factor of three when using $\gamma = 0.55$ compared with $\gamma = 0.5$. A higher value of γ was reported (0.6) experimentally by Hofmeister et al. [36] using a quasi-2D hourglass setup. We could speculate that net tangential frictional effects may depend on the strength of the gravitational field in real-world experiments, and then reflected in our simulations. This will certainly be an important area to explore in light of the anticipated applications of our code into different gravitational environments.

We also investigated the influence of static friction μ_s on the flow rate. Sims. 7, 18, 43, and 44 were made using a range of values of μ_s from 0 to 0.8 (see Table 1). We find that the flow rate decreases with increasing static friction (see Fig. 3). This finding, along with the dependence on k_t, seems to contradict the experimental findings made by Beverloo et al. [32], which indicate that the flow rate is independent of all material properties other than shape. However, this may only be true within the narrow range of static friction values that can be explored easily experimentally. Over the wider range of values that our computer simu-

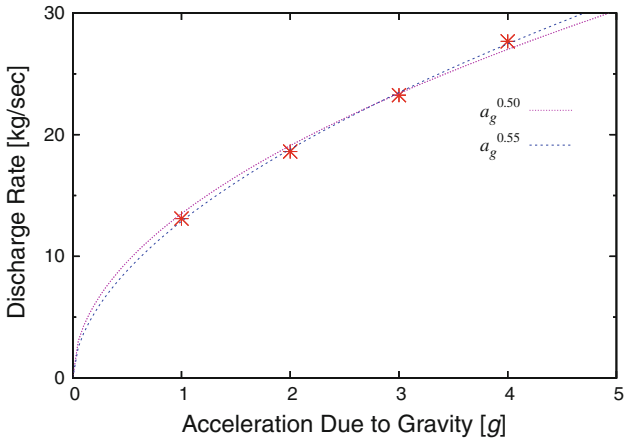

Fig. 8 Discharge rate as a function of the acceleration due to gravity, a_g. The (*red*) *asterisks* are data from the simulations; the (*magenta*) *dotted line* gives a $a_g^{0.5}$ dependence; the (*blue*) *dashed line* is the best fit to the data, $a_g^{0.55\pm0.02}$, found by minimizing χ^2. Note that the rate found at $4g$ used material with a higher k_n, which may affect the flow rate; however, at these values of k_n, the effect on flow rate is small (see discussion in Sect. 3.3). Data points come from Sims. 7, 53, 57, and 58. (Color figure online)

lations can investigate, we find that the experimental conclusion cannot be generalized, and that some values of the static friction can influence the flow rate. This is an inherent advantage of computer simulations over real-world experiments: the ability to explore a wide, and sometimes experimentally unreachable, parameter space. We also find that it takes longer (if even possible) to achieve a constant flow rate with μ_s set to zero (see Fig. 3). Similarly, setting the tangential damping parameter C_t to zero (Sim. 21 in Fig. 3) increases the flow rate and the time needed to achieve a steady flow. Moreover, Sims. 21, 7 and 24 were performed with $(C_t/\sqrt{k_n})$ equal to 0, 0.0176, and 0.0765 kg s^{-1}, respectively (see Table 1). They show that an increase in C_t may weakly inhibit flow, but the parameter significantly affects the packing and distribution of stresses, especially near the silo walls.

The influence of the parameter ϵ_n was also investigated. Sims. 22, 23, and 7 were performed using ϵ_n equal to 0.2, 0.5, and 0.8, respectively, with $(C_t/\sqrt{k_n}) = 0.0176\,\mathrm{kg\,s^{-1}}$ for each. They show that the flow rate has essentially no dependence on ϵ_n. However, a comparison of Sims. 25 and 24, which were performed with ϵ_n equal to 0.2 and 0.8, respectively, but with a higher value (0.0765 kg s^{-1}) of $(C_t/\sqrt{k_n})$, shows that there could be a greater influence of ϵ_n on the flow rate at high values of C_t, but this inference would need to be investigated further.

A simulation (Sim. 26) was also performed using non-zero values of μ_r and μ_t, both set equal to 0.5. It shows a decrease in flow rate, as expected for such a high value of these friction parameters.

Finally, we checked the sensitivity of the results on the timestep. As is shown by comparing the steady-state discharge rates of Sims. 7 and 14–17, no significant change in discharge rate was found when using smaller timesteps, but otherwise identical parameters, indicating that our choice of timestep (3 μs) is a reasonable one.

4 Conclusions and perspectives

We have implemented the soft-sphere discrete element method (SSDEM) in the N-body code PKDGRAV. SSDEM allows for the realistic modeling of contact forces between particles in granular material. To account for surface deformation of particles at contact, colliding particles are allowed to overlap, during which time they are subject to forces that work to oppose deformation, and which depend on the relative spins and velocities of the particles, their material properties, and the history of the contact. We take different frictional forces into account, including rolling and twisting friction, which are often neglected in SSDEM implementations. Moreover, the computation time is optimized thanks to the sophisticated parallelization and tree-code algorithms that are part of the PKDGRAV functionality, which allows all instances of particle overlap to be found in an efficient manner.

We performed a validation test for our numerical code and SSDEM implementation by reproducing successfully the dynamics of granular material flows in cylindrical hopper experiments. A series of empirical relations between mutual parameters involved in these systems have been formulated and rigorously tested by experimental studies, which allows us to test whether our granular physics code gives results that are consistent with those relations. The ability of our numerical code to consider wall boundaries with a wide range of geometries allows us to simulate with great precision the setup of the experiments, in particular the design of the cylindrical hopper. Using the same types of setups that were used in the experiments, we find that the empirical relations that describe the experimental outcomes can also be used to describe the outcomes of the simulations. Moreover, in simulations, we have the benefit of being able to track the instantaneous state of the system throughout, and seeing how the state evolves, something that cannot, in general, be done experimentally. In particular, we can trace the contact forces and construct a map of the force network, and see how this network evolves in time. In other words, we can investigate the dynamics of the system in great detail as it evolves, and better understand this dynamical evolution. Furthermore, we can determine the sensitivity to those parameters salient to the dynamics of the system. We performed 61 simulations of hopper discharges, covering a wide range in parameter space. In addition to matching experimental outcomes in most cases,

we find for instance that over a range of values of the static friction going from 0 to 0.8, the flow rate increases with decreasing static friction, while such an influence on a much narrower range of values was not identified experimentally, which led to the apparently incorrect conclusion that the flow rate is independent of all material properties other than shape. The influence of other parameters (such as the normal and tangential coefficients of the spring constant used to model the particle's deformation at contact, along with their respective viscous damping terms, and the acceleration due to gravity) on the flow rate was also explored.

Further comparisons to experiments will be performed, such as flows in a tumbler, avalanches, and other phenomena that will test the ability of our code to reproduce the behavior of granular materials in a wide range of contexts. Our ultimate goal is to be able to legitimately apply our method to planetary science studies (e.g., low-speed impacts on regolith, regolith evolution on solid celestial bodies' surfaces, etc.). We will then be able to provide important interpretation of images obtained by spacecraft of planetary and small bodies' surfaces, and to aid in the design of devices that will interact with extraterrestrial surfaces (anchors, sample collectors, and so on). Such devices are to be aboard sample-return missions to asteroids (e.g., the mission OSIRIS-Rex that will be launched by NASA in 2016, and MarcoPolo-R in selection phase at ESA). Also, since the shapes of the grains, as well as the cohesion between them, can have a great influence on their dynamics, one of the next steps will be to account for shape effects and to include cohesive forces in our numerical tool.

Acknowledgments This material is based on work supported by the National Aeronautics and Space Administration under Grant No. NNX08AM39G issued through the Office of Space Science and by the National Science Foundation under Grant No. AST0524875. Part of the work was performed by S.R.S. in France, supported by the Chateaubriand 2011 fellowship from the Embassy of France in the United States. P.M. acknowledges support from the french Programme de Planétologie. This study was performed as part of the International Team collaboration number 202 sponsored by the International Space Science Institute (ISSI) in Bern, Switzerland. Some simulations were performed on the `yorp` cluster administered by the Center for Theory and Computation of the Department of Astronomy at the University of Maryland in College Park, the `deepthought` cluster administered by the Office of Information Technology at the University of Maryland in College Park, and on the computer cluster `SIGAMM` at the Côte d'Azur Observatory in Nice (France). We also wish to extend our appreciation to participants at the 2011 Interdisciplinary Summer School on Granular Flows held at the University of Maryland for helpful discussions.

Appendix: Walls

Walls are used in PKDGRAV to provide hard-surface boundary conditions for granular dynamics simulations. Richardson et al. [3] describe the geometries and collision conditions used in their HSDEM simulations, for which collisions are predicted prior to contact, requiring often complex equations to be solved repeatedly. Here we provide the solutions for the same geometries but using SSDEM, with the principal advantage that overlaps are detected after the fact, and only once per timestep, greatly simplifying the detection algorithms and reducing the computational cost. Other geometries, such as the triangle, are currently being implemented into the code, but further testing is still required.

Briefly, during the force calculation of the integration step, every particle in an SSDEM simulation is checked to see whether it overlaps with another particle and/or wall. Corresponding SSDEM forces are applied that depend on the degree of overlap and that are directed along a line between the objects, which in turn depends on the overlap geometry. Particle-particle overlaps are simple to detect as they are just sphere intersect tests (cf. Eq. (2)). For walls, each supported geometry is handled separately, as detailed in the following.

To reduce the cost of wall-intersect tests, we first isolate regions of space that the wall does not occupy, progressively confining the wall until the point of closest contact is found. For example, when checking for an intersection with a finite planar wall, like a rectangle or a disk, it is usually of benefit to first consider the intersection with the wall as though it were simply the infinite plane that contains the finite wall. In this way, particles that are far above or below the plane can be ruled out without performing more computationally expensive wall-intersect checks. The regions were chosen with both simplicity and efficiency in mind (if regions of space where particles are likely to be found can be carved out with relatively few operations, this will save computational time).

Each particle is checked against each wall to see if an overlap exists. If it does, the point of closest contact is found and used to compute the forces on the particle. Restoring and frictional forces are applied to particles in contact with walls just as they are applied to particles in contact with neighbors as outlined in Sect. 2, but using a contact point on the wall surface (the walls have infinite linear and angular inertia and are not deformable). The contact point has a total relative velocity (Eq. (8)) given as the difference between the velocity of the wall at the contact point (taking into account its COM motion as well as any spin or oscillatory motion) and the velocity of the particle at the contact point (taking into account both its COM motion and its spin).

The following describes the primary set of boundary primitives, which can be combined in order to confine particles within certain geometries or to replicate specific mechanical devices.

Infinite plane

Starting from Richardson et al. [3], the parameters for the infinite plane are the origin $\mathbf{O}$ and normal $\hat{\mathbf{N}}$, plus optional

velocity $\mathbf{V}$, oscillation amplitude A, oscillation angular frequency Ω, oscillation normal vector $\hat{\mathbf{\Omega}}$ (so the relative vector displacement after time t due to oscillation, measured from the start of the simulation and evaluated at the start of the step, is $A\sin(\Omega t)\hat{\mathbf{\Omega}}$), and spin Σ around the orientation vector $\hat{\mathbf{N}}$. Note that for an infinite plane with $\Sigma = 0$, the origin can be any point in the plane (the choice is arbitrary).

To simplify the equations in this and subsequent derivations, we define the relative position vector $\boldsymbol{\rho} \equiv \mathbf{r}_p - \mathbf{O}$ and separate it into perpendicular and parallel components, $\boldsymbol{\rho}_N$ and $\boldsymbol{\rho}_T$, respectively, where $\boldsymbol{\rho}_N \equiv \rho_N \hat{\mathbf{N}}$, $\rho_N \equiv \boldsymbol{\rho} \cdot \hat{\mathbf{N}}$, and $\boldsymbol{\rho}_T \equiv \boldsymbol{\rho} - \boldsymbol{\rho}_N$ (so $\hat{\mathbf{T}} \equiv \boldsymbol{\rho}_T / |\boldsymbol{\rho}_T|$, which is only defined if $|\boldsymbol{\rho}_T| > 0$, and $\rho_T \equiv \boldsymbol{\rho} \cdot \hat{\mathbf{T}}$). Note, in Sect. 2, $\boldsymbol{\rho}$ was defined as the relative position between particle centers. Also note that $\boldsymbol{\rho}_N$ and $\boldsymbol{\rho}_T$ as defined here are *signed* quantities, i.e., vector components.

We similarly define the relative velocity as $\boldsymbol{\nu} = \mathbf{v}_p - \mathbf{V} - A\Omega\cos(\Omega t)\hat{\mathbf{\Omega}}$, with corresponding perpendicular and parallel components. If there is an overlap with a particle, then the total relative velocity between the particle and the wall is given as

$$\mathbf{u} = -\boldsymbol{\nu} + \Sigma\hat{\mathbf{N}} \times \boldsymbol{\rho}_T - l_p(\hat{\mathbf{n}} \times \boldsymbol{\omega}_p). \tag{A.1}$$

Also, rolling and twisting frictional terms arising from particle-wall contacts are calculated by following the methodology laid out in Sect. 2.5 and substituting the spin vector of the neighbor particle $\boldsymbol{\omega}_n$ with $\Sigma\hat{\mathbf{N}}$, and l_n with $\mathbf{Q} \cdot \hat{\mathbf{T}}$, where $\mathbf{Q}$ signifies the contact point between the particle and the wall.

In the specific case of the infinite plane, the overlap condition is simply $|\rho_N| < s$, where s is the radius of the particle. If this condition is met, the contact point is given by $\mathbf{Q} \equiv \mathbf{O} + \boldsymbol{\rho}_T$. As we do in the case of particle-particle contacts, we resolve the contact as though it were occurring at a single point (in the case of particle-particle contact, the contact point is taken to lie along the line that connects the centers of the two spheres; in the case of particle-wall contact, the contact point is taken to be the point on the boundary primitive closest to the particle center). In reality, contacts occur over areas, or within small volumes on the molecular scale, and give rise to a complex distribution of forces, to which we provide a rough approximation by including an array of frictional forces (cf. Sect. 2).

Disk

For a disk, we first isolate the test region to the corresponding infinite plane (all parameters in an infinite plane are included in the set of parameters of a disk), but now $\mathbf{O}$ defines the geometric center of the disk, and R_{out} and R_{in} define its outer and inner radii, respectively, where $R_{\text{out}} > R_{\text{in}}$ (i.e., the disk can have a central hole). For any particles that survive the first cut (i.e., those particles that would be in contact with the disk if it were infinite), $|\boldsymbol{\rho}_T|$ is compared against R_{out} and R_{in}. Three cases are considered: (A) $R_{\text{in}} \le |\rho_T| \le R_{\text{out}}$; (B) $R_{\text{in}} < R_{\text{out}} < |\rho_T|$; and (C) $|\rho_T| < R_{\text{in}} < R_{\text{out}}$. If case (A) is true, the particle is touching/overlapping the flat portion of the disk. If (B), the particle may be touching/overlapping the outer periphery of the disk; the potential overlap point is $\mathbf{O} + R_{\text{out}}\hat{\mathbf{T}}$, and the overlap condition is $|\boldsymbol{\rho} - R_{\text{out}}\hat{\mathbf{T}}| \le s$. If (C), the inner edge is the potential overlap point, given by $\mathbf{O} + R_{\text{in}}\hat{\mathbf{T}}$, with overlap condition $|\boldsymbol{\rho} - R_{\text{in}}\hat{\mathbf{T}}| \le s$.

A special case arises if $R_{\text{in}} > 0$ and $\rho_T = 0$. This is a subcase of (C) where the particle center lies on the disk orientation axis, above, below, or on $\mathbf{O}$. Here the overlap condition is $\rho^2 - R_{\text{in}}^2 \le s^2$. If the condition is satisfied, then the contact point $\mathbf{Q}$ is set to a "phantom" point at $\mathbf{O} - R_{\text{in}}\boldsymbol{\rho}/|\boldsymbol{\rho}|$. If $|\boldsymbol{\rho}| = 0$, meaning that the particle center is exactly at the origin, no net force is felt from the disk.

Rectangle

For a rectangle, like the disk, we first consider the region that corresponds to the infinite plane containing the rectangle. The four vertices of the rectangle are defined by three vectors, the origin $\mathbf{O}$, the vector Υ_1 that points from $\mathbf{O}$ to an adjacent corner, and the vector Υ_2 that points from $\mathbf{O}$ to the other adjacent corner. For simplicity, we require that Υ_1 and Υ_2 be orthogonal. Thus the four corners of the rectangle are $\mathbf{O}$, $\mathbf{O} + \Upsilon_1$, $\mathbf{O} + \Upsilon_2$, and $\mathbf{O} + \Upsilon_1 + \Upsilon_2$. Note the normal of the rectangle (used to define the infinite plane in which it lies) is just $\hat{\mathbf{N}} = (\Upsilon_1 \times \Upsilon_2)/|\Upsilon_1 \times \Upsilon_2|$. A necessary condition for the particle to be in contact with the rectangle is that the particle be in contact with the infinite plane containing it, i.e., $|\rho_N| < s$ (see above).

If the particle passes the plane-intersect test, we next need to find the point on the plane closest to the particle. If that point is closest to a point on the rectangle's face, then the particle is in overlap with the wall. If not, we must check to see if the particle is in contact with an edge or corner of the wall. To do this, we construct a coordinate system defined by unit vectors $\hat{\mathbf{a}} \equiv \Upsilon_1/|\Upsilon_1|$ and $\hat{\mathbf{b}} \equiv \Upsilon_2/|\Upsilon_2|$, with points (0,0), (1,0), (0,1), and (1,1) corresponding to the four corners of the rectangle. Note that all points on the infinite plane in which the rectangle lies can be described by real values (a, b).

In order to check for an overlap of the particle with the rectangle, the coordinate space defined above is divided into nine test regions. This is done by drawing four (infinite) lines: $(0, 0) + m\hat{\mathbf{a}}$; $(0, 1) + m\hat{\mathbf{a}}$; $(0, 0) + m\hat{\mathbf{b}}$; and $(1, 0) + m\hat{\mathbf{b}}$. The resulting nine regions are outlined in Table 2.

$\boldsymbol{\rho}_T$ can be transformed into this frame, and will be subject to a different overlap test depending on its coordinates (a, b). Case (A) describes a particle that is necessarily in overlap with the face of the rectangle, and $\mathbf{Q} = \mathbf{O} + \boldsymbol{\rho}_T$, or (a, b) in this frame. For the remaining cases, the potential overlap point, i.e., the point (x, y) on the rectangle closest to

Table 2 Overlap cases for a particle with a rectangle

Face	(A) $0 \le a \le 1, 0 \le b \le 1$			
Edge	(B) $a < 0, 0 \le b \le 1$	(C) $a > 1, 0 \le b \le 1$	(D) $0 \le a \le 1, b < 0$	(E) $0 \le a \le 1, b > 1$
Corner	(F) $a < 0, b < 0$	(G) $a < 0, b > 1$	(H) $a > 1, b < 0$	(I) $a > 1, b > 1$

Top row—the case where the particle is flush against a face of the rectangle; middle row—cases where the particle is closest to an edge of the rectangle; bottom row—cases where the particle is closest to a corner

the particle, is given for each case as: (B) $(0, b)$; (C) $(1, b)$; (D) $(a, 0)$; (E) $(a, 1)$; (F) $(0, 0)$; (G) $(0, 1)$; (H) $(1, 0)$; (I) $(1, 1)$. The overlap condition is $|\boldsymbol{\rho} - x\Upsilon_1 - y\Upsilon_2| \le s$. If the condition is met, $\mathbf{Q} = \mathbf{O} + x\Upsilon_1 + y\Upsilon_2$.

Infinite cylinder

The infinite cylinder is an infinitesimally thin and infinitely long, hollow circular shaft defined by taking $\mathbf{O}$ to be any point along the cylinder axis, $\hat{\mathbf{N}}$ as the orientation of the axis, and R as the radius of the cylinder. The overlap test is $\max\{R - s; 0\} \le \rho_T \le R + s$. If the condition is met, the contact point $\mathbf{Q} = \mathbf{O} + \boldsymbol{\rho}_N + R\hat{\mathbf{T}}$. If $\rho_T = 0$, which corresponds to the case where the particle is centered exactly on the cylinder axis, no force is felt by the particle, even if $s > R$.

Finite cylinder

The finite cylinder is defined by the same parameters as the infinite cylinder plus the total length along the cylinder axis, L, and the "taper," τ, which gives the ratio of the narrow end of the cylinder to the wide end of the cylinder. Allowed values of τ range from 0 to 1 inclusive, with 0 indicating constant cylinder radius and 1 indicating a closed funnel or cone. In addition, R, $\hat{\mathbf{N}}$, and $\mathbf{O}$ now have more specific meanings: R defines the radius of the wide end, $\hat{\mathbf{N}}$ gives the orientation of the cylinder—pointing along the cylinder axis from the wide end to the narrow end—and $\mathbf{O}$ defines the midpoint along the cylinder axis between the two ends.

For the overlap test, we first consider a cylindrical region of space aligned along the orientation axis with length $L + 2s$ centered on the origin, with an inner radius of $\max\{R_\tau - s; 0\}$, where $R_\tau \equiv \tau R$, and an outer radius of $R + s$. The entire finite cylinder, as defined by the parameters $\mathbf{O}$, $\hat{\mathbf{N}}$, R, L, and τ, as well as any particle of radius s that could be in contact with it, is contained within this region of space. If the particle in question has its center outside this region, it cannot be in overlap with the finite cylinder. Otherwise the particle is subject to the following overlap test.

We define $\mathbf{P}_{R_\tau}$ to be the point on the narrow rim of the cylinder that is closest to the particle's center; this is given as $\mathbf{P}_{R_\tau} = R_\tau\hat{\mathbf{T}} + \frac{L}{2}\hat{\mathbf{N}}$. We define $\mathbf{P}_R$ to be the point on the wide rim of the cylinder that is closest to the particle's center; this is given as $\mathbf{P}_R = R\hat{\mathbf{T}} - \frac{L}{2}\hat{\mathbf{N}}$. We next find the point on the line segment connecting points $\mathbf{P}_{R_\tau}$ and $\mathbf{P}_R$ that is closest to point $\boldsymbol{\rho}_T$:

$$c'' = \frac{\left(\mathbf{P}_{R_\tau} - \mathbf{P}_R\right) \cdot (\boldsymbol{\rho} - \mathbf{P}_R)}{\left|\mathbf{P}_{R_\tau} - \mathbf{P}_R\right|^2}, \tag{A.2}$$

$$c' = \max\{0; c''\}, \tag{A.3}$$

$$c = \min\{c'; 1\}, \tag{A.4}$$

$$\mathbf{P}_c = c\mathbf{P}_{R_\tau}, \tag{A.5}$$

where $\mathbf{P}_c$ is the potential contact point (relative to the origin). The overlap condition is $|\mathbf{P}_c - \boldsymbol{\rho}| < s$, where s again is the radius of the particle. If this condition is met, the contact point $\mathbf{Q}$ is taken to be $\mathbf{O} + \mathbf{P}_c$.

As in the case of the disk, there is the special case of $\rho_T = 0$, where the particle is centered on the axis of the cylinder. If there is an overlap in this case, the overlap occurs over a ring that is symmetric around the cylinder axis, and therefore the overlap is radially symmetric. The net force could only point along the axis, so the strategy used here is to create a "phantom" overlap point on the axis that penetrates the particle by the same amount as it is penetrated by the ring. In cylindrical coordinates, the angle is degenerate, and thus the computations can be done in two dimensions, saving time. After first ruling out the overlap ahead of time if $s < R_\tau$, we consider whether force should be applied in the positive $\hat{\mathbf{N}}$ direction or the negative $\hat{\mathbf{N}}$ direction. When $\rho_T = 0$ and $\boldsymbol{\rho}\cdot\hat{\mathbf{N}} > L/2$, this corresponds to the first case, where the particle is on the cylinder axis with its center outside of the cylinder on the side of the narrow rim (the centers of both rims, narrow and wide, lie in the same direction from the particle center, but the center of the narrow rim is closer). Here, the overlap condition is $R_\tau^2 + (L/2)^2 < s^2$, with $\mathbf{Q} = \mathbf{O} + \boldsymbol{\rho} - \left[R_\tau^2 + (L/2)^2\right]^{1/2}\hat{\mathbf{N}}$. Conversely, the force must be applied in the negative $\hat{\mathbf{N}}$ direction when the particle is centered on the cylinder axis and if either τ is nonzero or $\boldsymbol{\rho}\cdot\hat{\mathbf{N}} < -L/2$. Only tapered cylinders ($\tau > 0$) will push out a particle whose center is inside the cylinder and exactly on the axis, so we ignore particles inside a non-tapered cylinder. Strictly speaking, both tapered and non-tapered cylinders should provide frictional stability to particles with radii larger than the cylinders of which they are inside. The code has a way to handle this, but not in the case where the particle is centered exactly on the cylinder axis. This will be a future

feature if needed. In the meantime, for the overlap condition in this case, we use Eqs. (A.2)–(A.5) to determine $\mathbf{P}_c$, using corresponding two-dimensional values for $\boldsymbol{\rho}$, $\mathbf{P}_{R_\tau}$, and $\mathbf{P}_R$. If the overlap condition is met, $\mathbf{Q} = \mathbf{O} + \mathbf{P}_c$.

Using wall primitives to construct the hoppers

We used combinations of two of the primitives described above, the disk and the finite cylinder, to construct the hoppers. For each hopper of radius R_{hopper}, we used finite cylinders with radius $R = R_\tau = R_{\text{hopper}}$ (no taper) and height H, with a confining bottom disk of radius $R_{\text{out}} = R_{\text{hopper}}$ ($R_{\text{in}} = 0$ at this stage) to simulate the hoppers. In order to fill the hoppers, we attached a large finite cylinder of outer radius R_{funnel} and taper τ such that $R_\tau = \tau R = R_{\text{hopper}}$ to act as a funnel. To commence particle flow, we replaced the disk at the bottom of the hopper with one of identical properties, except that instead of having $R_{\text{in}} = 0$, we set R_{in} to be the desired radius of the aperture.

References

1. Yano, H. et al.: Touchdown of the Hayabusa Spacecraft at the Muses Sea on Itokawa. Science **312**, 1350–1353 (2006)
2. Richardson, J.E., Melosh, H.J., Greenberg, R.J., O'Brien, D.P.: The global effects of impact-induced seismic activity on fractured asteroid surface morphology. Icarus **179**, 325–349 (2005)
3. Richardson, D.C., Walsh, K.J., Murdoch, N., Michel, P.: Numerical simulations of granular dynamics: I. Hard-sphere discrete element method and tests. Icarus **212**, 427–437 (2011)
4. Mehta, A.J.: Granular Physics. Cambridge University Press, New York (2007)
5. Cleary, P.W., Sawley, M.L.: DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. Appl. Math. Model. **26**, 89–111 (2002)
6. Kacianauskas, R., Maknickas, A., Kaceniauskas, A., Markauskas, D., Balevicius, R.: Parallel discrete element simulation of poly-dispersed granular material. Adv. Eng. Softw. **41**, 52–63 (2010)
7. Elaskar, S.A., Godoy, L.A., Gray, D.D., Stiles, J.M.: A viscoplastic approach to model the flow of granular solids. Int. J. Solids Struct. **37**, 2185–2214 (2000)
8. Holsapple, K.A.: Equilibrium figures of spinning bodies with self-gravity. Icarus **172**, 272–303 (2004)
9. Holsapple, K.A., Michel, P.: Tidal disruptions. II. A continuum theory for solid bodies with strength, with applications to the solar system. Icarus **193**, 283–301 (2008)
10. Sharma, I., Jenkins, J.T., Burns, J.A.: Dynamical passage to approximate equilibrium shapes for spinning, gravitating rubble asteroids. Icarus **200**, 304–322 (2009)
11. Wada, K., Senshu, H., Matsui, T.: Numerical simulation of impact cratering on granular material. Icarus **180**, 528–545 (2006)
12. Hong, D.C., McLennan, J.A.: Molecular dynamics simulations of hard sphere granular particles. Phys. A **187**, 159–171 (1992)
13. Huilin, L., Yunhua, Z., Ding, J., Gidaspow, D., Wei, L.: Investigation of mixing/segregation of mixture particles in gas-solid fluidized beds. Chem. Eng. Sci. **62**, 301–317 (2007)
14. Kosinski, P., Hoffmann, A.C.: Extension of the hard-sphere particle-wall collision model to account for particle deposition. Phys. Rev. E **79**, 061302 (2009)
15. Tsuji, Y., Tanaka, T., Ishida, T.: Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. Powder Technol. **71**, 239–250 (1992)
16. Sànchez, P., Scheeres, D.J.: Simulating asteroid rubble piles with a self-gravitating soft-sphere distinct element method model. ApJ **727**, 120 (2011)
17. Tancredi, G., Maciel, A., Heredia, L., Richeri, P., Nesmachnow, S.: Granular physics in low-gravity environments using DEM. MNRAS **420**, 3368–3380 (2012)
18. Gallas, J.A.C., Hermann, H.J., Pöschel, T., Sokolowski, S.: Molecular dynamics simulation of size segregation in three dimensions. J. Stat. Phys. **82**, 443–450 (1996)
19. Silbert, L.E., Ertaş, D., Grest, G.S., Halsey, T.C., Levine, D., Plimpton, S.J.: Granular flow down an inclined plane: Bagnold scaling and rheology. Phys. Rev. E **64**, 051302 (2001)
20. Stadel, J.: Cosmological N-body simulations and their analysis. pp. 126 University of Washington, Washington, DC (2001)
21. Richardson, D.C., Quinn, T., Stadel, J., Lake, G.: Direct large-scale N-body simulations of planetesimal dynamics. Icarus **143**, 45–59 (2000)
22. Richardson, D.C., Michel, P., Walsh, K.J., Flynn, K.W.: Numerical simulations of asteroids modelled as gravitational aggregates with cohesion. Planet. Space Sci. **57**, 183–192 (2009)
23. Cundall, P.A., Strack, O.D.L.: A discrete numerical model for granular assemblies. Geotechnique **29**, 47–65 (1979)
24. Saha, P., Tremaine, S.: Symplectic integrators for solar system dynamics. Astron. J. **104**, 1633–1640 (1992)
25. Quinn, T., Perrine, R.P., Richardson, D.C., Barnes, R.: A Symplectic integrator for Hill's equations. ApJ **139**, 803–807 (2010)
26. Cleary, P.W.: Predicting charge motion, power draw, segregation and wear in ball mills using discrete element methods. Min. Eng. **11**, 1061–1080 (1998)
27. Zhou, Y.C., Wright, B.D., Yang, R.Y., Xu, B.H., Yu, A.B.: Rolling friction in the dynamic simulation of sandpile formation. Phys. A **269**, 536–553 (1999)
28. Kahn, K.M., Bushell, G.: Comment on rolling friction in the dynamic simulation of sandpile formation. Phys. A **352**, 522–524 (2005)
29. Zhu, H.P., Yu, A.B.: A theoretical analysis of the force models in discrete element method. Powder Technol. **161**, 122–129 (2006)
30. Nedderman, R.M., Tüzün, U., Savage, S.B., Houlsby, G.T.: The flow of granular materials–I: discharge rates from Hoppers. Chem. Eng. Sci. **37**, 1597–1609 (1982)
31. Bertrand, F., Leclaire, L.-A., Levecque, G.: DEM-based models for the mixing of granular materials. Chem. Eng. Sci. **60**, 2517–2531 (2005)
32. Beverloo, W.A., Leniger, H.A., Velde, J.van de : The flow of granular solids through orifices. Chem. Eng. Sci. **15**, 260–269 (1961)
33. Janssen, H.A.: Versuche über Getreidedruck in Silozellen. Ver. Dt. Ing. **39**, 1045–1049 (1895)
34. Shaxby, J.H., Evans, J.C.: The variation of pressure with depth in columns of powders. Trans. Faraday Soc. **19**, 60–72 (1923)
35. Rose, H.E., Tanaka, T.: Rate of discharge of granular materials from bins and hoppers. Engineer **208**, 465–469 (1959)
36. Hofmeister, P., Blum, J., Heißelmann, D.: The flow of granular matter under reduced-gravity conditions. In: Nakagawa M., Luding S. (eds.) Powders and Grains 2009: Proceedings of the 6th International Conference on Micromechanics of Granular Media, Hrsg. AIP Conference Proceedings vol. 1145, pp. 71–74 (2009)