Polyspherical grains and their dynamics *

Ulrich Mutze †
ulrichmutze@aol.com

This paper describes a method to simulate the dynamics of granular systems consisting of irregularly shaped grains. This method is a simplification and partial improvement of a method that has been developed and employed earlier in simulating the toning process in electro-photographic copiers. Here, grains are modeled as rigidly connected overlapping spherical particles. For each grain a bounding sphere around the center-of-mass is known. Therefore, it can efficiently be decided whether two particles intersect and thus exert contact forces upon each other. A second order time stepping algorithm for such grains is given which needs only one evaluation of the inter-particle forces in a time step. In a ten grain example system a favorable but poorly understood phenomenon of energy restoration is exemplified.

1 Introduction

Granular media consist of movable, relatively stable, entities that are usually referred to as *particles* or as *grains*. In this paper we consider a specific computational model according to which grains are made of overlapping spheres which are rigidly connected and which exchange forces with the spherical components of adjacent grains according to any suitable model for forces between spherical particles. This model of *polyspherical* grains combines the simplicity of spherical particle models [2] with the shape flexibility of polyhedral models [3], [4], [5] and spherosimplicial ones [6]. Polyspherical grains were considered recently by several authors [7], [8], [9].

The motivation for developing this model was an industrial project (which started in April 1998) concerning the simulation of the toning process in copying machines that employ rotating magnetic brush technology [10]. In these machines, *toner particles* are temporarily bound to hard-magnetic *carrier particles* by tribo-electric charges and get violently transported by a collective motion (*chain flipping*) that a *locally rotating magnetic*

*this is an pedagogically extended version of [24]

†retired from R&D for a subsidiary of Heidelberger Druckmaschinen AG, after R&D work for a subsidiary of Eastman Kodak Company

field induces in the system of carrier particles. Finally, the strong electric fields emanating from the *electrostatic image* on a *film loop*, together with mechanical de-acceleration forces, free the toner particles from their carriers and let them settle on the film loop where they contribute to the *development* of that image. (Toning is only one step in a process chain usually written as cleaning, charging, exposure, toning, transfer, fusing.)

As seen in the microscope, carrier and toner particles are quite irregularly shaped and one expects that the contact between these particles will be stabilized more by interference of visible asperities than by Coulomb friction between 'smooth' surfaces. Although one sees comparably irregular grains in sand or gravel successfully simulated with disks or spheres, it was tempting not to rely on fictitious friction laws between spherical bodies in a computational toning model but to mimic surface roughness on a geometrical level. It is a natural first idea to consider strongly bound agglomerates of spherical particles as a model for both toner and carrier particles. This does not work, however, since bounding forces that are strong enough to prevent frequent fragmentation of grains, enforce unacceptably small time steps in a dynamical simulation [1] . Fortunately it is possible to consider the bounds between the particles of a superparticle as perfectly rigid [2] (with no degree of freedom associated with them) as explained in those rare textbooks on classical mechanics that don't hold back the most important success of their formal methods concerning constraint motion and derive the model of a rigid body from rigidly connected point masses (e.g. [13]). Interaction and time evolution of rigidly connected spherical particles turned out to allow an efficient coding. Letting the spherical particles overlap, as suggested by [14], did not introduce any complications but strongly reduced the number of particles needed to generate useful grain shapes. In its final three-dimensional version the resulting program ran with 14000 particles on 40 processors for weeks and was able to show the formation of the magnetic brush and its efficiency in delivering toner particles to the right locations. Former two-dimensional versions of the program where able to cover the whole toning process in some detail. Various insights generated by these simulations led to actual machine improvements [1].

Meanwhile, I created a version of this program named *PaLa* (for *Pa*rticle *La*b) [3] which is free of the company-proprietary components related to copier technology and which has a focus on numerical experiments with granular systems. These experiments can be watched real time on screen and typically create an answer within minutes on a state-of-the-art personal computer if only a few hundred particles are involved.

The present paper can be considered a translation of the most basic parts of *PaLa* from commented C++ code to 'normal language'. These parts are concerned with the definition of polyspherical grains, of the repulsive contact forces between such grains, and the time-stepping algorithm defining their dynamics.

This paper is intended to enable applications of the polyspherical grain model in fields, in which the mechanical forces between contacting grains play the prominent

[1] There are applications, in which such grain models are useful, see [11], where such grains are called *superparticles* of type *cluster*

[2] in [11], these are the superparticles of type *clump*; such superparticles are also considered in [12]

[3] On request, I'll send a freely distributable Windows-executable of 1.5 MB by e-mail.

part and in which these forces may be given by formulas different from those implemented in *PaLa*. Such applications would then benefit from the striking stability and efficiency of the proposed time-stepping algorithm, especially from the high accuracy in conserving total mechanical energy in situations in which all friction descriptors are set to zero.

To indicate the capabilities of the present method, I list some features that it allowed me to implement in *PaLa*: (i) Switchability between 2D and 3D . (ii) Switchability between polyspherical grains and simple spherical particles. (iii) Particles can be enclosed in a rectangular or in a spherical cavity. (iv) The rectangular box may be divided by a grid as a model of a semi-permeable membrane for simulating osmosis; the faces of the box record the momentum transfer from particle impacts, thus allowing us to monitor momentum conservation and the building up of an osmotic pressure. (v) In addition to the contact forces between particles we have forces resulting from point masses, point charges, and electric and magnetic dipoles, all located at the center-of-mass of the particles. (vi) Arbitrary homogeneous gravitational, electric, and magnetic fields can be set. (vii) The total energy of the most general system configuration is implemented as an expression which is well conserved without noticeable trend if the timestep is not unreasonably large and if all friction forces are switched off. (viii) To most types of program runs one can create a sequence of program runs which cover the same time-interval with the number of integration steps multiplied by a selectable factor from one run to the next; the trajectory change from run to run can be visualized as a plane curve which gives the distance of synchronous configurations (in some natural metric) as a function of time. It becomes evident that for short time spans this distance is proportional to a power of the time step (which is the order of the integrator and which can be extracted automatically from any selected piece of the system trajectory) and evolves into a random function of the time step for long time spans. (ix) Also the variation of the total energy in such a run series can be analyzed for order and visualized. Both the order of the integrator, and of the energy deviation can be shown to be two by this analytical tool. Actually, the integrator algorithm, as defined in Sections 6, 7 would be consistent with my heuristics also if two substeps would be defined differently. To decide between equally plausible alternatives, I used numerical experiments based on this tool.

2 On vectors, points, and rotations

As the grains move through space, they translate and rotate. In this section we present the mathematics we use to describe this motion.

When dealing with *space* in classical physics it natural to employ affine geometry (e.g. [15]), in which points and vectors are different mathematical entities. [4] To begin with vectors, let $\mathcal{V}$ be a three-dimensional Euclidean oriented vector space, i.e. a three-dimensional linear space over $\mathbb{R}$ for which a scalar product $\cdot : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and a vector product $\times : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ are defined and have the properties implied by the names. For

[4] What in this paper is written as sets or maps translates directly to classes or functions in C++.

$\mathbf{v} \in \mathcal{V}$ one defines $|\mathbf{v}| := \sqrt{\mathbf{v}\cdot\mathbf{v}}$ and knows that the statements $\mathbf{v} = \mathbf{0}$ and $|\mathbf{v}| = 0$ imply each other. The determinant function, which usually is referred to in defining orientation in linear spaces is defined by combining the two product functions

$$\det : \mathcal{V} \times \mathcal{V} \times \mathcal{V} \to \mathbb{R}\,, \quad (\mathbf{u},\mathbf{v},\mathbf{w}) \mapsto (\mathbf{u}\times\mathbf{v})\cdot\mathbf{w}\,.$$

Despite its heterogeneous definition it treats the arguments on equal footing:

$$\det(\mathbf{u},\mathbf{v},\mathbf{w}) = \det(\mathbf{v},\mathbf{w},\mathbf{u}) = \det(\mathbf{w},\mathbf{u},\mathbf{v}) = -\det(\mathbf{u},\mathbf{w},\mathbf{v}) = -\det(\mathbf{w},\mathbf{v},\mathbf{u}) = -\det(\mathbf{v},\mathbf{u},\mathbf{w}).$$

An Euclidean point space $\mathcal{P}$ is related to $\mathcal{V}$ by the existence of a map $+ : \mathcal{P}\times\mathcal{V} \to \mathcal{P}$ for which the following holds:

1. For all $\mathbf{v},\mathbf{w} \in \mathcal{V}, p \in \mathcal{P}$ we have $(p+\mathbf{v})+\mathbf{w} = p+(\mathbf{v}+\mathbf{w})$.
2. For any two points $p,q \in \mathcal{P}$ there is a uniquely defined element of $\mathcal{V}$, conveniently written as $q-p$, for which $p+(q-p) = q$. For $p+\frac{1}{2}(q-p)$ one also writes $\frac{1}{2}(p+q)$.

With any two points p,q one associates the number $|p-q|$ $(= |q-p|)$ as their distance. With the functions defined so far, we define the *orientation* associated with a list (p_0,p_1,p_2,p_3) of points as the sign of $\det(p_1-p_0, p_2-p_0, p_3-p_0)$. This determinant is zero exactly if the four points lie on a *plane*, in which case the points fail to define an orientation. Notice that, by the same token, a list of three points never defines an orientation in three-dimensional space. Having already mentioned planes as subsets of $\mathcal{P}$, we consider the subsets of $\mathcal{P}$ which are most important in the present context: *spheres*

$$\mathcal{S}(r,c) := \{p \in \mathcal{P} : |p-c| \le r\} \text{ for all } r \in \mathbb{R}_+\,,\ c \in \mathcal{P}\,, \tag{1}$$

which excel by the computational cheapness of their indicator function. Nearly all physically relevant operations and relations in space are closely related to linear maps $\mathcal{V} \to \mathcal{V}$ which form a real associative involution algebra $\mathcal{L}$ with the linear operations carried over from $\mathcal{V}$ and with the composition $\circ$ of maps as the product, and the involution given by $L \mapsto L^*$, where L^* is the map adjoint to L. For the application of $L \in \mathcal{L}$ to a $\mathbf{v} \in \mathcal{V}$ we mostly write $L\mathbf{v}$ instead to the orthodox notation $L(\mathbf{v})$. To be sure, these definitions imply that we have

$$L\circ L'\,\mathbf{v} = LL'\,\mathbf{v}\,, \quad (\alpha L+\alpha' L')\,\mathbf{v} = \alpha L\mathbf{v}+\alpha' L'\,\mathbf{v}\,, \quad L\mathbf{v}\cdot\mathbf{w} = \mathbf{v}\cdot L^*\mathbf{w}$$

for all $L,L' \in \mathcal{L}, \mathbf{v},\mathbf{w} \in \mathcal{V}, \alpha,\alpha' \in \mathbb{R}$. The map $L \in \mathcal{L}$ is said to be *orthogonal* iff $L\mathbf{v}\cdot L\mathbf{w} = \mathbf{v}\cdot\mathbf{w}$ and a *rotation* iff, in addition, $L\mathbf{v}\times L\mathbf{w} = L(\mathbf{v}\times\mathbf{w})$. Further, $L \in \mathcal{L}$ is said to be *symmetric* iff $L\mathbf{v}\cdot\mathbf{w} = \mathbf{v}\cdot L\mathbf{w}$ and *skew-symmetric* iff $L\mathbf{v}\cdot\mathbf{w} = -\mathbf{v}\cdot L\mathbf{w}$. In these four definitions the quantification 'for all $\mathbf{v},\mathbf{w} \in \mathcal{V}$' is understood. It is an interesting aspect of our 3-dimensional case that the vector operations allow constructing the most useful elements of $\mathcal{L}$: For all $\lambda \in \mathbb{R}$ and $\mathbf{a},\mathbf{b} \in \mathcal{V}$ define

$$\underline{\lambda} : \mathcal{V} \to \mathcal{V}\,, \quad \mathbf{v} \mapsto \lambda\mathbf{v}\,, \tag{2}$$

$$|\mathbf{a}\rangle\langle\mathbf{b}| : \mathcal{V} \to \mathcal{V}\,, \quad \mathbf{v} \mapsto (\mathbf{b}\cdot\mathbf{v})\,\mathbf{a}\,, \tag{3}$$

$$A_{\mathbf{a}} : \mathcal{V} \to \mathcal{V}\,, \quad \mathbf{v} \mapsto \mathbf{a} \times \mathbf{v}\,. \tag{4}$$

These all are linear maps; $\underline{\lambda}$ is symmetric, $A_{\mathbf{a}}$ is skew-symmetric, and the adjoint of $|\mathbf{a}\rangle\langle\mathbf{b}|$ is $|\mathbf{b}\rangle\langle\mathbf{a}|$; therefore $P_{\mathbf{a}} := |\mathbf{a}\rangle\langle\mathbf{a}|$ is symmetric. For each linear skew-symmetric map $\mathcal{V} \to \mathcal{V}$ there is a uniquely defined $\mathbf{a}$ such that it equals $A_{\mathbf{a}}$. As is well known, for each skew-symmetric map A, the exponential $\exp(A)$ is orthogonal. So let us consider $\exp(A_{\mathbf{a}})$. Writing $\mathbf{a} = \varphi\,\mathbf{n}$, with a unit vector $\mathbf{n}$, we have to compute the powers of $A_{\mathbf{n}}$. This is surprisingly easy due to the following convenient relations: $A_{\mathbf{n}} \circ A_{\mathbf{n}} = P_{\mathbf{n}} - \underline{1}$, $P_{\mathbf{n}} \circ A_{\mathbf{n}} = A_{\mathbf{n}} \circ P_{\mathbf{n}} = \underline{0}$, and $P_{\mathbf{n}} \circ P_{\mathbf{n}} = P_{\mathbf{n}}$. They allow us to sum up the exponential series and to obtain

$$\exp(\varphi A_{\mathbf{n}}) = P_{\mathbf{n}} + \cos\varphi\,(\underline{1} - P_{\mathbf{n}}) + \sin\varphi\, A_{\mathbf{n}} =: R(\varphi, \mathbf{n})\,. \tag{5}$$

It turns out, that equation (5) gives the most general rotation. The representation of rotations by (5) is in terms of a *rotation angle* φ and a *rotation axis* $\mathbf{n}$. The angle φ is determined uniquely, if restricted to $0 \le \varphi \le \pi$. The rotation axis is determined uniquely if $\varphi \neq \pi$, otherwise $\mathbf{n}$ and $-\mathbf{n}$ correspond to the same rotation.

Rotations will play a decisive role in formulating a time-stepping algorithm for rigid bodies (and polyspherical grains in particular). In this algorithm, rotations have to act on many vectors and for many pairs of rotations R, R' one has to form the composition $R \circ R'$. It is therefore important to have fast algorithms for these operations. I use here the method of *Euler-Rodrigues parameters*, which is the most efficient one known. It seems to go back, (e.g. [16]), to the French mathematician Olinde Rodrigues, who wrote about it as early as 1840 [17]. Several rediscoveries happened in newer times, one—in 1986—by myself [18]. Here, I simply state the result by introducing two new operations $* : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ and $\circ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$:

$$\mathbf{r} * \mathbf{x} := \frac{(1 - \mathbf{r}\cdot\mathbf{r})\mathbf{x} + 2(\mathbf{r}\cdot\mathbf{x})\mathbf{r} + 2\mathbf{r}\times\mathbf{x}}{1 + \mathbf{r}\cdot\mathbf{r}}\,, \tag{6}$$

$$\mathbf{r} \circ \mathbf{r}' := \frac{\mathbf{r} + \mathbf{r}' + \mathbf{r}\times\mathbf{r}'}{1 - \mathbf{r}\cdot\mathbf{r}'}\,. \tag{7}$$

To avoid numerical exceptions in (7) one selects a small number ε, e.g. $\varepsilon = 10^{-12}$, and replaces the nominator by ε whenever $|1 - \mathbf{r}\cdot\mathbf{r}'| < \varepsilon$ [5]. The basic properties of these operations are

$$R(\varphi, \mathbf{n})(\mathbf{x}) = \mathbf{r} * \mathbf{x} \text{ for all } \mathbf{x} \in \mathcal{V} \quad \text{iff} \quad \mathbf{r} = \tan\left(\frac{\varphi}{2}\right)\mathbf{n}\,, \tag{8}$$

$$(\mathbf{r} \circ \mathbf{r}') * \mathbf{x} = \mathbf{r} * (\mathbf{r}' * \mathbf{x})\,, \quad \mathbf{r} \circ (-\mathbf{r}) = \mathbf{0}\,, \quad \mathbf{r} \circ \mathbf{0} = \mathbf{r}\,, \quad \mathbf{r} \circ \mathbf{r}' = (\mathbf{r} * \mathbf{r}') \circ \mathbf{r} \tag{9}$$

which can be used to define $\circ$ in terms of $*$ and vice versa. Conceptually this is an interesting step: we don't need a new type of objects to represent rotations, instead we have new operations of type $\mathcal{V} \times \mathcal{V} \to \mathcal{V}$ which allow elements of $\mathcal{V}$ doing everything that rotations are supposed to do. Nevertheless, we shall speak of *rotation vectors*, and

[5] See [18] for evidence that this works smoothly.

use the symbol $\mathbf{r}$, to indicate that the raison d'être of the vector is to act as a rotation $\mathbf{x} \mapsto \mathbf{r} * \mathbf{x}$. Using the symbol $\mathcal{R}(\mathbf{r})$ for this mapping (hence $\mathcal{R}(\mathbf{r})\,\mathbf{x} := \mathbf{r} * \mathbf{x}$) allows writing the first equation in (9) as

$$\mathcal{R}(\mathbf{r} \circ \mathbf{r}') = \mathcal{R}(\mathbf{r}) \circ \mathcal{R}(\mathbf{r}') \tag{10}$$

which motivates the notation $\mathbf{r} \circ \mathbf{r}'$. Further, (8) can be written as

$$R(\varphi, \mathbf{n}) = \mathcal{R}(\tan\left(\frac{\varphi}{2}\right)\,\mathbf{n})$$

and the remaining equations in (9) read

$$\mathcal{R}(-\mathbf{r}) = \mathcal{R}(-\mathbf{r})^{-1}\,, \quad \mathcal{R}(\mathbf{0}) = \underline{1}\,, \quad \mathcal{R}(\mathbf{r}) \circ \mathcal{R}(\mathbf{r}') \circ \mathcal{R}(\mathbf{r})^{-1} = \mathcal{R}(\mathcal{R}(\mathbf{r})\,\mathbf{r}')\,.$$

Let us consider the computational realization of these expressions: As (6) is written, it defines the term but taking it literally as a recipe for computation would not be wise. What one would actually do is

$$\begin{aligned} &s_0 := \mathbf{r} \cdot \mathbf{r}\,, \quad s_1 := 1/(1+s_0)\,, \quad s_2 := 1 - s_0\,, \quad \mathbf{x}' := 2\mathbf{x}\,, \quad s_3 := \mathbf{r} \cdot \mathbf{x}'\,, \\ &\mathbf{r} * \mathbf{x} := s_1(s_2\mathbf{x} + s_3\mathbf{r} + \mathbf{r} \times \mathbf{x}') \end{aligned}$$

If one needs to compute $\mathbf{r} * \mathbf{x}$ for some fixed $\mathbf{r}$ and a multitude of $\mathbf{x}$'s it pays first to compute the 3×3-matrix which represents $\mathcal{R}(\mathbf{r})$. Expressing everything in terms of components one can do even better: putting $\mathbf{r} = (r_1, r_2, r_3)$ and $\mathbf{x} = (x_1, x_2, x_3)$ we define $\mathbf{x}' = \mathbf{r} * \mathbf{x}$ by

$$R := \begin{pmatrix} {r_1}^2 + b & r_1 r_2 + r_3 & r_1 r_3 - r_2 \\ r_1 r_2 - r_3 & {r_2}^2 + b & r_2 r_3 + r_1 \\ r_1 r_3 + r_2 & r_2 r_3 - r_1 & {r_3}^2 + b \end{pmatrix} \tag{11}$$

$$x'_i := c(R_{i1}x_1 + R_{i2}x_2 + R_{i3}x_3)\,, \quad i = 1,2,3\,, \tag{12}$$

where the numbers b and c are defined as follows:

$$\begin{aligned} \rho :=& {r_1}^2 + {r_2}^2 + {r_3}^2 \\ b :=& (1-\rho)/2 \\ c :=& 2/(1+\rho)\,. \end{aligned} \tag{13}$$

Finally we shall need the function $\exp : \mathcal{V} \to \mathcal{V}$ that allows us to express the rotation vector $\mathbf{r}$ directly from the vector $\mathbf{a} := \varphi\mathbf{n}$ (rather than from φ and $\mathbf{n}$ separately) as $\mathbf{r} = \exp(\mathbf{a})$. This is how the connection between the Lie algebra and the Lie group of rotations is represented in the present formalism. Equations (5) and (8) imply

$$\exp(\mathbf{a}) = \tan\frac{|\mathbf{a}|}{2}\,\frac{\mathbf{a}}{|\mathbf{a}|} \approx \frac{\mathbf{a}}{2}\,, \tag{14}$$

where the approximation is for $|\mathbf{a}| \ll 1$. This function satisfies

$$\begin{aligned} \mathcal{R}(\exp(t\mathbf{a})) &= \exp(tA_{\mathbf{a}}) \\ \exp((t+t')\,\mathbf{a}) &= \exp(t\,\mathbf{a}) \circ \exp(t'\,\mathbf{a}) \text{ for all } \mathbf{a} \in \mathcal{V},\ t, t' \in \mathbb{R}\,. \end{aligned} \tag{15}$$

For $L \in \mathcal{L}$ and $\mathbf{r} \in \mathcal{V}$ we define the 'rotated operator' $L_{\mathbf{r}}$ by

$$L_{\mathbf{r}}\mathbf{v} := \mathbf{r} * L((-\mathbf{r}) * \mathbf{v}) \text{ for all } \mathbf{v} \in \mathcal{V}\,, \tag{16}$$

and get

$$L_{\mathbf{r}} = \mathcal{R}(\mathbf{r}) \circ L \circ \mathcal{R}(-\mathbf{r}) = \mathcal{R}(\mathbf{r}) \circ L \circ \mathcal{R}(\mathbf{r})^{-1}\,, \quad |\mathbf{a}\rangle\langle\mathbf{b}|_{\mathbf{r}} = |\mathbf{r} * \mathbf{a}\rangle\langle\mathbf{r} * \mathbf{b}|\,. \tag{17}$$

A time-dependent rotation vector $\mathbf{r}(t)$ determines the time-dependent *angular velocity* $\omega(t)$ by

$$\mathbf{r}(t+h) = \exp(h\omega(t)) \circ \mathbf{r}(t) + O(h^2) \tag{18}$$

which entails

$$\omega(t) = \frac{2}{1 + \mathbf{r}(t) \cdot \mathbf{r}(t)} \left(\dot{\mathbf{r}}(t) + \mathbf{r}(t) \times \dot{\mathbf{r}}(t)\right)\,. \tag{19}$$

For $I \in \mathcal{L}$ (not depending on t) we conclude from (15)

$$\frac{\mathrm{d}}{\mathrm{d}t} I_{\mathbf{r}(t)} = A_{\omega(t)} \circ I_{\mathbf{r}(t)} - I_{\mathbf{r}(t)} \circ A_{\omega(t)} \tag{20}$$

and thus

$$\frac{\mathrm{d}}{\mathrm{d}t} I_{\mathbf{r}(t)}\, \omega(t) = I_{\mathbf{r}(t)} \dot{\omega}(t) + \omega(t) \times I_{\mathbf{r}(t)}\, \omega(t) \tag{21}$$

which gives the rate of change of angular momentum if I is the tensor of inertia as introduced later.

3 Defining polyspherical grains

Just as a spherical particle, a polyspherical grain G has—by design—an invariable *shape*. Though elastic repulsion forces between grains will be computed from formulae which are based on the deformation of elastic spheres under forced contact, other effects of these deformations such as a shift of the center of mass or changes in the tensor of inertia are not taken into account, not to mention elastic waves and sound generation. Therefore, the mechanical degrees of freedom of G are just those of a *rigid body*. As a geometrical object, G is made of *overlapping* spheres which we shall call *components* of G and which we shall describe by a list $(r_i)_{i=1}^n \in \mathbb{R}_+^n$ of radii and a list $(c_i)_{i=1}^n \in \mathcal{P}^n$ of centers. The space occupied by the grain is the set union of these spheres and the grain is assumed to consist of a homogeneous material. From the various physical properties of this material we presently need only the mass density ρ. To allow for extensions, we introduce a material property list $\pi = (\rho, \ldots)$. In the specific form of the model to be presented in Section 5, there will be five more entries to π. Let Π be the set of all such lists that agree with the requirements of our intended application. These requirements should guide us to define a function

$$\gamma : \mathbb{N} \to \mathcal{C} := \bigcup_{n \in \mathbb{N}} \mathcal{C}_n := \bigcup_{n \in \mathbb{N}} \Pi \times \mathbb{R}_+^n \times \mathcal{P}^n \quad {}^{6} \tag{22}$$

[6] $\gamma(1), \gamma(2), \gamma(3), \ldots$ is thus a sequence of initial grain configurations to be used in building the granular model system we are interested in. The set $\mathcal{C}$ of configurations is a union of sets $\mathcal{C}_n$ with n components. This allows n to change from grain to grain; this n is not assumed to be related to the argument of the function in any way.

which acts as a 'grain factory' by generating a useful *initial configuration* of a grain for each value of its argument. It will typically try to let basic grain properties such as diameter or mass follow prescribed statistical distributions by making random choices. Natural conditions that one probably should enforce are:

1. The grain would not fall into parts if only overlap would make a stiff connection between components.
2. There are no spheres which lie completely in the interior of another sphere.

It is to be noticed, however, that the dynamics to be defined later on will always let the spheres move as a rigid assembly irrespective of whether components are overlapping or are separated by gaps. If there would be one sphere within another this would give the outer sphere a harder nucleus (or the inner sphere a soft hull). Agglomerates of soap-bubbles give a good idea of possible grain shapes. Using only a few spheres with radii that don't differ too much gives necessarily rather smooth shapes, whereas using dozens of spheres with quite different radii allows us to mimic also corners and edges. In the work [10] the grains were quite compact and built out of three to five spheres. It is possible, however, to build more elongated shapes like boomerangs or dumbbells, or even more complex things like a perforated shell that encloses a cavity.

An initial configuration $\gamma(l)$ provides the data

$$\rho \in \mathbb{R}_+ \,, \quad n \in \mathbb{N} \,, \quad (r_i)_{i=1}^n \in \mathbb{R}_+^{\,n} \,, \quad (c_i)_{i=1}^n \in \mathcal{P}^n \tag{23}$$

which are the input for our subsequent computation of all the quantities —such as mass, inertial moments, and principal axes— that are needed for an algorithmic definition of grain dynamics.

3.1 Computing descriptors for shape and inertia

It would restrict the allowed arrangements of the spheres probably too much if we would consider only cases for which we can find efficient explicit formulae for the volume, the center-of-mass, and the tensor of inertia of the configuration. Since we are interested in strongly overlapping spheres it would certainly not be acceptable to ignore the circumstance that the parts of space belonging to more than one sphere have the same mass density as the parts covered by only one sphere. The most natural response to this problem is to determine these quantities by Monte Carlo integration. The complete chain of formulae will be given, since making the few necessary remarks in general terms would not be much shorter. For simplicity we do the following computations with coordinates and not within pure geometry. For this purpose we pick a *frame* $(\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, o) \in \mathcal{V}^3 \times \mathcal{P}$, where $\mathbf{u_1}, \mathbf{u_2}$ are orthonormal and $\mathbf{u_3} := \mathbf{u_1} \times \mathbf{u_2}$, and define coordinates by

$$v_\alpha := \mathbf{v} \cdot \mathbf{u}_\alpha \,, \quad p_\alpha := (p - o) \cdot \mathbf{u}_\alpha \text{ for all } \alpha \in \{1,2,3\} \,, \; \mathbf{v} \in \mathcal{V}, p \in \mathcal{P} \,. \tag{24}$$

For each $\alpha \in \{1,2,3\}$ we form the two numbers

$$L_\alpha := \inf\{c_{i\alpha} - r_i \,:\, 1 \le i \le n\} \,, \quad U_\alpha := \sup\{c_{i\alpha} + r_i \,:\, 1 \le i \le n\} \,. \tag{25}$$

Then the set

$$B := \{p \in \mathcal{P} \,:\, L_\alpha \le p_\alpha \le U_\alpha \text{ for all } \alpha \in \{1,2,3\}\} \tag{26}$$

is an enclosing box for $S := \cup_{i=1}^{n} \mathcal{S}(r_i, c_i)$ and the volume of the box is

$$V_B := (U_1 - L_1)(U_2 - L_2)(U_3 - L_3)\,. \tag{27}$$

The indicator function $\chi_S : \mathcal{P} \to \mathbb{R}$ of the set S

$$\chi_S(p) := \text{ if } |p - c_i| \le r_i \text{ for some } i \in \{1,\ldots,n\} \text{ then } 1 \text{ else } 0 \;^{7} \tag{28}$$

is computationally cheap (if n is not too large) which is one of the major advantages of the present shape model. For Monte Carlo integration we define a *uniformly distributed random sequence* $\tilde{p} : \mathbb{N} \to B$

$$\tilde{p}_{k\alpha} := L_\alpha + \zeta_{3k+\alpha-1}(U_\alpha - L_\alpha)\,, \tag{29}$$

where $\zeta : \mathbb{N} \to [0,1]$ is a general-purpose uniformly distributed random sequence. A working choice is

$$\zeta_k := 10^6 \sin(k) - \text{floor}(10^6 \sin(k))\,.^{8} \tag{30}$$

Then, by replacing integrals $\int_S f(x)\,\mathrm{d}x$ by $\frac{V_B}{N}\sum_{k=1}^{N} \chi_S(\tilde{p}_k) f(\tilde{p}_k)$ for sufficiently large N, e.g. $N = 10^4$, we set for all $\alpha, \beta \in \{1,2,3\}$

$$\begin{aligned}
V :=& \frac{V_B}{N}\sum_{k=1}^{N} \chi_S(\tilde{p}_k)\,,\\
m :=& \rho V\,,\\
x_\alpha :=& \frac{1}{V}\frac{V_B}{N}\sum_{k=1}^{N} \chi_S(\tilde{p}_k)\tilde{p}_{k\alpha}\,,\\
I'_{\alpha\beta} :=& \frac{\rho V_B}{N}\sum_{k=1}^{N} \chi_S(\tilde{p}_k)(\tilde{p}_{k\gamma}\tilde{p}_{k\gamma}\delta_{\alpha\beta} - \tilde{p}_{k\alpha}\tilde{p}_{k\beta})\,,\\
I_{\alpha\beta} :=& I'_{\alpha\beta} - m(x_\gamma x_\gamma \delta_{\alpha\beta} - x_\alpha x_\beta)\,,
\end{aligned} \tag{31}$$

where sum convention over repeated Greek indexes is used. Define

$$x := o + \sum_{\alpha=1}^{3} x_\alpha \mathbf{u}_\alpha\,, \quad \mathbf{x}_i := c_i - x \text{ for all } i \in \{1,\ldots,n\}\,, \quad r := \sup\{|\mathbf{x}_i| + r_i \,:\, 1 \le i \le n\}\,. \tag{32}$$

[7] Although this linear notation of conditional terms may look unfamiliar in a non-programming context, I use it throughout this paper. Using the traditional multi-line version instead, creates a permanent pressure to minimize the occurrence of conditional terms for reasons of space savings at the potential cost of clarity and completeness.

[8] Traditionally, random generators avoid using floating point arithmetics and transcendental functions for efficiency reasons that I consider no longer relevant in scientific computing. Throwing off these restrictions, offers rich opportunities to define sufficiently chaotic one line functions as the one given here. The present definition mimics a procedure that was once common: taking as random numbers the successive entries in some printed function table, with the decimal point shifted a fixed number of places to the right and replacing everything left to the new decimal point by 0. It is very plausible that these 'less important digits' of different entries are effectively uncorrelated.

Obviously x is the position of the *center of mass*, and r is the smallest *radius* of a sphere that encloses S and has x as center. The vectors $\mathbf{x}_i$, which encode the relative position of the component centers to the center of mass, will be called *shape vectors*. The matrix $M := (I_{\alpha\beta})_{\alpha,\beta=1}^{3}$ is symmetric. Thus by a suitable algorithm we find an orthogonal matrix O such that $OMO^t =: D$ is diagonal. D then yields the *moments of inertia* and O yields the vectors associated with the *principal axes*

$$I_\alpha := D_{\alpha\alpha}\,, \quad \mathbf{e}_\alpha := \sum_{\beta=1}^{3} O_{\alpha\beta}\,\mathbf{u}_\beta\,, \qquad \text{for all } \alpha \in \{1,2,3\}\,. \tag{33}$$

The *tensor of inertia*, $I \in \mathcal{L}$, is

$$I := \sum_{\alpha=1}^{3} I_\alpha |\mathbf{e}_\alpha\rangle\langle\mathbf{e}_\alpha| = \sum_{\alpha,\beta=1}^{3} I_{\alpha\beta} |\mathbf{u}_\alpha\rangle\langle\mathbf{u}_\beta| \tag{34}$$

and its inverse is

$$I^{-1} := \sum_{\alpha=1}^{3} I_\alpha^{-1} |\mathbf{e}_\alpha\rangle\langle\mathbf{e}_\alpha|\,. \tag{35}$$

It will be convenient to refer to this algorithm as two functions, both operating on the generic grain configuration. Evaluating the center of mass defines the function

$$\begin{aligned} \xi_n \,:\, \mathcal{C}_n &\to \mathcal{P}\,, \\ (\pi,\, (r_i)_{i=1}^n\,,\, (c_i)_{i=1}^n) &\mapsto x\,, \end{aligned} \tag{36}$$

and evaluating bounding radius, shape vectors, mass, moments of inertia, and principal axes defines

$$\begin{aligned} \iota_n \,:\, \mathcal{C}_n &\to \mathcal{D}_n := \Pi \times \mathbb{R}_+^{\,n} \times \mathbb{R}_+ \times \mathcal{V}^n \times \mathbb{R}_+ \times \mathbb{R}_+^{\,3} \times \mathcal{V}^3\,, \\ (\pi,\, (r_i)_{i=1}^n\,,\, (c_i)_{i=1}^n) &\mapsto (\pi,\, (r_i)_{i=1}^n\,,\, r,\, (\mathbf{x}_i)_{i=1}^n\,,\, m,\, (I_\alpha)_{\alpha=1}^3\,,\, (\mathbf{e}_\alpha)_{\alpha=1}^3)\,. \end{aligned} \tag{37}$$

To get rid of the n-dependence, we paste these maps together in a natural manner

$$\xi := \bigcup_{n\in\mathbb{N}} \xi_n \,:\, \mathcal{C} \to \mathcal{P}\,, \tag{38}$$

$$\mathcal{D} := \bigcup_{n\in\mathbb{N}} \mathcal{D}_n\,, \quad \iota := \bigcup_{n\in\mathbb{N}} \iota_n \,:\, \mathcal{C} \to \mathcal{D}\,. \tag{39}$$

It is interesting to observe that the Monte Carlo integration does not introduce physical inconsistencies such as violating the established inequalities for the $I_{\alpha\beta}$. It only replaces the model system with continuous mass distribution by another consistent model system in which matter is concentrated in a dense cloud of points. If this cloud is dense enough (it needs not to be as dense as the cloud of atomic nuclei, though) this does not conflict with using the original continuous model for force generation and reaction to forces.

3.2 Adding kinematics

Consider G, that was at rest so far, moving around and record the configuration at times close to some point in time t. Since G moves as a rigid body, there are quantities

$$x_t \in \mathcal{P}\,,\ \mathbf{r}_t \in \mathcal{V}\,,\ \mathbf{v}_t \in \mathcal{V}\,,\ \omega_t \in \mathcal{V} \tag{40}$$

such that the occupied space is given by (see (18) for the representation of small rotations in terms of angular velocity)

$$S(t+\Delta t) \;=\; \cup_{i=1}^{n}\, \mathcal{S}(r_i\,,\, x_t + \Delta t\, \mathbf{v}_t + \exp(\Delta t\, \omega_t) * \mathbf{r}_t * \mathbf{x}_i) \tag{41}$$

to first order in Δt. Here, $\mathbf{v}_t$ and ω_t are the instantaneous values of *velocity* and *angular velocity* respectively. Thus it is natural to speak of x_t and $\mathbf{r}_t$ as instantaneous values of *position* and *angular position*. Especially for $\Delta t = 0$ we have

$$S(t) \;=\; \cup_{i=1}^{n}\, \mathcal{S}(r_i\,,\, x_t + \mathbf{r}_t * \mathbf{x}_i)\;. \tag{42}$$

It is instructive to consider the configuration belonging to $S(t)$ as an initial configuration and let the algorithm (ξ, ι) act on it. As was to be expected, one gets

$$\xi(\pi,\ (r_i)_{i=1}^{n}\,,\ (x_t + \mathbf{r}_t * \mathbf{x}_i)_{i=1}^{n}) \;=\; x_t\;, \tag{43}$$

$$\iota(\pi,\ (r_i)_{i=1}^{n}\,,\ (x_t + \mathbf{r}_t * \mathbf{x}_i)_{i=1}^{n}) \;=\; (\pi,\ (r_i)_{i=1}^{n}\,,\ r,\ (\mathbf{r}_t * \mathbf{x}_i)_{i=1}^{n}\,,\ m,\ (I_\alpha)_{\alpha=1}^{3}\,,\ (\mathbf{r}_t * \mathbf{e}_\alpha)_{\alpha=1}^{3})\;. \tag{44}$$

The quantities from (40) complete the list g of state descriptors of G:

$$g := (g_c\,,\, x\,,\, \mathbf{r}\,,\, \mathbf{v}\,,\, \omega) := (\pi,\ (r_i)_{i=1}^{n}\,,\ r,\ (\mathbf{x}_i)_{i=1}^{n}\,,\ m,\ (I_\alpha)_{\alpha=1}^{3}\,,\ (\mathbf{e}_\alpha)_{\alpha=1}^{3}\,,\ x,\ \mathbf{r}\,,\ \mathbf{v}\,,\ \omega)\;. \tag{45}$$

The first seven components g_c of g are constants (or parameters) which conserve the value that function (37) gave them with the initial configuration as input. The remaining components, however, are the *dynamical variables* of a rigid body. This means in particular, that x is now the center of mass of the grain in the particular *state* symbolized by g and not the center of mass of the grain's initial configuration. The set $\mathcal{G}$ of all such lists that possibly arise from the previous defining algorithm can be characterized as

$$\mathcal{G} := \mathcal{G}_c \times \mathcal{P} \times \mathcal{V} \times \mathcal{V} \times \mathcal{V} := \iota(\mathcal{C}) \times \mathcal{P} \times \mathcal{V} \times \mathcal{V} \times \mathcal{V}\;. \tag{46}$$

where the factors hold the *state constants*, the position, the angular position, the velocity, and the angular velocity respectively. With the rigid body in state g we associate the *momentum* $m\mathbf{v}$, the *angular momentum* $I_{\mathbf{r}}(\omega)$ (see equations (34) and (16)), the *kinetic energy* $\frac{1}{2}m\mathbf{v}\cdot\mathbf{v} + \frac{1}{2}\omega\cdot I_{\mathbf{r}}(\omega)$, and the velocity $\mathbf{v}(p) := \mathbf{v} + \omega \times (p - x)$ [9] of any body-fixed point p.

[9] In [24] ω and $(p-x)$ are interchanged by mistake.

4 Repulsive contact interaction of polyspherical grains

Consider polyspherical grains G and G' in states g and g'. Since grains are rigid bodies, their dynamics is not yet determined by giving the force $\mathbf{F}(g,g')$ which G feels due to the presence of G'. We need the torque $\mathbf{N}(g,g')$ too. It is a particularity of the polyspherical grain geometry that these two quantities are given by computationally simple formulae from the forces between spherical particles. In the present section we leave these forces unspecified and only assume that they vanish between particles the shapes of which don't intersect and that the geometrical center point of the intersection zone can be considered the point where this contact force acts. From the state descriptors g and g' only the components that are not related to inertia:

$$(\pi\,,\,(r_i)_{i=1}^n\,,\,r\,,\,(\mathbf{x}_i)_{i=1}^n\,,\,x\,,\,\mathbf{r}\,,\,\mathbf{v}\,,\,\omega)\,,\quad(\pi'\,,\,(r'_j)_{j=1}^{n'}\,,\,r'\,,\,(\mathbf{x}'_j)_{j=1}^{n'}\,,\,x'\,,\,\mathbf{r}'\,,\,\mathbf{v}'\,,\,\omega')\,, \tag{47}$$

enter the definition of $\mathbf{F}(g,g')$ and $\mathbf{N}(g,g')$:

$$\mathbf{F}(g,g') := \text{ if } |x-x'|-r-r'>0 \text{ then } \mathbf{0} \text{ else } \sum_{i=1}^{n}\sum_{j=1}^{n'}\mathbf{F}_{ij}\,, \tag{48}$$

where

$$\mathbf{F}_{ij} := \text{ if } |c_i-c'_j|-r_i-r'_j>0 \text{ then } \mathbf{0} \text{ else } \mathcal{F}(\pi,c_i,r_i;\,\pi',c'_j,r'_j;\,\mathbf{n}_{ij},\mathbf{w}_{ij})\,, \tag{49}$$

where $\mathcal{F}:(\Pi\times\mathcal{P}\times\mathbb{R}_+)^2\times\mathcal{V}\times\mathcal{V}\to\mathcal{V}$ is a model specific function [10] and its arguments are defined as follows [11]:

$$\begin{aligned}
c_i &:= x+\mathbf{r}*\mathbf{x}_i\,,\qquad c'_j := x'+\mathbf{r}'*\mathbf{x}'_j\,,\\
\mathbf{n}_{ij} &:= \frac{c_i-c'_j}{|c_i-c'_j|}\,,\\
c_{ij} &:= \frac{1}{2}(c_i+c'_j)+\frac{r'_j-r_i}{2}\,\mathbf{n}_{ij}\,,\\
\mathbf{w}_{ij} &:= \mathbf{v}'+\omega'\times(c_{ij}-x')-\mathbf{v}-\omega\times(c_{ij}-x)\,.
\end{aligned} \tag{50}$$

The *contact point* c_{ij} allows us to associate a well defined torque with each of the forces $\mathbf{F}_{ij}$:

$$\mathbf{N}(g,g') := \text{ if } |x-x'|-r-r'>0 \text{ then } \mathbf{0} \text{ else } \sum_{i=1}^{n}\sum_{j=1}^{n'}(c_{ij}-x)\times\mathbf{F}_{ij}\,. \tag{51}$$

It also allows us to define a single vector of relative velocity $\mathbf{w}_{ij}$ (instead of a distribution of such velocities) that can be used as the basic determinant of friction forces.

[10] the argument structure ('declaration' for programmers) of this function is part of the present framework, the underlying algorithm ('implementation') depends on the specifics of the intended application. The implementation (53) should be considered a pattern for importing the descriptors of any other force model of interest into the present framework.

[11] In [24] the arguments of the vector products are exchanged by mistake.

It is important to compute the present positions c_i, c'_j of the spheres always from the constant shape vectors and the dynamical variables x and $\mathbf{r}$. Only then the shape remains exactly constant during arbitrarily long simulations, wheres developing the positions of the spheres from time-step to time-step without a memory of the initial shape lets the shape undergo unacceptable deformations by numerical noise, unless some shape restoration strategy is implemented.

In many practical situations, deformations of grains in contact are tiny, so that in the present model the zones of overlap between components of different grains will be tiny. Then the concept of a contact point as used above is clearly appropriate. It should be noticed, however, that the spirit of the soft-particle model is to make particles softer than they are in reality in order to allow larger time steps in simulations of dynamics. Then the zones of overlap may cover a substantial part of the smooth surface fragments of grains and probably will extend over two or more such fragments; this amplifies the repelling force in the transition region between spheres so that the profile of the grain gets flattened. Also the concept of a contact point becomes vague. This is not a conceptual problem since the very definition of the shape as a union of spheres is also only an approximation to the shapes that occur in a real granular system for which the present framework is intended to provide a model.

The force $\mathbf{F}(g,t)$ and the torque $\mathbf{N}(g,t)$ which G feels at time t due to the presence of external fields—such as gravity—and of confining walls—such as the supporting ground for a pile of sand— can be easily introduced along these lines; this will not be carried out here but is implemented in *PaLa*.

5 A force model for polyspherical grains based on Hertz formulas for repulsion of elastic spheres

Here I describe the contact forces (excluding adhesion which significantly influences the toning process) and the friction forces in the form that worked best in toning simulations and which is implemented in *PaLa*. Here one uses the material data

$$\pi := (\rho\,,\, E\,,\, \sigma,\, \delta,\, \mu,\, \nu) \tag{52}$$

where the meaning is as follows:

1. E: Young's modulus ($0 \leq E$)
2. σ: Poisson's ratio ($0 \leq \sigma \leq 0.5$)
3. δ: square of the coefficient of normal restitution ($0 \leq \delta \leq 1$)
4. μ: coefficient of friction ($0 \leq \mu \leq 1$)
5. ν: a small regularizing velocity to smooth out the direction discontinuity of the friction law. For an integrating time step Δt one should have $0 \leq \nu\,\Delta t \ll r_i$ for all $i \in \{1,\ldots,n\}$.

The function $\mathcal{F}$ of the previous section is implemented as follows:

$$\mathcal{F}(\pi, c, r;\, \pi', c', r';\, \mathbf{n}, \mathbf{w}) := \text{ if } d > 0 \text{ then } \mathbf{0} \text{ else } F_n \mathbf{n} - \frac{\tilde{\mu} F_n}{\tilde{\nu} + w_t} \mathbf{w}_t \,, \tag{53}$$

where

$$d := |c - c'| - r - r' \,, \quad \mathbf{w}_t := \mathbf{w} - w_n \mathbf{n} \,, \text{ where } w_n := \mathbf{w} \cdot \mathbf{n} \,, \quad w_t := |\mathbf{w}_t| \,. \tag{54}$$

The step to define $\tilde{\mu}$ and $\tilde{\nu}$ will be formulated as to give also quantities to be needed in the next step: Take from π the data $E, \sigma, \delta, \mu, \nu$ and from π' the corresponding data $E', \sigma', \delta', \mu', \nu'$. Denoting by $H(x, y)$ the *harmonic mean* of x and y, we define

$$\tilde{E} := H(\frac{E}{1-\sigma^2}, \frac{E'}{1-\sigma'^2}) \,,\ \tilde{r} := H(r, r') \,,\ \tilde{\delta} := H(\delta, \delta') \,,\ \tilde{\mu} := H(\mu, \mu') \,,\ \tilde{\nu} := H(\nu, \nu') \tag{55}$$

and

$$F_n := \text{ if } w_n > 0 \text{ then } \tilde{\delta} \cdot F_{\text{Hertz}} \text{ else } F_{\text{Hertz}} \,, \tag{56}$$

where

$$F_{\text{Hertz}} := \frac{\sqrt{3}}{2} \tilde{E} \sqrt{\tilde{r}} (-d)^{3/2} \,. \tag{57}$$

Here the usage of the harmonic mean to combine data from various grains, is well-founded within the Hertz formula [19]; it is only a very formal device for the dissipation descriptors. In the application [10] I did not use different values for these quantities for different grains, so that the question did not arise. If different values *are* to be used, it is necessary to combine material data from different grains in a symmetric manner—otherwise one would violate $\mathbf{F}(g, g') = -\mathbf{F}(g', g)$. To be sure, the Hertz formula is here taken only as a semi-realistic model for elastic repulsion; certainly the areas where the surfaces of two overlapping spheres meet will be harder to deform than the surface of a free sphere, to which (57) applies exclusively.

Here we employ conventional sliding friction with a *coefficient of friction* μ; this has not to account for slip stick effects since sticking is here an effect of grain geometry which lets projecting parts of one grain get embedded in recessing parts of the other grain. Only if forces are strong enough to unhinge this connection, relative translation becomes possible. Relative motion then gets damped by usual sliding friction that is directed against the tangential relative velocity and proportional to the normal force F_n. In order to avoid instabilities in time-stepping dynamics one better lets the friction force go through zero when the tangential velocity changes direction. This is the role of the small velocity ν.

6 Free motion of the rigid body

An interesting fact concerning the free motion of rigid bodies is that—unlike the translational velocity—the angular velocity is not constant. However, conservation of angular momentum tells how angular velocity has to change during a time step. Here I give my

version of a time stepping algorithm based on the *direct midpoint method* [20],[21]. The time evolution in state space for a short time span Δt is written as

$$\Phi : \mathbb{R} \times \mathcal{G} \to \mathcal{G} , \quad (\Delta t , g_c , x , \mathbf{r} , \mathbf{v} , \omega) \mapsto (g_c , \underline{x} , \underline{\mathbf{r}} , \underline{\mathbf{v}} , \underline{\omega}) , \tag{58}$$

where

$$\begin{aligned}
\underline{x} &:= x + \Delta t \, \mathbf{v} , \\
\underline{\mathbf{v}} &:= \mathbf{v} , \\
\delta \mathbf{r} &:= \exp\left(\frac{\Delta t}{2}\omega\right) \approx \frac{\Delta t}{4}\omega , \\
\Delta \omega &:= \Delta t \, I_{\mathbf{r}}^{-1} \left(I_{\delta \mathbf{r} \circ \mathbf{r}}(\omega) \times \omega\right) , \\
\underline{\omega} &:= \omega + \Delta \omega , \\
\Delta \mathbf{r} &:= \exp\left(\Delta t \, \frac{\omega + \underline{\omega}}{2}\right) \approx \frac{\Delta t}{4}(\omega + \underline{\omega}) , \\
\underline{\mathbf{r}} &:= \Delta \mathbf{r} \circ \mathbf{r} .
\end{aligned} \tag{59}$$

Here, the exponential function is introduced only to make the terms easier to understand, actually the approximative terms are always sufficient. Recall (16) for the meaning of rotation vectors as subscripts to linear mappings and (7) for the operation $\circ$ for rotation vectors. In the limit $\Delta t \to 0$ this reduces to the following equations

$$\dot{\mathbf{v}} = \mathbf{0} , \quad \dot{\omega} = I_{\mathbf{r}}^{-1}(I_{\mathbf{r}}(\omega) \times \omega) \tag{60}$$

for the accelerations. These simply express conservation of the linear and angular momentum (see (21)). As it is characteristic for the direct midpoint method, the integrator step (59) is divided into three sub-steps: (1) motion to the midpoint (in time) with constant velocity (2) instantaneous change of the velocity according to the dynamical law, and (3) motion with the new velocity held constant during the last half of the time step. After the first substep the angular position is just $\delta \mathbf{r} \circ \mathbf{r}$ and we understand $\Delta \omega$ in (59) as $\dot{\omega} \Delta t$ with $\dot{\omega}$ from (60). That the angular position $\mathbf{r}$ is shifted to $\delta \mathbf{r} \circ \mathbf{r}$ only in one place and not also in $I_{\mathbf{r}}^{-1}$ is the result of experimentation as indicated at the end of the introduction.

7 Dynamics of systems of grains

As building blocks for the final time stepping algorithm we define, using the notation (45):

$$\Gamma_1 : \mathcal{G} \times \mathcal{V} \to \mathcal{V} , \quad (g , \mathbf{F}) \mapsto m^{-1} \mathbf{F} , \tag{61}$$

$$\Gamma_2 : \mathcal{G} \times \mathcal{V} \to \mathcal{V} , \quad (g , \mathbf{N}) \mapsto I_{\mathbf{r}}^{-1} \mathbf{N} , \tag{62}$$

$$\Gamma : \mathbb{R} \times \mathcal{G} \times \mathcal{V} \times \mathcal{V} \to \mathcal{G} , \quad (\Delta t , g , \alpha , \beta) \mapsto (g_c , x , \mathbf{r} , \mathbf{v} + \Delta t \, \alpha , \omega + \Delta t \, \beta) . \tag{63}$$

Now, we have in mind a system consisting of p grains and ask for an algorithm (integrator) for updating the state of the system after elapse of the time span Δt . Among

the state descriptors there is also the time, which is essential in situations where the influences of the environment to the grains depend on time. In order to get an integrator of second order also in the case of velocity-dependent forces, such as friction or Lorentz-forces on moving charges, we extend the state space $\mathcal{G}$ of grains to $\mathcal{G}_e := \mathcal{G} \times \mathcal{V} \times \mathcal{V}$ by adding the accelerations $\alpha := \dot{\mathbf{v}}$, $\beta := \dot{\omega}$. This introduces no new degrees of freedom since all these accelerations get set to $\mathbf{0}$ in the initial state. Then the integrator is a function

$$\Psi : \mathbb{R} \times \mathbb{R} \times \mathcal{G}_e^p \to \mathbb{R} \times \mathcal{G}_e^p , \quad (\Delta t , t , (g_k , \alpha_k , \beta_k)_{k=1}^p) \mapsto (\underline{t} , (\underline{g}_k , \underline{\alpha}_k , \underline{\beta}_k)_{k=1}^p) . \tag{64}$$

Again, we apply the direct midpoint method that changes the velocities in the middle of a time step in accordance with the equation of motion, wheres time evolution to this midpoint and from the midpoint is free [12] .

The first sub-step performs free motion for a span $\tau := \frac{\Delta t}{2}$ of time:

$$\hat{t} := t + \tau , \quad \text{do for } k = 1, \ldots, p \quad \acute{g}_k := \Phi(\tau , g_k) . \tag{65}$$

The second sub-step is interaction; it changes the velocities, not the positions. It consists of three phases.

Phase 1: Correcting the velocities according to the accelerations known from the previous step:

$$\text{do for } k = 1, \ldots, p \quad \hat{g}_k := \Gamma(\tau , \acute{g}_k , \alpha_k , \beta_k) . \tag{66}$$

Phase 2: Forming the forces and torques between grains and from the environment (confining walls and external fields):

$$\begin{aligned} \text{do for } k = 1, \ldots, p \quad \mathbf{F}_k &:= \sum_{l=1, l \neq k}^{p} \mathbf{F}(\hat{g}_k, \hat{g}_l) + \mathbf{F}(\hat{g}_k, \hat{t}) , \\ \mathbf{N}_k &:= \sum_{l=1, l \neq k}^{p} \mathbf{N}(\hat{g}_k, \hat{g}_l) + \mathbf{N}(\hat{g}_k, \hat{t}) , \\ \underline{\alpha}_k &:= \Gamma_1(\hat{g}_k , \mathbf{F}_k) , \quad \underline{\beta}_k := \Gamma_2(\hat{g}_k , \mathbf{N}_k) . \end{aligned} \tag{67}$$

Phase 3: Correcting the velocities according to new and old accelerations:

$$\text{do for } k = 1, \ldots, p \quad \grave{g}_k := \Gamma\Big(\tau , \hat{g}_k , 2\underline{\alpha}_k - \alpha_k , 2\underline{\beta}_k - \beta_k\Big) . \tag{68}$$

And the third sub-step, again, performs free motion:

$$\underline{t} := \hat{t} + \tau , \quad \text{do for } k = 1, \ldots, p \quad \underline{g}_k := \Phi(\tau , \grave{g}_k) . \tag{69}$$

Notice that this integrator can be applied to any many-body problem in which the angular position of bodies is described by rotation vectors and in which mutual forces and torques can be expressed in terms of the state variables of the bodies.

[12] This is analogous to the representation of propagators in quantum mechanical perturbation theory of first order, where this structure is reflected in the well-known first order Feynman diagrams showing two straight lines that meet at an 'interaction vertex'.

8 A numerical example on energy conservation

Although it is very important for a granular systems integrator to cope with friction, it is even more important that the total energy remains approximately constant in the absence of friction. Otherwise, the energy dissipation observed in a simulation would be an obscure mixture of effects originating from the dissipation mechanisms built into the model and from energy production due to numerical artifacts.

To get an impression of the numerical behavior of the algorithm of Sections 6, 7 we therefore consider conservation of energy for a small and simple system for which friction is disabled (by giving all friction coefficients the value 0 and all coefficients of normal restitution the value 1). The system consists of ten grains moving inside a spherical cavity (radius 0.0442, physical data are given as numbers that are to be understood relative to SI units). The grains, are all copies of a single master grain. This is made of four overlapping spheres. The volume of the grain is that of a sphere of radius 0.005 and the radius of a minimum bounding sphere around the center of mass is 0.00604. Density 1250 and Young's modulus $5 \cdot 10^8$ are those of silicon rubber. The grains are initially at rest at random positions near to an equatorial plane inside the sphere. A homogeneous acceleration field (like gravitation but about 2040 times stronger) perpendicular to the equatorial plane lets the particles collide with the enclosing wall which has the same value of Young's modulus as the grains. The concave spherical wall tends to concentrate the reflected particles. Therefore, many collisions happen in which at least three bodies are involved simultaneously. This is what the system is designed for: It should test conservation of total energy in a situation in which the distribution of energy over the various degrees of freedoms changes rapidly and all degrees of freedom get excited.

All data, especially the number of grains and the number of constituents of a grain are mere numbers in a text file that controls the execution of the *PaLa* program (see Section 1). More particles would make the figures 1 and 2 more busy and therefore harder to print and to understand.

Figure 1 shows only six of the ten trajectories in order to keep them separable at least in the beginning of the free fall phase. The wire frame representation of polyspherical grains is straightforward: From the center of mass of a grain lines are drawn to the centers of the spherical components. Each such component is represented by three orthogonal (in space) lines which are radii of the sphere. One of these radii has the same direction as the line from the center of mass to the center of the sphere. This representation is not easy to interpret when seen printed on paper but it is very suitable for an anaglyphic stereo representation. The *PaLa* program allows each run to be recorded as a 'movie'-textfile and to be replayed under flexible control of viewing geometry (stereo or planar, eye position, viewing direction, viewing angle) and viewing speed. Since a single grain is represented by only a few lines, one can look (in stereo mode) into a dense cluster of grains and gets a good impression of the relative positions of surprisingly many grains.

In order to test energy conservation, one has to derive an expression for the potential energy of the Hertz-forces introduced in Section 5 and to extend these formulas to a system that is enclosed in a cavity the walls of which repel the particles by Hertz-forces.

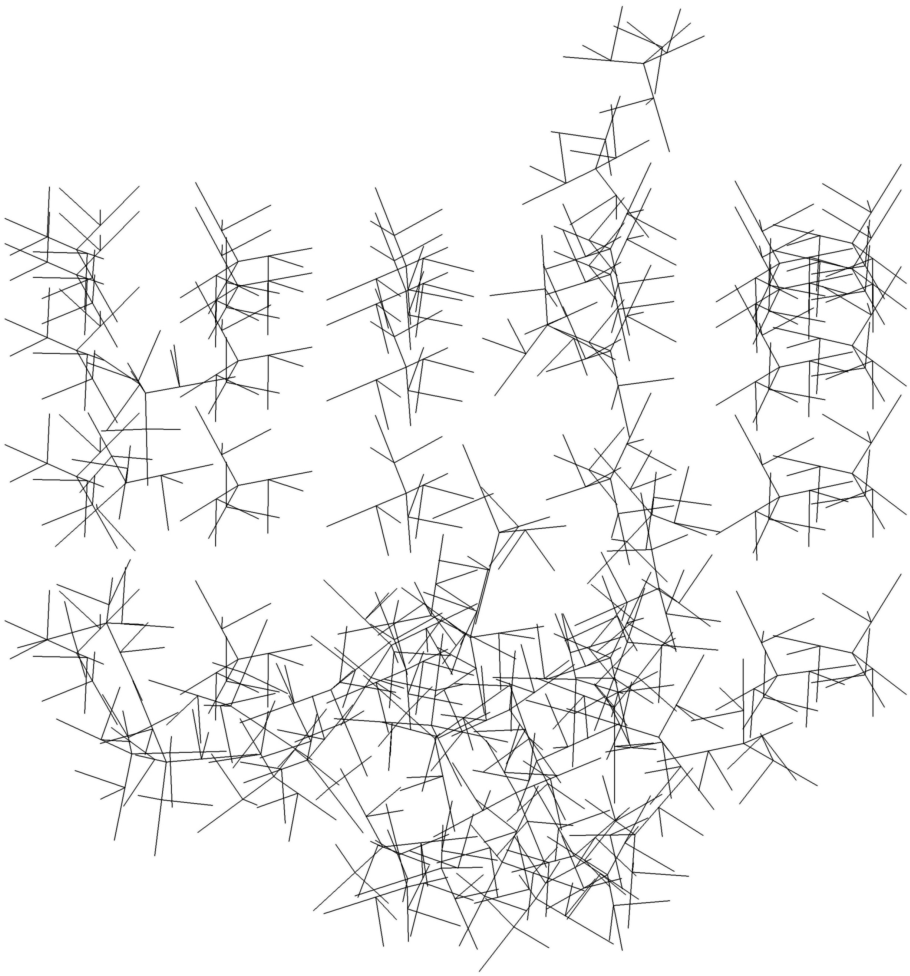

Figure 1: Trajectories of six of the ten particles

One also has to include into the energy expression the potential energy in the acceleration field. All this is straightforward and is implemented in *PaLa* . Figure 2 shows the following interesting behavior of the total energy: During the first impact of most of the grains with the wall the total energy deviates from constancy by an error term that is proportional to the square of the time step. After a phase of frequent collisions the energy comes back to the original value with a surprising accuracy. This phenomenon of *energy restoration* (see also [21], text following equation (86)) looks curious since the system seems to memorize this value although the direct memory of the time stepping algorithm lasts no longer than one time step. This indicates that there may be a truly

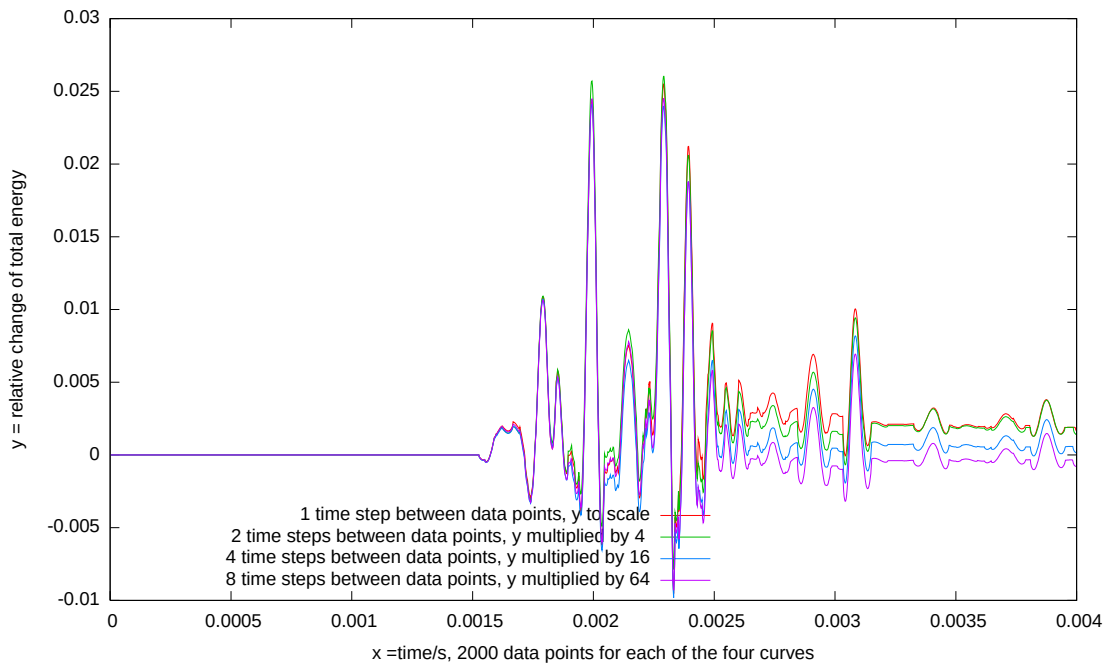

Figure 2: Scaled relative change of total energy for various values of the time step

conserved (up to roundoff errors) energy function in the present time-discrete model.

For symplectic integrators of Hamiltonian systems (which is not exactly our present framework) it has been shown (e.g. [23] equation (10.5) and [22] equation (38)) that exactly conserved energy functions for the time-discrete system exist if these are allowed to depend on the time step. When displaying the energy expression inferred from the time-continuous model one expects to get a curve that wiggles around the constant value of the exactly conserved function. In the formula in [22], the difference between the two energy expressions is second order in the time step which would give a scaling behavior of the wiggling curve just as in Figure 2.

It might be instructive to see the logical elements of this interpretation exemplified with a simple system in which everything can be done explicitly: Consider a particle of mass m moving in one-dimensional space under the influence of a potential $V(x)$. Among the standard methods for solving the equation of motion of this particle is to make use of conservation of energy which reduces the problem to a single integration. Imitating this method in a time-discrete framework we get the following implicit inte-

grator (which invites iteration, starting with $a = 0$ [13])

$$\begin{aligned} t_{n+1} &= t_n + h = 2t_n - t_{n-1} \ , \\ x(t_{n+1}) &= a h^2 + 2x(t_n) - x(t_{n-1}) \ , \\ m a &= -\frac{V(x(t_{n+1})) - V(x(t_{n-1}))}{x(t_{n+1}) - x(t_{n-1})} \ . \end{aligned} \tag{70}$$

If one associates with the time interval $[t_n, t_{n-1}]$ the energy

$$E_n := \frac{m}{2}\left(\frac{x(t_n) - x(t_{n-1})}{h}\right)^2 + \frac{V(x(t_n)) + V(x(t_{n-1}))}{2} \tag{71}$$

then one gets exact conservation:

$$E_{n+1} = E_n \tag{72}$$

independent of the size of the time step h. Assume, we got method (70) from somewhere or we invented it based on heuristics different from energy conservation. For the same reasons as in our previous granular system we want to test the integrator (70) for energy conservation. We probably would represent energy as

$$\mathcal{E}_n := \frac{m}{2}\left(\frac{x(t_{n+1}) - x(t_{n-1})}{2h}\right)^2 + V(x(t_n)) \tag{73}$$

and would find (73) wiggling around some constant value, which we would not easily identify as the constant value of (71). In this case the wiggle displacement can be computed quite explicitly

$$\mathcal{E}_n - E_n = \frac{h^2}{8}\left(m a^2 - 2V''_{n-1}\, v_n\, v_{n+1}\right) + O(h^3) \ , \tag{74}$$

where

$$V''_i := V''(x(t_i)) \ , \quad v_i := \frac{x(t_i) - x(t_{i-1})}{h} \ . \tag{75}$$

For sufficiently small h the $O(h^3)$-contribution can be neglected, and thus in all space regions of constant potential (where $a = 0$ and $V''_{n-1} = 0$) the value of $\mathcal{E}$ comes precisely back to the constant value of E, and we thus observe energy restoration.

[13] In [24] the minus sign in the formula for a is missing by mistake.

9 Conclusion

The polyspherical grain model, together with the rigid body integrator of Sections 6, 7 started as a conceptual experiment on a PC and grew into a large application running on a super computing cluster at Cornell University. The unexpectedly robust and flexible behavior of the program then motivated to shrink it again to the more manageable complexity of the *PaLa* program for careful analysis and optimization of its basic properties. With the status reported here, the method seems to be ready for getting again be applied to large granular systems. With the modest set of functions and variables as used in the present formulation, a new coding effort could take advantage of existing efficient data structures and functions for parallel programming (such as those of *MPI*) instead of following my method which was to take from *MPI* only two functions for sending and receiving character strings and to implement all higher communication patterns myself.

Acknowledgments

I would like to thank Sean McNamara for valuable suggestions that helped to increase the readability of this paper, and to Thomas Dera, Helmut Domes, Thomas M. Plutchak, Eric C. Stelter, and John A. Zollweg for support and valuable contributions to the toning simulation project.

References

[1] Eric C. Stelter, Joseph E. Guth, Matthias H. Regelsberger, Edward M. Eck, Ulrich Mutze : Electrophotographic image developing process with optimized average developer bulk velocity, US patent 6728503, 04/27/2004

[2] D.E.Wolf: Modelling and Computer Simulation of Granular Media; in K.H. Hoffmann, M. Schreiber (editors) Computational Physics, Springer 1996

[3] B. Muth, P. Eberhard, S. Luding: Collisions between particles of complex shape; in Powders and Grains (2005) - García-Rojo, Herrmann and McNamara (eds) Taylor & Francis Group, London, 2005 (cited in the following as P&G05), Volume 2, 1379-1383

[4] A.A. Peña, H.J. Herrmann, A. Lizcano, F. Alonso-Marroquín: Investigation of the asymptotic states of granular materials using a discrete model of anisotropic particles, P&G05, Volume 1, 697-700

[5] R. García-Rojo, S. McNamara, A.A. Peña, H.J. Herrmann: Sliding and localization in a biaxial test of granular material, P&G05, Volume 1, 705-708

[6] L. Pournin, T.M. Liebling: A generalization of distinct element method to tridimensional particles with complex shape, P&G05, Volume 2, 1375-1378

[7] T. Matsushima: Effects of irregular grain shape on quasi-static shear behavior of granular assembly; in Powders and Grains, P&G05), Volume 2, 1319-1323

[8] C. O'Sullivan: The importance of accurately capturing particle geometry in DEM simulations, P&G05, Volume 2, 1333-1337

[9] R. Deluzarche, B. Cambou: Modeling grain crushing in rockfill material using discrete 2D model, P&G05, Volume 2, 1409-1412

[10] Ulrich Mutze, Eric Stelter, Thomas Dera: Simulation of Electrophotographic Development, presented at the 19th conference on Non Impact Printing, Sept 28-October 4 2003, New Orleans

[11] L. Li, R.M. Holt: Approaching real grain shape in the simulation of sandstone using DEM, P&G05, Volume 2, 1368-1373

[12] Y.P. Cheng, M.D. Bolton, Y. Nakata: Grain crushing and critical states observed in DEM simulations, P&G05, Volume 2, 1393-1397

[13] Walter Weizel: Lehrbuch der Theoretischen Physik, Band I, Springer, 1955, pp. 68-71

[14] Thomas Dera, 1999, verbal communication

[15] Saunders Mac Lane, Garrett Birkhoff: Algebra, MACMILLAN, 1968, Chapter XII

[16] Thomas Haslwanter: Mathematics of Three-dimensional Eye Rotations, Vision Res. Vol 35, No 12, pp 1727-1739, 1995

[17] Olinde Rodrigues: Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ses déplacements considérés indépendamment des causes qui peuvent les produire, Journal de Mathématique Pures et Appliquées, 5, 380-440, 1840

[18] Ulrich Mutze: Quaternions - Redundancy + Efficiency = Ternions, Mathematical Physics Preprint Archive 05–53, February 2005
www.ma.utexas.edu/mp_arc/c/05/05-53.pdf

[19] L.D. Landau, E.M. Lifschitz: Lehrbuch der Theoretischen Physik VII, Elastizitätstheorie, 5. Auflage, Akademie-Verlag Berlin 1983, Ch. I, §9

[20] Ulrich Mutze: Predicting Classical Motion Directly from the Action Principle II, Mathematical Physics Preprint Archive 1999–271
www.ma.utexas.edu/mp_arc/c/99/99-271.pdf

[21] Ulrich Mutze: A Simple Variational Integrator for General Holonomic Mechanical Systems, Mathematical Physics Preprint Archive 2003–491
www.ma.utexas.edu/mp_arc/c/03/03-491.pdf

[22] J. David Brown: The Midpoint Rule as a Variational-Symplectic Integrator. I. Hamiltonian Systems
arXiv: gr-qc/0511018v1 3 Nov 2005

[23] J.M. Sanz-Serna and M.P. Calvo: Numerical Hamiltonian Problems, Chapman & Hall, 1994

[24] Ulrich Mutze: Rigidly connected overlapping spherical particles: a versatile grain model, Granular Matter Vol. 8, 185-194, 2006

Last modification: 2008-05-02.
The original version is www.ma.utexas.edu/mp_arc/c/07/07-252.pdf