A Letters Journal Exploring
the Frontiers of Physics

July 2008

EPL, **83** (2008) 14001
doi: 10.1209/0295-5075/83/14001

www.epljournal.org

Spheropolygons: A new method to simulate conservative and dissipative interactions between 2D complex-shaped rigid bodies

F. Alonso-Marroquín[(a)]

ESSCC & School of Physical Sciences, The University of Queensland - Qld. 4072, Brisbane, Australia

received 19 November 2007; accepted in final form 15 May 2008
published online 11 June 2008

PACS 45.70.-n – Granular systems
PACS 47.11.Mn – Molecular dynamics methods

Abstract – I present a method to simulate complex-shaped interacting bodies, a problem which appears in many areas, including molecular dynamics, material science, virtual reality, geo- and astrophysics. The particle shape is represented by the classical concept of a Minkowski sum, which permits the representation of complex shapes without the need to define the object as a composite of spherical or convex particles. A well-defined conservative and frictional interaction between these bodies is derived. The model (particles + interactions) is much more efficient, accurate and easier to implement than other models. Simulations with conservative interactions comply with the statistical mechanical principles for conservative systems. Simulations with frictional forces show that particle shape strongly affects the jamming phenomena in granular flow.

Introduction. – Realistic modeling of interacting bodies has a fundamental impact in several research areas. In most areas particle shape plays a key role: i) drug molecules often have to act as a key in a lock formed by a protein cavity, otherwise they lose their activity [1]; ii) liquid crystals consisting of rod-shaped, disk-shaped, or banana-shaped molecules exhibit transition to a nematic phase, which is strongly dependent on particle shape [2]; iii) the microscopic description of geological materials requires the modeling of cohesive-frictional interaction between particles with a wide range of shapes [3]; iv) computing the motion of rigid and articulated bodies can lead to new advances in robotics and automation [4]; v) dynamics simulation of complex-shaped objects has fundamental importance in virtual reality applications, as realistic force feedback between the user and the computer-simulated environment guarantees realism in the simulations [5].

The most typical approach for these applications is to solve the dynamics of interacting rigid bodies, where their real shapes are approximated by polyhedra [4,6,7]. The most difficult aspect for the simulations is to model contact interactions. In computer graphics, or other forms of interactive computing, the interaction between polyhedra is resolved by decomposing them in convex pieces, and applying penalty methods, impulse-based methods or dynamic constraints in the interaction between these pieces. Impulse-based methods allow real-time simulations, but they cannot handle permanent or lasting contacts [4]. On the other hand, constraint methods can handle resting contacts with infinite stiffness, but simulations are computationally expensive and lead in some cases to indeterminacy in the solution of contact forces [6]. This indeterminacy is removed by using penalty methods, where the bodies are allowed to interpenetrate each other and the force is calculated in terms of their overlap. However, the determination of such contact force is still heuristic and lacks physical correctness [7,8].

Scientific applications require high-precision dynamic simulations and reliable interaction models capturing the real physics of the problem. Robust penalty methods have been proposed for two-dimensional (2D) simulations using polygons [3,8]. They allow to gain insight in the further development of three-dimensional (3D) models. In the case of convex polygons, the force is calculated as a function of their overlap area [3]. However, the assumption that elastic force is a function of their overlap leads to a non-conservative elastic interaction [8]. An alternative approach is to assume that the potential elastic energy is a function of the overlap. Then forces and torques are derived from this potential [8]. This approach, however, leads to very expensive force calculations, partly because each particle needs to be decomposed in triangles, and partly because many different interactions cases need to be

[(a)] E-mail: fernando@esscc.uq.edu.au

considered separately. Both approaches are also extremely difficult to extend to 3D, because the calculation of the overlap between two polyhedra is computationally very expensive. This is the main reason why most of the commercial codes for particulate systems are still based on simulations with spheres, or clumps of spheres representing complex-shaped objects [9].

In this letter I present a solution to this problem in 2D, using the concept of spheropolygons. They are generated from the Minkowski sum of a polygon with a disk, which is nothing more than the object resulting from sweeping a disk around a polygon. This simple concept can be used to generate very complex shapes, including non-convex bodies, without the need to decompose them into spherical or convex parts. The interaction between spheropolygons is modeled by considering all possible vertex-edge interactions. The result is a simple and elegant model, where different interaction laws can be implemented straightforwardly. I anticipate that this model will be the starting point to the development of a new generation of particle-based models. These models will capture both complex particle shape and realistic interactions laws in a unified framework, allowing simulations of several systems involved in complex-shaped rigid bodies.

The model. – To solve the interaction between spheropolygons, we adopt the basic assumption of rigid-body dynamics: The particles do not change of shape, and interaction occurs when they overlap. Finite stiffness is attributed to the particles, so that the bodies are not truly rigid as those used in the Contact Dynamics method [10]. The model is implemented in a C++, fully customizable, object-oriented code.

Minkowski sum. Given two sets of points P and Q in an Euclidean space, their Minkowski sum is given by $P+Q=\{\vec{x}+\vec{y}\,|\,\vec{x}\in P,\ \vec{y}\in Q\}$. This operation is geometrically equivalent to the sweeping of one set around the profile of the other without changing the relative orientation. A special case is the sum of a polygon with a disk, which is defined here as spheropolygons. Other examples of a Minkowski sum are the spherocyllinder (sphere + line segment) [11], the spherosimplex (sphere + simplex) [12] and the spheropolyhedron (sphere + polyhedron) [13], which are used in simulations of particulate systems.

A prototype of the spheropolygons is the Minkowski cow shown in part (a) of fig. 1. The Minkowski sum is compared to the clump of disks technique shown in part (b) of fig. 1. The number of vertices required to represent a complex shape with the Minkowski sum is typically lower than the number of disks needed to reconstruct the clumpy object. This property holds when one removes the inner disks of the clumpy object. In the case of the clumpy cow, the total number of disks is 726, and the number of boundary disks is 296, which is still larger than the 62 vertices of the polygon. Another visible advantage of the Minkowski sum approach is that it removes the unwanted roughness of the surface in the clump-of-disks method.

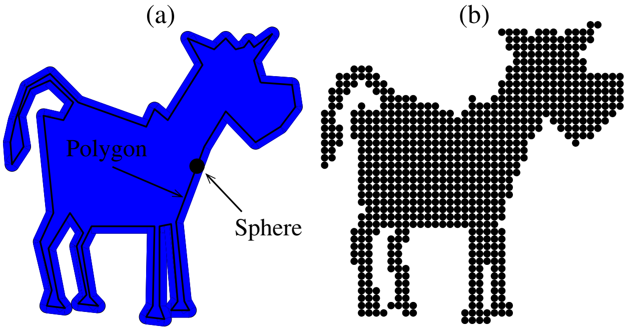

Fig. 1: (a) Minkowski cow obtained by sweeping a disk around a polygon of 62 edges. The disk has radius 10 cm and the length of the polygon is 3 m. (b) The cow as represented by a clump of disks using 726 disks.

The point-inside-spheropolygon test combined with a basic Monte Carlo method is used to evaluate the integral expressions for mass, center of mass and the moment of inertia. This numerical integration, whose details are explained elsewhere [14], does not represent much computational effort, because the mass properties are calculated only once and they are assumed to be constant throughout the simulation.

Interaction force. To solve the interaction between spheropolygons, we consider all vertex-edge distances between the polygons. Let us take two spheropolygons SP_i and SP_j with their respective polygons P_i and P_j and the radii of the disks r_i and r_j. Each polygon is defined by the set of vertices $\{V_i\}$ and edges $\{E_i\}$. The overlapping length between each pair of vertex-edge (V,E) is defined as

$$\delta(V,E)=\langle r_i+r_j-d(V,E)\rangle, \tag{1}$$

where $d(X,E)=\parallel \vec{Y}-\vec{X}\parallel$ is the Euclidean distance from the vertex V to the segment E. Here $\vec{X}$ is the position of the vertex V and $\vec{Y}$ is its closest point on the edge E. The ramp function $\langle x\rangle$ returns x if $x>0$ and zero otherwise.

The force $\vec{F}_{ij}$ acting on particle i by the particle j is defined by

$$\vec{F}_{ij}=-\vec{F}_{ji}=\sum_{V_iE_j}\vec{F}(V_i,E_j)+\sum_{V_jE_i}\vec{F}(V_j,E_i), \tag{2}$$

where $\vec{F}(V,E)$ represents the force between the vertex V and the edge E. If the vertex-edge overlapping length is zero, we take $\vec{F}(V,E)=0$. Different vertex-edge forces can be included in the model: linear dashpots, non-linear Hertzian laws, Lennard-Jones forces, dissipative viscous forces, sliding friction, etc.

The torque on particle i given by j is

$$\begin{aligned}\tau_{ij} &= \sum_{V_iE_j}(\vec{R}(V_i,E_j)-\vec{r}_i)\times\vec{F}(V_i,E_j)\\ &\quad+\sum_{V_jE_i}(\vec{R}(V_j,E_i)-\vec{r}_i)\times\vec{F}(V_j,E_i),\end{aligned} \tag{3}$$

where $\vec{r}_i$ is the center of mass of particle i, and $\vec{R}$ is the point of application of the force, which is taken as the middle point of the overlap region between the vertex and the edge:

$$\vec{R}(V,E) = \vec{X} + \left(r_i + \frac{1}{2}\delta(V,E) \right) \frac{\vec{X} - \vec{Y}}{\parallel \vec{Y} - \vec{X} \parallel}, \qquad (4)$$

The evolution of $\vec{r}_i$ and the orientation φ_i of the particle is governed by the equations of motion:

$$m_i \ddot{\vec{r}}_i = \sum_j \vec{F}_{ij} - m_i g \hat{y}, \quad I_i \ddot{\varphi}_i = \sum_j \tau_{ij}. \qquad (5)$$

Here m_i and I_i are the mass and moment of inertia of the particle. The sum is over all particles interacting with this particle; g is the gravity; and $\hat{y}$ is the unit vector along the vertical direction.

Efficient calculation of forces. The efficiency of the dynamics simulation is mainly determined by the method of contact detection. In a system of N particles, each one with M edges, the number of calculations for contact detection is $O(N^2M^2)$. Simulations therefore become very slow when either the number of particles or the number of vertices is large. In order to speed up the simulations, we create a *neighbor list* of all pair particles which are likely to be in contact. We also use for each element of this neighbor list a *contact list* of those vertex-edge pairs which are in potential overlap. The neighbor list is calculated as the collection of all pair particles whose distance between them is less than 2α, where alpha is called Verlet distance [14]. For each element of the neighbor list, a contact list is defined as the collection of all vertex-edge pairs whose distance is less than 2α. These lists are updated when the maximal displacement over all points of the particles after the last neighbor update is larger than α [14]. A *link cell* algorithm [8] is used to allow rapid calculation of the neighbor list. This method for neighborhood identification using neighbor and contact lists requires little memory storage, and reduce the amount of calculations of contact forces to $O(N)$, which is of the same order as in simulations with circular particles [8].

Time integration. The equations of motion of the system are numerically solved using a four order predictor-corrector algorithm [15]. A pseudocode with the basic procedures in each time step is shown in Algorithm 1. In each iteration, the algorithm visit each instance of the particle class and changes its position, its derivatives, and the vertices of its polygon. The algorithm is similar to the one used in simulations with circular particles, except that it needs an additional procedure to update the vertices of the polygons. This procedure is required also in simulations with polygons, and it involves $O(N,M)$ calculations. However, simulations with spheropolygons are faster than with polygons because the contact force given by eqs. (2) is much simpler than the one used in polygons [3,8]. Note also that the Minkowski sum does not need to be calculated during the time integration, which is, along with the neighborhood identification, one of the main reasons of the efficiency of the code.

Algorithm 1: One time step of the time integration scheme.

```
Input: state of the particles at time $t$
Output: state of the particles at time $t + \Delta t$
predict position of the particles and its derivatives using a Taylor expansion;
update vertices of the polygons;
if neighbor update condition is satisfied then
    calculate link cell;
    update neighbor;
    update contact list;
end
calculate contact forces between neighbor particles;
apply contact forces to the particles;
apply gravity forces to the particles;
correct positions and its derivatives using forces and torques.
```

Benchmark tests. – Simulations will be perform using Minkowski cows interacting via repulsive, frictional forces

$$\vec{F}(V,E) = k_n \delta_n \vec{N} + k_t \delta_t \vec{T}, \qquad (6)$$

where $\vec{N} = (\vec{Y} - \vec{X}) / \parallel \vec{Y} - \vec{X} \parallel$ is the normal unit vector. The tangential vector $\vec{T}$ is taken perpendicular to $\vec{n}$. δ_n is the overlapping length defined in eq. (1). The elastic displacement δ_t accounts frictional forces, and it must satisfy the sliding condition $|F_t| < \mu F_n$, where μ is the coefficient of friction [16]. The parameters of the simulations are the normal stiffness $k = 10^7\,\mathrm{N/m}$, the tangential stiffness $k_t = 0.1 k_n$, friction coefficient $\mu = 0.5$ gravity $g = 10\,\mathrm{m/s^2}$, density $\sigma = 1\,\mathrm{kg/m^2}$, time step $\Delta t = 10^{-5}\,\mathrm{s}$ and the Verlet distance $\alpha = 1\,\mathrm{m}$.

We perform three series of simulations. The first one corresponds to a test with two particles with only repulsive forces ($k_t = 0$). The second series involves many particles interacting via repulsive forces. The third set are granular flow simulations using gravity, along with repulsive and frictional forces.

Cow-landscape interaction. To demonstrate the capability of the model to solve interactions between complex, non-convex–shaped bodies, I simulate a Minkowski cow bouncing on a simple landscape (see fig. 2). The cow initially has zero angular and linear velocity. Periodic boundary conditions are applied in the horizontal directions, which means that when the particle leaves the space domain at one side it appears automatically at the other side. The space domain is given by $0 < x < 20\,\mathrm{m}$.

At time $t = 1.82\,\mathrm{s}$ one of the vertices of the cow get into contact with one of the edges of the landscape. The elastic force produced by this contact transfers part of its linear momentum into angular momentum. This makes the cow rotate. We note a clear difference of the Minkowski cow

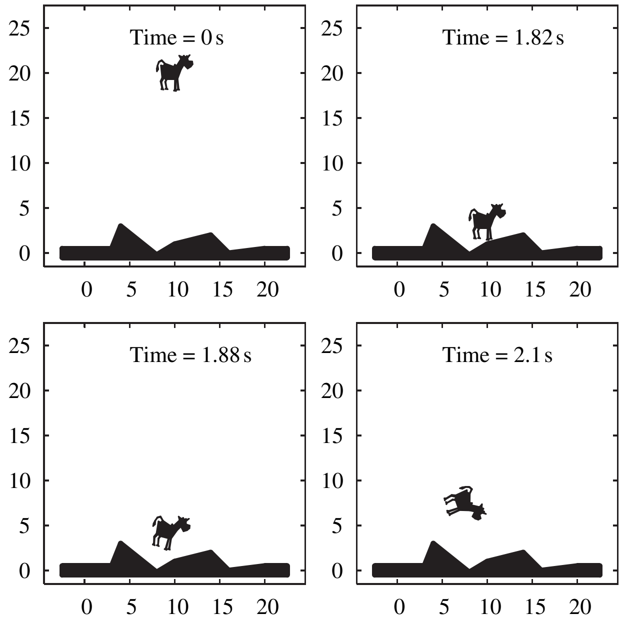

Fig. 2: Four different snapshots of the simulation of a Minkowski cow bouncing on a simple landscape.

with respect to the spherical particles: In the absence of frictional forces, the landscape can transfer angular momentum to the cow during the collision. This is because the force is not collinear with the line connecting the contact point to the center of mass of the object, as expected in non-spherical objects.

The time evolution of the energy components are presented in fig. 3. Elastic energy has a negligible contribution to the energy budget, as it differs from zero only for short times during the collisions. The effect of the interaction force is to produce exchanges of linear and angular kinetic energy during collision. Energy conservation is numerically verified by taking the sum of all energy contributions. This sum remains almost constant, except minute numerical fluctuations due to time discretization. Energy balance is also guaranteed when particles are represented by clump of disks. Yet we will see later that the main advantage of the Minkowski sum method is the high performance of the simulations.

Many-body simulation. I present here simulations with large number of particles with repulsive interactions and no gravity. The simulations will allow us to investigate the statistical-mechanical equilibrium, and to compare the efficiency of the Minkowski sum approach with other methods. First, I present the results of a simulation with 400 Minkowski cows confined by a fixed rectangular box (see fig. 4). Each particle occupies an area of $a = 6.52\,\mathrm{m}^2$ and the area of the box is $A = 10609\,\mathrm{m}^2$. These values lead to a volume fraction of $\Phi = Na/A = 0.246$. Gravity and frictional forces are set to zero. Initially, each particle has zero angular velocity and a linear velocity of 1 m/s with random direction. Due to collisions, the linear momentum of each particle changes and part of it is transferred to angular momentum. As a consequence, the rotational kinetic energy increases as shown in fig. 5. We observe a stationary regime, where the average of rotational kinetic energy reaches the limit of 1/2 the linear kinetic energy. This is in agreement with the theorem

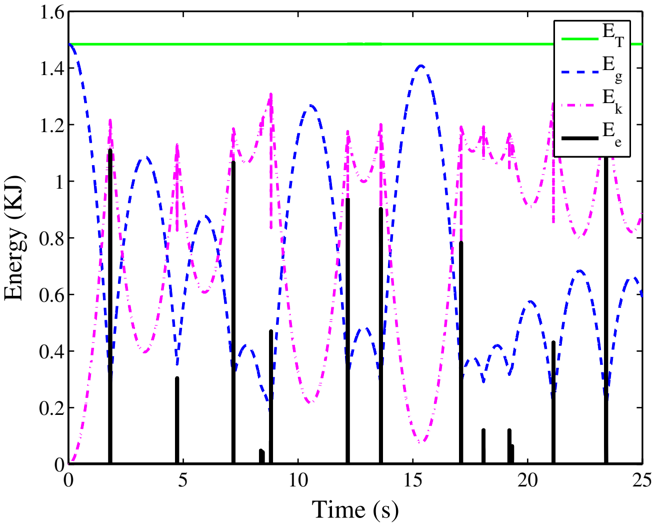

Fig. 3: Energy budget in bouncing cow simulation. The total energy E_T consists of the gravitational energy $E_g = mgy$, the kinetic energy $E_k = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2$, and the potential energy E_e given by $E^e = (k_n/2)(\sum_{V_i E_j} \delta(V_i, E_j) + \sum_{V_j E_i} \delta(V_j, E_i))$. As is expected for a conservative system, the sum of all energy contributions lead to a constant value, with and error which is lower than 0.0001% throughout all the simulation.

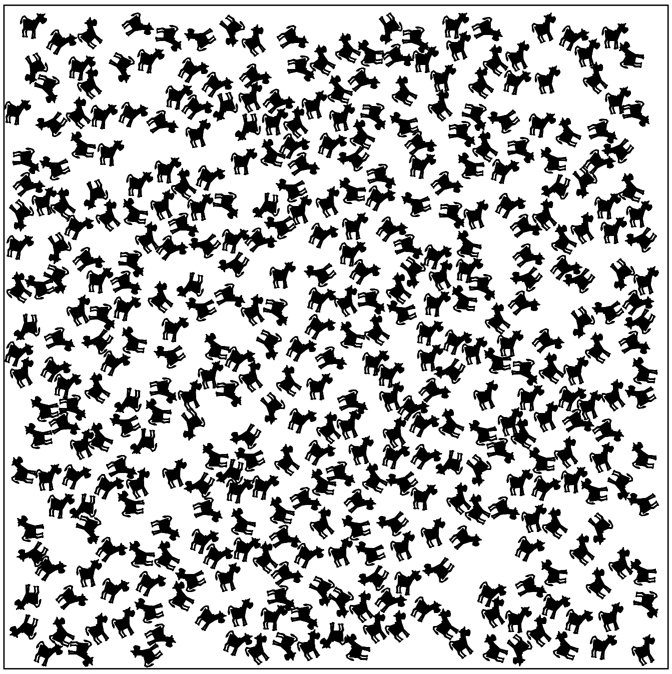

Fig. 4: Snapshot of the simulation with 400 Minkowski cows confined in a rectangular box.

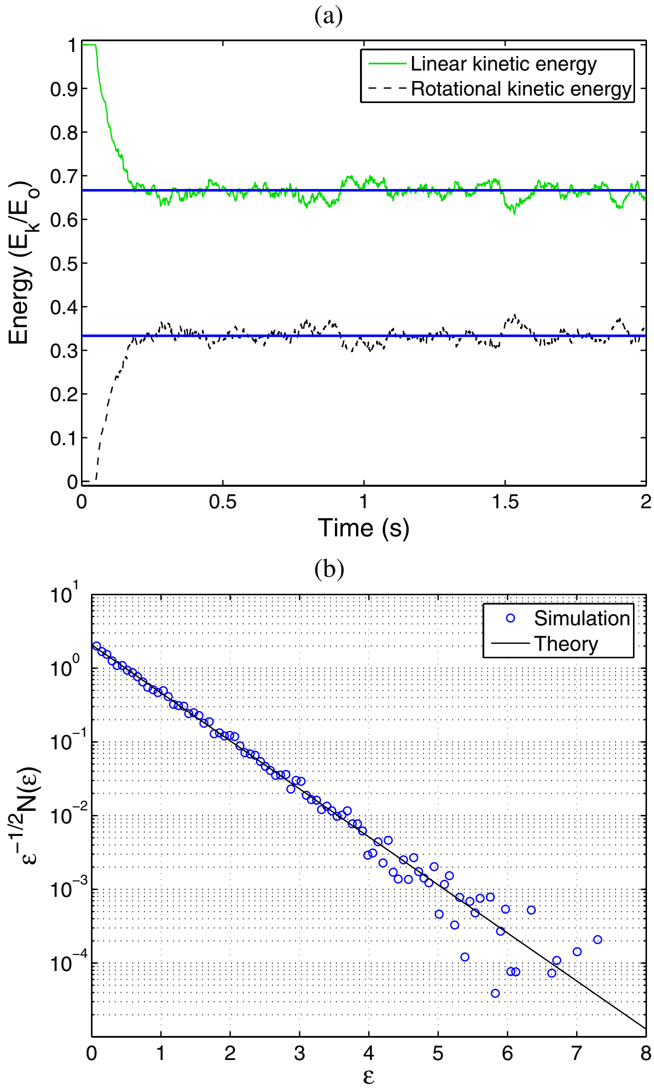

Fig. 5: (a) Time evolution of the total linear and rotational kinetic energy of the particles. E_0 is the initial value of the total kinetic energy. The horizontal lines correspond to the expected value by the equilibrium statistical mechanics. (b) Energy distribution $N(\epsilon)$ for the particles, where $\epsilon = E_k/\bar{E}_k$. The line corresponds to the Maxwell-Boltzmann distribution $n(\epsilon) = 2\beta\sqrt{\epsilon\beta/\pi}\,\exp(-\beta\epsilon)$, with $\beta = 1.5$.

of equipartition of energy [17], which states that, in the statistical equilibrium, each quadratic term in the energy should contribute with the same weight in the mean energy of the system.

We also calculate the energy distribution of the particles in the stationary regime. We take snapshots of the simulations between $t = 1\,$s and $t = 8\,$s distanced by 0.01 s. In each snapshot the kinetic energy of the individual particles is measured. The histogram of the variable $\epsilon = E_k/\bar{E}_k$ (where $E_k = \frac{1}{2}(mv_x^2 + mv_y^2 + I\omega^2)$ is the kinetic energy of the particle) is constructed using 100 identical bins between zero and the maximal value. As shown in fig. 5, the distribution of ϵ can be fitted by the Maxwell-Boltzmann distribution. The existence of statistical equilibrium for this simple system opens new doors to study the thermodynamics of dissipative granular materials subjected to an external driving. This model may provide decisive numerical experiments to evaluate the role of particle shape and interparticle interactions in the statistical-mechanical properties of these granular systems.

Lastly, we compare the efficiency of the many-body simulations of systems consisting on spherical particles, Minkowski cows and clumpy cows. The clumpy cows are simulating by imposing holonomic constraints on the disks, as is used classical molecular dynamics methods [18]. The performance of the simulations is estimated by running different processes in a Pentium 4, 3.0 GHz, and calculating the CPU time during $10s$ simulation in each one of them. The number of particles is ranged between $N = 100$ and $N = 1000$. Simulations with disks are around 50 times faster than simulation with Minkowski cows. However, the speed of the simulation with Minkowski cows is 40 times faster that simulations with clumpy cows. Therefore simulations with spheropolygons are more efficient than those ones with clumps of disks, because the former ones require less elements to represent the particle shape.

Granular flow. In this section we introduce another interesting application of this model: the study of the effect of particle shape on the jamming phenomenon of granular flow. Examples of 2D granular flow are pedestrian or vehicle traffic, transport of baggage in conveyor belts in airports, and gravity flow in a hopper or a hourglass. The flow may happen when particles are discharged through a small opening, but particles may became jammed when the opening is smaller than a critical value. Modeling of gravity flow has been done using circular or spherical particles [19], but the effect of shape on flow has not been thoughtfully investigated. In particular, non-convex particles is expected to jam more easily than convex, or circular particles.

Granular flow with non-convex particles is presented using 400 Minkowski cows within a hopper geometry, see fig. 6. We used apertures between 20 m and 40 m. The results were compared to simulations with disks. Circular particles flow continuously for all these apertures, whereas non-convex particles jam when the aperture is lower than 30 m. Jamming is produced by arches formed near the aperture. Simulations with convex polygons using Pöeschel model show that the length of these arches is around 4–6 particle diameters [8]. This contrasts to our simulations with non-convex particles, where arches may be larger than 20 particle diameters. A more detailed investigation of the flow and jamming of non-convex particle would certainly have interesting industrial applications.

Concluding remarks. – Modeling interacting particles using spheropolygons has several advantages with

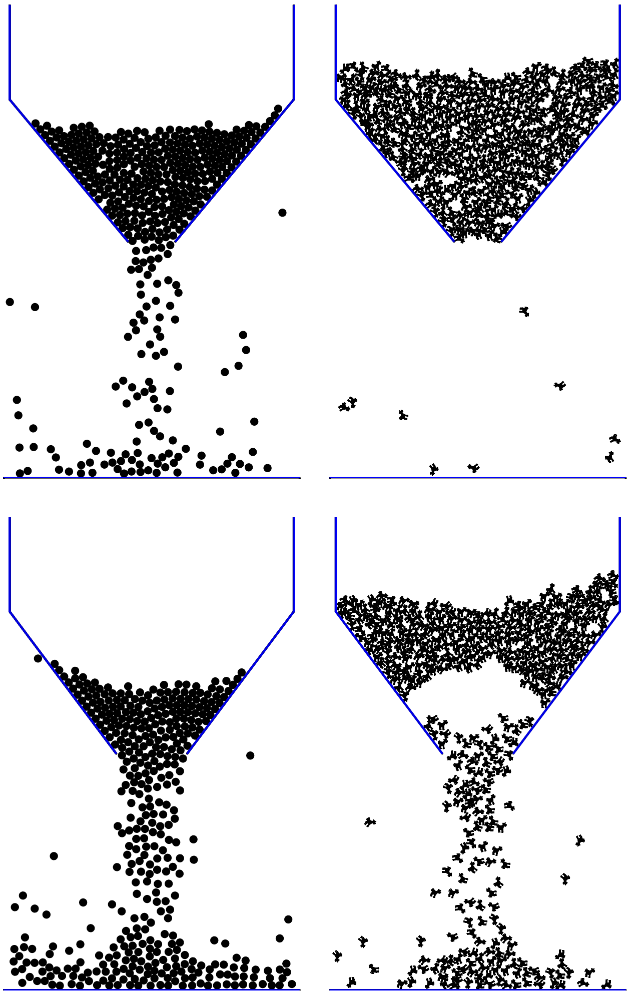

Fig. 6: Simulations of granular flow using disks (left) and Minkowski cows (right). Two different apertures are chosen: 20 m (top) and 30 m (bottom). The snapshots are taken eight seconds after discharge.

respect to other existing particle-based models: i) The possibility to model non-convex particles; ii) a realistic representation of the surface curvature of particles; iii) guaranteed compliance with physical and statistical mechanical laws; iv) balance between accuracy and efficiency. The model is still very simple, but extensions to 3D simulations and more complex interactions are achievable in the near future. 3D modeling using spheropolyhedra requires forces similar to eq. (2), where the sum is over all vertex-face and edge-edge interactions. Special attention is required for the case of two parallel edges in contact. This case lead to a non-uniqueness in the selection of contact points. This need to be resolved to get a physical correctness in the torque calculation.

Applicability of the method ranges to several branches of biology, physics, geo- or even astrophysics. Conservative models can be used to model dynamics of complex molecules. Dissipative systems with complex-shaped constituents may have interesting geophysical problems as in avalanches (snow, debris) and earthquakes, or in astrophysics as the dynamical evolution of precursors of planets or the icy particles forming planetary rings. This work is the first step for the realistic description of conservative and dissipative particulate systems with plenty of potential applications.

***

This work is supported by the Australian Research Council (project number DP0772409)

REFERENCES

[1] Carlson H. and Mccammon A., *Mol. Pharmacol.*, **57** (2000) 213.

[2] Pelzl G., Diele S. and Weissflog W., *Adv. Mater.*, **11** (1999) 707.

[3] Alonso-Marroquin F., Luding S., Herrmann H. and Vardoulakis I., *Phys. Rev. E*, **51** (2005) 051304.

[4] Mirtich B., *ACM Trans. Graph. (TOG)*, **15** (1998) 177.

[5] Ruspini D. and Khatib O., *J. Rob. Syst.*, **18** (2001) 769.

[6] Baraff D., *Algorithmica*, **10** (1993) 292.

[7] Hasegawa S. and Sato M., *Comput. Graph. Forum*, **23** (2004) 529.

[8] Poeschel T. and Schwager T., *Computational Granular Dynamics* (Springer, Berlin) 2004.

[9] Lu M. and McDowell G. R., *Granular Matter*, **9** (2007) 69.

[10] Moreau J., *Unilateral contact and dry friction in finite freedom dynamics,* in *International Centre for Mechanical Sciences, Courses and Lectures*, Vol. **302** (Springer, Vienna) 1988.

[11] Pournin L., Weber M., Tsukahara M., Ferrez J.-A., Ramaioli M. and Liebling T. M., *Granular Matter*, **7** (2005) 119.

[12] Pournin L. and Liebling T., *A generalization of distinct element method to tridimensional particles with complex shapes,* in *Powders & Grains 2005*, Vol. **1375** (Balkema, Leiden) 2005.

[13] Pournin L., *On the behavior of spherical and non-spherical grain assemblies, its modeling and numerical simulation*, PhD Thesis, École Polytechnique Fédérale de Lausanne (2005).

[14] Alonso-Marroquin F. and Wang Y., *An efficient algorithm for granular dynamics simulations with complex-shaped objects*, arXiv:0804.0474 (2008).

[15] Gear C., *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall PTR, Upper Saddle River, NJ) 1971.

[16] García-Rojo R., Alonso-Marroquín F. and Herrmann H., *Phys. Rev. E*, **72** (2005) 41302.

[17] Tolman R., *The Principles of Statistical Mechanics* (Dover Publications, Inc., New York) 1979.

[18] Ryckaert J., Ciccotti G. and Berendsen H., *J. Comput. Phys.*, **23** (1977) 327.

[19] To K., Lai P. and Pak H., *Phys. Rev. Lett.*, **86** (2001) 71.