EPJ Web of Conferences **140**, 06004 (2017) DOI: 10.1051/epjconf/201714006004
Powders & Grains 2017

A parallel Discrete Element Method to model collisions between non-convex particles

Andriarimina Daniel Rakotonirina[1,⋆], *Jean-Yves* Delenne[3,⋆⋆], and *Anthony* Wachs[1,2,⋆⋆⋆]

[1] *Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T 1Z4*
[2] *Department of Chemical & Biological Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z2*
[3] *IATE, UMR 1208 INRA – CIRAD – Montpellier Supagro – Université Montpellier 2, Montpellier, France*

Abstract. In many dry granular and suspension flow configurations, particles can be highly non-spherical. It is now well established in the literature that particle shape affects the flow dynamics or the microstructure of the particles assembly in assorted ways as e.g. compacity of packed bed or heap, dilation under shear, resistance to shear, momentum transfer between translational and angular motions, ability to form arches and block the flow. In this talk, we suggest an accurate and efficient way to model collisions between particles of (almost) arbitrary shape. For that purpose, we develop a Discrete Element Method (DEM) combined with a soft particle contact model. The collision detection algorithm handles contacts between bodies of various shape and size. For non-convex bodies, our strategy is based on decomposing a non-convex body into a set of convex ones. Therefore, our novel method can be called "glued-convex method" (in the sense clumping convex bodies together), as an extension of the popular "glued-spheres" method, and is implemented in our own granular dynamics code Grains3D. Since the whole problem is solved explicitly, our fully-MPI parallelized code Grains3D exhibits a very high scalability when dynamic load balancing is not required. In particular, simulations on up to a few thousands cores in configurations involving up to a few tens of millions of particles can readily be performed. We apply our enhanced numerical model to (i) the collapse of a granular column made of convex particles and (i) the microstructure of a heap of non-convex particles in a cylindrical reactor.

1 Introduction

The Discrete Element Method has been extensively used over the past 40 years for the numerical modelling of granular material dynamics. When combined with a soft-sphere approach, it is conceptually simple, easy to implement and has shown to supply computed results of satisfactory accuracy. DEM simulations have overall helped a lot to gain novel physical insight into granular flows and generally contributed significantly to advance the understanding of the intricate mechanisms involved (force chains, dilatance, blockage, compaction, etc). Although conceptually simple, the implementation of DEM is however more challenging for large systems and systems containing non-spherical, angular and potentially non-convex particles. In the following, we suggest to treat the former by distributed computing and the latter by an extension of our numerical strategy for convex bodies to non-convex bodies.

The rest of the paper is organized as follows. In Section 2, we shortly recall the basic governing equations. Then we elaborate in Section 3 on our glued-convex method to treat (almost) any body shape. Section 4 provides the main ingredients for a simple though extremely efficient domain decomposition based parallel implementation. We conclude by illustrating the computing capabilities of our code Grains3D in Section 5 and discuss what can be achieved in terms of novel physical insight with such a numerical tool in Section 6.

⋆ e-mail: adr.rkt@gmail.com
⋆⋆ e-mail: jean-yves.delenne@supagro.inra.fr
⋆⋆⋆ e-mail: wachs@math.ubc.ca

2 Numerical model

The motion of the granular material is determined by applying Newton's second law to each particle $i \in < 0, N-1 >$, where N is the total number of particles. The rigid body motion assumption leads to the decomposition of the velocity vector $\boldsymbol{v}$ as $\boldsymbol{v} = \boldsymbol{U} + \boldsymbol{\omega} \wedge \boldsymbol{R}$, where $\boldsymbol{U}$, $\boldsymbol{\omega}$ and $\boldsymbol{R}$ denote the translational velocity vector of the center of mass, the angular velocity vector of the center of mass and the position vector with respect to the center of mass, respectively. The complete set of equations to be considered is the following one:

$$M_i \frac{d\boldsymbol{U}_i}{dt} = \boldsymbol{F}_i \quad , \quad \boldsymbol{J}_i \frac{d\boldsymbol{\omega}_i}{dt} + \boldsymbol{\omega}_i \wedge \boldsymbol{J}_i \boldsymbol{\omega}_i = \boldsymbol{M}_i \qquad (1)$$

$$\frac{d\boldsymbol{x}_i}{dt} = \boldsymbol{U}_i \quad , \quad \frac{d\boldsymbol{\theta}_i}{dt} = \boldsymbol{\omega}_i \qquad (2)$$

where M_i, $\boldsymbol{J}_i$, $\boldsymbol{x}_i$ and $\boldsymbol{\theta}_i$ stand for the mass, inertia tensor, center of mass position and angular position of particle i. $\boldsymbol{F}_i$ and $\boldsymbol{M}_i$ are the sum of all forces and torques applied

on particle i, respectively, and can be further decomposed in purely granular dynamics (i.e., without accounting for any external forcing as e.g. hydrodynamic or electrostatic) into a torque-free gravity contribution and a contact force contribution as:

$$\boldsymbol{F}_i = M_i\boldsymbol{g} + \sum_{j=0, j\neq i}^{N-1} \boldsymbol{F}_{ij} \tag{3}$$

$$\boldsymbol{M}_i = \sum_{j=0, j\neq i}^{N-1} \boldsymbol{R}_j \wedge \boldsymbol{F}_{ij} \tag{4}$$

where $\boldsymbol{F}_{ij}$ is the force due to collision with particle j and $\boldsymbol{R}_j$ is a vector pointing from the center of mass of particle i to the contact point with particle j. In our model, $\boldsymbol{F}_{ij}$ comprises a normal Hookean elastic restoring force, a normal dissipative force and a tangential friction force.

The set of equations Eq. (1) is integrated in time using a second order leap-frog Verlet scheme. Rotations of particles are computed using quaternions for computational efficiency as well as to avoid any gimbal lock configurations. The collision detection algorithm is a classical two-step process. Potential collisions are first detected via a linked-cell list and then actual collisions are determined using a GJK algorithm. Our GJK-based collision detection strategy enables us to consider any convex shape and size. For more detail, see [1] and the references therein.

3 Extension to non-convex bodies

In [1], we suggested an original and efficient way to detect collisions between particles of arbitrary convex shape. Our strategy utilizes the Gilbert-Johnson-Keerthi algorithm to compute the minimal distance between two convex bodies and a sequence of shrinking-swelling of bodies. This 3-step collision detection strategy has proven to be well adapted and to preserve shape angularity (unlike, e.g., a superquadric representation). Once combined with a soft-sphere approach in our granular solver Grains3D, the method constitutes a powerful tool to compute granular flows. In [1], we applied it to cubes, cylinders and tetrahedra in (i) the filling of a small container and (ii) rotating drum dynamics.

We achieve the extension to more complex shapes and in particular to non-convex bodies by recognizing that any non-convex body can be decomposed into a set of convex bodies as shown in Fig. 1. We call the original convex body the composite particle and the set of convex bodies the elementary components. Our method is thus an extension of the popular glued-sphere method [2]. We call it the glued-convex method, in the sense that non-convex bodies are considered by clumping convex bodies. There is however a major improvement compared to the glued-sphere approach: because elementary components are convex bodies the original shape of the non-convex particle is perfectly accounted for (no rounded corner, no artificial rugosity). The main asset is the ability to use the same collision detection strategy as for convex bodies (with a linked-cell spatial sorting and circumscribed spheres). In fact, if

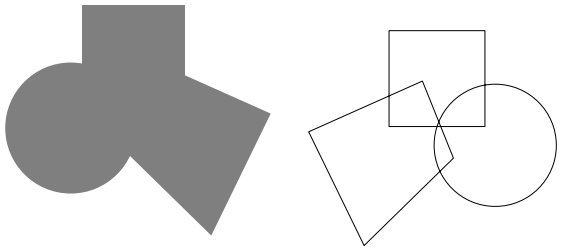


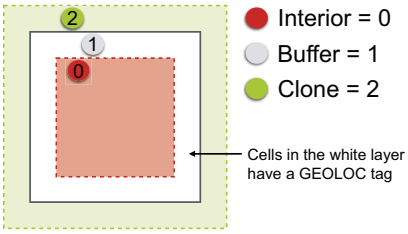
Figure 1: Decomposition strategy illustrated in two dimension

two non-convex bodies touch (or overlap), then there exists at least one pair of convex elementary components that touch. In practice, to detect collisions between two non-convex bodies, we test potential collisions over all pairs (E_i, E_j), $i = 0, ..., N-1$, $j = 0, ..., M-1$ of elementary (convex) components, where N is the total number of elementary components of the first composite particle and M is the total number of elementary components of the second composite particle. The only real drawback of the glued-convex method is that the computing cost theoretically scales as $N \times M$.

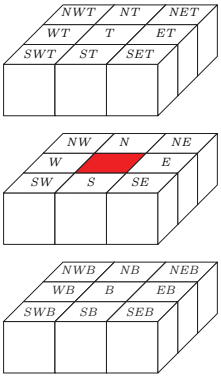
Finally, since mass properties (volume, center of mass, tensor of inertia) of a complex non-convex particle cannot be calculated analytically, we employ a parallel numerical integration method to evaluate these quantities. In practice, we embed a non-convex body in a regular cartesian fine mesh and compute the integrals numerically based on a simple cell center in/out criterion. From a 500^3 mesh, mass properties are computed with an error less than 0.1% and this takes less than a few minutes in parallel on a supercomputer. These mass properties are computed once for a non-convex shape and are stored in a database.

4 A MPI/Domain Decomposition parallel implementation

A key to simulate systems with a large number of particles is distributed computing [3, 4]. Our parallel strategy is classical and is based on a constant in time domain decomposition technique. We employ a cartesian domain decomposition. Each process hosts a single subdomain and we hence define a cartesian MPI communicator using the `MPI_Cart_create` command. On each subdomain, we construct a cartesian linked-cell list with an additional layer of cells at the boundary with neighbouring subdomain to serve as an overlapping zone. This overlapping zone hosts clone particles used to compute collisions with particles located on a neighbouring subdomain (process). As a consequence, cells in a linked-cell list are tagged based on their location on the subdomain: 0 = interior, 1 = buffer and 2 = clone, as illustrated on Fig. 2(a). At each time step, clone particles are either created, deleted or updated. All particles are tagged based on the cell they belong to. Hence they consistently change status as they



(a) 2D illustration of the status of a particle depending on their location on the domain, i.e., depending on the tag of the cell it belongs to



(b) GEOLOC tag of cells in the buffer zone. N, S, W, E, T and B denote respectively the North, South, West, East, Top and Bottom directions.

Figure 2: Tags (status and geolocalisation) of particles in the linked-cell grid.

move in the subdomain. Corresponding operations are performed on neighbouring subdomains when a particle change status, e.g., if a particle moves from an interior cell (tag = 0) to a buffer cell (tag = 1), a clone particle (tag = 2) is automatically created on the neighbouring subdomain.

Not only cells (and hence particles belonging to these cells) are tagged in terms of their status but cells in the buffer zone are also tagged in terms of their location with respect to the neighbouring subdomains using a second tag, named `GEOLOC` for geographic location, that takes the 26 following values (whose meaning is rather obvious on a 3D cartesian grid as can be seen in Fig. 2(b)): `EAST` & `WEST` in the x direction, `NORTH` & `SOUTH` in the y direction, `TOP` & `BOTTOM` in the z direction are the main neighbours, `NORTH_EAST`, `NORTH_WEST`, `SOUTH_EAST`, `SOUTH_WEST`, `NORTH_BOTTOM`, `NORTH_TOP`, `SOUTH_BOTTOM`, `SOUTH_TOP` are the edge neighbours, and `NORTH_WEST_TOP`, `NORTH_WEST_BOTTOM` and so on are the corner neighbours. The aftermath is an exact tailoring of sent MPI messages with the appropriate information only, thus reducing the size of each sent message to the minimum, as illustrated in Fig. 3.

5 Results

5.1 Parallel performance

We illustrate the parallel performance of Grains3D on a granular column collapse (also called granular dam break

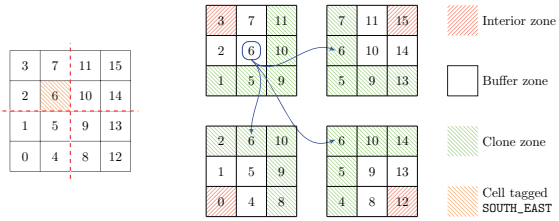
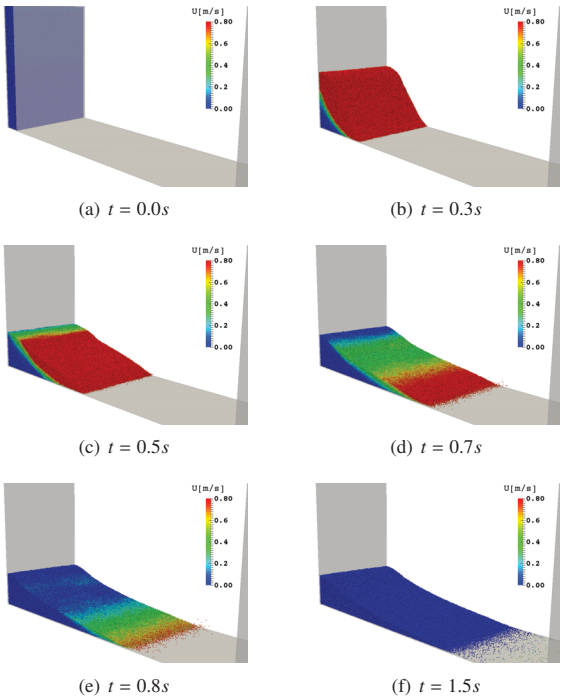


Figure 3: 2D illustration of inter-process communication for a particle tagged SOUTH_EAST.



(a) $t = 0.0s$ (b) $t = 0.3s$ (c) $t = 0.5s$ (d) $t = 0.7s$ (e) $t = 0.8s$ (f) $t = 1.5s$

Figure 4: 3D view of the granular dam break flow with 1,627,500 icosahedra.

in the literature). The column is made of icosahedral particles that mimic sand grains. We perform weak scaling tests that involve increasing the number of cores and the size of the system simultaneously. In other words, we set the load of particles per core constant. Here, the load of particles per core is set to 102,850 and we perform computations with up to 52,080,000 icosahedra on 512 cores. The domain is periodic in the transverse direction and the initial aspect ratio of the column is set to 7.3. Fig. 4 shows snapshots of the collapse. The parallel performance of Grains3D is presented in Fig. 5. Grains3D exhibits a very satisfactory scaling factor of around 0.9 up to 512 cores (that corresponds to the largest system containing 52 million of icosahedron).

5.2 Filling a container with non-convex shapes

We expand here the results obtained in [1] on the filling of a cylindrical reactor with particles of various shape but

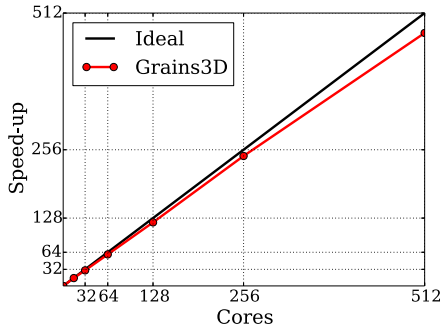
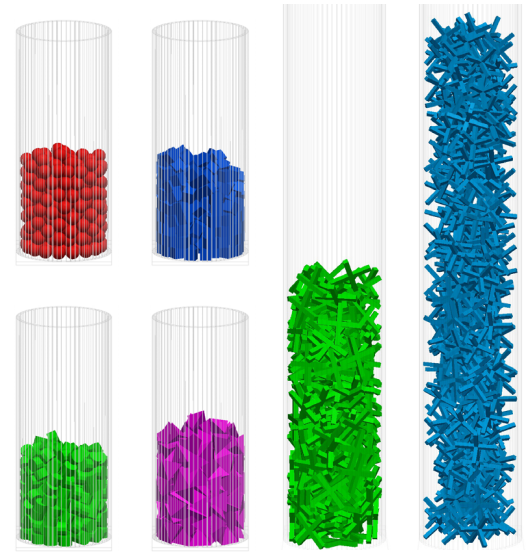


Figure 5: Weak scaling parallel performance of Grains3D in granular dam break computations up to 512 cores.



(a) Sphere (top) and Cylinder (bottom) (b) Cube (top) and Tetrahedron (bottom) (c) "2D-cross" (d) "3D-cross"

Figure 6: Packing of 250 particles of six different shapes in a cylindrical container. Shapes in (a,b) are from [1]

same volume (spheres, cylinders, cubes and tetrahedra). Here we create two new shapes: a 2D cross and a 3D cross of same volume. These two new shapes are indeed non-convex and are modelled by clumping beams of square cross section together (2 in 2D and 3 in 3D). We run the same simulations for these two new cross-like shapes as in [1] for the 4 aforementioned convex shapes. All physical and numerical parameters are the same, only the shape changes. Please also note that the reactor diameter to particle equivalent diameter is pretty small, which promotes high porosity close to the wall. The height of the pack of particles is a measure of the level of compaction, or conversely of porosity, in the system. It is quite spectacular in Fig. 6 that 2D and even more 3D crosses lead to a level of porosity 2 to 4 times higher than for any of the isometric convex shapes. This nicely illustrates the effect of non-convexity on the microstructure of a pack of particles.

6 Discussion and Perspectives

The GJK-based collision detection strategy suggested in [1] already enabled us to examine granular flows with particles of multiple size and multiple convex shapes. We have now extended the modelling capabilities of our code Grains3D to non-convex shapes and massively parallel computing. Our glued-convex method for the extension to non-convex bodies is very robust and accurate as it uses the same tools as for convex bodies. Its computational cost is however quite high as it scales with the product of the number of elementary components in two colliding particles. But it also opens up unprecedented perspectives in the numerical modelling of granular flows. In particular, highly non-convex particles as 3D crosses provide a sort of cohesion to the granular media that significantly modifies the way it flows.

In systems in which the load of particles per core does not change too much, our parallel strategy shows very satisfactory weak scaling properties. Systems of up to a few hundreds of millions of convex particles can now be computed on a few hundreds to a few thousands of cores in a reasonable time (i.e., of the order of a few days). Systems with a few tens of millions of non-convex particles are also attainable. High Performance Computing and supercomputers combined to highly scalable codes are really becoming game-shifters in the field of granular flow simulation. The next stage in terms of improving even further our numerical model is to develop a dynamic load balancing algorithm in order to offer a good scalability even in systems with a high level of particle volume fraction heterogeneity in space and in time. This is an on-going work in our group.

Acknowledgements

This work was granted access to the HPC resources of CINES under the allocations 2013-c20132b6699 and 2014-c20142b6699 made by GENCI.

References

[1] A. Wachs, L. Girolami, G. Vinay, G. Ferrer, Powder Technology **224**, 374 (2012)
[2] M. Abbaspour-Fard, Biosystems Engineering **88**, 153 (2004)
[3] J.H. Walther, I.F. Sbalzarini, Engineering Computations **26**, 688 (2009)
[4] K. Iglberger, U. Rüde, Multibody System Dynamics **25**, 81 (2011)