



# DEM Formulation

Václav Šmilauer, Bruno Chareyre

---

**Citing this document:**

Šmilauer V. and Chareyre B. (2015). DEM Formulation. In: *Yade Documentation 2nd ed.*  
doi:10.5281/zenodo.34044. <http://yade-dem.org>  
See also <http://yade-dem/doc/citing.html>.



# Contents

<b>1</b>	<b>DEM Background</b>	<b>1</b>
1.1	Collision detection . . . . .	1
1.2	Creating interaction between particles . . . . .	5
1.3	Strain evaluation . . . . .	6
1.4	Stress evaluation (example) . . . . .	9
1.5	Motion integration . . . . .	10
1.6	Periodic boundary conditions . . . . .	17
1.7	Computational aspects . . . . .	21
	<b>Bibliography</b>	<b>23</b>



# Chapter 1

## DEM Background

In this chapter, we mathematically describe general features of explicit DEM simulations, with some reference to Yade implementation of these algorithms. They are given roughly in the order as they appear in simulation; first, two particles might establish a new interaction, which consists in

1. detecting collision between particles;
2. creating new interaction and determining its properties (such as stiffness); they are either precomputed or derived from properties of both particles;

Then, for already existing interactions, the following is performed:

1. strain evaluation;
2. stress computation based on strains;
3. force application to particles in interaction.

This simplified description serves only to give meaning to the ordering of sections within this chapter. A more detailed description of this *simulation loop* is given later.

In this chapter we refer to kinematic variables of the contacts as “strains“, although at this scale it is also common to speak of “displacements“. Which semantic is more appropriate depends on the conceptual model one is starting from, and therefore it cannot be decided independently of specific problems. The reader familiar with displacements can mentally replace normal strain and shear strain by normal displacement and shear displacement, respectively, without altering the meaning of what follows.

### 1.1 Collision detection

#### 1.1.1 Generalities

Exact computation of collision configuration between two particles can be relatively expensive (for instance between Sphere and Facet). Taking a general pair of bodies  $i$  and  $j$  and their “exact“ (In the sense of precision admissible by numerical implementation.) spatial predicates (called Shape in Yade) represented by point sets  $P_i, P_j$  the detection generally proceeds in 2 passes:

1. fast collision detection using approximate predicate  $\tilde{P}_i$  and  $\tilde{P}_j$ ; they are pre-constructed in such a way as to abstract away individual features of  $P_i$  and  $P_j$  and satisfy the condition

$$\forall \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} \in P_i \Rightarrow \mathbf{x} \in \tilde{P}_i \quad (1.1)$$

(likewise for  $P_j$ ). The approximate predicate is called “bounding volume” (Bound in Yade) since it bounds any particle’s volume from outside (by virtue of the implication). It follows that  $(P_i \cap P_j) \neq \emptyset \Rightarrow (\tilde{P}_i \cap \tilde{P}_j) \neq \emptyset$  and, by applying *modus tollens*,

$$(\tilde{P}_i \cap \tilde{P}_j) = \emptyset \Rightarrow (P_i \cap P_j) = \emptyset \quad (1.2)$$

which is a candidate exclusion rule in the proper sense.

2. By filtering away impossible collisions in (1.2), a more expensive, exact collision detection algorithms can be run on possible interactions, filtering out remaining spurious couples  $(\tilde{P}_i \cap \tilde{P}_j) \neq \emptyset \wedge (P_i \cap P_j) = \emptyset$ . These algorithms operate on  $P_i$  and  $P_j$  and have to be able to handle all possible combinations of shape types.

It is only the first step we are concerned with here.

### 1.1.2 Algorithms

Collision evaluation algorithms have been the subject of extensive research in fields such as robotics, computer graphics and simulations. They can be roughly divided in two groups:

**Hierarchical algorithms** which recursively subdivide space and restrict the number of approximate checks in the first pass, knowing that lower-level bounding volumes can intersect only if they are part of the same higher-level bounding volume. Hierarchy elements are bounding volumes of different kinds: octrees [Jung1997], bounding spheres [Hubbard1996], k-DOP's [Klosowski1998].

**Flat algorithms** work directly with bounding volumes without grouping them in hierarchies first; let us only mention two kinds commonly used in particle simulations:

**Sweep and prune** algorithm operates on axis-aligned bounding boxes, which overlap if and only if they overlap along all axes. These algorithms have roughly  $\mathcal{O}(n \log n)$  complexity, where  $n$  is number of particles as long as they exploit *temporal coherence* of the simulation.

**Grid algorithms** represent continuous  $\mathbb{R}^3$  space by a finite set of regularly spaced points, leading to very fast neighbor search; they can reach the  $\mathcal{O}(n)$  complexity [Munjiza1998] and recent research suggests ways to overcome one of the major drawbacks of this method, which is the necessity to adjust grid cell size to the largest particle in the simulation ([Munjiza2006], the “multistep” extension).

**Temporal coherence** expresses the fact that motion of particles in simulation is not arbitrary but governed by physical laws. This knowledge can be exploited to optimize performance.

Numerical stability of integrating motion equations dictates an upper limit on  $\Delta t$  (sect. *Stability considerations*) and, by consequence, on displacement of particles during one step. This consideration is taken into account in [Munjiza2006], implying that any particle may not move further than to a neighboring grid cell during one step allowing the  $\mathcal{O}(n)$  complexity; it is also explored in the periodic variant of the sweep and prune algorithm described below.

On a finer level, it is common to enlarge  $\tilde{P}_i$  predicates in such a way that they satisfy the (1.1) condition during *several* timesteps; the first collision detection pass might then be run with stride, speeding up the simulation considerably. The original publication of this optimization by Verlet [Verlet1967] used enlarged list of neighbors, giving this technique the name *Verlet list*. In general cases, however, where neighbor lists are not necessarily used, the term *Verlet distance* is employed.

### 1.1.3 Sweep and prune

Let us describe in detail the sweep and prune algorithm used for collision detection in Yade (class InsertionSortCollider). Axis-aligned bounding boxes (Aabb) are used as  $\tilde{P}_i$ ; each Aabb is given by lower and upper corner  $\in \mathbb{R}^3$  (in the following,  $\tilde{P}_i^{x0}$ ,  $\tilde{P}_i^{x1}$  are minimum/maximum coordinates of  $\tilde{P}_i$  along the  $x$ -axis and so on). Construction of Aabb from various particle Shape's (such as Sphere, Facet, Wall) is straightforward, handled by appropriate classes deriving from BoundFunctor (Bo1\_Sphere\_Aabb, Bo1\_Facet\_Aabb, ...).

Presence of overlap of two Aabb's can be determined from conjunction of separate overlaps of intervals along each axis (*fig-sweep-and-prune*):

$$(\tilde{P}_i \cap \tilde{P}_j) \neq \emptyset \Leftrightarrow \bigwedge_{w \in \{x, y, z\}} \left[ \left( (\tilde{P}_i^{w0}, \tilde{P}_i^{w1}) \cap (\tilde{P}_j^{w0}, \tilde{P}_j^{w1}) \right) \neq \emptyset \right]$$

where  $(a, b)$  denotes interval in  $\mathbb{R}$ .

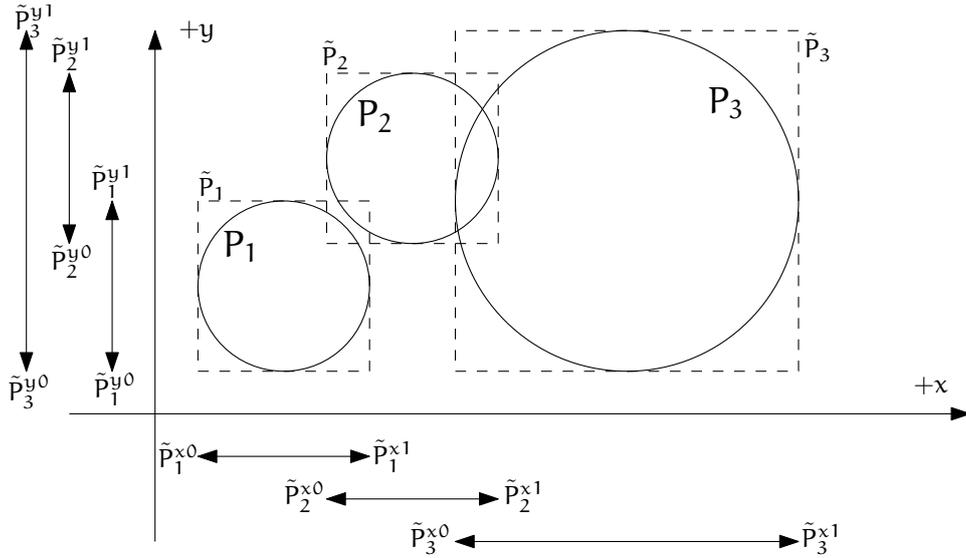


Figure 1.1: Sweep and prune algorithm (shown in 2D), where Aabb of each sphere is represented by minimum and maximum value along each axis. Spatial overlap of Aabb's is present if they overlap along all axes. In this case,  $\tilde{P}_1 \cap \tilde{P}_2 \neq \emptyset$  (but note that  $P_1 \cap P_2 = \emptyset$ ) and  $\tilde{P}_2 \cap \tilde{P}_3 \neq \emptyset$ .

The collider keeps 3 separate lists (arrays)  $L_w$  for each axis  $w \in \{x, y, z\}$

$$L_w = \bigcup_i \{ \tilde{P}_i^{w0}, \tilde{P}_i^{w1} \}$$

where  $i$  traverses all particles.  $L_w$  arrays (sorted sets) contain respective coordinates of minimum and maximum corners for each Aabb (we call these coordinates *bound* in the following); besides bound, each of list elements further carries *id* referring to particle it belongs to, and a flag whether it is lower or upper bound.

In the initial step, all lists are sorted (using quicksort, average  $\mathcal{O}(n \log n)$ ) and one axis is used to create initial interactions: the range between lower and upper bound for each body is traversed, while bounds in-between indicate potential Aabb overlaps which must be checked on the remaining axes as well.

At each successive step, lists are already pre-sorted. Inversions occur where a particle's coordinate has just crossed another particle's coordinate; this number is limited by numerical stability of simulation and its physical meaning (giving spatio-temporal coherence to the algorithm). The insertion sort algorithm swaps neighboring elements if they are inverted, and has complexity between  $\text{bigO}\{n\}$  and  $\text{bigO}\{n^2\}$ , for pre-sorted and unsorted lists respectively. For our purposes, we need only to handle inversions, which by nature of the sort algorithm are detected inside the sort loop. An inversion might signify:

- overlap along the current axis, if an upper bound inverts (swaps) with a lower bound (i.e. that the upper bound with a higher coordinate was out of order in coming before the lower bound with a lower coordinate). Overlap along the other 2 axes is checked and if there is overlap along all axes, a new potential interaction is created.
- End of overlap along the current axis, if lower bound inverts (swaps) with an upper bound. If there is only potential interaction between the two particles in question, it is deleted.
- Nothing if both bounds are upper or both lower.

### Aperiodic insertion sort

Let us show the sort algorithm on a sample sequence of numbers:

$$\| \ 3 \quad 7 \quad 2 \quad 4 \ \|$$

Elements are traversed from left to right; each of them keeps inverting (swapping) with neighbors to the left, moving left itself, until any of the following conditions is satisfied:

$(\leq)$	the sorting order with the left neighbor is correct, or
$(\ )$	the element is at the beginning of the sequence.

We start at the leftmost element (the current element is marked  $\boxed{i}$ )

$$\| \boxed{3} \quad 7 \quad 2 \quad 4 \ \|.$$

It obviously immediately satisfies  $(\|)$ , and we move to the next element:

$$\| 3 \quad \boxed{7} \quad 2 \quad 4 \ \|.$$

$\swarrow$   
 $\leq$

Condition  $(\leq)$  holds, therefore we move to the right. The  $\boxed{2}$  is not in order (violating  $(\leq)$ ) and two inversions take place; after that,  $(\|)$  holds:

$$\| 3 \quad 7 \quad \boxed{2} \quad 4 \ \|,$$

$$\| 3 \quad \boxed{2} \quad 7 \quad 4 \ \|,$$

$\swarrow$   
 $\leq$

$$\| \boxed{2} \quad 3 \quad 7 \quad 4 \ \|.$$

The last element  $\boxed{4}$  first violates  $(\leq)$ , but satisfies it after one inversion

$$\| 2 \quad 3 \quad 7 \quad \boxed{4} \ \|,$$

$$\| 2 \quad 3 \quad \boxed{4} \quad 7 \ \|.$$

$\swarrow$   
 $\leq$

All elements having been traversed, the sequence is now sorted.

It is obvious that if the initial sequence were sorted, elements only would have to be traversed without any inversion to handle (that happens in  $\mathcal{O}(n)$  time).

For each inversion during the sort in simulation, the function that investigates change in Aabb overlap is invoked, creating or deleting interactions.

The periodic variant of the sort algorithm is described in *Periodic insertion sort algorithm*, along with other periodic-boundary related topics.

### Optimization with Verlet distances

As noted above, [Verlet1967] explored the possibility of running the collision detection only sparsely by enlarging predicates  $\tilde{P}_i$ .

In Yade, this is achieved by enlarging Aabb of particles by fixed relative length (or Verlet’s distance) in all dimensions  $\Delta L$  (InsertionSortCollider.sweepLength). Suppose the collider run last time at step  $m$  and the current step is  $n$ . NewtonIntegrator tracks the cummulated distance traversed by each particle between  $m$  and  $n$  by comparing the current position with the reference position from time  $n$  (Bound::refPos),

$$L_{mn} = |X^n - X^m| \tag{1.3}$$

triggering the collider re-run as soon as one particle gives:

$$L_{mn} > \Delta L. \tag{1.4}$$

InsertionSortCollider.targetInterv is used to adjust  $\Delta L$  independently for each particle. Larger  $\Delta L$  will be assigned to the fastest ones, so that all particles would ideally reach the edge of their bounds after this “target” number of iterations. Results of using Verlet distance depend highly on the nature of simulation and choice of InsertionSortCollider.targetInterv. Adjusting the sizes independently for each particle is especially efficient if some parts of a problem have high-speed particles will others are not moving. If

it is not the case, no significant gain should be expected as compared to `targetInterv=0` (assigning the same  $\Delta L$  to all particles).

The number of particles and the number of available threads is also to be considered for choosing an appropriate Verlet's distance. A larger distance will result in less time spent in the collider (which runs single-threaded) and more time in computing interactions (multi-threaded). Typically, large  $\Delta L$  will be used for large simulations with more than  $10^5$  particles on multi-core computers. On the other hand simulations with less than  $10^4$  particles on single processor will probably benefit from smaller  $\Delta L$ . Users benchmarks may be found on Yade's wiki (see e.g. [https://yade-dem.org/wiki/Colliders\\_performance](https://yade-dem.org/wiki/Colliders_performance)).

## 1.2 Creating interaction between particles

Collision detection described above is only approximate. Exact collision detection depends on the geometry of individual particles and is handled separately. In Yade terminology, the Collider creates only *potential* interactions; potential interactions are evaluated exactly using specialized algorithms for collision of two spheres or other combinations. Exact collision detection must be run at every timestep since it is at every step that particles can change their mutual position (the collider is only run sometimes if the Verlet distance optimization is in use). Some exact collision detection algorithms are described in *Strain evaluation*; in Yade, they are implemented in classes deriving from `IGeomFunctor` (prefixed with `Ig2`).

Besides detection of geometrical overlap (which corresponds to `IGeom` in Yade), there are also non-geometrical properties of the interaction to be determined (`IPhys`). In Yade, they are computed for every new interaction by calling a functor deriving from `IPhysFunctor` (prefixed with `Ip2`) which accepts the given combination of Material types of both particles.

### 1.2.1 Stiffnesses

Basic DEM interaction defines two stiffnesses: normal stiffness  $K_N$  and shear (tangent) stiffness  $K_T$ . It is desirable that  $K_N$  be related to fictitious Young's modulus of the particles' material, while  $K_T$  is typically determined as a given fraction of computed  $K_N$ . The  $K_T/K_N$  ratio determines macroscopic Poisson's ratio of the arrangement, which can be shown by dimensional analysis: elastic continuum has two parameters ( $E$  and  $\nu$ ) and basic DEM model also has 2 parameters with the same dimensions  $K_N$  and  $K_T/K_N$ ; macroscopic Poisson's ratio is therefore determined solely by  $K_T/K_N$  and macroscopic Young's modulus is then proportional to  $K_N$  and affected by  $K_T/K_N$ .

Naturally, such analysis is highly simplifying and does not account for particle radius distribution, packing configuration and other possible parameters such as the interaction radius introduced later.

#### Normal stiffness

The algorithm commonly used in Yade computes normal interaction stiffness as stiffness of two springs in serial configuration with lengths equal to the sphere radii (*fig-spheres-contact-stiffness*).

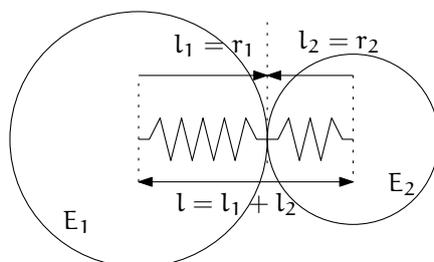


Figure 1.2: Series of 2 springs representing normal stiffness of contact between 2 spheres.

Let us define distance  $l = l_1 + l_2$ , where  $l_i$  are distances between contact point and sphere centers, which are initially (roughly speaking) equal to sphere radii. Change of distance between the sphere centers  $\Delta l$

is distributed onto deformations of both spheres  $\Delta l = \Delta l_1 + \Delta l_2$  proportionally to their compliances. Displacement change  $\Delta l_i$  generates force  $F_i = K_i \Delta l_i$ , where  $K_i$  assures proportionality and has physical meaning and dimension of stiffness;  $K_i$  is related to the sphere material modulus  $E_i$  and some length  $\tilde{l}_i$  proportional to  $r_i$ .

$$\begin{aligned}\Delta l &= \Delta l_1 + \Delta l_2 \\ K_i &= E_i \tilde{l}_i \\ K_N \Delta l &= F = F_1 = F_2 \\ K_N (\Delta l_1 + \Delta l_2) &= F \\ K_N \left( \frac{F}{K_1} + \frac{F}{K_2} \right) &= F \\ K_1^{-1} + K_2^{-1} &= K_N^{-1} \\ K_N &= \frac{K_1 K_2}{K_1 + K_2} \\ K_N &= \frac{E_1 \tilde{l}_1 E_2 \tilde{l}_2}{E_1 \tilde{l}_1 + E_2 \tilde{l}_2}\end{aligned}$$

The most used class computing interaction properties `Ip2_FrictMat_FrictMat_FrictPhys` uses  $\tilde{l}_i = 2r_i$ . Some formulations define an equivalent cross-section  $A_{eq}$ , which in that case appears in the  $\tilde{l}_i$  term as  $K_i = E_i \tilde{l}_i = E_i \frac{A_{eq}}{\tilde{l}_i}$ . Such is the case for the concrete model (`Ip2_CpmMat_CpmMat_CpmPhys`), where  $A_{eq} = \min(r_1, r_2)$ .

For reasons given above, no pretense about equality of particle-level  $E_i$  and macroscopic modulus  $E$  should be made. Some formulations, such as [Hentz2003], introduce parameters to match them numerically. This is not appropriate, in our opinion, since it binds those values to particular features of the sphere arrangement that was used for calibration.

## 1.2.2 Other parameters

Non-elastic parameters differ for various material models. Usually, though, they are averaged from the particles' material properties, if it makes sense. For instance, `Ip2_CpmMat_CpmMat_CpmPhys` averages most quantities, while `Ip2_FrictMat_FrictMat_FrictPhys` computes internal friction angle as  $\varphi = \min(\varphi_1, \varphi_2)$  to avoid friction with bodies that are frictionless.

## 1.3 Strain evaluation

In the general case, mutual configuration of two particles has 6 degrees of freedom (DoFs) just like a beam in 3D space: both particles have 6 DoFs each, but the interaction itself is free to move and rotate in space (with both spheres) having 6 DoFs itself; then  $12 - 6 = 6$ . They are shown at [fig-spheres-dofs](#).

We will only describe normal and shear components of strain in the following, leaving torsion and bending aside. The reason is that most constitutive laws for contacts do not use the latter two.

### 1.3.1 Normal strain

#### Constants

Let us consider two spheres with *initial* centers  $\bar{\mathbf{C}}_1$ ,  $\bar{\mathbf{C}}_2$  and radii  $r_1$ ,  $r_2$  that enter into contact. The order of spheres within the contact is arbitrary and has no influence on the behavior. Then we define lengths

$$\begin{aligned}d_0 &= |\bar{\mathbf{C}}_2 - \bar{\mathbf{C}}_1| \\ d_1 &= r_1 + \frac{d_0 - r_1 - r_2}{2}, & d_2 &= d_0 - d_1.\end{aligned}$$

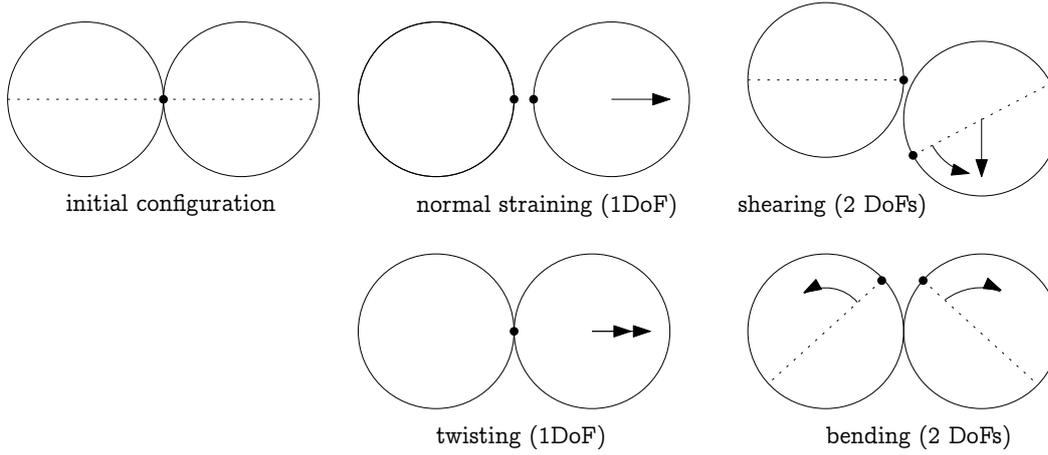


Figure 1.3: Degrees of freedom of configuration of two spheres. Normal strain appears if there is a difference of linear velocity along the interaction axis ( $\mathbf{n}$ ); shearing originates from the difference of linear velocities perpendicular to  $\mathbf{n}$  and from the part of  $\boldsymbol{\omega}_1 + \boldsymbol{\omega}_2$  perpendicular to  $\mathbf{n}$ ; twisting is caused by the part of  $\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2$  parallel with  $\mathbf{n}$ ; bending comes from the part of  $\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2$  perpendicular to  $\mathbf{n}$ .

These quantities are *constant* throughout the life of the interaction and are computed only once when the interaction is established. The distance  $d_0$  is the *reference distance* and is used for the conversion of absolute displacements to dimensionless strain, for instance. It is also the distance where (for usual contact laws) there is neither repulsive nor attractive force between the spheres, whence the name *equilibrium distance*.

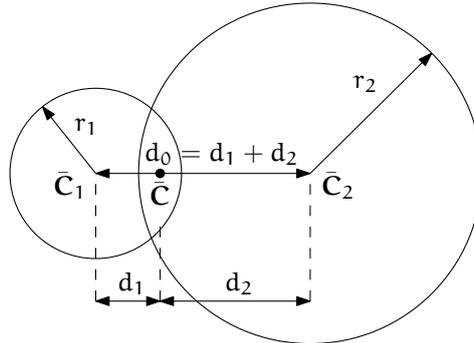


Figure 1.4: Geometry of the initial contact of 2 spheres; this case pictures spheres which already overlap when the contact is created (which can be the case at the beginning of a simulation) for the sake of generality. The initial contact point  $\mathbf{C}$  is in the middle of the overlap zone.

Distances  $d_1$  and  $d_2$  define reduced (or expanded) radii of spheres; geometrical radii  $r_1$  and  $r_2$  are used only for collision detection and may not be the same as  $d_1$  and  $d_2$ , as shown in fig. [fig-sphere-sphere](#). This difference is exploited in cases where the average number of contacts between spheres should be increased, e.g. to influence the response in compression or to stabilize the packing. In such case, interactions will be created also for spheres that do not geometrically overlap based on the *interaction radius*  $R_I$ , a dimensionless parameter determining „non-locality“ of contact detection. For  $R_I = 1$ , only spheres that touch are considered in contact; the general condition reads

$$d_0 \leq R_I(r_1 + r_2). \quad (1.5)$$

The value of  $R_I$  directly influences the average number of interactions per sphere (percolation), which for some models is necessary in order to achieve realistic results. In such cases, `Aabb` (or  $\tilde{P}_i$  predicates in general) must be enlarged accordingly (`Bo1_Sphere_Aabb.aabbEnlargeFactor`).

### Contact cross-section

Some constitutive laws are formulated with strains and stresses (Law2\_ScGeom\_CpmPhys\_Cpm, the concrete model described later, for instance); in that case, equivalent cross-section of the contact must be introduced for the sake of dimensionality. The exact definition is rather arbitrary; the CPM model (Ip2\_CpmMat\_CpmMat\_CpmPhys) uses the relation

$$A_{\text{eq}} = \pi \min(r_1, r_2)^2 \quad (1.6)$$

which will be used to convert stresses to forces, if the constitutive law used is formulated in terms of stresses and strains. Note that other values than  $\pi$  can be used; it will merely scale macroscopic packing stiffness; it is only for the intuitive notion of a truss-like element between the particle centers that we choose  $A_{\text{eq}}$  representing the circle area. Besides that, another function than  $\min(r_1, r_2)$  can be used, although the result should depend linearly on  $r_1$  and  $r_2$  so that the equation gives consistent results if the particle dimensions are scaled.

### Variables

The following state variables are updated as spheres undergo motion during the simulation (as  $\mathbf{C}_1^\circ$  and  $\mathbf{C}_2^\circ$  change):

$$\mathbf{n}^\circ = \frac{\mathbf{C}_2^\circ - \mathbf{C}_1^\circ}{|\mathbf{C}_2^\circ - \mathbf{C}_1^\circ|} \equiv \widehat{\mathbf{C}_2^\circ - \mathbf{C}_1^\circ} \quad (1.7)$$

and

$$\mathbf{C}^\circ = \mathbf{C}_1^\circ + \left( d_1 - \frac{d_0 - |\mathbf{C}_2^\circ - \mathbf{C}_1^\circ|}{2} \right) \mathbf{n}. \quad (1.8)$$

The contact point  $\mathbf{C}^\circ$  is always in the middle of the spheres' overlap zone (even if the overlap is negative, when it is in the middle of the empty space between the spheres). The *contact plane* is always perpendicular to the contact plane normal  $\mathbf{n}^\circ$  and passes through  $\mathbf{C}^\circ$ .

Normal displacement and strain can be defined as

$$\begin{aligned} \mathbf{u}_N &= |\mathbf{C}_2^\circ - \mathbf{C}_1^\circ| - d_0, \\ \varepsilon_N &= \frac{\mathbf{u}_N}{d_0} = \frac{|\mathbf{C}_2^\circ - \mathbf{C}_1^\circ|}{d_0} - 1. \end{aligned}$$

Since  $\mathbf{u}_N$  is always aligned with  $\mathbf{n}$ , it can be stored as a scalar value multiplied by  $\mathbf{n}$  if necessary.

For massively compressive simulations, it might be beneficial to use the logarithmic strain, such that the strain tends to  $-\infty$  (rather than  $-1$ ) as centers of both spheres approach. Otherwise, repulsive force would remain finite and the spheres could penetrate through each other. Therefore, we can adjust the definition of normal strain as follows:

$$\varepsilon_N = \begin{cases} \log\left(\frac{|\mathbf{C}_2^\circ - \mathbf{C}_1^\circ|}{d_0}\right) & \text{if } |\mathbf{C}_2^\circ - \mathbf{C}_1^\circ| < d_0 \\ \frac{|\mathbf{C}_2^\circ - \mathbf{C}_1^\circ|}{d_0} - 1 & \text{otherwise.} \end{cases}$$

Such definition, however, has the disadvantage of effectively increasing rigidity (up to infinity) of contacts, requiring  $\Delta t$  to be adjusted, lest the simulation becomes unstable. Such dynamic adjustment is possible using a stiffness-based time-stepper (GlobalStiffnessTimeStepper in Yade).

### 1.3.2 Shear strain

In order to keep  $\mathbf{u}_T$  consistent (e.g. that  $\mathbf{u}_T$  must be constant if two spheres retain mutually constant configuration but move arbitrarily in space), then either  $\mathbf{u}_T$  must track spheres' spatial motion or must (somehow) rely on sphere-local data exclusively.

Geometrical meaning of shear strain is shown in [fig-shear-2d](#).

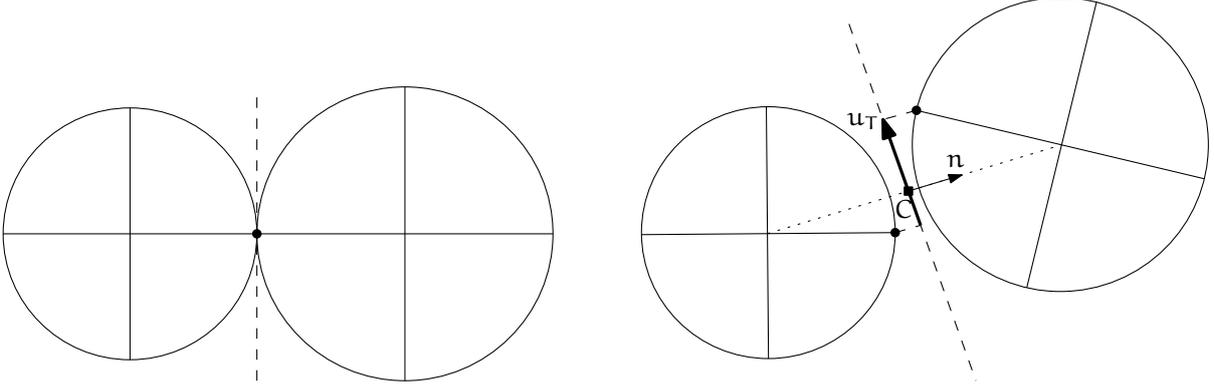


Figure 1.5: Evolution of shear displacement  $\mathbf{u}_T$  due to mutual motion of spheres, both linear and rotational. Left configuration is the initial contact, right configuration is after displacement and rotation of one particle.

The classical incremental algorithm is widely used in DEM codes and is described frequently ([Luding2008], [Alonso2004]). Yade implements this algorithm in the ScGeom class. At each step, shear displacement  $\mathbf{u}_T$  is updated; the update increment can be decomposed in 2 parts: motion of the interaction (i.e.  $\mathbf{C}$  and  $\mathbf{n}$ ) in global space and mutual motion of spheres.

1. Contact moves due to changes of the spheres' positions  $\mathbf{C}_1$  and  $\mathbf{C}_2$ , which updates current  $\mathbf{C}^\circ$  and  $\mathbf{n}^\circ$  as per (1.8) and (1.7).  $\mathbf{u}_T^-$  is perpendicular to the contact plane at the previous step  $\mathbf{n}^-$  and must be updated so that  $\mathbf{u}_T^- + (\Delta\mathbf{u}_T) = \mathbf{u}_T^\circ \perp \mathbf{n}^\circ$ ; this is done by perpendicular projection to the plane first (which might decrease  $|\mathbf{u}_T^-|$ ) and adding what corresponds to spatial rotation of the interaction instead:

$$(\Delta\mathbf{u}_T)_1 = -\mathbf{u}_T^- \times (\mathbf{n}^- \times \mathbf{n}^\circ)$$

$$(\Delta\mathbf{u}_T)_2 = -\mathbf{u}_T^- \times \left( \frac{\Delta t}{2} \mathbf{n}^\circ \cdot (\boldsymbol{\omega}_1^\circ + \boldsymbol{\omega}_2^\circ) \right) \mathbf{n}^\circ$$

2. Mutual movement of spheres, using only its part perpendicular to  $\mathbf{n}^\circ$ ;  $\mathbf{v}_{12}$  denotes mutual velocity of spheres at the contact point:

$$\mathbf{v}_{12} = (\mathbf{v}_2^\circ + \boldsymbol{\omega}_2^\circ \times (-d_2\mathbf{n}^\circ)) - (\mathbf{v}_1^\circ + \boldsymbol{\omega}_1^\circ \times (d_1\mathbf{n}^\circ))$$

$$\mathbf{v}_{12}^\perp = \mathbf{v}_{12} - (\mathbf{n}^\circ \cdot \mathbf{v}_{12})\mathbf{n}^\circ$$

$$(\Delta\mathbf{u}_T)_3 = -\Delta t \mathbf{v}_{12}^\perp$$

Finally, we compute

$$\mathbf{u}_T^\circ = \mathbf{u}_T^- + (\Delta\mathbf{u}_T)_1 + (\Delta\mathbf{u}_T)_2 + (\Delta\mathbf{u}_T)_3.$$

## 1.4 Stress evaluation (example)

Once strain on a contact is computed, it can be used to compute stresses/forces acting on both spheres.

The constitutive law presented here is the most usual DEM formulation, originally proposed by Cundall. While the strain evaluation will be similar to algorithms described in the previous section regardless of stress evaluation, stress evaluation itself depends on the nature of the material being modeled. The constitutive law presented here is the most simple non-cohesive elastic case with dry friction, which Yade implements in Law2\_ScGeom\_FrictPhys\_CundallStrack (all constitutive laws derive from base class LawFunctor).

In DEM generally, some constitutive laws are expressed using strains and stresses while others prefer displacement/force formulation. The law described here falls in the latter category.

When new contact is established (discussed in *sect-simulation-loop*) it has its properties (IPhys) computed from Materials associated with both particles. In the simple case of frictional material FrictMat, Ip2\_FrictMat\_FrictMat\_FrictPhys creates a new FrictPhys instance, which defines normal stiffness  $K_N$ , shear stiffness  $K_T$  and friction angle  $\varphi$ .

At each step, given normal and shear displacements  $\mathbf{u}_N$ ,  $\mathbf{u}_T$ , normal and shear forces are computed (if  $u_N > 0$ , the contact is deleted without generating any forces):

$$\begin{aligned}\mathbf{F}_N &= K_N \mathbf{u}_N \mathbf{n}, \\ \mathbf{F}_T^t &= K_T \mathbf{u}_T\end{aligned}$$

where  $\mathbf{F}_N$  is normal force and  $\mathbf{F}_T$  is trial shear force. A simple non-associated stress return algorithm is applied to compute final shear force

$$\mathbf{F}_T = \begin{cases} \mathbf{F}_T^t \frac{|\mathbf{F}_N| \tan \varphi}{F_T^t} & \text{if } |\mathbf{F}_T| > |\mathbf{F}_N| \tan \varphi, \\ \mathbf{F}_T^t & \text{otherwise.} \end{cases}$$

Summary force  $\mathbf{F} = \mathbf{F}_N + \mathbf{F}_T$  is then applied to both particles – each particle accumulates forces and torques acting on it in the course of each step. Because the force computed acts at contact point  $\mathbf{C}$ , which is difference from spheres' centers, torque generated by  $\mathbf{F}$  must also be considered.

$$\begin{aligned}\mathbf{F}_{1+} &= \mathbf{F} & \mathbf{F}_{2+} &= -\mathbf{F} \\ \mathbf{T}_{1+} &= \mathbf{d}_1(-\mathbf{n}) \times \mathbf{F} & \mathbf{T}_{2+} &= \mathbf{d}_2 \mathbf{n} \times \mathbf{F}.\end{aligned}$$

## 1.5 Motion integration

Each particle accumulates generalized forces (forces and torques) from the contacts in which it participates. These generalized forces are then used to integrate motion equations for each particle separately; therefore, we omit  $i$  indices denoting the  $i$ -th particle in this section.

The customary leapfrog scheme (also known as the Verlet scheme) is used, with some adjustments for rotation of non-spherical particles, as explained below. The ‘‘leapfrog’’ name comes from the fact that even derivatives of position/orientation are known at on-step points, whereas odd derivatives are known at mid-step points. Let us recall that we use  $\mathbf{a}^-$ ,  $\mathbf{a}^\circ$ ,  $\mathbf{a}^+$  for on-step values of  $\mathbf{a}$  at  $t - \Delta t$ ,  $t$  and  $t + \Delta t$  respectively; and  $\mathbf{a}^\ominus$ ,  $\mathbf{a}^\oplus$  for mid-step values of  $\mathbf{a}$  at  $t - \Delta t/2$ ,  $t + \Delta t/2$ .

Described integration algorithms are implemented in the NewtonIntegrator class in Yade.

### 1.5.1 Position

Integrating motion consists in using current acceleration  $\ddot{\mathbf{u}}^\circ$  on a particle to update its position from the current value  $\mathbf{u}^\circ$  to its value at the next timestep  $\mathbf{u}^+$ . Computation of acceleration, knowing current forces  $\mathbf{F}$  acting on the particle in question and its mass  $m$ , is simply

$$\ddot{\mathbf{u}}^\circ = \mathbf{F}/m.$$

Using the 2nd order finite difference with step  $\Delta t$ , we obtain

$$\ddot{\mathbf{u}}^\circ \cong \frac{\mathbf{u}^- - 2\mathbf{u}^\circ + \mathbf{u}^+}{\Delta t^2}$$

from which we express

$$\begin{aligned}\mathbf{u}^+ &= 2\mathbf{u}^\circ - \mathbf{u}^- + \ddot{\mathbf{u}}^\circ \Delta t^2 = \\ &= \mathbf{u}^\circ + \Delta t \underbrace{\left( \frac{\mathbf{u}^\circ - \mathbf{u}^-}{\Delta t} + \ddot{\mathbf{u}}^\circ \Delta t \right)}_{(\dot{\mathbf{u}})}.\end{aligned}$$

Typically,  $\mathbf{u}^-$  is already not known (only  $\mathbf{u}^\circ$  is); we notice, however, that

$$\dot{\mathbf{u}}^\ominus \simeq \frac{\mathbf{u}^\circ - \mathbf{u}^-}{\Delta t},$$

i.e. the mean velocity during the previous step, which is known. Plugging this approximate into the (†) term, we also notice that mean velocity during the current step can be approximated as

$$\dot{\mathbf{u}}^\oplus \simeq \dot{\mathbf{u}}^\ominus + \ddot{\mathbf{u}}^\circ \Delta t,$$

which is (‡); we arrive finally at

$$\mathbf{u}^+ = \mathbf{u}^\circ + \Delta t (\dot{\mathbf{u}}^\ominus + \ddot{\mathbf{u}}^\circ \Delta t).$$

The algorithm can then be written down by first computing current mean velocity  $\dot{\mathbf{u}}^\oplus$  which we need to store for the next step (just as we use its old value  $\dot{\mathbf{u}}^\ominus$  now), then computing the position for the next time step  $\mathbf{u}^+$ :

$$\begin{aligned} \dot{\mathbf{u}}^\oplus &= \dot{\mathbf{u}}^\ominus + \ddot{\mathbf{u}}^\circ \Delta t \\ \mathbf{u}^+ &= \mathbf{u}^\circ + \dot{\mathbf{u}}^\oplus \Delta t. \end{aligned}$$

Positions are known at times  $i\Delta t$  (if  $\Delta t$  is constant) while velocities are known at  $i\Delta t + \frac{\Delta t}{2}$ . The facet that they interleave (jump over each other) in such way gave rise to the colloquial name “leapfrog” scheme.

### 1.5.2 Orientation (spherical)

Updating particle orientation  $\mathbf{q}^\circ$  proceeds in an analogous way to position update. First, we compute current angular acceleration  $\dot{\boldsymbol{\omega}}^\circ$  from known current torque  $\mathbf{T}$ . For spherical particles where the inertia tensor is diagonal in any orientation (therefore also in current global orientation), satisfying  $\mathbf{I}_{11} = \mathbf{I}_{22} = \mathbf{I}_{33}$ , we can write

$$\dot{\omega}_i^\circ = \mathbf{T}_i / \mathbf{I}_{11},$$

We use the same approximation scheme, obtaining an equation analogous to (??)

$$\boldsymbol{\omega}^\oplus = \boldsymbol{\omega}^\ominus + \Delta t \dot{\boldsymbol{\omega}}^\circ.$$

The quaternion  $\Delta \mathbf{q}$  representing rotation vector  $\boldsymbol{\omega}^\oplus \Delta t$  is constructed, i.e. such that

$$\begin{aligned} (\Delta \mathbf{q})_{\mathfrak{g}} &= |\boldsymbol{\omega}^\oplus|, \\ (\Delta \mathbf{q})_{\mathbf{u}} &= \widehat{\boldsymbol{\omega}^\oplus} \end{aligned}$$

Finally, we compute the next orientation  $\mathbf{q}^+$  by rotation composition

$$\mathbf{q}^+ = \Delta \mathbf{q} \mathbf{q}^\circ.$$

### 1.5.3 Orientation (aspherical)

Integrating rotation of aspherical particles is considerably more complicated than their position, as their local reference frame is not inertial. Rotation of rigid body in the local frame, where inertia matrix  $\mathbf{I}$  is diagonal, is described in the continuous form by Euler’s equations ( $i \in \{1, 2, 3\}$  and  $i, j, k$  are subsequent indices):

$$\mathbf{T}_i = \mathbf{I}_{ii} \dot{\omega}_i + (\mathbf{I}_{kk} - \mathbf{I}_{jj}) \omega_j \omega_k.$$

Due to the presence of the current values of both  $\boldsymbol{\omega}$  and  $\dot{\boldsymbol{\omega}}$ , they cannot be solved using the standard leapfrog algorithm (that was the case for translational motion and also for the spherical bodies’ rotation where this equation reduced to  $\mathbf{T} = \mathbf{I} \dot{\boldsymbol{\omega}}$ ).

The algorithm presented here is described by [Allen1989] (pg. 84–89) and was designed by Fincham for molecular dynamics problems; it is based on extending the leapfrog algorithm by mid-step/on-step

estimators of quantities known at on-step/mid-step points in the basic formulation. Although it has received criticism and more precise algorithms are known ([Omelyan1999], [Neto2006], [Johnson2008]), this one is currently implemented in Yade for its relative simplicity.

Each body has its local coordinate system based on the principal axes of inertia for that body. We use  $\tilde{\bullet}$  to denote vectors in local coordinates. The orientation of the local system is given by the current particle's orientation  $q^\circ$  as a quaternion; this quaternion can be expressed as the (current) rotation matrix  $\mathbf{A}$ . Therefore, every vector  $\mathbf{a}$  is transformed as  $\tilde{\mathbf{a}} = \mathbf{q}\mathbf{a}\mathbf{q}^* = \mathbf{A}\mathbf{a}$ . Since  $\mathbf{A}$  is a rotation (orthogonal) matrix, the inverse rotation  $\mathbf{A}^{-1} = \mathbf{A}^\top$ .

For given particle in question, we know

- $\tilde{\mathbf{I}}^\circ$  (constant) inertia matrix; diagonal, since in local, principal coordinates,
- $\mathbf{T}^\circ$  external torque,
- $q^\circ$  current orientation (and its equivalent rotation matrix  $\mathbf{A}$ ),
- $\boldsymbol{\omega}^\ominus$  mid-step angular velocity,
- $\mathbf{L}^\ominus$  mid-step angular momentum; this is an auxiliary variable that must be tracked in addition for use in this algorithm. It will be zero in the initial step.

Our goal is to compute new values of the latter three, that is  $\mathbf{L}^\oplus$ ,  $q^\oplus$ ,  $\boldsymbol{\omega}^\oplus$ . We first estimate current angular momentum and compute current local angular velocity:

$$\begin{aligned}\mathbf{L}^\circ &= \mathbf{L}^\ominus + \mathbf{T}^\circ \frac{\Delta t}{2}, & \tilde{\mathbf{L}}^\circ &= \mathbf{A}\mathbf{L}^\circ, \\ \mathbf{L}^\oplus &= \mathbf{L}^\ominus + \mathbf{T}^\circ \Delta t, & \tilde{\mathbf{L}}^\oplus &= \mathbf{A}\mathbf{L}^\oplus, \\ \tilde{\boldsymbol{\omega}}^\circ &= \tilde{\mathbf{I}}^\circ^{-1} \tilde{\mathbf{L}}^\circ, \\ \tilde{\boldsymbol{\omega}}^\oplus &= \tilde{\mathbf{I}}^\circ^{-1} \tilde{\mathbf{L}}^\oplus.\end{aligned}$$

Then we compute  $\dot{q}^\circ$ , using  $q^\circ$  and  $\tilde{\boldsymbol{\omega}}^\circ$ :

$$\begin{aligned}\begin{pmatrix} \dot{q}_w^\circ \\ \dot{q}_x^\circ \\ \dot{q}_y^\circ \\ \dot{q}_z^\circ \end{pmatrix} &= \frac{1}{2} \begin{pmatrix} q_w^\circ & -q_x^\circ & -q_y^\circ & -q_z^\circ \\ q_x^\circ & q_w^\circ & -q_z^\circ & q_y^\circ \\ q_y^\circ & q_z^\circ & q_w^\circ & -q_x^\circ \\ q_z^\circ & -q_y^\circ & q_x^\circ & q_w^\circ \end{pmatrix} \begin{pmatrix} 0 \\ \tilde{\omega}_x^\circ \\ \tilde{\omega}_y^\circ \\ \tilde{\omega}_z^\circ \end{pmatrix}, \\ q^\oplus &= q^\circ + \dot{q}^\circ \frac{\Delta t}{2}.\end{aligned}$$

We evaluate  $\dot{q}^\oplus$  from  $q^\oplus$  and  $\tilde{\boldsymbol{\omega}}^\oplus$  in the same way as in (??) but shifted by  $\Delta t/2$  ahead. Then we can finally compute the desired values

$$\begin{aligned}q^+ &= q^\circ + \dot{q}^\oplus \Delta t, \\ \boldsymbol{\omega}^\oplus &= \mathbf{A}^{-1} \tilde{\boldsymbol{\omega}}^\oplus\end{aligned}$$

### 1.5.4 Clumps (rigid aggregates)

DEM simulations frequently make use of rigid aggregates of particles to model complex shapes [Price2007] called *clumps*, typically composed of many spheres. Dynamic properties of clumps are computed from the properties of its members:

- For non-overlapping clump members the clump's mass  $m_c$  is summed over members, the inertia tensor  $\mathbf{I}_c$  is computed using the parallel axes theorem:  $\mathbf{I}_c = \sum_i (m_i * d_i^2 + \mathbf{I}_i)$ , where  $m_i$  is the mass of clump member  $i$ ,  $d_i$  is the distance from center of clump member  $i$  to clump's centroid and  $\mathbf{I}_i$  is the inertia tensor of the clump member  $i$ .

- For overlapping clump members the clump's mass  $m_c$  is summed over cells using a regular grid spacing inside axis-aligned bounding box (Aabb) of the clump, the inertia tensor is computed using the parallel axes theorem:  $\mathbf{I}_c = \sum_j (m_j * d_j^2 + I_j)$ , where  $m_j$  is the mass of cell  $j$ ,  $d_j$  is the distance from cell center to clump's centroid and  $I_j$  is the inertia tensor of the cell  $j$ .

Local axes are oriented such that they are principal and inertia tensor is diagonal and clump's orientation is changed to compensate rotation of the local system, as to not change the clump members' positions in global space. Initial positions and orientations of all clump members in local coordinate system are stored.

In Yade (class Clump), clump members behave as stand-alone particles during simulation for purposes of collision detection and contact resolution, except that they have no contacts created among themselves within one clump. It is at the stage of motion integration that they are treated specially. Instead of integrating each of them separately, forces/torques on those particles  $\mathbf{F}_i$ ,  $\mathbf{T}_i$  are converted to forces/torques on the clump itself. Let us denote  $\mathbf{r}_i$  relative position of each particle with regards to clump's centroid, in global orientation. Then summary force and torque on the clump are

$$\begin{aligned}\mathbf{F}_c &= \sum \mathbf{F}_i, \\ \mathbf{T}_c &= \sum \mathbf{r}_i \times \mathbf{F}_i + \mathbf{T}_i.\end{aligned}$$

Motion of the clump is then integrated, using aspherical rotation integration. Afterwards, clump members are displaced in global space, to keep their initial positions and orientations in the clump's local coordinate system. In such a way, relative positions of clump members are always the same, resulting in the behavior of a rigid aggregate.

### 1.5.5 Numerical damping

In simulations of quasi-static phenomena, it is desirable to dissipate kinetic energy of particles. Since most constitutive laws (including `Law_ScGeom_FrictPhys_Basic` shown above, *Stress evaluation (example)*) do not include velocity-based damping (such as one in [Addetta2001]), it is possible to use artificial numerical damping. The formulation is described in [Pfc3dManual30], although our version is slightly adapted. The basic idea is to decrease forces which increase the particle velocities and vice versa by  $(\Delta F)_d$ , comparing the current acceleration sense and particle velocity sense. This is done by component, which makes the damping scheme clearly non-physical, as it is not invariant with respect to coordinate system rotation; on the other hand, it is very easy to compute. Cundall proposed the form (we omit particle indices  $i$  since it applies to all of them separately):

$$\frac{(\Delta F)_{dw}}{\mathbf{F}_w} = -\lambda_d \operatorname{sgn}(\mathbf{F}_w \dot{\mathbf{u}}_w^\ominus), \quad w \in \{x, y, z\}$$

where  $\lambda_d$  is the damping coefficient. This formulation has several advantages [Hentz2003]:

- it acts on forces (accelerations), not constraining uniform motion;
- it is independent of eigenfrequencies of particles, they will be all damped equally;
- it needs only the dimensionless parameter  $\lambda_d$  which does not have to be scaled.

In Yade, we use the adapted form

$$\frac{(\Delta F)_{dw}}{\mathbf{F}_w} = -\lambda_d \operatorname{sgn} \mathbf{F}_w \underbrace{\left( \dot{\mathbf{u}}_w^\ominus + \frac{\ddot{\mathbf{u}}_w^\ominus \Delta t}{2} \right)}_{\simeq \dot{\mathbf{u}}_w^\ominus}, \quad (1.9)$$

where we replaced the previous mid-step velocity  $\dot{\mathbf{u}}^\ominus$  by its on-step estimate in parentheses. This is to avoid locked-in forces that appear if the velocity changes its sign due to force application at each step, i.e. when the particle in question oscillates around the position of equilibrium with  $2\Delta t$  period.

In Yade, damping (1.9) is implemented in the `NewtonIntegrator` engine; the damping coefficient  $\lambda_d$  is `NewtonIntegrator.damping`.

## 1.5.6 Stability considerations

### Critical timestep

In order to ensure stability for the explicit integration scheme, an upper limit is imposed on  $\Delta t$ :

$$\Delta t_{\text{cr}} = \frac{2}{\omega_{\text{max}}} \quad (1.10)$$

where  $\omega_{\text{max}}$  is the highest eigenfrequency within the system.

### Single mass-spring system

Single 1D mass-spring system with mass  $m$  and stiffness  $K$  is governed by the equation

$$m\ddot{x} = -Kx$$

where  $x$  is displacement from the mean (equilibrium) position. The solution of harmonic oscillation is  $x(t) = A \cos(\omega t + \varphi)$  where phase  $\varphi$  and amplitude  $A$  are determined by initial conditions. The angular frequency

$$\omega^{(1)} = \sqrt{\frac{K}{m}} \quad (1.11)$$

does not depend on initial conditions. Since there is one single mass,  $\omega_{\text{max}}^{(1)} = \omega^{(1)}$ . Plugging (1.11) into (1.10), we obtain

$$\Delta t_{\text{cr}}^{(1)} = 2/\omega_{\text{max}}^{(1)} = 2\sqrt{m/K}$$

for a single oscillator.

### General mass-spring system

In a general mass-spring system, the highest frequency occurs if two connected masses  $m_i$ ,  $m_j$  are in opposite motion; let us suppose they have equal velocities (which is conservative) and they are connected by a spring with stiffness  $K_i$ : displacement  $\Delta x_i$  of  $m_i$  will be accompanied by  $\Delta x_j = -\Delta x_i$  of  $m_j$ , giving  $\Delta F_i = -K_i(\Delta x_i - (-\Delta x_i)) = -2K_i\Delta x_i$ . That results in apparent stiffness  $K_i^{(2)} = 2K_i$ , giving maximum eigenfrequency of the whole system

$$\omega_{\text{max}} = \max_i \sqrt{K_i^{(2)}/m_i}.$$

The overall critical timestep is then

$$\Delta t_{\text{cr}} = \frac{2}{\omega_{\text{max}}} = \min_i 2\sqrt{\frac{m_i}{K_i^{(2)}}} = \min_i 2\sqrt{\frac{m_i}{2K_i}} = \min_i \sqrt{2}\sqrt{\frac{m_i}{K_i}}. \quad (1.12)$$

This equation can be used for all 6 degrees of freedom (DOF) in translation and rotation, by considering generalized mass and stiffness matrices  $M$  and  $K$ , and replacing fractions  $\frac{m_i}{K_i}$  by eigen values of  $M.K^{-1}$ . The critical timestep is then associated to the eigen mode with highest frequency :

$$\Delta t_{\text{cr}} = \min \Delta t_{\text{cr}k}, \quad k \in \{1, \dots, 6\}. \quad (1.13)$$

### DEM simulations

In DEM simulations, per-particle stiffness  $K_{ij}$  is determined from the stiffnesses of contacts in which it participates [Chareyre2005]. Suppose each contact has normal stiffness  $K_{Nk}$ , shear stiffness  $K_{Tk} =$

$\xi K_{Nk}$  and is oriented by normal  $\mathbf{n}_k$ . A translational stiffness matrix  $\mathbf{K}_{ij}$  can be defined as the sum of contributions of all contacts in which it participates (indices  $k$ ), as

$$\mathbf{K}_{ij} = \sum_k (K_{Nk} - K_{Tk}) \mathbf{n}_i \mathbf{n}_j + K_{Tk} = \sum_j K_{Nk} ((1 - \xi) \mathbf{n}_i \mathbf{n}_j + \xi) \quad (1.14)$$

with  $i$  and  $j \in \{x, y, z\}$ . Equations (1.13) and (1.14) determine  $\Delta t_{cr}$  in a simulation. A similar approach generalized to all 6 DOFs is implemented by the GlobalStiffnessTimeStepper engine in Yade. The derivation of generalized stiffness including rotational terms is very similar but not developed here, for simplicity. For full reference, see “PFC3D - Theoretical Background”.

Note that for computation efficiency reasons, eigenvalues of the stiffness matrices are not computed. They are only approximated assuming that DOF’s are uncoupled, and using diagonal terms of  $\mathbf{K} \cdot \mathbf{M}^{-1}$ . They give good approximates in typical mechanical systems.

There is one important condition that  $\omega_{max} > 0$ : if there are no contacts between particles and  $\omega_{max} = 0$ , we would obtain value  $\Delta t_{cr} = \infty$ . While formally correct, this value is numerically erroneous: we were silently supposing that stiffness remains constant during each timestep, which is not true if contacts are created as particles collide. In case of no contact, therefore, stiffness must be pre-estimated based on future interactions, as shown in the next section.

### Estimation of $\Delta t_{cr}$ by wave propagation speed

Estimating timestep in absence of interactions is based on the connection between interaction stiffnesses and the particle’s properties. Note that in this section, symbols  $E$  and  $\rho$  refer exceptionally to Young’s modulus and density of *particles*, not of macroscopic arrangement.

In Yade, particles have associated Material which defines density  $\rho$  (Material.density), and also may define (in ElastMat and derived classes) particle’s “Young’s modulus”  $E$  (ElastMat.young).  $\rho$  is used when particle’s mass  $m$  is initially computed from its  $\rho$ , while  $E$  is taken in account when creating new interaction between particles, affecting stiffness  $K_N$ . Knowing  $m$  and  $K_N$ , we can estimate (1.14) for each particle; we obviously neglect

- number of interactions per particle  $N_i$ ; for a “reasonable” radius distribution, however, there is a geometrically imposed upper limit (6 for a 2D-packing of spheres with equal radii, for instance);
- the exact relationship the between particles’ rigidities  $E_i$ ,  $E_j$ , supposing only that  $K_N$  is somehow proportional to them.

By defining  $E$  and  $\rho$ , particles have continuum-like quantities. Explicit integration schemes for continuum equations impose a critical timestep based on sonic speed  $\sqrt{E/\rho}$ ; the elastic wave must not propagate farther than the minimum distance of integration points  $l_{min}$  during one step. Since  $E$ ,  $\rho$  are parameters of the elastic continuum and  $l_{min}$  is fixed beforehand, we obtain

$$\Delta t_{cr}^{(c)} = l_{min} \sqrt{\frac{\rho}{E}}$$

For our purposes, we define  $E$  and  $\rho$  for each particle separately;  $l_{min}$  can be replaced by the sphere’s radius  $R_i$ ; technically,  $l_{min} = 2R_i$  could be used, but because of possible interactions of spheres and facets (which have zero thickness), we consider  $l_{min} = R_i$  instead. Then

$$\Delta t_{cr}^{(p)} = \min_i R_i \sqrt{\frac{\rho_i}{E_i}}$$

This algorithm is implemented in the utils.PWaveTimeStep function.

Let us compare this result to (1.12); this necessitates making several simplifying hypotheses:

- all particles are spherical and have the same radius  $R$ ;
- the sphere’s material has the same  $E$  and  $\rho$ ;
- the average number of contacts per sphere is  $N$ ;
- the contacts have sufficiently uniform spatial distribution around each particle;

- the  $\xi = K_N/K_T$  ratio is constant for all interactions;
- contact stiffness  $K_N$  is computed from  $E$  using a formula of the form

$$K_N = E\pi'R', \quad (1.15)$$

where  $\pi'$  is some constant depending on the algorithm in use<sup>footnote</sup>{For example,  $\pi' = \pi/2$  in the concrete particle model (Ip2\_CpmMat\_CpmMat\_CpmPhys), while  $\pi' = 2$  in the classical DEM model (Ip2\_FrictMat\_FrictMat\_FrictPhys) as implemented in Yade.} and  $R'$  is half-distance between spheres in contact, equal to  $R$  for the case of interaction radius  $R_I = 1$ . If  $R_I = 1$  (and  $R' \equiv R$  by consequence), all interactions will have the same stiffness  $K_N$ . In other cases, we will consider  $K_N$  as the average stiffness computed from average  $R'$  (see below).

As all particles have the same parameters, we drop the  $i$  index in the following formulas.

We try to express the average per-particle stiffness from (1.14). It is a sum over all interactions where  $K_N$  and  $\xi$  are scalars that will not rotate with interaction, while  $\mathbf{n}_w$  is  $w$ -th component of unit interaction normal  $\mathbf{n}$ . Since we supposed uniform spatial distribution, we can replace  $\mathbf{n}_w^2$  by its average value  $\bar{\mathbf{n}}_w^2$ . Recognizing components of  $\mathbf{n}$  as direction cosines, the average values of  $\mathbf{n}_w^2$  is  $1/3$ . %we find the average value by integrating over all possible orientations, which are uniformly distributed in space:

Moreover, since all directions are equal, we can write the per-body stiffness as  $K = \mathbf{K}_w$  for all  $w \in \{x, y, z\}$ . We obtain

$$K = \sum K_N \left( (1 - \xi) \frac{1}{3} + \xi \right) = \sum K_N \frac{1 - 2\xi}{3}$$

and can put constant terms (everything) in front of the summation.  $\sum 1$  equals the number of contacts per sphere, i.e.  $N$ . Arriving at

$$K = NK_N \frac{1 - 2\xi}{3},$$

we substitute  $K$  into (1.12) using (1.15):

$$\Delta t_{cr} = \sqrt{2} \sqrt{\frac{m}{K}} = \sqrt{2} \sqrt{\frac{\frac{4}{3}\pi R^3 \rho}{NE\pi'R \frac{1-2\xi}{3}}} = \underbrace{R \sqrt{\frac{\rho}{E}}}_{\Delta t_{cr}^{(p)}} 2 \sqrt{\frac{\pi/\pi'}{N(1-2\xi)}}.$$

The ratio of timestep  $\Delta t_{cr}^{(p)}$  predicted by the p-wave velocity and numerically stable timestep  $\Delta t_{cr}$  is the inverse value of the last (dimensionless) term:

$$\frac{\Delta t_{cr}^{(p)}}{\Delta t_{cr}} = 2 \sqrt{\frac{N(1+\xi)}{\pi/\pi'}}.$$

Actual values of this ratio depend on characteristics of packing  $N$ ,  $K_N/K_T = \xi$  ratio and the way of computing contact stiffness from particle rigidity. Let us show it for two models in Yade:

**Concrete particle model** computes contact stiffness from the equivalent area  $A_{eq}$  first (1.6),

$$A_{eq} = \pi R^2 K_N = \frac{A_{eq} E}{d_0}.$$

$d_0$  is the initial contact length, which will be, for interaction radius (1.5)  $R_I > 1$ , in average larger than  $2R$ . For  $R_I = 1.5$  (sect. *sect-calibration-elastic-properties*), we can roughly estimate  $\bar{d}_0 = 1.25 \cdot 2R = \frac{5}{2}R$ , getting

$$K_N = E \left( \frac{2}{5}\pi \right) R$$

where  $\frac{2}{5}\pi = \pi'$  by comparison with (1.15).

Interaction radius  $R_I = 1.5$  leads to average  $N \approx 12$  interactions per sphere for dense packing of spheres with the same radius  $R$ .  $\xi = 0.2$  is calibrated (sect. *sect-calibration-elastic-properties*) to match the desired macroscopic Poisson's ratio  $\nu = 0.2$ .

Finally, we obtain the ratio

$$\frac{\Delta t_{\text{cr}}^{(\text{p})}}{\Delta t_{\text{cr}}} = 2\sqrt{\frac{12(1 - 2 \cdot 0.2)}{\frac{\pi}{(2/5)\pi}}} = 3.39,$$

showing significant overestimation by the p-wave algorithm.

**Non-cohesive dry friction model** is the basic model proposed by Cundall explained in *Stress evaluation (example)*. Supposing almost-constant sphere radius  $R$  and rather dense packing, each sphere will have  $N = 6$  interactions on average (that corresponds to maximally dense packing of spheres with a constant radius). If we use the `Ip2_FrictMat_FrictMat_FrictPhys` class, we have  $\pi' = 2$ , as  $K_N = E2R$ ; we again use  $\xi = 0.2$  (for lack of a more significant value). In this case, we obtain the result

$$\frac{\Delta t_{\text{cr}}^{(\text{p})}}{\Delta t_{\text{cr}}} = 2\sqrt{\frac{6(1 - 2 \cdot 0.2)}{\pi/2}} = 3.02$$

which again overestimates the numerical critical timestep.

To conclude, p-wave timestep gives estimate proportional to the real  $\Delta t_{\text{cr}}$ , but in the cases shown, the value of about  $\Delta t = 0.3\Delta t_{\text{cr}}^{(\text{p})}$  should be used to guarantee stable simulation.

### Non-elastic $\Delta t$ constraints

Let us note at this place that not only  $\Delta t_{\text{cr}}$  assuring numerical stability of motion integration is a constraint. In systems where particles move at relatively high velocities, position change during one timestep can lead to non-elastic irreversible effects such as damage. The  $\Delta t$  needed for reasonable result can be lower  $\Delta t_{\text{cr}}$ . We have no rigorously derived rules for such cases.

## 1.6 Periodic boundary conditions

While most DEM simulations happen in  $R^3$  space, it is frequently useful to avoid boundary effects by using periodic space instead. In order to satisfy periodicity conditions, periodic space is created by repetition of parallelepiped-shaped cell. In Yade, periodic space is implemented in the `Cell` class. The geometry of the cell in the reference coordinates system is defined by three edges of the parallelepiped. The corresponding base vectors are stored in the columns of matrix  $\mathbf{H}$  (`Cell.hSize`).

The initial  $\mathbf{H}$  can be explicitly defined as a  $3 \times 3$  matrix at the beginning of the simulation. There are no restrictions on the possible shapes: any parallelepiped is accepted as the initial cell. If the base vectors are axis-aligned, defining only their sizes can be more convenient than defining the full  $\mathbf{H}$  matrix; in that case it is enough to define the norms of columns in  $\mathbf{H}$  (see `Cell.size`).

After the definition of the initial cell's geometry,  $\mathbf{H}$  should generally not be modified by direct assignment. Instead, its deformation rate will be defined via the velocity gradient `Cell.velGrad` described below. It is the only variable that let the period deformation be correctly accounted for in constitutive laws and Newton integrator (`NewtonIntegrator`).

### 1.6.1 Deformations handling

The deformation of the cell over time is defined via a matrix representing the gradient of an homogeneous velocity field  $\nabla \mathbf{v}$  (`Cell.velGrad`). This gradient represents arbitrary combinations of rotations and stretches. It can be imposed externally or updated by boundary controllers (see `PeriTriaxController` or `Peri3dController`) in order to reach target strain values or to maintain some prescribed stress.

The velocity gradient is integrated automatically over time, and the cumulated transformation is reflected in the transformation matrix  $\mathbf{F}$  (`Cell.trsf`) and the current shape of the cell  $\mathbf{H}$ . The per-step transformation update reads (it is similar for  $\mathbf{H}$ ), with  $\mathbf{I}$  the identity matrix:

$$\mathbf{F}^+ = (\mathbf{I} + \nabla \mathbf{v} \Delta t) \mathbf{F}^\circ.$$

$\mathbf{F}$  can be set back to identity at any point in simulations, in order to define the current state as reference for strains definition in boundary controllers. It will have no effect on  $\mathbf{H}$ .

Along with the automatic integration of cell transformation, there is an option to homothetically displace all particles so that  $\nabla\mathbf{v}$  is applied over the whole simulation (enabled via `Cell.homoDeform`). This avoids all boundary effects coming from change of the velocity gradient.

## 1.6.2 Collision detection in periodic cell

In usual implementations, particle positions are forced to be inside the cell by wrapping their positions if they get over the boundary (so that they appear on the other side). As we wanted to avoid abrupt changes of position (it would make particle's velocity inconsistent with step displacement change), a different method was chosen.

### Approximate collision detection

Pass 1 collision detection (based on sweep and prune algorithm, sect. *Sweep and prune*) operates on axis-aligned bounding boxes (Aabb) of particles. During the collision detection phase, bounds of all Aabb's are wrapped inside the cell in the first step. At subsequent runs, every bound remembers by how many cells it was initially shifted from coordinate given by the Aabb and uses this offset repeatedly as it is being updated from Aabb during particle's motion. Bounds are sorted using the periodic insertion sort algorithm (sect. *Periodic insertion sort algorithm*), which tracks periodic cell boundary  $\parallel$ .

Upon inversion of two Aabb's, their collision along all three axes is checked, wrapping real coordinates inside the cell for that purpose.

This algorithm detects collisions as if all particles were inside the cell but without the need of constructing "ghost particles" (to represent periodic image of a particle which enters the cell from the other side) or changing the particle's positions.

It is required by the implementation (and partly by the algorithm itself) that particles do not span more than half of the current cell size along any axis; the reason is that otherwise two (or more) contacts between both particles could appear, on each side. Since Yade identifies contacts by `Body.id` of both bodies, they would not be distinguishable.

In presence of shear, the sweep-and-prune collider could not sort bounds independently along three axes: collision along  $x$  axis depends on the mutual position of particles on the  $y$  axis. Therefore, bounding boxes *are expressed in transformed coordinates* which are perpendicular in the sense of collision detection. This requires some extra computation: Aabb of sphere in transformed coordinates will no longer be cube, but cuboid, as the sphere itself will appear as ellipsoid after transformation. Inversely, the sphere in simulation space will have a parallelepiped bounding "box", which is cuboid around the ellipsoid in transformed axes (the Aabb has axes aligned with transformed cell basis). This is shown in fig. [fig-cell-shear-aabb](#).

The restriction of a single particle not spanning more than half of the transformed axis becomes stringent as Aabb is enlarged due to shear. Considering Aabb of a sphere with radius  $r$  in the cell where  $x' \equiv x$ ,  $z' \equiv z$ , but  $\angle(\mathbf{y}, \mathbf{y}') = \varphi$ , the  $x$ -span of the Aabb will be multiplied by  $1/\cos\varphi$ . For the infinite shear  $\varphi \rightarrow \pi/2$ , which can be desirable to simulate, we have  $1/\cos\varphi \rightarrow \infty$ . Fortunately, this limitation can be easily circumvented by realizing the quasi-identity of all periodic cells which, if repeated in space, create the same grid with their corners: the periodic cell can be flipped, keeping all particle interactions intact, as shown in fig. [fig-cell-flip](#). It only necessitates adjusting the `Interaction.cellDist` of interactions and re-initialization of the collider (`Collider::invalidatePersistentData`). Cell flipping is implemented in the `utils.flipCell` function.

This algorithm is implemented in `InsertionSortCollider` and is used whenever simulation is periodic (`Omega.isPeriodic`); individual `BoundFunctor`'s are responsible for computing sheared Aabb's; currently it is implemented for spheres and facets (in `Bo1_Sphere_Aabb` and `Bo1_Facet_Aabb` respectively).

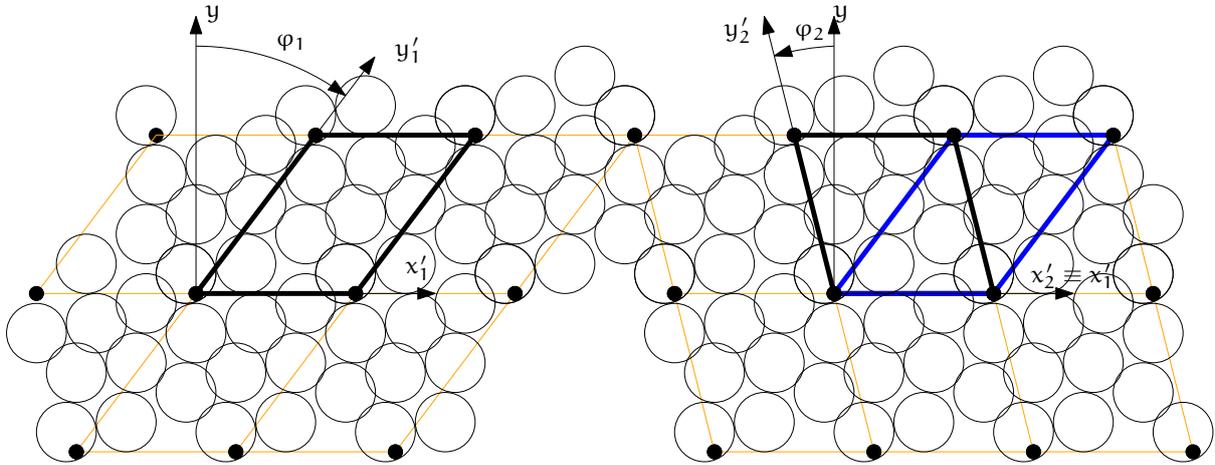


Figure 1.6: Flipping cell (`utils.flipCell`) to avoid infinite stretch of the bounding boxes' spans with growing  $\varphi$ . Cell flip does not affect interactions from the point of view of the simulation. The periodic arrangement on the left is the same as the one on the right, only the cell is situated differently between identical grid points of repetition; at the same time  $|\varphi_2| < |\varphi_1|$  and sphere bounding box's  $x$ -span stretched by  $1/\cos \varphi$  becomes smaller. Flipping can be repeated, making effective infinite shear possible.

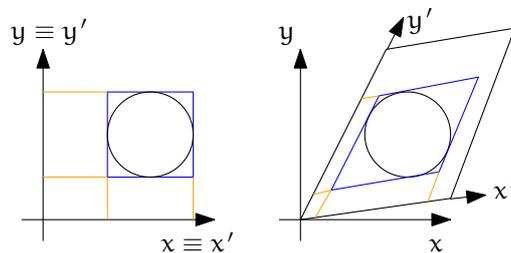


Figure 1.7: Constructing axis-aligned bounding box (Aabb) of a sphere in simulation space coordinates (without periodic cell – left) and transformed cell coordinates (right), where collision detection axes  $x'$ ,  $y'$  are not identical with simulation space axes  $x$ ,  $y$ . Bounds' projection to axes is shown by orange lines.

### Exact collision detection

When the collider detects approximate contact (on the Aabb level) and the contact does not yet exist, it creates *potential* contact, which is subsequently checked by exact collision algorithms (depending on the combination of Shapes). Since particles can interact over many periodic cells (recall we never change their positions in simulation space), the collider embeds the relative cell coordinate of particles in the interaction itself (Interaction.cellDist) as an *integer* vector  $\mathbf{c}$ . Multiplying current cell size  $\mathbf{T}\mathbf{s}$  by  $\mathbf{c}$  component-wise, we obtain particle offset  $\Delta\mathbf{x}$  in aperiodic  $\mathbb{R}^3$ ; this value is passed (from InteractionLoop) to the functor computing exact collision (IGeomFunctor), which adds it to the position of the particle Interaction.id2.

By storing the integral offset  $\mathbf{c}$ ,  $\Delta\mathbf{x}$  automatically updates as cell parameters change.

### Periodic insertion sort algorithm

The extension of sweep and prune algorithm (described in *Sweep and prune*) to periodic boundary conditions is non-trivial. Its cornerstone is a periodic variant of the insertion sort algorithm, which involves keeping track of the “period” of each boundary; e.g. taking period  $(0, 10)$ , then  $8_1 \equiv -2_2 < 2_2$  (subscript indicating period). Doing so efficiently (without shuffling data in memory around as bound wraps from one period to another) requires moving period boundary rather than bounds themselves and making the comparison work transparently at the edge of the container.

This algorithm was also extended to handle non-orthogonal periodic Cell boundaries by working in transformed rather than Cartesian coordinates; this modifies computation of Aabb from Cartesian coordinates in which bodies are positioned (treated in detail in *Approximate collision detection*).

The sort algorithm is tracking Aabb extrema along all axes. At the collider’s initialization, each value is assigned an integral period, i.e. its distance from the cell’s interior expressed in the cell’s dimension along its respective axis, and is wrapped to a value inside the cell. We put the period number in subscript.

Let us give an example of coordinate sequence along x axis (in a real case, the number of elements would be even, as there is maximum and minimum value couple for each particle; this demonstration only shows the sorting algorithm, however.)

$$4_1 \quad 12_2 \quad || \quad -1_2 \quad -2_4 \quad 5_0$$

with cell x-size  $s_x = 10$ . The  $4_1$  value then means that the real coordinate  $x_i$  of this extremum is  $x_i + 1 \cdot 10 = 4$ , i.e.  $x_i = -4$ . The  $||$  symbol denotes the periodic cell boundary.

Sorting starts from the first element in the cell, i.e. right of  $||$ , and inverts elements as in the aperiodic variant. The rules are, however, more complicated due to the presence of the boundary  $||$ :

$(\leq)$	stop inverting if neighbors are ordered;
$(  \bullet)$	current element left of $  $ is below 0 (lower period boundary); in this case, decrement element’s period, decrease its coordinate by $s_x$ and move $  $ right;
$(\bullet  )$	current element right of $  $ is above $s_x$ (upper period boundary); increment element’s period, increase its coordinate by $s_x$ and move $  $ left;
$(\#)$	inversion across $  $ must subtract $s_x$ from the left coordinate during comparison. If the elements are not in order, they are swapped, but they must have their periods changed as they traverse $  $ . Apply $(  \bullet)$ if necessary;
$(  \circ)$	if after $(\#)$ the element that is now right of $  $ has $x_i < s_x$ , decrease its coordinate by $s_x$ and decrement its period. Do not move $  $ .

In the first step,  $(||\bullet)$  is applied, and inversion with  $12_2$  happens; then we stop because of  $(\leq)$ :

$$\begin{array}{cccccc}
 4_1 & 12_2 & || & \boxed{-1_2} & -2_4 & 5_0, \\
 4_1 & 12_2 & \leftarrow & \boxed{9_1} & || & -2_4 \quad 5_0, \\
 & & & \leq & & \\
 4_1 & \leftarrow & \boxed{9_1} & 12_2 & || & -2_4 \quad 5_0. \\
 & \leq & & & & 
 \end{array}$$

We move to next element  $\boxed{-2_4}$ ; first, we apply ( $\|\bullet$ ), then invert until ( $\leq$ ):

$$\begin{array}{ccccccc}
 4_1 & 9_1 & 12_2 & \parallel & \boxed{-2_4} & 5_0, \\
 4_1 & 9_1 & 12_2 & \leftarrow & \boxed{8_3} & \parallel & 5_0, \\
 & & & \leq & & & \\
 4_1 & 9_1 & \leftarrow & \boxed{8_3} & 12_2 & \parallel & 5_0, \\
 & & & \leq & & & \\
 4_1 & \leftarrow & \boxed{8_3} & 9_1 & 12_2 & \parallel & 5_0. \\
 & & & \leq & & & 
 \end{array}$$

The next element is  $\boxed{5_0}$ ; we satisfy ( $\#$ ), therefore instead of comparing  $12_2 > 5_0$ , we must do  $(12_2 - s_x) = 2_3 \leq 5$ ; we adjust periods when swapping over  $\parallel$  and apply ( $\|\circ$ ), turning  $12_2$  into  $2_3$ ; then we keep inverting, until ( $\leq$ ):

$$\begin{array}{ccccccc}
 4_1 & 8_3 & 9_1 & 12_2 & \parallel & \boxed{5_0}, \\
 & & & \leftarrow & \leq & & \\
 4_1 & 8_3 & 9_1 & \leftarrow & \boxed{5_{-1}} & \parallel & 2_3, \\
 & & & \leq & & & \\
 4_1 & 8_3 & \leftarrow & \boxed{5_{-1}} & 9_1 & \parallel & 2_3, \\
 & & & \leq & & & \\
 4_1 & \leftarrow & \boxed{5_{-1}} & 8_3 & 9_1 & \parallel & 2_3. \\
 & & & \leq & & & 
 \end{array}$$

We move (wrapping around) to  $\boxed{4_1}$ , which is ordered:

$$\boxed{4_1} \quad 5_{-1} \quad 8_3 \quad 9_1 \quad \parallel \quad 2_3$$

$\curvearrowright \geq$

and so is the last element

$$4_1 \leftarrow \boxed{5_{-1}} \quad 8_3 \quad 9_1 \quad \parallel \quad 2_3.$$

$\leq$

## 1.7 Computational aspects

### 1.7.1 Cost

The DEM computation using an explicit integration scheme demands a relatively high number of steps during simulation, compared to implicit schemes. The total computation time  $Z$  of simulation spanning  $T$  seconds (of simulated time), containing  $N$  particles in volume  $V$  depends on:

- linearly, the number of steps  $i = T/(s_t \Delta t_{cr})$ , where  $s_t$  is timestep safety factor;  $\Delta t_{cr}$  can be estimated by p-wave velocity using  $E$  and  $\rho$  (sect. *Estimation of by wave propagation speed*) as  $\Delta t_{cr}^{(p)} = r \sqrt{\frac{\rho}{E}}$ . Therefore

$$i = \frac{T}{s_t r} \sqrt{\frac{E}{\rho}}.$$

- the number of particles  $N$ ; for fixed value of simulated domain volume  $V$  and particle radius  $r$

$$N = p \frac{V}{\frac{4}{3} \pi r^3},$$

where  $p$  is packing porosity, roughly  $\frac{1}{2}$  for dense irregular packings of spheres of similar radius.

The dependency is not strictly linear (which would be the best case), as some algorithms do not scale linearly; a case in point is the sweep and prune collision detection algorithm introduced in sect. *Sweep and prune*, with scaling roughly  $\mathcal{O}(N \log N)$ .

The number of interactions scales with  $N$ , as long as packing characteristics are the same.

- the number of computational cores  $n_{\text{cpu}}$ ; in the ideal case, the dependency would be inverse-linear were all algorithms parallelized (in Yade, collision detection is not).

Let us suppose linear scaling. Additionally, let us suppose that the material to be simulated ( $E, \rho$ ) and the simulation setup ( $V, T$ ) are given in advance. Finally, dimensionless constants  $s_t, p$  and  $n_{\text{cpu}}$  will have a fixed value. This leaves us with one last degree of freedom,  $r$ . We may write

$$Z \propto iN \frac{1}{n_{\text{cpu}}} = \frac{T}{s_t r} \sqrt{\frac{E}{\rho}} p \frac{V}{\frac{4}{3}\pi r^3} \frac{1}{n_{\text{cpu}}} \propto \frac{1}{r} \frac{1}{r^3} = \frac{1}{r^4}.$$

This (rather trivial) result is essential to realize DEM scaling; if we want to have finer results, refining the “mesh” by halving  $r$ , the computation time will grow  $2^4 = 16$  times.

For very crude estimates, one can use a known simulation to obtain a machine “constant”

$$\mu = \frac{Z}{Ni}$$

with the meaning of time per particle and per timestep (in the order of  $10^{-6}$  s for current machines).  $\mu$  will be only useful if simulation characteristics are similar and non-linearities in scaling do not have major influence, i.e.  $N$  should be in the same order of magnitude as in the reference case.

### 1.7.2 Result indeterminism

It is naturally expected that running the same simulation several times will give exactly the same results: although the computation is done with finite precision, round-off errors would be deterministically the same at every run. While this is true for *single-threaded* computation where exact order of all operations is given by the simulation itself, it is not true anymore in *multi-threaded* computation which is described in detail in later sections.

The straight-forward manner of parallel processing in explicit DEM is given by the possibility of treating interactions in arbitrary order. Strain and stress is evaluated for each interaction independently, but forces from interactions have to be summed up. If summation order is also arbitrary (in Yade, forces are accumulated for each thread in the order interactions are processed, then summed together), then the results can be slightly different. For instance

```
(1/10.)+(1/13.)+(1/17.)=0.23574660633484162
(1/17.)+(1/13.)+(1/10.)=0.23574660633484165
```

As forces generated by interactions are assigned to bodies in quasi-random order, summary force  $F_i$  on the body can be different between single-threaded and multi-threaded computations, but also between different runs of multi-threaded computation with exactly the same parameters. Exact thread scheduling by the kernel is not predictable since it depends on asynchronous events (hardware interrupts) and other unrelated tasks running on the system; and it is thread scheduling that ultimately determines summation order of force contributions from interactions.

### Numerical damping influence

The effect of summation order can be significantly amplified by the usage of a *discontinuous* damping function in NewtonIntegrator given in (1.9) as

$$\frac{(\Delta F)_{dw}}{F_w} = -\lambda_d \operatorname{sgn} F_w \left( \dot{u}_w^\ominus + \frac{\ddot{u}_w^\circ \Delta t}{2} \right).$$

If the  $\operatorname{sgn}$  argument is close to zero then the least significant finite precision artifact can determine whether the equation (relative increment of  $F_w$ ) is  $+\lambda_d$  or  $-\lambda_d$ . Given commonly used values of  $\lambda_d = 0.4$ , it means that such artifact propagates from least significant place to the most significant one at once.

# Bibliography

- [yade:background] V. Šmilauer, B. Chareyre (2010), (Yade dem formulation). In *Yade Documentation* ( V. Šmilauer, ed.), The Yade Project , 1st ed. (fulltext) (<http://yade-dem.org/doc/formulation.html>)
- [yade:doc] V. Šmilauer, E. Catalano, B. Chareyre, S. Dorofeenko, J. Duriez, A. Gladky, J. Kozicki, C. Modenese, L. Scholtès, L. Sibille, J. Stránský, K. Thoeni (2010), (Yade Documentation). The Yade Project. (<http://yade-dem.org/doc/>)
- [yade>manual] V. Šmilauer, A. Gladky, J. Kozicki, C. Modenese, J. Stránský (2010), (Yade, using and programming). In *Yade Documentation* ( V. Šmilauer, ed.), The Yade Project , 1st ed. (fulltext) (<http://yade-dem.org/doc/>)
- [yade:reference] V. Šmilauer, E. Catalano, B. Chareyre, S. Dorofeenko, J. Duriez, A. Gladky, J. Kozicki, C. Modenese, L. Scholtès, L. Sibille, J. Stránský, K. Thoeni (2010), (Yade Reference Documentation). In *Yade Documentation* ( V. Šmilauer, ed.), The Yade Project , 1st ed. (fulltext) (<http://yade-dem.org/doc/>)
- [Bance2014] Bance, S., Fischbacher, J., Schrefl, T., Zins, I., Rieger, G., Cassignol, C. (2014), (Micromagnetics of shape anisotropy based permanent magnets). *Journal of Magnetism and Magnetic Materials* (363), pages 121–124.
- [Bonilla2015] Bonilla-Sierra, V., Scholtès, L., Donzé, F.V., Elmoultie, M.K. (2015), (Rock slope stability analysis using photogrammetric data and dfn–dem modelling). *Acta Geotechnica*, pages 1-15. DOI 10.1007/s11440-015-0374-z (fulltext)
- [Boon2012a] Boon, C.W., Houlsby, G.T., Utili, S. (2012), (A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method). *Computers and Geotechnics* (44), pages 73 - 82. DOI 10.1016/j.comgeo.2012.03.012 (fulltext)
- [Boon2012b] Boon, C.W., Houlsby, G.T., Utili, S. (2013), (A new contact detection algorithm for three-dimensional non-spherical particles). *Powder Technology*. DOI 10.1016/j.powtec.2012.12.040 (fulltext)
- [Boon2014] Boon, C.W., Houlsby, G.T., Utili, S. (2014), (New insights into the 1963 vajont slide using 2d and 3d distinct-element method analyses). *Géotechnique* (64), pages 800–816.
- [Boon2015] Boon, C.W., Houlsby, G.T., Utili, S. (2015), (A new rock slicing method based on linear programming). *Computers and Geotechnics* (65), pages 12–29.
- [Bourrier2013] Bourrier, F., Kneib, F., Chareyre, B., Fourcaud, T. (2013), (Discrete modeling of granular soils reinforcement by plant roots). *Ecological Engineering*. DOI 10.1016/j.ecoleng.2013.05.002 (fulltext)
- [Bourrier2015] Bourrier, F., Lambert, S., Baroth, J. (2015), (A reliability-based approach for the design of rockfall protection fences). *Rock Mechanics and Rock Engineering* (48), pages 247–259.
- [Catalano2014a] Catalano, E., Chareyre, B., Barthélémy, E. (2014), (Pore-scale modeling of fluid-particles interaction and emerging poromechanical effects). *International Journal for Numerical and Analytical Methods in Geomechanics* (38), pages 51–71. DOI 10.1002/nag.2198 (fulltext) (<http://arxiv.org/pdf/1304.4895.pdf>)
- [Chareyre2012a] Chareyre, B., Cortis, A., Catalano, E., Barthélemy, E. (2012), (Pore-scale modeling of viscous flow and induced forces in dense sphere packings). *Transport in Porous Media* (92), pages 473-493. DOI 10.1007/s11242-011-9915-6 (fulltext)

- [Chen2007] Chen, F., Drumm, E. C., Guiochon, G. (2007), (Prediction/verification of particle motion in one dimension with the discrete-element method). *International Journal of Geomechanics, ASCE* (7), pages 344–352. DOI [10.1061/\(ASCE\)1532-3641\(2007\)7:5\(344\)](https://doi.org/10.1061/(ASCE)1532-3641(2007)7:5(344))
- [Chen2011a] Chen, F., Drumm, E., Guiochon G. (2011), (Coupled discrete element and finite volume solution of two classical soil mechanics problems). *Computers and Geotechnics*. DOI [10.1016/j.compgeo.2011.03.009](https://doi.org/10.1016/j.compgeo.2011.03.009) (fulltext)
- [Chen2012] Chen, Jingsong, Huang, Baoshan, Chen, Feng, Shu, Xiang (2012), (Application of discrete element method to superpave gyratory compaction). *Road Materials and Pavement Design* (13), pages 480-500. DOI [10.1080/14680629.2012.694160](https://doi.org/10.1080/14680629.2012.694160) (fulltext)
- [Chen2014] Chen, J., Huang, B., Shu, X., Hu, C. (2014), (Dem simulation of laboratory compaction of asphalt mixtures using an open source code). *Journal of Materials in Civil Engineering*.
- [Dang2010a] Dang, H. K., Meguid, M. A. (2010), (Algorithm to generate a discrete element specimen with predefined properties). *International Journal of Geomechanics* (10), pages 85-91. DOI [10.1061/\(ASCE\)GM.1943-5622.0000028](https://doi.org/10.1061/(ASCE)GM.1943-5622.0000028)
- [Dang2010b] Dang, H. K., Meguid, M. A. (2010), (Evaluating the performance of an explicit dynamic relaxation technique in analyzing non-linear geotechnical engineering problems). *Computers and Geotechnics* (37), pages 125 - 131. DOI [10.1016/j.compgeo.2009.08.004](https://doi.org/10.1016/j.compgeo.2009.08.004)
- [Donze2008] Donzé, F.V. (2008), (Impacts on cohesive frictional geomaterials). *European Journal of Environmental and Civil Engineering* (12), pages 967–985.
- [Duriez2011] Duriez, J., Darve, F., Donzé, F.V. (2011), (A discrete modeling-based constitutive relation for infilled rock joints). *International Journal of Rock Mechanics & Mining Sciences* (48), pages 458–468. DOI [10.1016/j.ijrmms.2010.09.008](https://doi.org/10.1016/j.ijrmms.2010.09.008)
- [Duriez2013] Duriez, J., Darve, F., Donzé, F.V. (2013), (Incrementally non-linear plasticity applied to rock joint modelling). *International Journal for Numerical and Analytical Methods in Geomechanics* (37), pages 453–477. DOI [10.1002/nag.1105](https://doi.org/10.1002/nag.1105) (fulltext)
- [Dyck2015] Dyck, N. J., Straatman, A.G. (2015), (A new approach to digital generation of spherical void phase porous media microstructures). *International Journal of Heat and Mass Transfer* (81), pages 470–477.
- [Elias2014] Jan Elias (2014), (Simulation of railway ballast using crushable polyhedral particles). *Powder Technology* (264), pages 458–465. DOI [10.1016/j.powtec.2014.05.052](https://doi.org/10.1016/j.powtec.2014.05.052)
- [Epifancev2013] Epifancev, K., Nikulin, A., Kovshov, S., Mozer, S., Brigadnov, I. (2013), (Modeling of peat mass process formation based on 3d analysis of the screw machine by the code yade). *American Journal of Mechanical Engineering* (1), pages 73–75. DOI [10.12691/ajme-1-3-3](https://doi.org/10.12691/ajme-1-3-3) (fulltext)
- [Epifantsev2012] Epifantsev, K., Mikhailov, A., Gladky, A. (2012), (Proizvodstvo kuskovogo torfa, ekstrudirovanie, forma zakhodnoi i kalibriruyushchei chasti fil'ery matritsy, metod diskretnykh elementov [rus]). *Mining informational and analytical bulletin (scientific and technical journal)*, pages 212-219.
- [Favier2009a] Favier, L., Daudon, D., Donzé, F.V., Mazars, J. (2009), (Predicting the drag coefficient of a granular flow using the discrete element method). *Journal of Statistical Mechanics: Theory and Experiment* (2009), pages P06012.
- [Favier2012] Favier, L., Daudon, D., Donzé, F.V. (2012), (Rigid obstacle impacted by a supercritical cohesive granular flow using a 3d discrete element model). *Cold Regions Science and Technology* (85), pages 232–241. (fulltext)
- [Gladky2014] Gladky, Anton, Schwarze, Rüdiger (2014), (Comparison of different capillary bridge models for application in the discrete element method). *Granular Matter*, pages 1-10. DOI [10.1007/s10035-014-0527-z](https://doi.org/10.1007/s10035-014-0527-z) (fulltext)
- [Grujicic2013] Grujicic, M, Snipes, JS, Ramaswami, S, Yavari, R (2013), (Discrete element modeling and analysis of structural collapse/survivability of a building subjected to improvised explosive device (ied) attack). *Advances in Materials Science and Applications* (2), pages 9–24.

- [Guo2014] Guo, Ning, Zhao, Jidong (2014), (A coupled fem/dem approach for hierarchical multiscale modelling of granular media). *International Journal for Numerical Methods in Engineering* (99), pages 789–818. DOI [10.1002/nme.4702](https://doi.org/10.1002/nme.4702) (fulltext)
- [Guo2015] N. Guo, J. Zhao (2015), (Multiscale insights into classical geomechanics problems). *International Journal for Numerical and Analytical Methods in Geomechanics*. (under review)
- [Gusenbauer2012] Gusenbauer, M., Kovacs, A., Reichel, F., Exl, L., Bance, S., Özelt, H., Schrefl, T. (2012), (Self-organizing magnetic beads for biomedical applications). *Journal of Magnetism and Magnetic Materials* (324), pages 977–982.
- [Gusenbauer2014] Gusenbauer, M., Nguyen, H., Reichel, F., Exl, L., Bance, S., Fischbacher, J., Özelt, H., Kovacs, A., Brandl, M., Schrefl, T. (2014), (Guided self-assembly of magnetic beads for biomedical applications). *Physica B: Condensed Matter* (435), pages 21–24.
- [Hadda2013] Hadda, Nejib, Nicot, François, Bourrier, Franck, Sibille, Luc, Radjai, Farhang, Darve, Félix (2013), (Micromechanical analysis of second order work in granular media). *Granular Matter* (15), pages 221–235. DOI [10.1007/s10035-013-0402-3](https://doi.org/10.1007/s10035-013-0402-3) (fulltext)
- [Hadda2015] Hadda, N., Nicot, F., Wan, R., Darve, F. (2015), (Microstructural self-organization in granular materials during failure). *Comptes Rendus Mécanique*.
- [Harthong2009] Harthong, B., Jerier, J.F., Doremus, P., Imbault, D., Donzé, F.V. (2009), (Modeling of high-density compaction of granular materials by the discrete element method). *International Journal of Solids and Structures* (46), pages 3357–3364. DOI [10.1016/j.ijsolstr.2009.05.008](https://doi.org/10.1016/j.ijsolstr.2009.05.008)
- [Harthong2012b] Harthong, B., Jerier, J.-F., Richefeu, V., Chareyre, B., Doremus, P., Imbault, D., Donzé, F.V. (2012), (Contact impingement in packings of elastic-plastic spheres, application to powder compaction). *International Journal of Mechanical Sciences* (61), pages 32–43.
- [Harthong2012a] Harthong, B., Scholtès, L., Donzé, F.-V. (2012), (Strength characterization of rock masses, using a coupled dem-dfn model). *Geophysical Journal International* (191), pages 467–480. DOI [10.1111/j.1365-246X.2012.05642.x](https://doi.org/10.1111/j.1365-246X.2012.05642.x) (fulltext)
- [Hassan2010] Hassan, A., Chareyre, B., Darve, F., Meyssonier, J., Flin, F. (2010 (submitted)), (Microtomography-based discrete element modelling of creep in snow). *Granular Matter*.
- [Hilton2013] Hilton, J. E., Tordesillas, A. (2013), (Drag force on a spherical intruder in a granular bed at low froude number). *Phys. Rev. E* (88), pages 062203. DOI [10.1103/PhysRevE.88.062203](https://doi.org/10.1103/PhysRevE.88.062203) (fulltext)
- [Jerier2009] Jerier, J.-F., Imbault, D. and Donzé, F.V., Doremus, P. (2009), (A geometric algorithm based on tetrahedral meshes to generate a dense polydisperse sphere packing). *Granular Matter* (11). DOI [10.1007/s10035-008-0116-0](https://doi.org/10.1007/s10035-008-0116-0)
- [Jerier2010a] Jerier, J.-F., Richefeu, V., Imbault, D., Donzé, F.V. (2010), (Packing spherical discrete elements for large scale simulations). *Computer Methods in Applied Mechanics and Engineering*. DOI [10.1016/j.cma.2010.01.016](https://doi.org/10.1016/j.cma.2010.01.016)
- [Jerier2010b] Jerier, J.-F., Harthong, B., Richefeu, V., Chareyre, B., Imbault, D., Donzé, F.-V., Doremus, P. (2010), (Study of cold powder compaction by using the discrete element method). *Powder Technology* (In Press). DOI [10.1016/j.powtec.2010.08.056](https://doi.org/10.1016/j.powtec.2010.08.056)
- [Kozicki2006a] Kozicki, J., Tejchman, J. (2006), (2D lattice model for fracture in brittle materials). *Archives of Hydro-Engineering and Environmental Mechanics* (53), pages 71–88. (fulltext)
- [Kozicki2007a] Kozicki, J., Tejchman, J. (2007), (Effect of aggregate structure on fracture process in concrete using 2d lattice model). *Archives of Mechanics* (59), pages 365–384. (fulltext)
- [Kozicki2008] Kozicki, J., Donzé, F.V. (2008), (A new open-source software developed for numerical simulations using discrete modeling methods). *Computer Methods in Applied Mechanics and Engineering* (197), pages 4429–4443. DOI [10.1016/j.cma.2008.05.023](https://doi.org/10.1016/j.cma.2008.05.023) (fulltext)
- [Kozicki2009] Kozicki, J., Donzé, F.V. (2009), (Yade-open dem: an open-source software using a discrete element method to simulate granular material). *Engineering Computations* (26), pages 786–805. DOI [10.1108/02644400910985170](https://doi.org/10.1108/02644400910985170) (fulltext)
- [Lomine2013] Lominé, F., Scholtès, L., Sibille, L., Poullain, P. (2013), (Modelling of fluid-solid interaction in granular media with coupled lb/de methods: application to piping erosion). *Internation*

- tional Journal for Numerical and Analytical Methods in Geomechanics* (37), pages 577-596. DOI [10.1002/nag.1109](https://doi.org/10.1002/nag.1109)
- [Nicot2011] Nicot, F., Hadda, N., Bourrier, F., Sibille, L., Darve, F. (2011), (Failure mechanisms in granular media: a discrete element analysis). *Granular Matter* (13), pages 255-260. DOI [10.1007/s10035-010-0242-3](https://doi.org/10.1007/s10035-010-0242-3)
- [Nicot2012] Nicot, F., Sibille, L., Darve, F. (2012), (Failure in rate-independent granular materials as a bifurcation toward a dynamic regime). *International Journal of Plasticity* (29), pages 136-154. DOI [10.1016/j.ijplas.2011.08.002](https://doi.org/10.1016/j.ijplas.2011.08.002)
- [Nicot2013a] Nicot, F., Hadda, N., Darve, F. (2013), (Second-order work analysis for granular materials using a multiscale approach). *International Journal for Numerical and Analytical Methods in Geomechanics*. DOI [10.1002/nag.2175](https://doi.org/10.1002/nag.2175)
- [Nicot2013b] Nicot, F., Hadda, N., Guessasma, M., Fortin, J., Millet, O. (2013), (On the definition of the stress tensor in granular media). *International Journal of Solids and Structures*. DOI [10.1016/j.ijsolstr.2013.04.001](https://doi.org/10.1016/j.ijsolstr.2013.04.001) (fulltext)
- [Nitka2015] Nitka, M., Tejchman, J. (2015), (Modelling of concrete behaviour in uniaxial compression and tension with dem). *Granular Matter*, pages 1–20.
- [Puckett2011] Puckett, J.G., Lechenault, F., Daniels, K.E. (2011), (Local origins of volume fraction fluctuations in dense granular materials). *Physical Review E* (83), pages 041301. DOI [10.1103/PhysRevE.83.041301](https://doi.org/10.1103/PhysRevE.83.041301) (fulltext)
- [Sayeed2011] Sayeed, M.A., Suzuki, K., Rahman, M.M., Mohamad, W.H.W., Razlan, M.A., Ahmad, Z., Thumrongvut, J., Seangatith, S., Sobhan, MA, Mofiz, SA, others (2011), (Strength and deformation characteristics of granular materials under extremely low to high confining pressures in triaxial compression). *International Journal of Civil & Environmental Engineering IJCEE-IJENS* (11).
- [Scholtes2009a] Scholtès, L., Chareyre, B., Nicot, F., Darve, F. (2009), (Micromechanics of granular materials with capillary effects). *International Journal of Engineering Science* (47), pages 64–75. DOI [10.1016/j.ijengsci.2008.07.002](https://doi.org/10.1016/j.ijengsci.2008.07.002) (fulltext)
- [Scholtes2009b] Scholtès, L., Hicher, P.-Y., Chareyre, B., Nicot, F., Darve, F. (2009), (On the capillary stress tensor in wet granular materials). *International Journal for Numerical and Analytical Methods in Geomechanics* (33), pages 1289–1313. DOI [10.1002/nag.767](https://doi.org/10.1002/nag.767) (fulltext)
- [Scholtes2009c] Scholtès, L., Chareyre, B., Nicot, F., Darve, F. (2009), (Discrete modelling of capillary mechanisms in multi-phase granular media). *Computer Modeling in Engineering and Sciences* (52), pages 297–318. (fulltext)
- [Scholtes2010] Scholtès, L., Hicher, P.-Y., Sibille, L. (2010), (Multiscale approaches to describe mechanical responses induced by particle removal in granular materials). *Comptes Rendus Mécanique* (338), pages 627–638. DOI [10.1016/j.crme.2010.10.003](https://doi.org/10.1016/j.crme.2010.10.003) (fulltext)
- [Scholtes2011] Scholtès, L., Donzé, F.V., Khanal, M. (2011), (Scale effects on strength of geomaterials, case study: coal). *Journal of the Mechanics and Physics of Solids* (59), pages 1131–1146. DOI [10.1016/j.jmps.2011.01.009](https://doi.org/10.1016/j.jmps.2011.01.009) (fulltext)
- [Scholtes2012] Scholtès, L., Donzé, F.V. (2012), (Modelling progressive failure in fractured rock masses using a 3d discrete element method). *International Journal of Rock Mechanics and Mining Sciences* (52), pages 18–30. DOI [10.1016/j.ijrmms.2012.02.009](https://doi.org/10.1016/j.ijrmms.2012.02.009) (fulltext)
- [Scholtes2013] Scholtès, L., Donzé, F.V. (2013), (A DEM model for soft and hard rocks: role of grain interlocking on strength). *Journal of the Mechanics and Physics of Solids* (61), pages 352–369. DOI [10.1016/j.jmps.2012.10.005](https://doi.org/10.1016/j.jmps.2012.10.005) (fulltext)
- [Scholtes2015a] Scholtès, L., Chareyre, B., Michallet, H., Catalano, E., Marzougui, D. (2015), (Modeling wave-induced pore pressure and effective stress in a granular seabed). *Continuum Mechanics and Thermodynamics* (27), pages 305–323. DOI <http://dx.doi.org/10.1007/s00161-014-0377-2>
- [Scholtes2015b] Scholtès, L., Donzé, F., V. (2015), (A dem analysis of step-path failure in jointed rock slopes). *Comptes rendus - Mécanique* (343), pages 155–165. DOI <http://dx.doi.org/10.1016/j.crme.2014.11.002>

- [Shiu2008] Shiu, W., Donzé, F.V., Daudeville, L. (2008), (Compaction process in concrete during missile impact: a dem analysis). *Computers and Concrete* (5), pages 329–342.
- [Shiu2009] Shiu, W., Donzé, F.V., Daudeville, L. (2009), (Discrete element modelling of missile impacts on a reinforced concrete target). *International Journal of Computer Applications in Technology* (34), pages 33–41.
- [Sibille2014] Sibille, L., Lominé, F., Poullain, P., Sail, Y., Marot, D. (2014), (Internal erosion in granular media: direct numerical simulations and energy interpretation). *Hydrological Processes*. DOI 10.1002/hyp.10351 (fulltext) (First published online Oct. 2014)
- [Sibille2015] Sibille, L., Hadda, N., Nicot, F., Tordesillas, A., Darve, F. (2015), (Granular plasticity, a contribution from discrete mechanics). *Journal of the Mechanics and Physics of Solids* (75), pages 119–139. DOI 10.1016/j.jmps.2014.09.010 (fulltext)
- [Smilauer2006] Václav Šmilauer (2006), (The splendors and miseries of yade design). *Annual Report of Discrete Element Group for Hazard Mitigation*. (fulltext)
- [Thoeni2013] K. Thoeni, C. Lambert, A. Giacomini, S.W. Sloan (2013), (Discrete modelling of hexagonal wire meshes with a stochastically distorted contact model). *Computers and Geotechnics* (49), pages 158–169. DOI 10.1016/j.compgeo.2012.10.014 (fulltext)
- [Thoeni2014] K. Thoeni, A. Giacomini, C. Lambert, S.W. Sloan, J.P. Carter (2014), (A 3D discrete element modelling approach for rockfall analysis with drapery systems). *International Journal of Rock Mechanics and Mining Sciences* (68), pages 107–119. DOI 10.1016/j.ijrmms.2014.02.008 (fulltext)
- [Tong2012] Tong, A.-T., Catalano, E., Chareyre, B. (2012), (Pore-scale flow simulations: model predictions compared with experiments on bi-dispersed granular assemblies). *Oil & Gas Science and Technology - Rev. IFP Energies nouvelles*. DOI 10.2516/ogst/2012032 (fulltext)
- [Tran2011] Tran, V.T., Donzé, F.V., Marin, P. (2011), (A discrete element model of concrete under high triaxial loading). *Cement and Concrete Composites*.
- [Tran2012] Tran, V.D.H., Meguid, M.A., Chouinard, L.E. (2012), (An algorithm for the propagation of uncertainty in soils using the discrete element method). *The Electronic Journal of Geotechnical Engineering*. (fulltext)
- [Tran2012c] Tran, V.D.H., Meguid, M.A., Chouinard, L.E. (2012), (Discrete element and experimental investigations of the earth pressure distribution on cylindrical shafts). *International Journal of Geomechanics*. DOI 10.1061/(ASCE)GM.1943-5622.0000277
- [Tran2013] Tran, V.D.H., Meguid, M.A., Chouinard, L.E. (2013), (A finite–discrete element framework for the 3d modeling of geogrid–soil interaction under pullout loading conditions). *Geotextiles and Geomembranes* (37), pages 1-9. DOI 10.1016/j.geotexmem.2013.01.003
- [Tran2014] Tran, V.D.H., Meguid, M.A., Chouinard, L.E. (2014), (Three-dimensional analysis of geogrid-reinforced soil using a finite-discrete element framework). *International Journal of Geomechanics*.
- [Wan2014] Wan, R., Khosravani, S., Pouragha, M. (2014), (Micromechanical analysis of force transport in wet granular soils). *Vadose Zone Journal* (13).
- [Wang2014] Wang, XiaoLiang, Li, JiaChun (2014), (Simulation of triaxial response of granular materials by modified dem). *Science China Physics, Mechanics & Astronomy* (57), pages 2297–2308.
- [Zhao2015] J. Zhao, N. Guo (2015), (The interplay between anisotropy and strain localisation in granular soils: a multiscale insight). *Géotechnique*. (under review)
- [kozicki2014] Kozicki, Jan, Tejchman, Jacek, Mühlhaus, Hans-Bernd (2014), (Discrete simulations of a triaxial compression test for sand by dem). *International Journal for Numerical and Analytical Methods in Geomechanics* (38), pages 1923–1952.
- [Catalano2008a] E. Catalano (2008), (Infiltration effects on a partially saturated slope - an application of the discrete element method and its implementation in the open-source software yade). Master thesis at *UJF-Grenoble*. (fulltext)
- [Catalano2012] Emanuele Catalano (2012), (A pore-scale coupled hydromechanical model for biphasic granular media). PhD thesis at *Université de Grenoble*. (fulltext)

- [Charlas2013] Benoit Charlas (2013), (Etude du comportement mécanique d'un hydrure intermétallique utilisé pour le stockage d'hydrogène). PhD thesis at *Université de Grenoble*. ([fulltext](#))
- [Chen2009a] Chen, F. (2009), (Coupled flow discrete element method application in granular porous media using open source codes). PhD thesis at *University of Tennessee, Knoxville*. ([fulltext](#))
- [Chen2011b] Chen, J. (2011), (Discrete element method (dem) analyses for hot-mix asphalt (hma) mixture compaction). PhD thesis at *University of Tennessee, Knoxville*. ([fulltext](#))
- [Duriez2009a] J. Duriez (2009), (Stabilité des massifs rocheux : une approche mécanique). PhD thesis at *Institut polytechnique de Grenoble*. ([fulltext](#))
- [Favier2009c] Favier, L. (2009), (Approche numérique par éléments discrets 3d de la sollicitation d'un écoulement granulaire sur un obstacle). PhD thesis at *Université Grenoble I – Joseph Fourier*.
- [Guo2014c] N. Guo (2014), (Multiscale characterization of the shear behavior of granular media). PhD thesis at *The Hong Kong University of Science and Technology*.
- [Jerier2009b] Jerier, J.F. (2009), (Modélisation de la compression haute densité des poudres métalliques ductiles par la méthode des éléments discrets (in french)). PhD thesis at *Université Grenoble I – Joseph Fourier*. ([fulltext](#))
- [Kozicki2007b] J. Kozicki (2007), (Application of discrete models to describe the fracture process in brittle materials). PhD thesis at *Gdansk University of Technology*. ([fulltext](#))
- [Marzougui2011] Marzougui, D. (2011), (Hydromechanical modeling of the transport and deformation in bed load sediment with discrete elements and finite volume). Master thesis at *Ecole Nationale d'Ingénieur de Tunis*. ([fulltext](#))
- [Scholtes2009d] Luc Scholtès (2009), (modélisation micromécanique des milieux granulaires partiellement saturés). PhD thesis at *Institut National Polytechnique de Grenoble*. ([fulltext](#))
- [Smilauer2010b] Václav Šmilauer (2010), (Cohesive particle model using the discrete element method on the yade platform). PhD thesis at *Czech Technical University in Prague, Faculty of Civil Engineering & Université Grenoble I – Joseph Fourier, École doctorale I-MEP2*. ([fulltext](#)) ([LaTeX sources](#))
- [Smilauer2010c] Václav Šmilauer (2010), (Doctoral thesis statement). (*PhD thesis summary*). ([fulltext](#)) ([LaTeX sources](#))
- [Tran2011b] Van Tieng TRAN (2011), (Structures en béton soumises à des chargements mécaniques extrêmes: modélisation de la réponse locale par la méthode des éléments discrets (in french)). PhD thesis at *Université Grenoble I – Joseph Fourier*. ([fulltext](#))
- [Addetta2001] G.A. D'Addetta, F. Kun, E. Ramm, H.J. Herrmann (2001), (From solids to granulates - Discrete element simulations of fracture and fragmentation processes in geomaterials.). In *Continuous and Discontinuous Modelling of Cohesive-Frictional Materials*. ([fulltext](#))
- [Allen1989] M. P. Allen, D. J. Tildesley (1989), (Computer simulation of liquids). Clarendon Press.
- [Alonso2004] F. Alonso-Marroquin, R. Garcia-Rojo, H.J. Herrmann (2004), (Micro-mechanical investigation of the granular ratcheting). In *Cyclic Behaviour of Soils and Liquefaction Phenomena*. ([fulltext](#))
- [Antypov2011] D. Antypov, J. A. Elliott (2011), (On an analytical solution for the damped hertzian spring). *EPL (Europhysics Letters)* (94), pages 50004. ([fulltext](#))
- [Bagi2006] Katalin Bagi (2006), (Analysis of microstructural strain tensors for granular assemblies). *International Journal of Solids and Structures* (43), pages 3166 - 3184. DOI 10.1016/j.ijsolstr.2005.07.016
- [Bertrand2005] D. Bertrand, F. Nicot, P. Gotteland, S. Lambert (2005), (Modelling a geo-composite cell using discrete analysis). *Computers and Geotechnics* (32), pages 564–577.
- [Bertrand2008] D. Bertrand, F. Nicot, P. Gotteland, S. Lambert (2008), (Discrete element method (dem) numerical modeling of double-twisted hexagonal mesh). *Canadian Geotechnical Journal* (45), pages 1104–1117.
- [Calvetti1997] Calvetti, F., Combe, G., Lanier, J. (1997), (Experimental micromechanical analysis of a 2d granular material: relation between structure evolution and loading path). *Mechanics of Cohesive-frictional Materials* (2), pages 121–163.

- [Camborde2000a] F. Camborde, C. Mariotti, F.V. Donzé (2000), (Numerical study of rock and concrete behaviour by discrete element modelling). *Computers and Geotechnics* (27), pages 225–247.
- [Chan2011] D. Chan, E. Klaseboer, R. Manica (2011), (Film drainage and coalescence between deformable drops and bubbles.). *Soft Matter* (7), pages 2235–2264.
- [Chareyre2002a] B. Chareyre, L. Briancon, P. Villard (2002), (Theoretical versus experimental modeling of the anchorage capacity of geotextiles in trenches.). *Geosynthet. Int.* (9), pages 97–123.
- [Chareyre2002b] B. Chareyre, P. Villard (2002), (Discrete element modeling of curved geosynthetic anchorages with known macro-properties.). In *Proc., First Int. PFC Symposium, Gelsenkirchen, Germany*.
- [Chareyre2003] Bruno Chareyre (2003), (Modélisation du comportement d’ouvrages composites sol-géosynthétique par éléments discrets - application aux tranchées d’ancrage en tête de talus.). PhD thesis at *Grenoble University*. ([fulltext](#))
- [Chareyre2005] Bruno Chareyre, Pascal Villard (2005), (Dynamic spar elements and discrete element methods in two dimensions for the modeling of soil-inclusion problems). *Journal of Engineering Mechanics* (131), pages 689–698. DOI [10.1061/\(ASCE\)0733-9399\(2005\)131:7\(689\)](https://doi.org/10.1061/(ASCE)0733-9399(2005)131:7(689)) ([fulltext](#))
- [CundallStrack1979] P.A. Cundall, O.D.L. Strack (1979), (A discrete numerical model for granular assemblies). *Geotechnique* (), pages 47–65. DOI [10.1680/geot.1979.29.1.47](https://doi.org/10.1680/geot.1979.29.1.47)
- [Dallavalle1948] J. M. DallaValle (1948), (Micrometrics : the technology of fine particles). Pitman Pub. Corp.
- [DeghmReport2006] F. V. Donzé (ed.), (Annual report 2006) (2006). *Discrete Element Group for Hazard Mitigation*. Université Joseph Fourier, Grenoble ([fulltext](#))
- [Donze1994a] F.V. Donzé, P. Mora, S.A. Magnier (1994), (Numerical simulation of faults and shear zones). *Geophys. J. Int.* (116), pages 46–52.
- [Donze1995a] F.V. Donzé, S.A. Magnier (1995), (Formulation of a three-dimensional numerical model of brittle behavior). *Geophys. J. Int.* (122), pages 790–802.
- [Donze1999a] F.V. Donzé, S.A. Magnier, L. Daudeville, C. Mariotti, L. Davenne (1999), (Study of the behavior of concrete at high strain rate compressions by a discrete element method). *ASCE J. of Eng. Mech* (125), pages 1154–1163. DOI [10.1016/S0266-352X\(00\)00013-6](https://doi.org/10.1016/S0266-352X(00)00013-6)
- [Donze2004a] F.V. Donzé, P. Bernasconi (2004), (Simulation of the blasting patterns in shaft sinking using a discrete element method). *Electronic Journal of Geotechnical Engineering* (9), pages 1–44.
- [GarciaRojo2004] R. García-Rojo, S. McNamara, H. J. Herrmann (2004), (Discrete element methods for the micro-mechanical investigation of granular ratcheting). In *Proceedings ECCOMAS 2004*. ([fulltext](#))
- [Hentz2003] Sébastien Hentz (2003), (Modélisation d’une structure en béton armé soumise à un choc par la méthode des éléments discrets). PhD thesis at *Université Grenoble 1 – Joseph Fourier*.
- [Hentz2004a] S. Hentz, F.V. Donzé, L. Daudeville (2004), (Discrete element modelling of concrete submitted to dynamic loading at high strain rates). *Computers and Structures* (82), pages 2509–2524. DOI [10.1016/j.compstruc.2004.05.016](https://doi.org/10.1016/j.compstruc.2004.05.016)
- [Hentz2004b] S. Hentz, L. Daudeville, F.V. Donzé (2004), (Identification and validation of a discrete element model for concrete). *ASCE Journal of Engineering Mechanics* (130), pages 709–719. DOI [10.1061/\(ASCE\)0733-9399\(2004\)130:6\(709\)](https://doi.org/10.1061/(ASCE)0733-9399(2004)130:6(709))
- [Hentz2005a] S. Hentz, F.V. Donzé, L. Daudeville (2005), (Discrete elements modeling of a reinforced concrete structure submitted to a rock impact). *Italian Geotechnical Journal* (XXXIX), pages 83–94.
- [Herminghaus2005] Herminghaus, S. (2005), (Dynamics of wet granular matter). *Advances in Physics* (54), pages 221–261. DOI [10.1080/00018730500167855](https://doi.org/10.1080/00018730500167855) ([fulltext](#))
- [Hubbard1996] Philip M. Hubbard (1996), (Approximating polyhedra with spheres for time-critical collision detection). *ACM Trans. Graph.* (15), pages 179–210. DOI [10.1145/231731.231732](https://doi.org/10.1145/231731.231732)
- [Ivars2011] Diego Mas Ivars, Matthew E. Pierce, Caroline Darcel, Juan Reyes-Montes, David O. Potyondy, R. Paul Young, Peter A. Cundall (2011), (The synthetic rock mass approach for jointed rock mass modelling). *International Journal of Rock Mechanics and Mining Sciences* (48), pages 219 – 244. DOI [10.1016/j.ijrmms.2010.11.014](https://doi.org/10.1016/j.ijrmms.2010.11.014)

- [Johnson2008] Scott M. Johnson, John R. Williams, Benjamin K. Cook (2008), (Quaternion-based rigid body rotation integration algorithms for use in particle methods). *International Journal for Numerical Methods in Engineering* (74), pages 1303–1313. DOI [10.1002/nme.2210](https://doi.org/10.1002/nme.2210)
- [Jung1997] Derek Jung, Kamal K. Gupta (1997), (Octree-based hierarchical distance maps for collision detection). *Journal of Robotic Systems* (14), pages 789–806. DOI [10.1002/\(SICI\)1097-4563\(199711\)14:11<789::AID-ROB3>3.0.CO;2-Q](https://doi.org/10.1002/(SICI)1097-4563(199711)14:11<789::AID-ROB3>3.0.CO;2-Q)
- [Kettner2011] Lutz Kettner, Andreas Meyer, Afra Zomorodian (2011), (Intersecting sequences of dD iso-oriented boxes). In *CGAL User and Reference Manual*. ([fulltext](#))
- [Klosowski1998] James T. Klosowski, Martin Held, Joseph S. B. Mitchell, Henry Sowizral, Karel Zikan (1998), (Efficient collision detection using bounding volume hierarchies of k-dops). *IEEE Transactions on Visualization and Computer Graphics* (4), pages 21–36. ([fulltext](#))
- [Kuhl2001] E. Kuhl, G. A. D’Addetta, M. Leukart, E. Ramm (2001), (Microplane modelling and particle modelling of cohesive-frictional materials). In *Continuous and Discontinuous Modelling of Cohesive-Frictional Materials*. DOI [10.1007/3-540-44424-6\\_3](https://doi.org/10.1007/3-540-44424-6_3) ([fulltext](#))
- [Lambert2008] Lambert, Pierre, Chau, Alexandre, Delchambre, Alain, Régnier, Stéphane (2008), (Comparison between two capillary forces models). *Langmuir* (24), pages 3157–3163.
- [Lu1998] Ya Yan Lu (1998), (Computing the logarithm of a symmetric positive definite matrix). *Appl. Numer. Math* (26), pages 483–496. DOI [10.1016/S0168-9274\(97\)00103-7](https://doi.org/10.1016/S0168-9274(97)00103-7) ([fulltext](#))
- [Lucy1977] Lucy, L.~B. (1977), (A numerical approach to the testing of the fission hypothesis). *aj* (82), pages 1013-1024. DOI [10.1086/112164](https://doi.org/10.1086/112164) ([fulltext](#))
- [Luding2008] Stefan Luding (2008), (Introduction to discrete element methods). In *European Journal of Environmental and Civil Engineering*.
- [Luding2008b] Luding, Stefan (2008), (Cohesive, frictional powders: contact models for tension). *Granular Matter* (10), pages 235-246. DOI [10.1007/s10035-008-0099-x](https://doi.org/10.1007/s10035-008-0099-x) ([fulltext](#))
- [Magnier1998a] S.A. Magnier, F.V. Donzé (1998), (Numerical simulation of impacts using a discrete element method). *Mech. Cohes.-frict. Mater.* (3), pages 257–276. DOI [10.1002/\(SICI\)1099-1484\(199807\)3:3<257::AID-CFM50>3.0.CO;2-Z](https://doi.org/10.1002/(SICI)1099-1484(199807)3:3<257::AID-CFM50>3.0.CO;2-Z)
- [Mani2013] Mani, Roman, Kadau, Dirk, Herrmann, HansJ. (2013), (Liquid migration in sheared unsaturated granular media). *Granular Matter* (15), pages 447-454. DOI [10.1007/s10035-012-0387-3](https://doi.org/10.1007/s10035-012-0387-3) ([fulltext](#))
- [McNamara2008] S. McNamara, R. García-Rojo, H. J. Herrmann (2008), (Microscopic origin of granular ratcheting). *Physical Review E* (77). DOI [11.1103/PhysRevE.77.031304](https://doi.org/10.1103/PhysRevE.77.031304)
- [Monaghan1985] Monaghan, J.~J., Lattanzio, J.~C. (1985), (A refined particle method for astrophysical problems). *aap* (149), pages 135-143. ([fulltext](#))
- [Monaghan1992] Monaghan, J.~J. (1992), (Smoothed particle hydrodynamics). *araa* (30), pages 543-574. DOI [10.1146/annurev.aa.30.090192.002551](https://doi.org/10.1146/annurev.aa.30.090192.002551)
- [Morris1997] (1997), (Modeling low reynolds number incompressible flows using {sph}). *Journal of Computational Physics* (136), pages 214 - 226. DOI <http://dx.doi.org/10.1006/jcph.1997.5776> ([fulltext](#))  
( )
- [Mueller2003] Müller, Matthias, Charypar, David, Gross, Markus (2003), (Particle-based fluid simulation for interactive applications). In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ([fulltext](#))
- [Munjiza1998] A. Munjiza, K. R. F. Andrews (1998), (Nbs contact detection algorithm for bodies of similar size). *International Journal for Numerical Methods in Engineering* (43), pages 131–149. DOI [10.1002/\(SICI\)1097-0207\(19980915\)43:1<131::AID-NME447>3.0.CO;2-S](https://doi.org/10.1002/(SICI)1097-0207(19980915)43:1<131::AID-NME447>3.0.CO;2-S)
- [Munjiza2006] A. Munjiza, E. Rougier, N. W. M. John (2006), (Mr linear contact detection algorithm). *International Journal for Numerical Methods in Engineering* (66), pages 46–71. DOI [10.1002/nme.1538](https://doi.org/10.1002/nme.1538)
- [Neto2006] Natale Neto, Luca Bellucci (2006), (A new algorithm for rigid body molecular dynamics). *Chemical Physics* (328), pages 259–268. DOI [10.1016/j.chemphys.2006.07.009](https://doi.org/10.1016/j.chemphys.2006.07.009)

- [Omelyan1999] Igor P. Omelyan (1999), (A new leapfrog integrator of rotational motion. the revised angular-momentum approach). *Molecular Simulation* (22). DOI [10.1080/08927029908022097](https://doi.org/10.1080/08927029908022097) (fulltext)
- [Pfc3dManual30] ICG (2003), (Pfc3d (particle flow code in 3d) theory and background manual, version 3.0). Itasca Consulting Group.
- [Pion2011] Sylvain Pion, Monique Teillaud (2011), (3D triangulations). In *CGAL User and Reference Manual*. (fulltext)
- [Potyondy2004] D.O. Potyondy, P.A. Cundall (2004), (A bonded-particle model for rock). *International Journal of Rock Mechanics and Mining Sciences* (41), pages 1329 - 1364. DOI [10.1016/j.ijrmms.2004.09.011](https://doi.org/10.1016/j.ijrmms.2004.09.011)
- [Pournin2001] L. Pournin, Th. M. Liebling, A. Mocellin (2001), (Molecular-dynamics force models for better control of energy dissipation in numerical simulations of dense granular media). *Phys. Rev. E* (65), pages 011302. DOI [10.1103/PhysRevE.65.011302](https://doi.org/10.1103/PhysRevE.65.011302)
- [Price2007] Mathew Price, Vasile Murariu, Garry Morrison (2007), (Sphere clump generation and trajectory comparison for real particles). In *Proceedings of Discrete Element Modelling 2007*. (fulltext)
- [Rabinov2005] RABINOVICH Yakov I., ESAYANUR Madhavan S., MOUDGIL Brij M. (2005), (Capillary forces between two spheres with a fixed volume liquid bridge : theory and experiment). *Langmuir* (21), pages 10992–10997. (fulltext) (eng)
- [Radjai2011] Radjai, F., Dubois, F. (2011), (Discrete-element modeling of granular materials). John Wiley & Sons. (fulltext)
- [RevilBaudard2013] Revil-Baudard, T., Chauchat, J. (2013), (A two-phase model for sheet flow regime based on dense granular flow rheology). *Journal of Geophysical Research: Oceans* (118), pages 619–634.
- [Richardson1954] Richardson, J. F., W. N. Zaki (1954), (Sedimentation and fluidization: part i). *Trans. Instn. Chem. Engrs* (32).
- [Satake1982] M. Satake (1982), (Fabric tensor in granular materials.). In *Proc., IUTAM Symp. on Deformation and Failure of Granular materials, Delft, The Netherlands*.
- [Schmeeckle2007] Schmeeckle, Mark W., Nelson, Jonathan M., Shreve, Ronald L. (2007), (Forces on stationary particles in near-bed turbulent flows). *Journal of Geophysical Research: Earth Surface* (112). DOI [10.1029/2006JF000536](https://doi.org/10.1029/2006JF000536) (fulltext)
- [Schwager2007] Schwager, Thomas, Pöschel, Thorsten (2007), (Coefficient of restitution and linear-dashpot model revisited). *Granular Matter* (9), pages 465–469. DOI [10.1007/s10035-007-0065-z](https://doi.org/10.1007/s10035-007-0065-z) (fulltext)
- [Soulié2006] Soulié, F., Cherblanc, F., El Youssoufi, M.S., Saix, C. (2006), (Influence of liquid bridges on the mechanical behaviour of polydisperse granular materials). *International Journal for Numerical and Analytical Methods in Geomechanics* (30), pages 213–228. DOI [10.1002/nag.476](https://doi.org/10.1002/nag.476) (fulltext)
- [Thornton1991] Colin Thornton, K. K. Yin (1991), (Impact of elastic spheres with and without adhesion). *Powder technology* (65), pages 153–166. DOI [10.1016/0032-5910\(91\)80178-L](https://doi.org/10.1016/0032-5910(91)80178-L)
- [Thornton2000] Colin Thornton (2000), (Numerical simulations of deviatoric shear deformation of granular media). *Géotechnique* (50), pages 43–53. DOI [10.1680/geot.2000.50.1.43](https://doi.org/10.1680/geot.2000.50.1.43)
- [Verlet1967] Loup Verlet (1967), (Computer “experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules). *Phys. Rev.* (159), pages 98. DOI [10.1103/PhysRev.159.98](https://doi.org/10.1103/PhysRev.159.98)
- [Villard2004a] P. Villard, B. Chareyre (2004), (Design methods for geosynthetic anchor trenches on the basis of true scale experiments and discrete element modelling). *Canadian Geotechnical Journal* (41), pages 1193–1205.
- [Wang2009] Yucang Wang (2009), (A new algorithm to model the dynamics of 3-d bonded rigid bodies with rotations). *Acta Geotechnica* (4), pages 117–127. DOI [10.1007/s11440-008-0072-1](https://doi.org/10.1007/s11440-008-0072-1) (fulltext)
- [Weigert1999] Weigert, Tom, Ripperger, Siegfried (1999), (Calculation of the liquid bridge volume and bulk saturation from the half-filling angle). *Particle & Particle Systems Characterization* (16), pages 238–242. DOI [10.1002/\(SICI\)1521-4117\(199910\)16:5<238::AID-PPSC238>3.0.CO;2-E](https://doi.org/10.1002/(SICI)1521-4117(199910)16:5<238::AID-PPSC238>3.0.CO;2-E) (fulltext)

- [Wiberg1985] Wiberg, Patricia L., Smith, J. Dungan (1985), (A theoretical model for saltating grains in water). *Journal of Geophysical Research: Oceans* (90), pages 7341–7354.
- [Willett2000] Willett, Christopher D., Adams, Michael J., Johnson, Simon A., Seville, Jonathan P. K. (2000), (Capillary bridges between two spherical bodies). *Langmuir* (16), pages 9396-9405. DOI [10.1021/la000657y](https://doi.org/10.1021/la000657y) ([fulltext](#))
- [Zhou1999536] Y.C. Zhou, B.D. Wright, R.Y. Yang, B.H. Xu, A.B. Yu (1999), (Rolling friction in the dynamic simulation of sandpile formation). *Physica A: Statistical Mechanics and its Applications* (269), pages 536–553. DOI [10.1016/S0378-4371\(99\)00183-1](https://doi.org/10.1016/S0378-4371(99)00183-1) ([fulltext](#))
- [cgal] Jean-Daniel Boissonnat, Olivier Devillers, Sylvain Pion, Monique Teillaud, Mariette Yvinec (2002), (Triangulations in cgal). *Computational Geometry: Theory and Applications* (22), pages 5–19.