A granular Discrete Element Method for arbitrary convex particle shapes
Method and packing generation

Seelen, L. J.H.; Padding, J. T.; Kuipers, J. A.M.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A granular Discrete Element Method for arbitrary convex particle shapes: method and packing generation

L.J.H. Seelen[a,*], J.T. Padding[b], J.A.M. Kuipers[a]

[a] *Multiphase Reactors Group, Department of Chemical Engineering & Chemistry, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*
[b] *Intensified Reaction & Separation Systems, Process and Energy Department, Delft University of Technology, Leeghwaterstraat 39, 2628 CB Delft, The Netherlands*

Abstract

A novel granular discrete element method (DEM) is introduced to simulate mixtures of particles of any convex shape. To quickly identify pairs of particles in contact, the method first uses a broad-phase and a narrow-phase contact detection strategy. After this, a contact resolution phase finds the contact normal and penetration depth. A new algorithm is introduced to effectively locate the contact point in the geometric center of flat faces in partial contact. This is important for a correct evaluation of the torque on each particle, leading to a much higher stability of stacks of particles than with previous algorithms. The granular DEM is used to generate random packings in a cylindrical vessel. The simulated shapes include non-spherical particles with different aspect ratio cuboids, cylinders and ellipsoids. More complex polyhedral shapes representing sand and woodchip particles are also used. The latter particles all have a unique shape and size, resembling real granular particle packings. All packings are analyzed extensively by investigating positional and orientational ordering.

Keywords: Discrete element method, Non-spherical particle, Contact detection, Packing, Solid volume fraction, Orientational ordering

Nomenclature

Variables

a half-length in x-direction, [m]

---

*Corresponding author e-mail address: L.J.H.Seelen@gmail.com

A	set of points
$\mathbf{a}$, $\mathbf{A}$	triangle vertex
b	half-length in y-direction, [m]
B	set of points
$\mathbf{b}$, $\mathbf{B}$	triangle vertex
c	half-length in z-direction, [m]
C	set of points
$\mathbf{c}$, $\mathbf{C}$	triangle vertex
$\mathbf{d}$	direction vector
e	restitution coefficient
E_{kin}	kinetic energy, [J]
E_t	tangential spring energy, [J]
$\mathbf{F}$	force, [N]
	vector field
h	height, [m]
I_1, I_2, I_3	principal inertia components, $[\mathrm{kg\,m^2}]$
$\mathbf{I}$	inertia tensor, $[\mathrm{kg\,m^2}]$
	unit tensor
k	spring stiffness, $[\mathrm{N\,m^{-1}}]$
m	mass, [kg]
	average number of broad-phase objects per cell or node
N	total number of particles or points
$\mathbf{n}$	normal
$\hat{\mathbf{n}}$	unit normal
O	order of magnitude
$\mathbf{O}$	origin, [m]
$\mathbf{p}$	point, [m]
q	rotation quaternion
r	radial coordinate, [m]
r_p	particle radius, [m]

R_{dom}	domain radius, [m]
$\mathbf{r}$	particle center of mass position vector, [m]
S_2, S_4	nematic and orthogonal order parameters
$\mathbf{s}$	support mapping, [m]
t	time, [s]
δt	time step, [s]
$\mathbf{t}$	tangent
$\hat{\mathbf{t}}$	unit tangent
$\mathbf{u}_{ij}$	relative velocity at contact point, $[\mathrm{m\,s^{-1}}]$
V	volume, $[\mathrm{m^3}]$
$\mathbf{v}$	center of mass velocity, $[\mathrm{m\,s^{-1}}]$
	vertex of polyhedron
x	Cartesian component
$\mathbf{x}$	point, [m]
y	Cartesian component
z	Cartesian component

Greek letters

α	angle, [deg]
δ	penetration depth magnitude, [m]
δ_p	point based porosity
$\boldsymbol{\delta}$	penetration depth vector, [m]
ε_p	porosity
ε_s	solid volume fraction
$\bar{\varepsilon}_s$	overall solid volume fraction
η	damping coefficient, $[\mathrm{kg\,s^{-1}}]$
μ	reduced mass, [kg]
μ_c	Coulomb friction coefficient
ξ	dimensionless distance from the wall
ρ	mass density, $[\mathrm{kg\,m^{-3}}]$

3

σ	standard deviation	
$\boldsymbol{\tau}$	torque, $[\mathrm{N\,m}]$	
$\boldsymbol{\omega}$	angular velocity, $[\mathrm{rad\,s^{-1}}]$	

Subscripts and superscripts

ann	annular
b	body frame
c	contact
	core
cm	center of mass
dom	domain
i	particle index
j	particle index
max	maximum
n	normal
nw	near-wall
p	particle
t	tangential
x	x direction
y	y direction
z	z direction
0	at the start of the relevant time period

Abbreviations

AABB	Axis Aligned Bounding Box
AR	Aspect Ratio
DEM	Discrete Element Method
EPA	Expanding Polytope Algorithm
GJK	Gilbert-Johnson-Keerthi
OBB	Oriented Bounding Box

4

RSA   Random Sequential Addition

1. Introduction

Granular packings are commonly found in industry and nature. Closer to home, our solid food is stored in the form of a granular packing. Rice grains or dehydrated pasta is stored in containers. There is a whole variety of beans that are differently shaped, again stored in a packed state. In the chemical industry, packings of particles are also common. In a packed bed reactor, pellets of different shapes are used as support material for the catalytically active materials such as metals. Common shapes for these particles are cylinders, spheres, and thin rectangular plates. In packed bed reactors, the distribution of the porous space around the particles is particularly important because it determines the pressure drop across the reactor and the severity of flow inhomogeneities which lead to suboptimal operation of the reactor.

A granular packing can be defined as a mechanically stable collection of particles. This means that the packed structure does not move under the influence of gravity. There are two important classes of granular packings, the structured packings and random packings [1]. A structured or crystalline packing has a long range repeating order. Such a packing can only be created in reality by carefully placing each particle at a specific position relative to the other particles. An example is a stack of cannonballs in a hexagonal close packing (hcp) lattice structure. With these crystalline packings it is possible to achieve high packing densities. For example the hexagonal close packing of equally sized spheres is the maximum packing density possible, with a solid volume fraction of 0.74 ([2, 3]).

Current research focuses on non-spherical shapes such as ellipsoids and spherocylinders ([4, 5, 6, 7]). Donev et al. [4] provide experiments and molecular dynamics simulations for ellipsoids with an aspect ratio ranging from 0.25 to 3.5. The closest packing density of ellipsoids can become higher than that of spheres. A maximum packing fraction is obtained for aspect ratios 0.6 and 1.5. For more extreme aspect ratios the closest packing solid volume fraction decreases again. Delaney and Cleary [5] used an expansion method to generate close packings of superellipsoids. These particles have shapes varying from spheres to rounded cuboids. The closest packing fraction as a function of the aspect ratio has the same trend for

the different shapes as obtained for ellipsoids. Wouterse et al. [6] obtained comparable results for ellipsoids with a mechanical contraction method. They also simulated spherocylinders with different aspect ratios, for which there is a single maximum in the closest packing fraction at an aspect ratio of about 1.5. The simulation methods to obtain these packings commonly use periodic boundary conditions, meaning they apply to bulk regions, far away from possible influences of walls. The advantage of using periodic boundary conditions is that accurate and efficient simulations of the random close packed state are possible, with detailed information of contact numbers and ordering. However, wall effects are very important for many practical situations.

A common computational technique to model the dynamics of granular matter is the discrete element method (DEM), introduced by Cundall and Strack[8]. This method is designed for spherical particles, as this is the most commonly studied particle shape in literature. The important part of these type of models is the contact model. Two approaches are used for spherical DEM: an event driven hard-sphere model [9] or a molecular dynamics type of time driven soft-sphere model [10]. More details on the hard and soft sphere approach for DEM can be found in [11]. Recently, a novel model for spherical DEM has been proposed by Buist et al. [12] which uses a combined hard and soft sphere approach. This hybrid model is much faster compared to classical soft-sphere approaches and also more accurate.

Unfortunately, as most particle shapes in nature and industry are non-spherical, the spherical DEM approach is not directly applicable. Therefore, there is great interest into a class of non-spherical discrete element models. The review of Lu et al. [13] details the state of the art in non-spherical DEM. A few methods should be mentioned as they are commonly used.

One of the most commonly used methods to create complex granular shapes is the *sphere assembly* approach [14, 15]. Also called the *multi-sphere* approach, this method represents a general shape with multiple spheres of different sizes. The idea is that when enough spheres are placed in the correct position, any shape can be approximated. The major advantage is that contact detection and resolution can be carried out in the same way as the spherical DEM. One of the difficulties is how to actually generate good shape approximations [16]. However good the algorithm may be, it is not possible to generate real angular granular particles, because a sphere has no sharp edges. Another difficulty is related to the energy

6

dissipation that these particles experience during multi-particle contacts ([17]).

The sphere assembly model is a logical extension to the classical spherical DEM. It allows for simple contact tests and does not require a completely new contact resolution strategy. A more advanced method is that of the super-quadrics [18, 19]. This method uses a generalization of the ellipsoid equation to generate particle shapes that range from ellipsoids to rounded boxes. Contact detection is already much more involved compared to the sphere assembly approach, and is based on the common-normal approach [20].

A third important class of methods is based on polyhedral representations of particles. Note that particles obtained from the sphere-assembly and super-quadric approaches tend to be smooth surfaced. Blocky and edged type of particles (such as common sand) are not well represented by this class. The polyhedron particle is well suited to represent particles with sharp edges and corners, such as cuboids [21, 22, 23]. The recent work of Wachs et al. [23] introduces a non-spherical DEM based on rounded polyhedra.

In this paper we will present a new way to deal with non-spherical DEM. Our approach is somewhat related to the approach of Wachs et al. [23], but our method can use truly sharp edged and cornered particles *in combination with* quadric type of particles. It is possible to mix all these different shapes, the only restriction being that all particle shapes must be *convex*, although in future work concave shapes could be created by assembly of multiple convex shapes. We provide a complete picture, including several optimization strategies and validation cases. We apply this granular DEM to the problem of generating packings in cylindrical vessels.

The organization of the paper is as follows: sections 2 to 6 detail the granular DEM method. Specifically, section 2 gives an overview of the method. Sections 3-4 deal with contact detection and section 5 with contact resolution. The contact force model is detailed in section 6. The second part of the work consists of the application of the method for random packing generation. Elongated non-spherical packings are presented in section 7 and a proof of concept for more realistic granular shapes is presented in section 8. We end with the conclusions in section 9.

2. Model overview

An overview of the new granular discrete element method (DEM) is schematically shown in figure 1. The most important blocks are the contact detection, contact resolution, force and torque calculations and numerical integration of the Newton-Euler equations. The contact detection strategies are split in a broad- and narrow-phase, to speed up the simulation. After the contact detection phase, the contact properties such as penetration normal and penetration depth must be determined. This phase also includes the determination of the location of the contact point, vital for the application of correct torques on the particles. With all contact variables known, the forces and torques can be calculated by using a specific force model. With the forces and torques it is possible to advance the simulation to the next time step by numerical integration of the equations of motion for the rigid bodies.

The next sections will detail the contact detection and resolution strategies and the force model used. In the remainder of this section we will detail the equations of motion that need to be solved and discuss the required input at the start of the granular DEM method of figure 1.

2.1. Newton-Euler equations

Since the granular DEM can simulate any *convex* particle shape, the Newton-Euler equations are used [24]:

$$\ddot{\mathbf{r}} = m^{-1}\mathbf{F}$$

$$\dot{\mathbf{r}} = \mathbf{v}$$

$$\dot{\boldsymbol{\omega}}^b = (\mathbf{I}^b)^{-1}\left(\boldsymbol{\tau}^b - \boldsymbol{\omega}^b \times \mathbf{I}^b\boldsymbol{\omega}^b\right). \tag{1}$$

In this notation we introduce a lab frame and body frame (superscript b). The lab frame variables do not have a superscript b. Each particle has a mass m, center of mass position $\mathbf{r}$, center of mass velocity $\mathbf{v}$, angular velocity $\boldsymbol{\omega}$, and experiences a force $\mathbf{F}$ and torque $\boldsymbol{\tau}$. $\mathbf{I}^b$ is the moment of inertia tensor in the body frame. The description of equation 1 additionally needs the specification of the particle orientation, which in this work is done using rotation quaternions. We employ our improved quaternion based integration scheme for rigid bodies, as described in our previous work [25]. Without going into the details, the main advantage
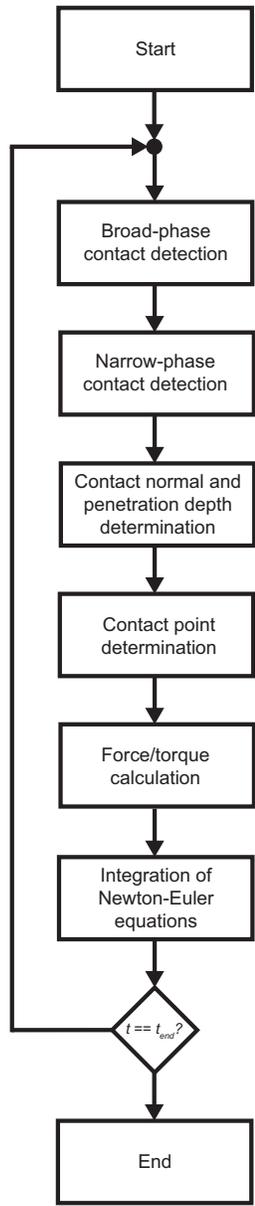
Figure 1: Flow diagram of the granular discrete element method.

is that rotations from and to the body frame are carried out directly by the quaternions. This means there is no more need for converting the quaternion to a rotation matrix and using this for the actual rotations. Our previous work [25] also showed that the method is second order accurate in time, compared to the original method as detailed by Zhao et al. [26]. This accuracy is reached with a single force/torque evaluation per time step, which is a requirement to keep the computational cost low for the granular DEM.

2.2. Particle input parameters

There are a number of parameters that are required as input for the granular DEM. These include the particle *mass* and the *inertia tensor in the principle axes*. The axes correspond to the body frame coordinate system, by design. This reduces the inertia tensor to a diagonal tensor with three principal inertia components I_1, I_2 and I_3.

The particle shape description is another important input set. The benefit of our new granular DEM method is that to specify the shape, only a *support mapping* must be available, as explained in detail in section 4.2. For quadric shapes such as spheres, ellipsoids, cylinders, and cones there are simple analytical expressions for the support mapping. Any other convex shapes that can be represented as *polyhedra* have support mapping based on the vertex locations.

The last important input property that is required is also related to the particle shape. It is the *mean squared distance from the particle center of mass to the particle surface*. Details of why this is needed are presented in the force model section.

3. Broad-phase contact detection

For non-spherical DEM contact detection is one of the hardest steps. To simplify this we employ a two-stage approach: a broad-phase method builds a list of *possible* contact pairs, and the narrow-phase algorithm determines which of these potential contact pairs are actually in contact.

The broad-phase method speeds up contact detection by first simplifying the actual non-spherical particle shape to a simpler geometrical shape that allows for faster overlap detection [27, 28]. This shape is a *bounding volume*, and it completely encloses the actual particle.
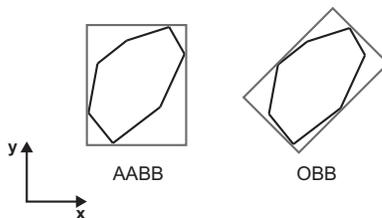
Figure 2: Two bounding volumes in two dimensions. To the left an axis aligned bounding box (AABB) is shown and to the right an oriented bounding box (OBB).

There are many possible bounding volume shapes and Ericson [27] describes them in detail. In this work we employ axis aligned bounding boxes (AABBs) and oriented bounding boxes (OBBs), see figure 2. To build the potential contact pairs list we use a *spatial hash grid* [29, 30] or an *octree* [28].

4. Narrow-phase contact detection

In the narrow-phase the actual particle geometry is used. Before introducing the narrow-phase algorithm two addition concepts are explained which are required for understanding the method.

4.1. Minkowski difference

The first concept of the *Minkowski difference* comes from set theory. Consider two sets A and B of points in $\mathbb{R}^3$. The *Minkowski difference* is a mathematical operation defined as:

$$C = A \ominus B = \{\mathbf{a} - \mathbf{b} \mid \mathbf{a} \in A, \ \mathbf{b} \in B\}. \tag{2}$$

The Minkowski difference is another set C that contains all vectors that can be constructed by taking a vector from B and subtracting it from a vector from A. A visual representation is given in figure 3, where a 2D version of the Minkowski difference is shown. The left schematic shows two overlapping polytopes, which are visual representations of two sets where all points on the edge and interior belong to that set. The right schematic shows the Minkowski difference set. The origin point is now part of this polytope. This leads to the fundamental property of the Minkowski difference:

11
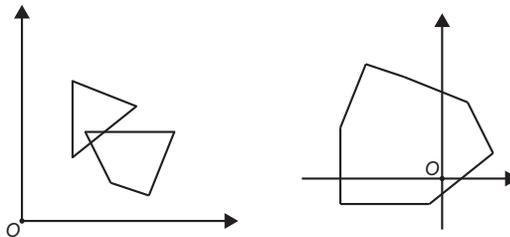
Figure 3: Two dimensional example of the Minkowski difference. To the left, two overlapping polytopes. The right shows the Minkowski difference of the polytopes, including the origin inside the Minkowski difference.

- If set C contains the origin $\mathbf{0}$, the original sets A and B have overlap.

The complementary property to this is that when the origin is not part of the new polytope (set C), there will be no overlap between the original polytopes (sets A and B).

4.2. Support mapping

The second mathematical concept is that of the support mapping, which is a function defined for each particle shape. The support mapping takes as input a search direction (as a vector) and returns a point on the particle surface that is most extreme in the input search direction. Let $\mathbf{d}$ be a search direction vector, and A a particle. The support mapping of particle A is denoted as $\mathbf{s}_A(\mathbf{d})$. A useful property of the support mapping function is:

$$\mathbf{d} \cdot \mathbf{s}_A(\mathbf{d}) = \max\{\mathbf{v} \cdot \mathbf{d} \mid \mathbf{v} \in A\}. \tag{3}$$

Equation (3) states that the point that is returned by the support mapping, has a maximum inner product with the search direction. Figure 4 shows two examples of the support mapping. The black arrows are the input search directions and the grey points are the support mapping outputs: points extreme in the given search direction. For a polygon (left) the most extreme point will always be a vertex. For quadric shapes such as the ellipse (right) any point on the edge can be an extreme point.

The exact form of the support mapping $\mathbf{s}_A$ depends on the particle shape. Table 5 lists the support mappings for a number of different particle shapes. The polyhedron definition gives rise to the algorithm listed in 1 that can find the support mapping for a polyhedron with n vertices. This algorithm is $O(n)$ in the number of vertices. The particles that will be

12
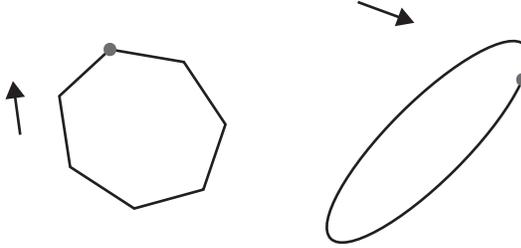
Figure 4: Support mapping of a polygon (left) and an ellipse (right). The arrows indicate the search direction and the grey points are the corresponding extreme points.

Algorithm 1 Support mapping for polyhedral particles.

1: **procedure** SUPPORT-MAPPING-POLYHEDRA($\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}, \mathbf{d}$)
2:      $d_{max} = \mathsf{dot}(\mathbf{v}_1, \mathbf{d})$
3:      $id\_max = 1$
4:      **for** $i = 2 \ldots n$ **do**
5:          $d = \mathsf{dot}(\mathbf{v}_i, \mathbf{d})$
6:          **if** $d > d_{max}$ **then**
7:              $id\_max = i$
8:              $d_{max} = d$
9:          **end if**
10:      **end for**
11:      **return** $\mathbf{v}_{id\_max}$
12: **end procedure**

---

used in this work only have about 10 to 20 vertices, so algorithm 1 will be sufficiently fast. A faster algorithm that could be used is the *hill-climbing algorithm* [31].

The connection between the support mapping and the Minkowski difference is relevant in the narrow-phase algorithm. Instead of calculating the Minkowski difference explicitly for all possible particle shapes, only the following property is required:

$$\mathbf{s}_{A \ominus B}(\mathbf{d}) = \mathbf{s}_A(\mathbf{d}) - \mathbf{s}_B(-\mathbf{d}). \tag{4}$$

Equation (4) states that the support mapping of a Minkowski difference object is equal to the difference of the support mappings of the original objects, where the search direction

Table 5: Support mapping functions for different particle shapes. The ellipsoid, cylinder and cuboid are assumed to be oriented along the Cartesian directions (as in their body frames). The sgn is the sign function which is defined as $\mathsf{sgn}(x) = -1$ if $x < 0$, and 1 otherwise.

Shape	Parameters	Support Mapping $\mathbf{s}_A(\mathbf{d})$
Point	vertex $\mathbf{v}$	$\mathbf{v}$
Sphere	radius R	$\dfrac{R}{\|\|\mathbf{d}\|\|}\mathbf{d}$ if $\mathbf{d} \neq \mathbf{0}$, $\mathbf{0}$ otherwise
Ellipsoid	semi-major axes a, b, c	$\dfrac{(a^2 d_x,\ b^2 d_y,\ c^2 d_z)}{\|\|(a d_x,\ b d_y,\ c d_z)\|\|}$ if $\mathbf{d} \neq \mathbf{0}$, $\mathbf{0}$ otherwise
Cylinder	radius R, height h	$\sigma = \sqrt{d_x^2 + d_y^2},\quad (\dfrac{R}{\sigma}d_x,\ \dfrac{R}{\sigma}d_y,\ \mathsf{sgn}(d_z)\dfrac{h}{2})$ if $\sigma >$ 0, $(0,\ 0,\ \mathsf{sgn}(d_z)\dfrac{h}{2})$ otherwise
Cuboid	side-lengths a, b, c	$(\mathsf{sgn}(d_x)\dfrac{a}{2},\ \mathsf{sgn}(d_y)\dfrac{b}{2},\ \mathsf{sgn}(d_z)\dfrac{c}{2})$
Polyhedron	set of vertices $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$	$\mathbf{v}_i$ for which $\max\{\mathbf{v} \cdot \mathbf{d} \mid \mathbf{v} \in \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}\}$

should be negated for object B.

4.3. Gilbert-Johnson-Keerthi's overlap algorithm

The narrow-phase algorithm that can determine if there is overlap between two convex particle is the *Gilbert-Johnson-Keerthi* (GJK) overlap algorithm [32, 33]. The GJK algorithm attempts to construct a *simplex* that encloses the origin of the Minkowski difference of the two particles. If the GJK algorithm succeeds that means there is overlap. If no simplex can be constructed that encloses the origin, the particles are not in contact.

For the two dimensional version of the GJK method, a triangle (2-simplex) is needed to enclose the origin. In 3D a tetrahedron needs to be formed. The iterative procedure is detailed in algorithm 2.

The most important steps are in lines $12-15$. In these lines the current simplex is checked for enclosing the origin. If it does not enclose the origin, an attempt is made to update the simplex. For this update there are a number of options. If the simplex is a triangle it cannot contain the origin (for a three dimensional case). In this situation, the update function will attempt to add a fourth point creating a tetrahedron. If the simplex

Algorithm 2 Gilbert-Johnson-Keerthi (GJK) overlap algorithm.

1: **procedure** GJK-OVERLAP($\mathbf{s}_A$, $\mathbf{s}_B$)

- search direction $\mathbf{d} = \mathsf{randomUnitVec}$

- $simplex = \{\emptyset\}$

- $\mathbf{C} = \mathbf{s}_{A\ominus B}(\mathbf{d})$

- $simplex = \{\mathbf{C}\}$

2:     **if** $\mathbf{C} \cdot \mathbf{d} < 0$ **then**

3:         **return** false

4:     **end if**

- set the new search direction equal to $\mathbf{d} = -\mathbf{C}$

- $\mathbf{B} = \mathbf{s}_{A\ominus B}(\mathbf{d})$

- $simplex = \{\mathbf{C}, \mathbf{B}\}$

5:     **if** $\mathbf{B} \cdot \mathbf{d} < 0$ **then**

6:         **return** false

7:     **end if**

- set the new search direction equal to $\mathbf{d} = (\mathbf{C} - \mathbf{B}) \times -\mathbf{B} \times (\mathbf{C} - \mathbf{B})$

8:     **while** true **do**

- $\mathbf{A} = \mathbf{s}_{A\ominus B}(\mathbf{d})$

- $simplex = simplex + \{\mathbf{A}\}$

9:         **if** $\mathbf{A} \cdot \mathbf{d} < 0$ **then**

10:             **return** false

11:         **end if**

12:         updateSimplexAndSearchDirection($simplex$, $\mathbf{d}$)

13:         **if** isOriginInsideSimplex($simplex$) **then**

14:             **return** true

15:         **end if**

16:     **end while**

17: **end procedure**
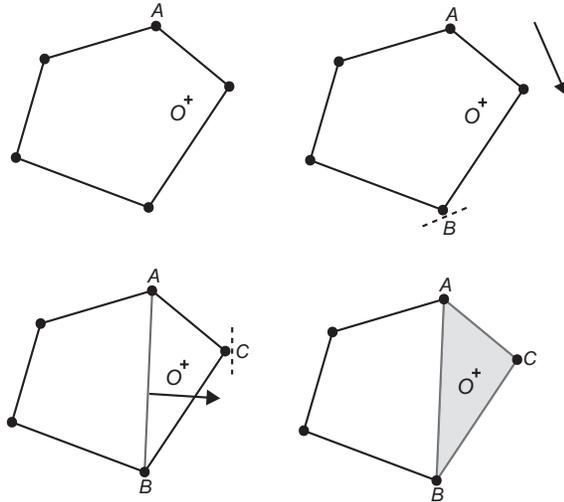
Figure 5: GJK algorithm in two dimensions. The initial support point on the Minkowski difference is **A**. By searching for a new support point in the direction of −**A** (note the location of the origin O) the second point **B** is found. The third point is added by looking orthogonal to the line segment **AB** in the direction of the origin. The support point found is **C**. In two dimensions, a triangle is enough to enclose the origin, and in this case triangle **ABC** encloses the origin, which terminates the algorithm.

already was a tetrahedron, there are two options. First, if the origin is inside, the GJK while loop is terminated. Second, if the tetrahedron does not contain the origin, again the update function tries to create a new simplex that does contain the origin. If at one point in the while loop the new support point has a negative dot product with the current search direction, the GJK while loop is also terminated because it can be concluded that the origin cannot lie inside the Minkowski difference.

Figure 5 visualizes how the GJK algorithm works in two dimensions. In two dimensions, the GJK algorithm needs to construct a triangle to be able to enclose the origin. Starting from a support mapping point **A**, the next point added is **B**. At this point a line segment is formed, which cannot enclose the origin. So a third point needs to be added, by finding a support mapping point in the direction of the origin, orthogonal to the line segment. Thus point **C** is found, which creates the 2-simplex **ABC**. This simplex contains the origin, hence we can conclude that the Minkowski difference contains the origin. This means that the two particles that created the Minkowski difference have overlap and are in fact contacting.

5. Contact resolution

The next step in the granular DEM is to resolve the contact parameters. These are the *contact normal* and *penetration depth*, and the *contact point*. The normal and penetration depth are obtained by the *Expanding Polytope Algorithm* (EPA) [33].

5.1. Expanding Polytype Algorithm (EPA)

The Expanding Polytope Algorithm starts with the terminating simplex of the GJK algorithm. This simplex is a tetrahedron which contains the origin. EPA expands this tetrahedron in the direction of the closest feature from the origin to the simplex. In two dimensions this feature is an edge and in three dimensions it is a triangle. This process of expanding the simplex is iterative and continues until the actual closest feature on the Minkowski difference is found. The normal of this feature is the required contact normal and the distance from the origin to the closest point on the feature is the penetration depth. Algorithm 3 shows the procedure of EPA.

The most important part of the algorithm is given in lines $3-14$. This is the iterative section in which the polytope gets expanded. This update also removes obsolete triangles.

Figure 6 visualizes the working of EPA in two dimensions. The closest feature will be an edge in this situation. The first edge that is tested is far from being the true closest edge. This is a feature of the algorithm: progress is made by expanding the polytope in seemingly the wrong direction. After a new support mapping point is added, the old edge that can be seen from this new point is removed. This creates a hole in the polytope, which is filled by creating two new edges. The new polytope approximates the shape of the Minkowski difference much closer than before. In the simple case of figure 6, the actual closest edge is now part of the polytope, and this causes the EPA to terminate and to return the contact normal and penetration depth.

5.2. Contact point generation

The last quantity that needs to be determined in the contact resolution phase is the contact point. For non-spherical particles the contact does not have to be a unique point. It can be an edge or plane. However, even in the case of a contact edge or plane, we will still apply the contact force at a single contact point within this edge or plane. To

Algorithm 3 Expanding Polytope Algorithm (EPA) for finding the penetration depth and contact normal.

1: **procedure** EPA($\mathbf{s}_A$, $\mathbf{s}_B$, *simplex*)

2:    initialize the EPA polytope with the terminating GJK simplex

3:    **while** true **do**

4:        findClosestFaceOfPolytope()

5:        **if** $|closestDistance - oldClosestDistance| < \epsilon$ **then**

6:            $penetrationDepth = closestDistance$

7:            $normal = $ GetNormal()

8:            **break**

9:        **end if**

10:       updateSearchDir()

11:      getNewSupportPoint($\mathbf{s}_A$, $\mathbf{s}_B$)

12:      removeOldTriangles()

13:      updatePolytope()

14:    **end while**

15: **end procedure**

---

apply the correct torque it is important that the correct location is chosen for this contact point. Suppose that the contact geometry is a plane formed by two cubes stacked on top of each other. The correct contact point is the centroid of the contact plane. Similarly, for a contact edge the centroid of the edge segment should be chosen. An erroneous contact point determination will lead to mechanical instabilities in stacks of particles. In fact, for many previous non-spherical DEM algorithms, tall towers of stacked particles (e.g. cubes) tend to collapse because of such instabilities.

We present a new algorithm to determine the contact geometry of the two overlapping particles. Once the contact geometry is found, the centroid can be calculated to obtain the contact point. This method works for both quadric and polyhedral particle shapes. Algorithm 4 explains how the correct contact point is obtained between the contacting particles A and B.

The first part of the algorithm (lines $3-9$) finds points on the objects that are part of
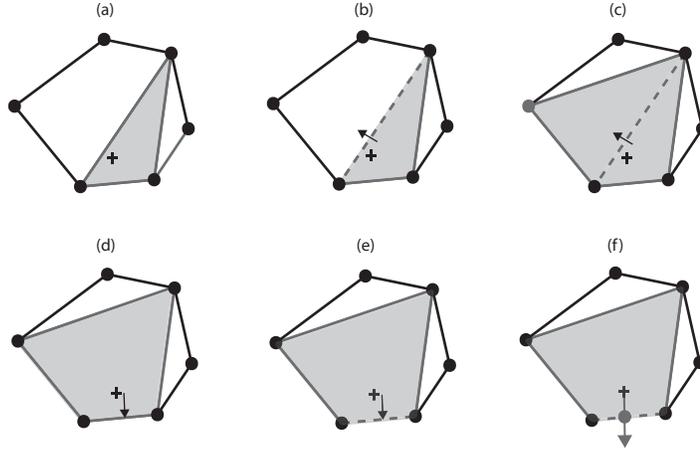
Figure 6: EPA algorithm in two dimensions. (a) The initial polytope containing the origin is a triangle (grey). (b) The closest face (in 2D an edge) is found to be the dotted edge of the triangle. (c) Searching in the normal direction a new support mapping point is found. (d) The edge that can be seen from this new support mapping point is removed. Two new edges are created such that the polytope is again closed. (e) A new closest edge is now found (dotted grey line segment). (f) This new closest edge turns out to be the actual closest feature on the Minkowski difference shape. This yields the contact normal (arrow) and penetration depth.

the contact geometry. This is based on the fact that the distance in the normal direction between a point of A and a point of B must be less or equal to the penetration depth δ. Figure 7 shows this for two cuboid objects. To the left the xz projection (side view) of the objects is shown. Clearly, the four grey points are the points that will be found by the algorithm. Because these are three dimensional particles, a total of four points will be found for each particle, these four points form two polygons. In the case of figure 7 these polygons are two rectangles as shown in the top view of the right figure.

At this point two polygons are found, each from one of the contacting particles. To simplify the following computations the polygons are rotated to a two dimensional plane (either xy, xz or yz plane). It is now much more straightforward to calculate the intersection of the two polygons (which are now 2D). Figure 7 shows the intersection polygon in the right figure, it is equal to the grey shaded rectangle in this example. The intersection polygon is the contact geometry, and the centroid can now be calculated. Note that this point will still be in the rotated two dimensional plane. The last step is therefore to rotate the centroid

Algorithm 4 Contact point generation algorithm.

1: **procedure** CONTACT-POINT-GENERATION($\mathbf{n}$, δ)

Let A be the set of points of object A and B the set of points of object B.

2:   **for** $i \in A$ **do**

3:     **for** $j \in B$ **do**

4:       **if** $(\mathbf{r}[i] - \mathbf{r}[j]) \cdot \mathbf{n} \leq \delta$ **then**

5:         Add point $\mathbf{r}[i]$ to polygon P_A and $\mathbf{r}[j]$ to polygon P_B

6:       **end if**

7:     **end for**

8:   **end for**

9:   rotate(P_A, P_B)

10:   $intersectionPolygon = $ computeIntersection()

11:   $centroidPlane = $ computeCentroid($intersectionPolygon$)

12:   $centroid = $ rotatePointTo3DSpace($centroidPlane$)

13: **end procedure**
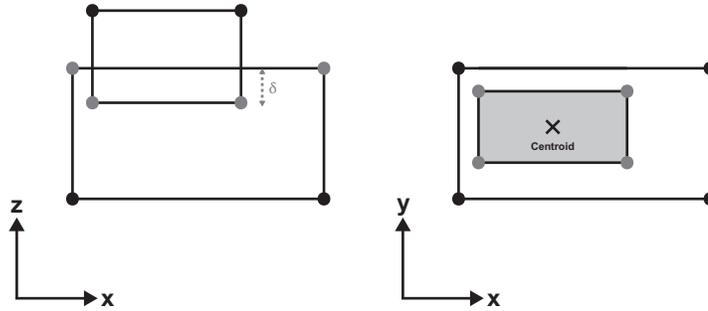
Figure 7: Contact geometry for two colliding cuboids. Left: xz projection (side view) showing the points of the objects (grey) that will be part of the contact geometry. Right: xy projection (top view) showing the two contact polygons and the (grey shaded) intersection polygon, with its centroid.
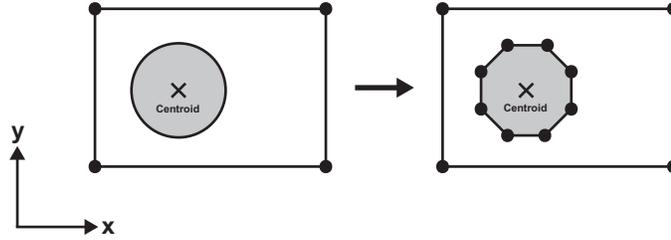
Figure 8: The xy projection of the contact geometry for a cuboid colliding with a cylinder. Left: the actual intersection polygon is a circle (grey shaded). Right: intersection polygon used in algorithm 4 is a polygon representation of the circle (grey shaded).

back to the three dimensional space to obtain the desired contact point.

The algorithm listed in 4 requires points on the objects, for polyhedra these will always be the vertices. For quadric shapes it is not that clear how to find the points, because the surface contains an infinite amount of points. For quadric shapes we therefore use the support mapping in the direction of the contact normal to obtain the correct points. Note that it is possible to get multiple points that are extreme in the normal direction. For example, the circular base of a cylinder constitutes of points that are all extreme in the axial direction. If such a situation occurs, we place a finite amount of points along the circle, which then approximates the actual circle that would be required. This is depicted in figure 8.

5.3. Narrow-phase and contact resolution tests

The narrow-phase GJK algorithm and the contact resolution EPA and contact point algorithms are tested in a particle *stability* test. This requires all the detection and resolution algorithms to work correctly.

One hundred cuboids were initialized some height above the bottom of the cylindrical domain. No initial velocity or force/torque was applied to the particles. As the particles fall, they will hit the bottom of the vessel and start to dissipate energy during these contacts. This process continues until the particles obtain a stable equilibrium position in the stack. The total simulation time was $250\,\mathrm{s}$, which means that 250 million time steps have been calculated. Figure 9 shows the v_z of three cuboids in the stack of 100 cuboids. There is a mechanical equilibrium after an initial dissipation phase, after which the velocities become
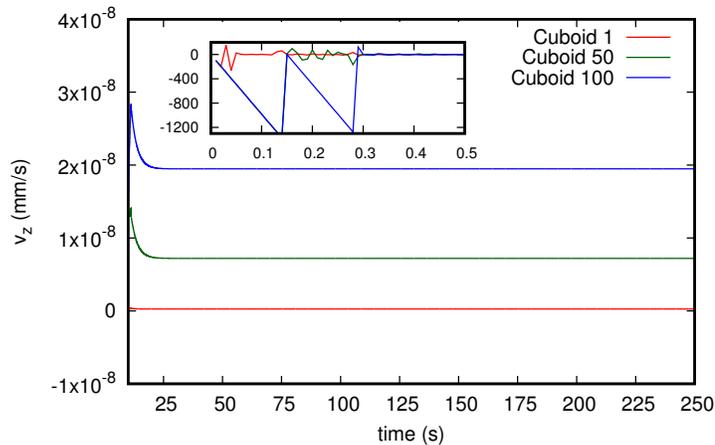
Figure 9: Linear velocity z-component for three cuboids in a stack of 100. The bottom cuboid is numbered 1 (red), the middle cuboid is nr. 50 (green) and the cuboid on the top of the stack is nr. 100 (blue). The inset plots show the time evolution of v_z for the first 0.5 s.

very small and constant. This proves that the stack is mechanically stable and the contact detection and resolution is working correctly.

6. Contact force model

A particle i has a total force acting on it defined as:

$$\mathbf{F}_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \mathbf{F}_{ij}^{c} + m_i \mathbf{g}, \tag{5}$$

where $\mathbf{F}_{ij}^{c}$ is the *contact force* from particle j acting on particle i. The second term on the right-hand side of equation (5) is the gravity. The torque about the particle center of mass $\mathbf{r}_i$ acting on a particle i is defined as:

$$\boldsymbol{\tau}_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \left(\mathbf{r}_{ij}^{c} - \mathbf{r}_i \right) \times \mathbf{F}_{ij}^{c}, \tag{6}$$

where $\mathbf{r}_{ij}^{c}$ is the contact point between colliding pairs i and j. Note that the torque is defined with respect to the center of mass of the particle $\mathbf{r}_i$.

Different contact models are reported in literature. For example, Deen et al. [11] describe contact models for spherical discrete element models. More advanced contact force models

are reviewed in [34, 35]. These models range from the most simple linear spring-dashpot model to non-linear models with hysteresis taken into account. Many spherical models also take into account rolling resistance [36], which takes the form of a torque acting on the particle which is opposing the rolling motion.

Much less is written about contact force models specifically for non-spherical particles. The non-spherical DEM of Wachs et al. [23] uses a linear-spring model (designed for spheres) without any modifications. Similarly, Dvziugys and Peters [37] present a DEM for ellipsoidal particles, and present several contact force models, both linear and non-linear, but without mentioning how to adapt these for non-spherical shapes. The work of Pournin et al. [7] describes a sphero-cylinder discrete element method, again using a linear spring-dashpot model.

Most non-spherical discrete element models use linear contact models that are designed for spherical particles. We follow this approach and use a linear spring-dashpot model for the calculation of the contact forces, but we note that our methodology does not exclude the use of other contact models. Other models can be implemented, but the energy conservation of a non-dissipative ($e_n = 1$) contact should be verified. If the force model parameters of the chosen model are not correctly set, energy will be lost during such contacts.

6.1. Contact force

The contact normal $\mathbf{n}_{ij}$ and normal overlap $\boldsymbol{\delta}_{ij,n}$ between a pair of objects i and j are obtained from the EPA algorithm. Together with the contact point $\mathbf{r}_{ij}^c$ from algorithm 4 all information is available to calculate the contact force. First, the relative velocity at the contact point is needed:

$$\mathbf{u}_{ij} = \mathbf{v}_i + \boldsymbol{\omega}_i \times \left(\mathbf{r}_{ij}^c - \mathbf{r}_i\right) - \mathbf{v}_j - \boldsymbol{\omega}_j \times \left(\mathbf{r}_{ij}^c - \mathbf{r}_j\right). \tag{7}$$

The relative velocity at the contact point can be decomposed into a normal and tangential component as given in equation (8).

$$\mathbf{u}_{ij,n} = \left(\mathbf{u}_{ij} \cdot \mathbf{n}_{ij}\right) \mathbf{n}_{ij}$$
$$\mathbf{u}_{ij,t} = \mathbf{u}_{ij} - \mathbf{u}_{ij,n} \tag{8}$$

23

The tangential unit vector $\mathbf{t}_{ij}$ can be calculated by equation (9).

$$\mathbf{t}_{ij} = \begin{cases} \frac{\mathbf{u}_{ij,t}}{\|\mathbf{u}_{ij,t}\|} & \text{if } \|\mathbf{u}_{ij,t}\| \neq 0, \\ \frac{\boldsymbol{\delta}_{ij,t}}{\|\boldsymbol{\delta}_{ij,t}\|} & \text{if } \|\mathbf{u}_{ij,t}\| = 0 \text{ and } \|\boldsymbol{\delta}_{ij,t}\| \neq 0, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{9}$$

The normal component of the contact force is given by:

$$\mathbf{F}_{ij,n}^c = -k_n \boldsymbol{\delta}_{ij,n} - \eta_n \mathbf{u}_{ij,n}. \tag{10}$$

The parameters k_n and η_n are the normal spring stiffness and normal damping coefficient, respectively. The tangential component of the contact force, which takes into account the Coulomb friction criterion, is defined in equation (11).

$$\mathbf{F}_{ij,t}^c = \begin{cases} -k_t \boldsymbol{\delta}_{ij,t} - \eta_t \mathbf{u}_{ij,t} & \text{if } \|\mathbf{F}_{ij,t}^c\| \leq \mu_c \|\mathbf{F}_{ij,n}^c\|, \\ -\mu_c \|\mathbf{F}_{ij,n}^c\| \mathbf{t}_{ij} & \text{if } \|\mathbf{F}_{ij,t}^c\| > \mu_c \|\mathbf{F}_{ij,n}^c\|. \end{cases} \tag{11}$$

Equation (11) introduces three new model parameters, the tangential spring stiffness k_t, the tangential damping coefficient η_t and the Coulomb friction coefficient μ_c.

The tangential displacement $\boldsymbol{\delta}_{ij,t}$ is updated during the contact using equation (12), starting with $\boldsymbol{\delta}_{ij,t} = \mathbf{0}$ at the beginning of the contact.

$$\boldsymbol{\delta}_{ij,t}^{n+1} = \begin{cases} q_{rot} \boldsymbol{\delta}_{ij,t}^n q_{rot}^{-1} + \displaystyle\int_{t^n}^{t^{n+1}} \mathbf{u}_{ij,t} dt & \text{if } \|\mathbf{F}_{ij,t}^c\| \leq \mu_c \|\mathbf{F}_{ij,n}^c\|, \\ \frac{1}{k_t} \left(-\mu_c \|\mathbf{F}_{ij,n}^c\| \mathbf{t}_{ij} - \eta_t \mathbf{u}_{ij,t} \right) & \text{if } \|\mathbf{F}_{ij,t}^c\| > \mu_c \|\mathbf{F}_{ij,n}^c\|. \end{cases} \tag{12}$$

The first equation in (12) is valid for the sticking regime. The rotation quaternion q_{rot} takes into account the effect that the contact normal does not stay in the same plane during the contact. This means that the old displacement value needs to be rotated into the new, slightly rotated frame [38]. In the sliding regime, the tangential displacement should not be increased further. Otherwise, this would lead to a too large displacement should there be a transition back to the sticking regime. Hence the tangential displacement is set to a value that is consistent with the Coulomb criterion [39].

Equations (10)-(11) give the expressions for the contact force. They contain parameters that need to be set. For spheres several derivations have been carried out before, see e.g.

[12, 38]. This work will present a derivation that is analogous to that of Pournin et al. [40], where these authors assumed spherical particles. The result derived in this work will have the desirable property that when a spherical shape is used, the original spherical results of [12, 38, 40] are obtained. The final equations are:

1. Set values for k_n, e_n and e_t.

2. Calculate k_t:

$$k_t = k_n \frac{1}{\left(\dfrac{1}{\mu} + \dfrac{\langle \|\mathbf{r}_{ij}^c - \mathbf{r}_i\|^2 \rangle}{\langle I_i \rangle} + \dfrac{\langle \|\mathbf{r}_{ij}^c - \mathbf{r}_j\|^2 \rangle}{\langle I_j \rangle} \right)} \frac{\pi^2 + \ln^2(e_t)}{\pi^2 + \ln^2(e_n)}. \tag{13}$$

3. Determine the contact time t_c:

$$t_c = \sqrt{\frac{\mu \left[\pi^2 + \ln^2(e_n) \right]}{k_n}}. \tag{14}$$

4. Calculate η_n:

$$\eta_n = -\frac{2\mu}{t_c} \ln(e_n). \tag{15}$$

5. Calculate η_t:

$$\eta_t = -\frac{2}{t_c \left(\dfrac{1}{\mu} + \dfrac{\langle \|\mathbf{r}_{ij}^c - \mathbf{r}_i\|^2 \rangle}{\langle I_i \rangle} + \dfrac{\langle \|\mathbf{r}_{ij}^c - \mathbf{r}_j\|^2 \rangle}{\langle I_j \rangle} \right)} \ln(e_t). \tag{16}$$

6.2. Binary contact energy conservation test

The contact force parameters derived in the previous section are not exact for non-spherical particles. This means that the requirement of equal contact times for tangential and normal components will not hold exactly. This requirement is important because otherwise one of the springs is still loaded while the contact has already ended. If the two contact times are different, a non-dissipative system ($e_n = e_t = 1$) with no Coulomb friction would still loose energy during a contact. Therefore, binary contacts between non-spherical particles are simulated and the conservation of energy is measured during these contacts.

The tests consist of two particles in a cylindrical domain with no gravitation. The particles are given initial linear and angular velocities randomly drawn from a Gaussian

Table 6: Average energy loss $\langle E_{loss} \rangle$ percentage after a binary contact with no dissipation. The averaging is taken over 1000 simulation samples.

Particle combination	$\langle E_{loss} \rangle$ (%)
Sphere-Sphere	0.09
Ellipsoid-Ellipsoid	0.32
Cylinder-Cylinder	0.29
Tetrahedron-Tetrahedron	0.20
Cube-Cube	0.37
Cube-Sphere	0.17
Cube-Cylinder	0.27
Ellipsoid-Tetrahedron	0.15

distribution. For each contact the pre-collisional kinetic energy E_k^0 is determined, along with the post-collisional residual tangential spring energy E_t, see equations $(17)-(18)$.

$$E_k^0 = \sum_{i=1}^{2} \frac{1}{2} m_i \mathbf{v}_i \cdot \mathbf{v}_i + \frac{1}{2} \left(I_{1,i} \omega_{x,i} + I_{2,i} \omega_{y,i} + I_{3,i} \omega_{z,i} \right) \tag{17}$$

$$E_t = 0.5 k_t \delta_t^2 \tag{18}$$

The residual energy loss error can then be defined as:

$$E_{loss} [\%] = \frac{E_t}{E_k^0} \cdot 100\%. \tag{19}$$

The results of the simulations are energy loss percentages averaged over 1000 independent simulations. Table 6 shows these results. The sphere-sphere contacts have the lowest energy loss error. The derivation is exact for sphere-sphere contacts, which means that per definition it should have a very low energy loss. All other values are also small in the range of $0.1 - 0.4\%$. This an acceptable error and in fact proves that assumptions for the force model parameters derivation are reasonable.

7. Elongated non-spherical packings

This section will use the granular DEM to generate random loose packings of cuboids, cylinders and ellipsoids with five different aspect ratios. Because the packings are generated

Table 7: Simulation parameters for the cuboid, cylinder and ellipsoid packing simulations. The spatial hash grid method was used as broad-phase for all simulations.

Parameter	Values	Unit
Cylindrical domain		
radius R_{dom}	25	mm
z-height	$50-400$	mm
Non-spherical particles		
Aspect ratio (AR)	(0.25, 0.5, 1.0, 2.0, 3.0)	
Cuboid half-length x, y	(1.5, 1.5)	mm
Cuboid half-length z	(0.375, 0.75, 1.5, 3.0, 4.5)	mm
Cylinder radius	1.5	mm
Cylinder height	(0.75, 1.5, 3.0, 6.0, 9.0)	mm
Ellipsoid semi-axes x, y	(1.5, 1.5)	mm
Ellipsoid semi-axes z	(0.375, 0.75, 1.5, 3.0, 4.5)	mm
Simulation settings		
time step	1.0×10^{-6}	s
gravity	9.81	$\mathrm{m\,s^{-2}}$
batch size	$20-60$	
Contact parameters		
k_n	1.0×10^4	$\mathrm{N\,m^{-1}}$
e_n	0.8	
e_t	0.6	
μ_c	0.1	

in a cylindrical vessel, the wall influence on positional and orientational ordering will follow naturally. Appendix A shows how the positional structure is obtained for non-spherical packings and Appendix B gives additional information about the orientational structure that is used in the results.

7.1. Simulation setup

The particles each have a different aspect ratio (AR). The definition used for the aspect ratio is the z-length of the particle (vertical size in the figure) divided by the x-length of the particle (horizontal size in the figure). For all three shapes the x and y size of the particles

27

was chosen to be equal, hence the definition of AR gives the same result when using the y-length. Table 7 shows the settings for all simulations. Batches were dropped and frozen after consolidation of the packing, to speed up the packing process.

In the cases of larger aspect ratios $AR \geq 2$ large overlap (about 0.2 mm) was found about once or twice in the entire simulation time. This overlap is so large that particles in contact are launched beyond the domain bounds. When such a numerical explosion developed, the simulation was restarted two insertion batches back in time. The reason of this large overlap was also investigated. It was found that this large overlap was caused by the fact that particles penetrated into each other while rotating. This can happen when both rotational velocities are large and the duration of the contact is very long. The proper solution of this problem would be to increase the stiffness of the particles, which decreases the contact time. This however would also require a smaller time step, which means even longer simulation times and therefore this option was discarded.

7.2. Positional structure results

The radial porosity profiles are shown in figure 10. The sawtooth pattern is much less pronounced for the cuboids and cylinders. The reason is that for the z-profile, the only stable configuration for the first layer is a cuboid and cylinder resting on a flat face. In the radial direction, any other orientation is also possible. The profiles for the cuboid have the largest fluctuations, as can be seen from figures $10\,(d)-(e)$. There is an influence from the aspect ratio, but only in the range from AR = 0.25 to AR = 1.0 it is pronounced, as for these cases the profile has smaller magnitude oscillations.

Zhang et al. [41] performed an experimental study on packings of equilateral cylinders and also determined the radial profiles. They used X-ray microtomography to scan the packings of equilateral cylinders. Figure 11 shows the radial profile of their work in blue. It is compared to the cylinder packing (AR = 1) of this work. Good agreement is obtained, however the experimental packing had a slightly higher overall packing fraction, which is why the blue graph is on average shifted downward compared to the simulations.

The solid volume fraction as a function of the aspect ratio has been reported in literature by several authors. Most studies use superballs to approximate a real cuboid [42, 43]. This means that the particles are more like a rounded cube, with no sharp edges. A number of
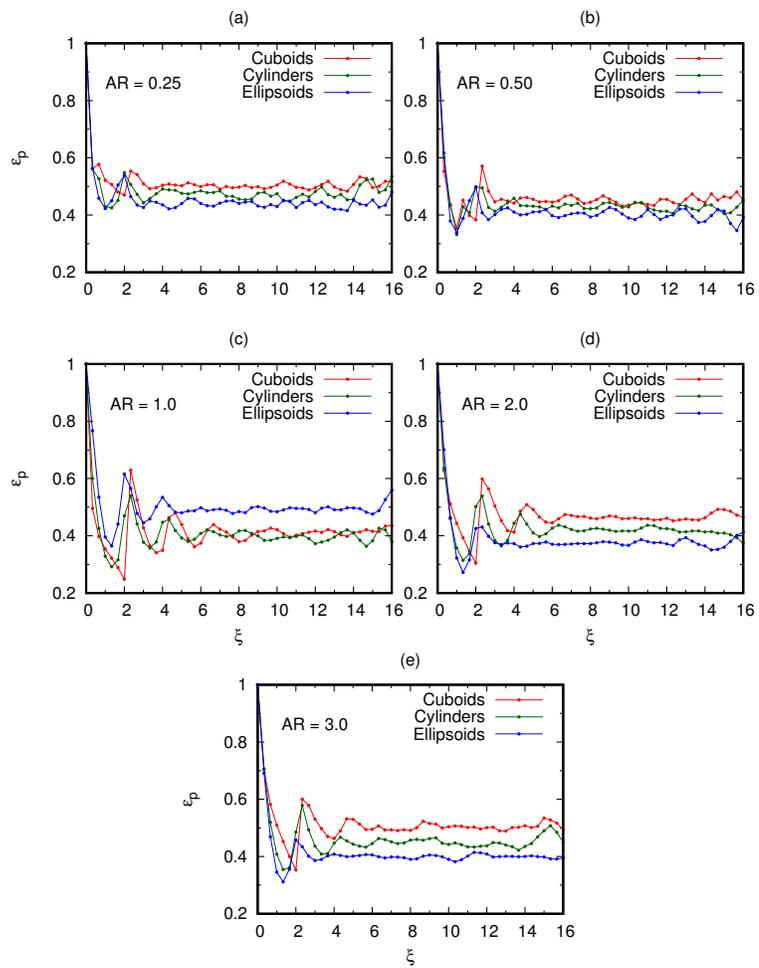
Figure 10: Radial porosity profile for five different aspect ratios. The particle shapes are color coded with in red for cuboids, green for cylinder and blue for ellipsoids. The aspect ratio is 0.25 for (a), 0.5 for (b), 1.0 for(c), 2.0 for (d) and 3.0 for (e).

29
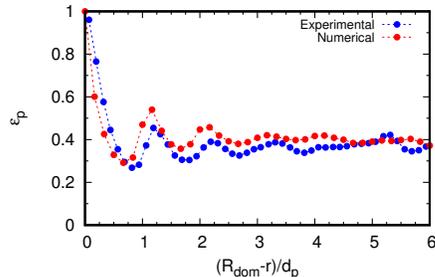
Figure 11: Comparison between the experimental (blue) and numerical (red) porosity profiles for a packing of equilateral cylinders (AR = 1). On the horizontal axis is the dimensionless distance from the wall, made dimensionless by dividing by the cylinder diameter d_p. The experimental results are from [41].

polyhedral based representations are also used. These create real cuboids with sharp edges [44, 45]. The recent work of Liu et al. [46] reports the packing fraction versus the aspect ratio of cuboids. Their trend shows a double maximum with a local minimum in between. This local minimum is located at AR = 1. Although this local minimum is not present in figure 12 (a), it is also not possible to see because the resolution in AR is not large enough.

For cylinders the trend is similar to that of Li et al. [47], a single maximum for the packing fraction is reached around AR = 1. As their resolution was larger than that used in this work it is not likely that the cylinders actually have a local minimum packing value at AR = 1, as is the case for the cuboids. It should be noted however that spherocylinders do exhibit this behaviour, as reported by Wouterse et al. [6].

Donev et al. [4] determined the solid volume fraction of ellipsoids for a larger range of aspect ratios, and found the same trend as obtained in this work. Figure 12 (c) shows that there is a minimum in the packing fraction for AR = 1 (spheres). The numerical values of [4] are higher, which is again due to the loose packing generated with our granular DEM.

7.3. Orientational structure results

To probe the ordering structure in the radial direction, the order parameters $S_{2,r}$ and $S_{4,r}$ (see Appendix B) are plotted as a function of the dimensionless distance from the side walls in figure 14. Focusing on the aspect ratio of 0.25, plots (a), (d) and (g) show a near-wall region of $\xi \leq 2$ for which the values are largest and positive. Since both $S_{2,r}$ and $S_{4,r}$
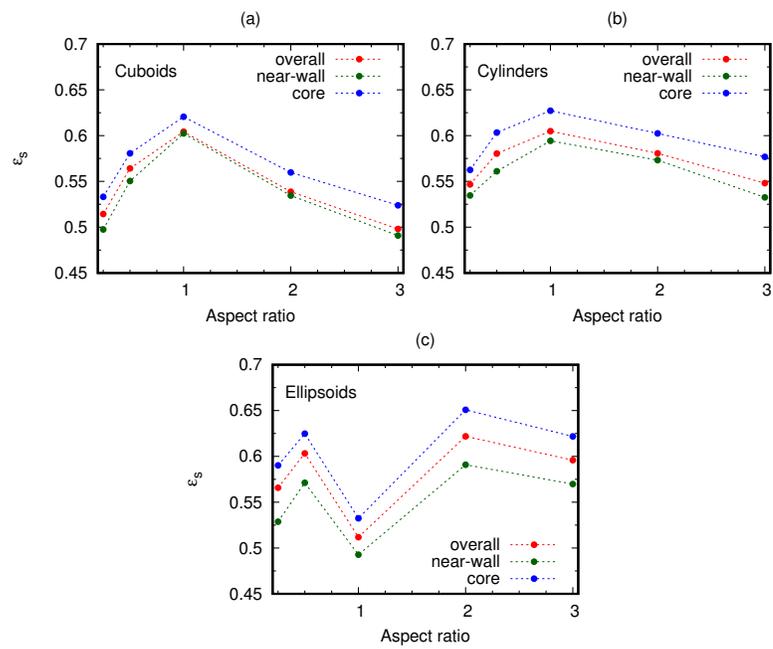
Figure 12: Solid volume fraction as a function of the particle aspect ratio for (a) cuboids, (b) cylinders and (c) ellipsoids. The red points indicate the overall value, the green points the near-wall region and the blue points the core region.
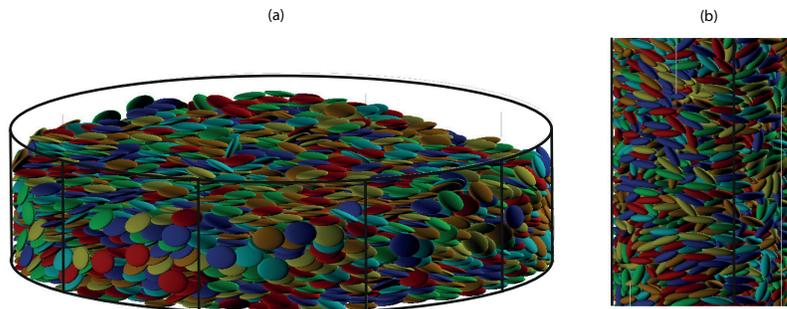
Figure 13: Ellipsoid packings for aspect ratio 0.25 (a) and 3.0 (b). The domain size was the same for both packings.

are positive it indicates that the particle directors are parallel to the radial direction. For the core region, a uniform ordering is present, for all three shapes. The situation is different for $AR = 1$ (plots (b), (e) and (h)), excluding a near-wall region in which the cuboids and cylinders have some ordering, there is little ordering for the core. The cubes and equilateral cylinders therefore behave sphere-like in this region. For the elongated $AR = 3$ shapes shown in (c), (f) and (i) there is a positive orthogonal ordering and negative nematic ordering in the near-wall region for all three shapes. This means that the director of the shapes is mainly located in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. The reason is that this is the only way in which the particle can be orthogonal to the radial direction, which would give a negative nematic value and positive orthogonal value.

To see how fast the nematic and orthogonal order parameters in the radial direction change as a function of distance to the wall, the cumulative plots need to be analyzed. Figure 15 (a) shows that the nematic order parameter reaches its global value in a range of $\xi \leq 2$ for cuboids. Note the inversion for $AR = 0.25$, which indicates going from an aligned situation in the near-wall region to an orthogonal orientation in the core. The cylinder and ellipsoid packings (b) – (c) show a similar trend. The orthogonal parameter $S_{4,r}$ varies over a much larger range of about $\xi \leq 6$. This is especially true for the aspect ratios of 1 and 3, as can be seen from the green and blue lines of (d) and (e). The ellipsoid case (f) is different, there is a point of minimum ordering after which the order increases again in the bed. What is clear from all these plots is that the extent of the orientational ordering change is larger than the positional ordering. So the near-wall region for the orientational structure has a
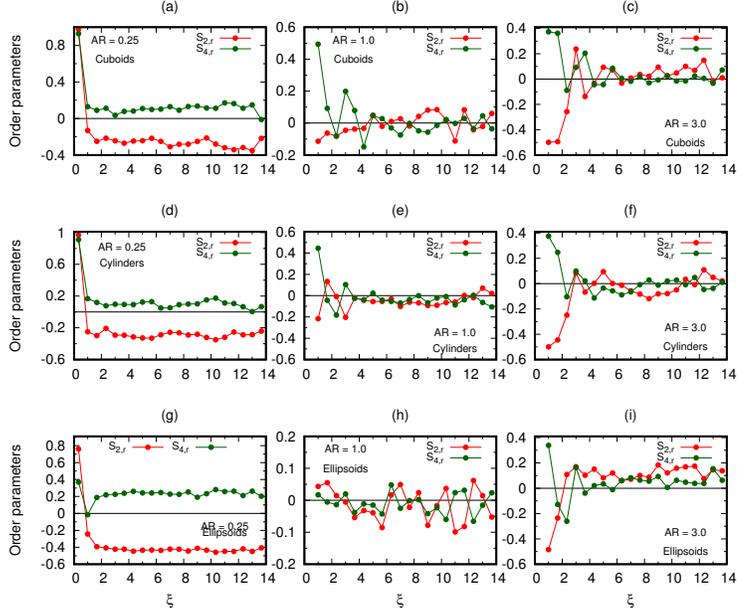
Figure 14: Radial profiles of the cylindrical order parameters $S_{2,r}$ and $S_{4,r}$. The profiles of (a), (b) and (c) are for cuboids and have an AR of 0.25, 1.0 and 3.0. The profiles of (d), (e) and (f) are for cylinders with AR 0.25, 1.0 and 3.0. The last three profiles of (g), (h) and (i) are for the ellipsoids with AR 0.25, 1.0 and 3.0, respectively.
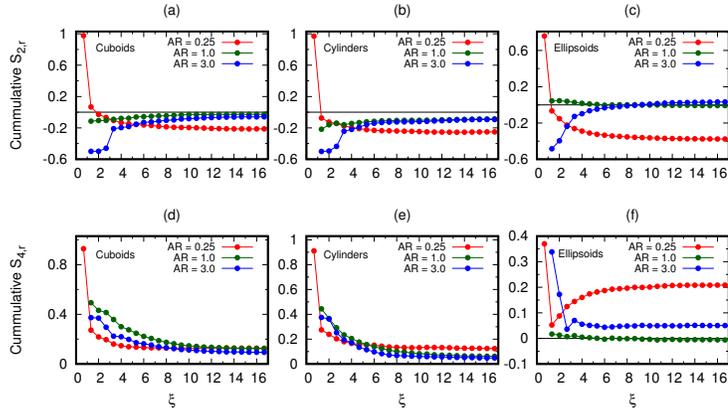
Figure 15: Cumulative radial profiles of the cylindrical order parameters $S_{2,r}$ and $S_{4,r}$. The aspect ratio 0.25 colored red, the AR 1.0 green and the AR 3.0 is blue.

33
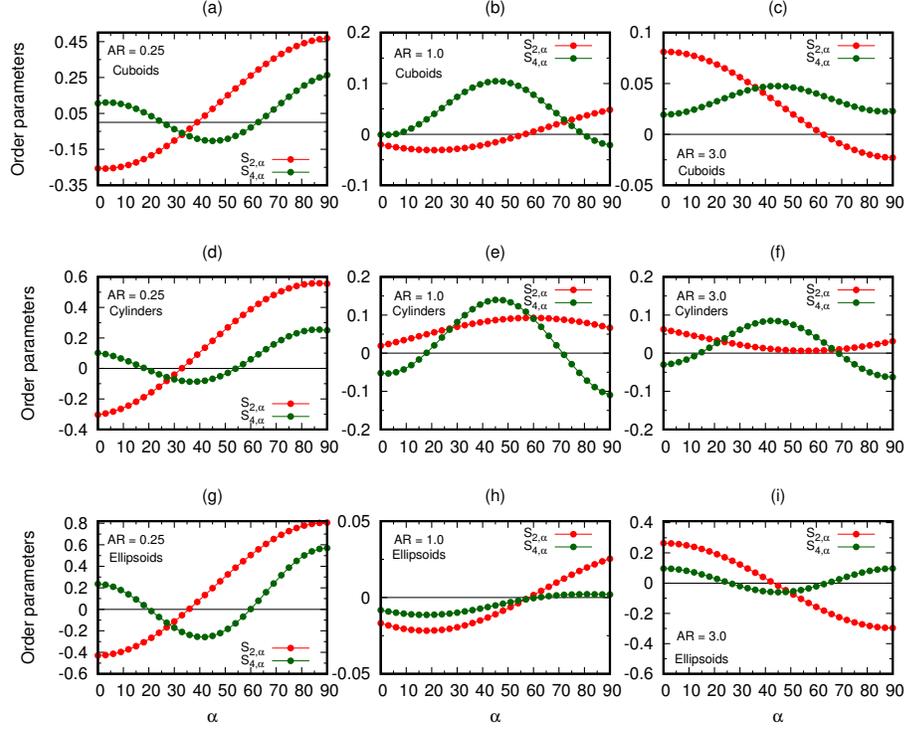
Figure 16: $S_{2,\alpha}$ and $S_{4,\alpha}$ as a function of the angle α in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. The plots (a), (b) and (c) are for cuboids with aspect ratios 0.25, 1.0 and 3.0. The plots (d), (e) and (f) are for cylinders with aspect ratios 0.25, 1.0 and 3.0. The last three plots (g), (h) and (i) are for ellipsoids with aspect ratios 0.25, 1.0 and 3.0.

larger size compared to the positional structure.

The radial profiles of 14 showed that particles can be oriented orthogonal to the radial coordinate. It is also interesting to determine whether or not there is a preferential orientation in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. Figure 16 shows the nematic and orthogonal order parameters $S_{2,\alpha}$ and $S_{4,\alpha}$ as a function of the angle α (with respect to $\hat{\boldsymbol{\theta}}$) in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. For the cuboids there is strong nematic ordering for AR = 0.25 with anti-azimuthal ordering ($\alpha = 0°$) and ordering parallel to z ($\alpha = 90°$). A smooth transition occurs between these two extremes. This profile is similar for the cubes (AR = 1), but in magnitude there is less ordering. The cuboids rods with AR = 3 have an opposite ordering, compared to the other aspect ratios: there is a positive azimuthal alignment and a small anti z alignment. The $S_{4,\alpha}$ values for the cuboids show a maximum at $\alpha = 45°$. This ordering is shown in figure 17, where near

34
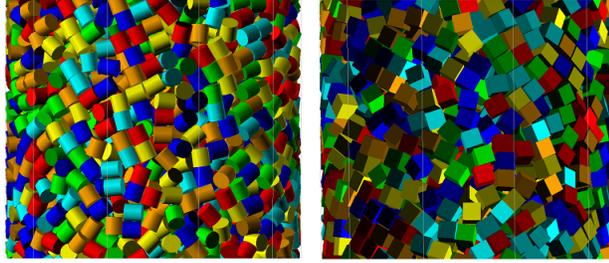
Figure 17: Cylinder (left) and cuboid packings (right) with an aspect ratio of 1. The particles near the wall are oriented mainly in orthogonal to the radial coordinate and have a 45° angle in the $\theta - z$ plane.

the wall a preferential ordering of 45° in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane is visible. This particular alignment was also found in the experimental work on cylinders of [41]. Their packings showed the same profile for $S_{4,\alpha}$ and the numerical values obtained for the maximum at $\alpha = 45°$ was in the range $0.1 - 0.2$, which is the same as the results from this work (see fig. 16 (e)).

8. Granular particle packings

8.1. Creating granular particles

To generate granular particle packings, a particle generation method is required. It is possible to use digital data of experimentally scanned particles. As an example, a 3D laser ranging (LADAR) technique is used by Latham et al. [48] to create digital images of rock samples commonly found in mining industry and civil engineering. The authors have compiled a shape library of processed rocks of varying detail and shapes.

Another option is using X-ray tomography to generate an image, as is done by Wang et al. [49] and Taylor et al. [50]. They generated voxel-images of the complete interior of the samples, which is relevant for porous structures. This data can then be processed to generate surface meshes to be used in DEM applications. The downside of using experimentally generated shapes is that they are non-convex and have thousands of vertices. A convex version of the shape should then be created first, which can destroy the original shape.

This section describes two *numerical* options to create granular particles. The first method generates particles two types of *sand*-like particles and the second method generates *woodchips*.

8.1.1. Sand-shaped particles

The goal is to create granular particles that are all unique, but still have a common overall shape. Another requirement is that there are at most 20 vertices that describe these shapes. The common overall shape serves as a kind of template, and for this work a sphere and prolate ellipsoid is used. The first step is to generate vertices on the template object. Between 5 and 20 vertices are used. These points are placed randomly on the surface of either the sphere or ellipsoid. The second phase uses an *energy minimization* method which redistributes the points on the surface, in a more uniform way. This is achieved by introducing a repulsion force between the vertices, which is large when the vertices are close together and smaller if they are further apart. The minimization is *not* allowed to fully converge, but instead is terminated early. The *convex hull* of the the configuration of vertices that is obtained is determined. After a final additional check for convexity, the particle is suitable as a polyhedron class for the granular DEM code. The reason why the energy minimization on the surface of the template shape is not carried out until convergence, is to avoid generating the same, usually very symmetric, shapes when the number of vertices is the same.

8.1.2. Woodchips-shaped particles

Woodchips are solids created by chipping larger pieces of wood. They are a source of biomass, and are therefore relevant to the energy production. The shapes are generated in the following way. The height h of the particles is chosen randomly in the range of 5 to 7 mm. This is the longest extent of the particle. Next, four points on a circle are randomly selected, this circle has a radius randomly chosen from a uniform distribution between 0.4 and 0.7 mm. The center of this (imaginary) circle is at $(0, 0, \frac{h}{2})$. Another four points are randomly drawn from a second circle, this circle has a radius which is chosen randomly in the range of $1.0 - 1.3$ mm. The location of this circle is at $(0, 0, -\frac{h}{2})$. In total this generates 8 points, which form the woodchip particle. Because the random selection of the parameters, each woodchip particle will be unique, but all will have an elongated shape.
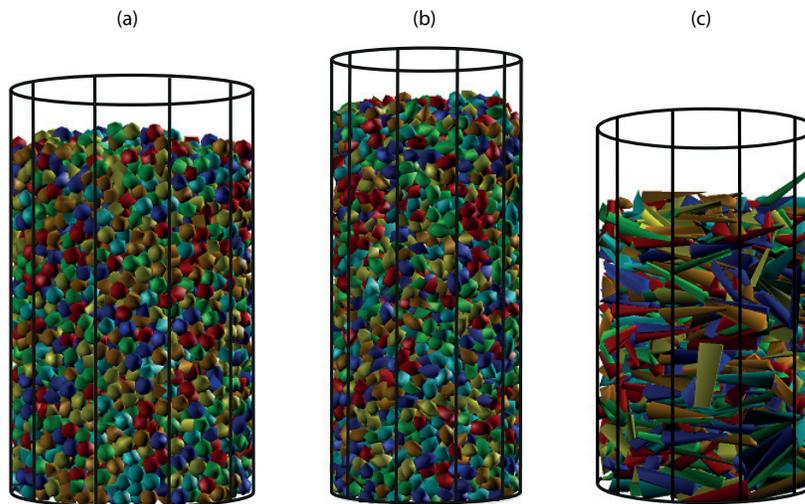
Figure 18: Random packings of three types of granular particles. The sand-shaped particles based on a spherical template (*granular packing 1*) are visualized in (a), the sand-shaped particles based on an ellipsoidal template (*granular packing 2*) in (b), and woodchips-shaped particles (*granular packing 3*) in (c).

8.2. Results

The simulations where carried out in the same way as for the elongated non-spherical particles. The granular particles that have been generated with the energy minimization method were packed in a cylinder with a radius of 25 mm. The woodchips were packed in a vessel with a radius of 10 mm. The spherical-based granular packing is referred to as *granular packing 1*, the ellipsoidal-based granular packing as *granular packing 2* and the woodchips packing as *granular packing 3*. The packings are visualized in figure 18.

The radial porosity profiles are almost completely flat as is apparent from figure 19. The near-wall region is visible for the sand-shaped particles (granular packings 1 and 2) but is nearly absent for the woodchips-shaped particles (granular packing 3). This effect is related to the fact that these are all three poly-disperse packings.

Lastly, the overall solid volume fraction was determined for all three packings. Table 8 shows that the overall solid volume fraction of both sand-shaped particles (granular packings 1 and 2) is almost the same. The elongated woodchips do have a much lower packing fraction. These shapes have the lowest solid volume fraction of all the shapes simulated, which must be related to the fact that they also have the largest aspect ratio. As discussed in the
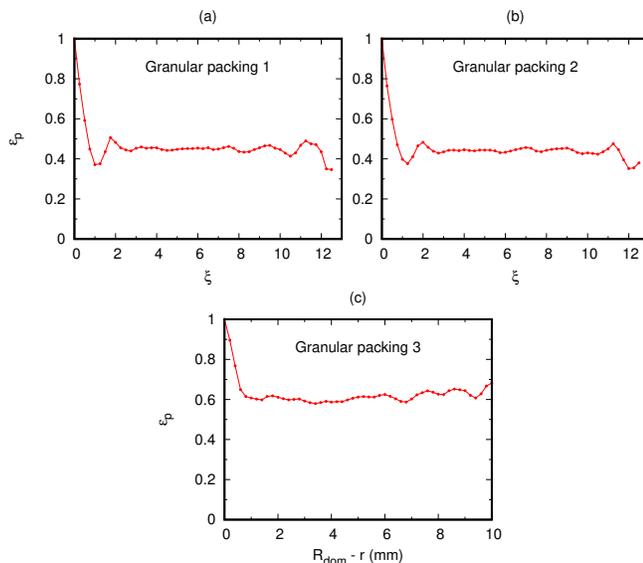
Figure 19: Radial porosity profiles for the sand-shaped particles (labelled granular packings 1 and 2) and the woodchips-shaped particles (granular packing 3).

Table 8: Overall solid volume fraction for the three granular packings.

Particle type	$\bar{\varepsilon}_{\mathbf{s}}$
Sand-shaped, spherical template (*Granular packing 1*)	0.546
Sand-shaped, ellipsoidal template (*Granular packing 2*)	0.550
Woodchips-shaped (*Granular packing 3*)	0.382

elongated non-spherical particles section, for larger AR > 2 the packing fraction decreases for cuboids. Another reason might be related to confinement issues, as the domain radius is only 10 mm, only a single woodchip fits along the radius when oriented in the xy-plane.

9. Conclusions

A granular discrete element method is presented and applied to the generation of random packings of non-spherical particles. The method is very flexible as it allows for the simulation of any convex particle shape. This includes both the quadric types and polyhedral type of particles. This is achieved by using a contact detection and resolution strategy

38

based on the GJK and EPA algorithms. Both require only a support mapping of the particle shape, which is readily available. The contact resolution phase also includes a novel algorithm to find the contact point, as this is a crucial parameter in the correct application of the torque. Verification tests showed that the method achieves unconditional mechanical stability, which proves the correct working of the contact detection and resolution. We also derived expressions for the tangential spring stiffness and tangential damping coefficient for non-spherical particles, based on the requirement that the contact times in tangential and normal direction should be equal.

Random packings of cuboids, cylinders and ellipsoids have been generated for different aspect ratios. Comparing the radial porosity profiles, there was a clear influence on the profiles as the aspect ratio increased. As these non-spherical shapes have a distinguishable orientation, we also looked at orientational ordering throughout the packed beds. Near the side walls, interesting ordering patterns were apparent for equilateral cylinders and cubes (both aspect ratio 1 particles). There was a v-shaped alignment structure which quickly broke up as particles where positioned further from the side walls. The walls also had a clear influence on the orientational ordering of M&M like (oblate ellipsoidal) particles. Near the side walls, there was a preferential vertical alignment, while in the core regions there was a preferential horizontal alignment.

Lastly, three packings of uniquely shaped and sized particles were simulated by using polyhedra. The first two packings resembled sand particles with aspect ratio of 1 and 3, and the third packing represented elongated woodchip particles. These packings show that it is possible to use the granular DEM to generate complex shapes that are all unique and have a specific size.

10. Acknowledgements

Appendix A. Global and local porosity measurements

A fundamental quantity of a random packing of particles is the solid volume fraction ε_s. The global average solid volume fraction in a macroscopic volume V_t containing N particles is defined as $\varepsilon_s = \sum_{i=1}^{N} V_{p,i}/V_t$, where $V_{p,i}$ is the volume of particle i. It is important to note that when a cylindrical vessel is only partially filled with particles, for V_t we take the vessel volume up to the filling height. The porosity (or voidage) is the complementary of the solid volume fraction, $\varepsilon_p = 1 - \varepsilon_s$.

It is important to also study the local porosity distribution because a significant amount of flow bypassing may occur in packed beds with heterogeneous porosity distribution. Particularly, at the walls of packed beds the local porosity values can reach a value of (almost) unity. To determine the local porosity, we introduce the point based porosity defined by Roblee et al. [51]:

$$\delta_p(\mathbf{x}) = \begin{cases} 0 & \text{if the point } \mathbf{x} \text{ is inside a particle,} \\ 1 & \text{if the point } \mathbf{x} \text{ is not inside a particle.} \end{cases} \tag{A.1}$$

This point porosity δ_p can be easily determined for any point in space $\mathbf{x}$. The GJK algorithm [32] used in our DEM code to detect particle collisions can handle the point-particle intersection query like any particle-particle intersection query. In fact, a point-particle GJK query will terminate with the outcome after a *single* iteration, making this query much faster than particle-particle intersection queries. With the definition of the point porosity, it is straightforward to determine the radial porosity profile $\varepsilon_p(r)$. The cylindrical vessel is divided into disks at discrete z-positions. In the plane of each disk, circles of increasingly larger radii are sampled with points. For all the points on a circle the δ_p is determined. Figure A.20 illustrates this procedure, where the grey points are inside a particle. With the points classified, it is possible to identify arc segments on the circle which are formed by grey points. Note that the smaller inner circle has a single arc segment that is defined by the grey points. The larger outer circle has three such segments of varying length. To get an azimuthal average solid volume fraction for a circle, the following formula is used:

$$\tilde{\varepsilon}_s(r = R, z = h) = \frac{\sum_i s_i}{2\pi R}. \tag{A.2}$$
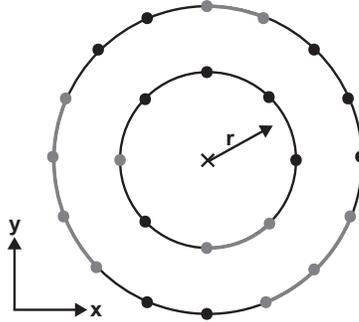
Figure A.20: Schematic representation of the circles used to determine the radial porosity profile at a certain z-height. Two circles are shown at two different radial distances r. The circles are sampled with points (in black and grey). The points that are inside a particle are colored grey. The line segments between grey points are considered to be inside a particle (grey line segments). The azimuthally averaged porosity of the small circle is 0.875 and for the large circle 0.625.

The length of the ith identified arc is denoted by s_i. The azimuthal average is still a function of the z-height, because a specific disc is being considered. To arrive at the final value $\varepsilon_s(r)$, we simply average all the azimuthally averaged values $\tilde{\varepsilon}_s$ over the height, making sure that the average is taken only over circles with the same radius. The radial porosity profile can then be obtained as $\varepsilon_p(r) = 1 - \varepsilon_s(r)$.

Appendix B. Orientational properties of random packings

Packings of non-spherical particles not only exhibit positional structure, but can also have orientational structure. In this section a number of orientational order parameters are introduced, which probe for different types of orientational structure.

When considering order, it is useful to first define a symmetry axis or *director* of the particles of interest. Three particle shapes are used, the *cuboid*, *cylinder* and *ellipsoid*. The aspect ratio is varied from 0.25 to 3, creating plates, coins and M&M's for the low aspect ratio and square rods, cylindrical rods and cigars for the high aspect ratios. The director $\mathbf{u}$ for all these particles is the minor axis for $AR < 1$, and the major axis for $AR \geq 1$. Since there is no essential difference between a particle characterized by a director $\mathbf{u}$ and a particle defined by the opposite director $-\mathbf{u}$, we are using orientational order parameters that are insensitive to the polarity of $\mathbf{u}$.

41

We define our order parameters with respect to a fixed cylindrical coordinate system. This is relevant because the vessel is cylindrical and ordering in the radial and z-direction is to be expected. The two cylindrical order parameters that are used are $S_{2,r}$ and $S_{4,r}$. They are essentially the second and fourth Legendre polynomials of the dot product of the particle director with the cylindrical basis vector $\hat{\mathbf{r}}$:

$$
\begin{aligned}
S_{2,r} &= \frac{1}{2} \left\langle 3(\mathbf{u}_i \cdot \hat{\mathbf{r}})^2 - 1 \right\rangle, \\
S_{4,r} &= \frac{1}{8} \left\langle 35(\mathbf{u}_i \cdot \hat{\mathbf{r}})^4 - 30(\mathbf{u}_i \cdot \hat{\mathbf{r}})^2 + 3 \right\rangle.
\end{aligned}
\tag{B.1}
$$

The averaging $\langle \ldots \rangle$ is carried out over all particles i.

The definitions of the above cylindrical order parameters can be used to find the radial profile of the cylindrical order parameters $S_{2,r}$ and $S_{4,r}$. This measurement can be done by first dividing the cylindrical domain into annular bins. For each bin the average $S_{2,r}$ and $S_{4,r}$ can be determined.

Similarly, the bins can be made increasingly larger when viewed from the cylindrical vessel wall. This yields the *cumulative* distribution of $S_{2,r}$ and $S_{4,r}$. The cumulative distribution varies between the near-wall value to the global domain value and gives insight into how fast this variation evolves.

When there is a tendency of particles to orient orthogonal to the radial coordinate $\hat{\mathbf{r}}$, it is interesting to see if there is any preferential orientation in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. To probe for this type of ordering, we introduce the angle α in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane. By varying alpha between $0°$ and $90°$ a distribution of the in-plane ordering can be made. To calculate these values the unit vector $\hat{\mathbf{a}}$ is needed. It is defined as a vector in the $\hat{\boldsymbol{\theta}}$-$\hat{\mathbf{z}}$ plane which makes an angle α with the $\hat{\boldsymbol{\theta}}$ axis. With this direction vector, we define $S_{2,\alpha}$ and $S_{4,\alpha}$:

$$
\begin{aligned}
S_{2,\alpha} &= \frac{1}{2} \left\langle 3(\mathbf{u}_i \cdot \hat{\mathbf{a}})^2 - 1 \right\rangle, \\
S_{4,\alpha} &= \frac{1}{8} \left\langle 35(\mathbf{u}_i \cdot \hat{\mathbf{a}})^4 - 30(\mathbf{u}_i \cdot \hat{\mathbf{a}})^2 + 3 \right\rangle.
\end{aligned}
\tag{B.2}
$$

References

[1] J. D. Bernal, J. L. Finney, Random close-packed hard-sphere model. ii. geometry of random packing of hard spheres, Discuss. Faraday Soc. 43 (1967) 62–69.

[2] T. C. Hales, A proof of the kepler conjecture, Annals of mathematics 162 (3) (2005) 1065–1185.

[3] T. C. Hales, J. Harrison, S. McLaughlin, T. Nipkow, S. Obua, R. Zumkeller, A revision of the proof of the kepler conjecture, in: The Kepler Conjecture, Springer, 2011, pp. 341–376.

[4] A. Donev, I. Cisse, D. Sachs, E. A. Variano, F. H. Stillinger, R. Connelly, S. Torquato, P. M. Chaikin, Improving the density of jammed disordered packings using ellipsoids, Science 303 (5660) (2004) 990–993.

[5] G. W. Delaney, P. W. Cleary, The packing properties of superellipsoids, EPL (Europhysics Letters) 89 (3) (2010) 34002.

[6] A. Wouterse, S. R. Williams, A. P. Philipse, Effect of particle shape on the density and microstructure of random packings, Journal of Physics: Condensed Matter 19 (40) (2007) 406215.

[7] L. Pournin, M. Weber, M. Tsukahara, J.-A. Ferrez, M. Ramaioli, T. M. Liebling, Three-dimensional distinct element simulation of spherocylinder crystallization, Granular Matter 7 (2-3) (2005) 119–126.

[8] P. A. Cundall, O. D. Strack, A discrete numerical model for granular assemblies, Geotechnique 29 (1) (1979) 47–65.

[9] B. Hoomans, J. Kuipers, W. Briels, W. Van Swaaij, Discrete particle simulation of bubble and slug formation in a two-dimensional gas-fluidised bed: a hard-sphere approach, Chemical Engineering Science 51 (1) (1996) 99–118.

[10] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, Powder technology 77 (1) (1993) 79–87.

[11] N. Deen, M. V. S. Annaland, M. Van der Hoef, J. Kuipers, Review of discrete particle modeling of fluidized beds, Chemical Engineering Science 62 (1) (2007) 28–44.

[12] K. A. Buist, L. Seelen, N. Deen, J. Padding, J. Kuipers, On an efficient hybrid soft and hard sphere collision integration scheme for dem, Chemical Engineering Science 153 (2016) 363–373.

[13] G. Lu, J. Third, C. Müller, Discrete element models for non-spherical particle systems: From theoretical developments to applications, Chemical Engineering Science 127 (2015) 425–465.

[14] X. Garcia, L. T. Akanji, M. J. Blunt, S. K. Matthai, J. P. Latham, Numerical study of the effects of particle shape and polydispersity on permeability, Physical Review E 80 (2) (2009) 021304.

[15] R. Gao, X. Du, Y. Zeng, Y. Li, J. Yan, A new method to simulate irregular particles by discrete element method, Journal of Rock Mechanics and Geotechnical Engineering 4 (3) (2012) 276–281.

[16] J.-F. Ferellec, G. R. McDowell, A method to model realistic particle shape and inertia in dem, Granular Matter 12 (5) (2010) 459–467.

[17] D. Höhner, S. Wirtz, H. Kruggel-Emden, V. Scherer, Comparison of the multi-sphere and polyhedral approach to simulate non-spherical particles within the discrete element method: Influence on temporal force evolution for multiple contacts, Powder Technology 208 (3) (2011) 643–656.

[18] J. R. Williams, A. P. Pentland, Superquadrics and modal dynamics for discrete elements in interactive design, Engineering Computations 9 (2) (1992) 115–127.

[19] P. W. Cleary, N. Stokes, J. Hurley, Efficient collision detection for three dimensional super-ellipsoidal particles, in: Proceedings of 8th international computational techniques and applications conference CTAC97, Adelaide, 1997.

[20] C. Wellmann, C. Lillie, P. Wriggers, A contact detection algorithm for superellipsoids based on the common-normal concept, Engineering Computations 25 (5) (2008) 432–442.

[21] R. Hart, P. Cundall, J. Lemos, Formulation of a three-dimensional distinct element model—part ii. mechanical calculations for motion and interaction of a system composed of many polyhedral blocks, in: International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, Vol. 25, Elsevier, 1988, pp. 117–125.

[22] C. W. Boon, G. Houlsby, S. Utili, A new algorithm for contact detection between convex polygonal and polyhedral particles in the discrete element method, Computers and Geotechnics 44 (2012) 73–82.

[23] A. Wachs, L. Girolami, G. Vinay, G. Ferrer, Grains3d, a flexible dem approach for particles of arbitrary convex shape—part i: Numerical model and validations, Powder Technology 224 (2012) 374–389.

[24] L. Landau, E. Lifshitz, Mechanics, vol. 1, Course of theoretical physics (1976) 84–93.

[25] L. Seelen, J. Padding, J. Kuipers, Improved quaternion-based integration scheme for rigid body motion, Acta Mechanica 227 (12) (2016) 3381–3389.

[26] F. Zhao, B. van Wachem, A novel quaternion integration approach for describing the behaviour of non-spherical particles, Acta Mechanica 224 (12) (2013) 3091–3109.

[27] C. Ericson, Real-Time Collision Detection, CRC Press, 2004.

[28] G. v. d. Bergen, Efficient collision detection of complex deformable models using aabb trees, Journal of Graphics Tools 2 (4) (1997) 1–13.

[29] M. Eitz, G. Lixu, Hierarchical spatial hashing for real-time collision detection, in: Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on, IEEE, 2007, pp. 61–70.

[30] E. J. Hastings, J. Mesit, R. K. Guha, Optimization of large-scale, real-time simulations by spatial hashing, in: Proc. 2005 Summer Computer Simulation Conference, Vol. 37, 2005, pp. 9–17.

[31] S. Cameron, A comparison of two fast algorithms for computing the distance between convex polyhedra, IEEE Transactions on Robotics and Automation 13 (6) (1997) 915–920.

[32] G. v. d. Bergen, A fast and robust gjk implementation for collision detection of convex objects, Journal of Graphics Tools 4 (2) (1999) 7–25.

[33] G. v. d. Bergen, Proximity queries and penetration depth computation on 3d game objects, in: Game developers conference, Vol. 170, 2001.

[34] H. Kruggel-Emden, E. Simsek, S. Rickelt, S. Wirtz, V. Scherer, Review and extension of normal force models for the discrete element method, Powder Technology 171 (3) (2007) 157–173.

[35] H. Kruggel-Emden, S. Wirtz, V. Scherer, A study on tangential force laws applicable to the discrete element method (dem) for materials with viscoelastic or plastic behavior, Chemical Engineering Science 63 (6) (2008) 1523–1541.

[36] J. Ai, J. Chen, J. Rotter, J. Ooi, Assessment of rolling resistance models in discrete element simulations, Powder Technology 206 (3) (2011) 269–282.

[37] A. Džiugys, B. Peters, An approach to simulate the motion of spherical and non-spherical fuel particles in combustion chambers, Granular matter 3 (4) (2001) 231–266.

[38] M. Ye, Multi-level modeling of dense gas-solid two-phase flows, Ph.D. thesis, University of Twente (2005).

[39] P. Costa, B. J. Boersma, J. Westerweel, W.-P. Breugem, Collision model for fully resolved simulations of flows laden with finite-size particles, Physical Review E 92 (5) (2015) 053012.

[40] L. Pournin, T. M. Liebling, A. Mocellin, Molecular-dynamics force models for better control of energy dissipation in numerical simulations of dense granular media, Physical Review E 65 (1) (2001) 011302.

[41] W. Zhang, K. E. Thompson, A. H. Reed, L. Beenken, Relationship between packing structure and porosity in fixed beds of equilateral cylindrical particles, Chemical Engineering Science 61 (24) (2006) 8060–8074.

[42] R. D. Batten, F. H. Stillinger, S. Torquato, Phase behavior of colloidal superballs: Shape interpolation from spheres to cubes, Physical Review E 81 (6) (2010) 061105.

[43] R. Ni, A. P. Gantapara, J. de Graaf, R. van Roij, M. Dijkstra, Phase diagram of colloidal hard superballs: from cubes via spheres to octahedra, Soft Matter 8 (34) (2012) 8826–8834.

[44] U. Agarwal, F. A. Escobedo, Mesophase behaviour of polyhedral particles, Nature materials 10 (3) (2011) 230–235.

[45] B. S. John, C. Juhlin, F. A. Escobedo, Phase behavior of colloidal hard perfect tetragonal parallelepipeds, The Journal of chemical physics 128 (4) (2008) 044909.

[46] L. Liu, Z. Li, Y. Jiao, S. Li, Maximally dense random packings of cubes and cuboids via a novel inverse packing method, Soft Matter.

[47] S. Li, J. Zhao, P. Lu, Y. Xie, Maximum packing densities of basic 3d objects, Chinese Science Bulletin 55 (2) (2010) 114–119.

[48] J.-P. Latham, A. Munjiza, X. Garcia, J. Xiang, R. Guises, Three-dimensional particle shape acquisition and use of shape library for dem and fem/dem simulation, Minerals Engineering 21 (11) (2008) 797–805.

[49] L. Wang, J.-Y. Park, Y. Fu, Representation of real particles for dem simulation using x-ray tomography, Construction and Building Materials 21 (2) (2007) 338–346.

[50] M. A. Taylor, E. J. Garboczi, S. Erdogan, D. Fowler, Some properties of irregular 3-d particles, Powder Technology 162 (1) (2006) 1–15.

[51] L. Roblee, R. Baird, J. Tierney, Radial porosity variations in packed beds, AIChE Journal 4 (4) (1958) 460–464.