Diploma Thesis

(Diplomarbeit zur Erlangung des akademischen Grades Master of Science der Studienrichtung Verfahrenstechnik an der Technischen Universität Graz)

Modeling of non-spherical particles in the Discrete Element Method (DEM) simulations

Peter Böhling

Advisor:

Univ.- Prof. Dr. tech. Dipl.-Ing. Johannes Khinast

Co Advisor:

Dipl.-Ing Toschkoff Gregor

Der Autor übernimmt keine Haftung für Schäden, die aus der Verwenduung dieses Dokuments und/oder eventuell auf einem Datenträger beigefügter Software entstehen.

Dieses Dokument enthält Verweise auf Websites, die von Dritten eingerichtet wurden. Der Autor hat keinerlei Kontrolle über die Websites und die dort angebotenen Informationen, Waren oder Dienstleistungen. Der Autor übernimmt daher keinerlei Verantwortung, aus welchem Rechtsgrund auch immer, für den Inhalt der Websites Dritter

Statutory Declaration

Declaration

The work presented in this diploma thesis is to the best of my knowledge and belief original, except as acknowledged in the text. This material has not been submitted, either in the whole or in part, for a degree at this or any other university.

Place, Date

Signature

Eidestattliche Erklärung

Ich erkläre an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Ich versichere, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Ort, Datum

Unterschrift

Abstract

The discrete element method (DEM) is used in engineering to simulate a wide area of processes: from storage and packing to mixing and even in the field of environmental particle physics for land slide probability computation. In pharmaceutical engineering, it is used for example to simulate the mixing of granular media or the behavior of tablets in a drum.

The first DEM simulations some 35 years ago were two-dimensional and used circles to discretize particle. With the rising computational power, the simulations got more and more complex. First, ellipses were introduced to simulate more sophisticated forms. Later, the simulations where expanded to the third dimension. The number of simulated particles also increased from a few hundred at the first DEM simulations to over 24 million spheres in state-of-the-art applications.

To emulate the real physical world in more detail, the particles have to become more complex. An important step is to move away from spherical particles. One way of making more sophisticated shapes is the multi element model; stacking together simple shapes to get more sophisticated particles.

In this work, we introduce the multi element model to the “eXtended Particle System”. The XPS program uses GPUs for the parallelization of the DEM method.

The implementation was tested to see if the model works as intended. Different tests were performed for evaluation, starting with particle - particle collisions and going to particle cluster behavior.

Table of Content

Figure table

Figure table

Abbreviations

API	Active pharmaceutical Ingridient
CAD	computer aided design
CAM	computer aided manufacturing

CFD	Computational Fluid Dynamics
CPU	central processing unit
DEM	Discrete Element Method
GPU	graphics processor unit
MSM	Multi Sphere Method
NP	number of Particles
RCPE	Research Center Pharmaceutical Engineering
XPS	eXtended Particle System

Nomenclature

a	acceleration
A	abitrary particle points
B	rotated abitrary particle points
c	center
C	number of contacts
c	contact number
d	distance
d	damping coefficient
d	distance of the center of the particle from the centroid
F	force
f	frequenz
f	force
g	gravitation constant
I	distance of the contact from the centroid of the particle
I	Moment of inertia
k	spring constant
m	mass

M	momentum
N	number
n	unit vector
n	number
R	radius of the particle
r	distance of the contact point from the center of the sub - particle
R	global rotated abitray particle points
r	restitution
R	resistance
S	number of spheres
s	sphere number
t	time
V	volume
v	velocity
X	center of mass on the X axis
x	center of the subparticle on the x axis
Y	center of mass on the Y axis
y	center of the subparticle on the y axis
Z	center of mass on the Z axis
z	center of the subparticle on the z axis
α	rotational acceleration
α	viscoelastic material property
Δ	difference
δ	overlap between the particles
η	damping ocefficient
θ	directional angle of the unit vector
ω	rotational velocity

Indices

a	particle describtion
ab	effective
ab	relative
b	particle describtion
c	coeffiecent
crit	critical
g	global
i	number of particle
l	local
N	of the time step
N	normalized
n	new normalized
N+1/2	of the next time step
N-1/2	of the before time step
nc	normal contact force
new	neu
nn	next neighbouring
p	particle
pc	particle contact
ps	sub particle
psc	contact of the sub particle
rel	relative
roll	rolling
t	tangential
tps	tangential component of the sub - particle
tpsc	tangential contact force of the sub - particle

X	x component of the vector
XX	x component of the vector
Y	y component of the vector
YY	y component of the vector
Z	z component of the vector
ZZ	z component of the vector

1. Introduction

Granular materials are commonly used in a wide variety of fields, like agriculture, chemical engineering, civil engineering, oil and gas engineering, mining, mineral processing and pharmaceutical engineering. They make up a large part of all natural explored resource [1], and are used widely in the pharmaceutical industry, for example in the form of powders, granules, or tablets. The understanding of the behavior of particle assemblies and the particle interactions are therefore critical for the correct execution of many pharmaceutical processes.

Handling and storage of granular matter is used widely in pharmaceutical engineering Suzzi, Radl, and Khinast [2]. The processes which work with granular matter range from drying, spraying granulation, mixing, transportation, silo storing and emptying.

These processes can be modeled using the Discrete Element Method. For modelling the process the process parameters have to be known. For this the particle shape is one of the most eminent. Holubec and D'Appolonia [3] showed in experiments with sand that a varying shape could have different mechanical behavior due to angularity. This is even more a concern in the pharmaceutical Industry.

From a modeling point of view, the particles interact with each other following Newton's laws of motion. Based on this, Cundall and Strack [4] first presented the "Discrete Element Method" (DEM) as a way to computationally calculate and analyze the behavior of particle assemblies. Nowadays, this method is widely used for particle system simulation. In their work, Cundall and Strack used circles in a two-dimensional simulation to save computational effort and time. With the increasing computational power of the last decades, efforts were made to make the simulations more realistic.

First uses for the DEM were periodic cells and chute-flows, and also small hoppers and shear cells, mostly in two dimensions[4]. Later, ball mills and hoppers were introduced. In this time, the number of simulated particles has risen from some hundreds [4] in the beginning of the DEM to the 10.000 -100.000 range for two dimensional models in the nineties to now regularly 100.000 in three dimensions, going up to a few millions for special cases (Chung, Liao, and Hsiau [5] Adam et al[6] and Radeke, Glasser, and Khinast [7]). DEM is nowadays used widely to understand the behavior of particle flow. Cleary [8] gives examples for applications of the DEM software, such as separation technique (sieving twin deck banana screen), mixing, excavation, transfers on conveyor belts, commination, crushing grinding(for example in a SAG mill). Further uses are the simulation of landslides and the flow of particles in a fluid. Zhu et al [9] mention that the DEM is used to study the behavior of chopper discharge, or the particle flow in a mixer. They also investigate the behavior of particles in a drum or mixer. Freireich, Litster, and Wassgren [10] used it to measure and predict the collision frequency and the impact velocity distribution of spheres.

Today, spheres are widely used in the DEM. Using spheres implies some simplifications. For example, the particle rotation is only diminished by frictional forces. To model translational and rotational forces arising due to a non-spherical shapes, the simulated particles have to be modeled more accurately. In the early two-dimensional days of DEM simulation, circles were replaced by polygons (Walton [11]). Later, in three dimensions, spheres and polyhedral were used Cundall [12] and Hocking [13]. Contemporaneous to polygons, Ting et al. [14] used ellipse in two dimensional cases, which Ng and Lin [15] expanded to ellipsoid for three dimensional cases. In addition, super quadric functions were introduced by Williams and Pentland [16]. All these method have advantages in describing the particle more accurate, but lack in accuracy of the contact detection and cost a lot of computational power.

One promising way of modeling curved surface particles is the multi element model, also known as glued sphere approach or stacked sphere approach It was implanted successfully by Favier, J.F. et al.[17] and Abbaspour-fard [1]. In this model, a real particle is replaced with spheres of constant or varying sizes. They can overlap, and are arranged in such a manner that they resemble the surface of the real particles. The spheres are locked in position relatively to each other. The main advantage of this model is that the contact detection still uses the simple sphere algorithm (basically comparing the positions and radii of the spheres) and is therefore computationally efficient. To calculate the total force acting on the multi element particle, the forces and momenta acting on every single sphere is calculated and transferred to the center of mass of the model particle. The accuracy of the simulation of such a modeled particle depends heavily on choosing the correct parameters of the contact force models. Höhner et al. [18] validated the multi sphere and polyhedral approaches for modeling non spherical particles and concluded that both have their advantages, but pointed out that it is important how the contacts are calculated to get good results in the simulation. Markauskas et al. [19] investigated the adequacy of the multi sphere approach to model elliptical particles and also concluded that the multi element method is suitable for most applications. The multi sphere approach is used successfully in the pharmaceutical process engineering to study the behavior of tablets in a drum as shown by Toschkoff et al. [20], [21]. Here, the motion of the tablets differs heavily from the motion of a single sphere. Abbaspour-Fard [22] tested the validity of the multi-sphere approach for biomaterials and phenomena like sliding friction or rotation. . Kruggel-Emden et al. [23] studied the validity of the multi-sphere approach in modeling a master sphere and comparing the results with an actual sphere as a model particle. A special case related to the multi-sphere approach is described by Song, Turton, and Kayihan [24], where three intersecting spheres are used to describe the tablet form.

In this work, the implementation of the multi-sphere approach in an existing massively parallel DEM software is investigated. The used DEM software is termed “eXtended Particle System (XPS)”, and is developed in-house at the Research Center Pharmaceutical Engineering. The special characteristic is that it uses graphical processor units (GPU) for massive parallelization [7]. For this, XPS takes advantage of the special architecture of the GPU and the CUDA programming language (a C extension developed for nVidia GPUs). This has been shown to be very efficient in parallelizing the algorithms used in the Discrete Element Method, ([7]). It is also possible to couple the XPS software to a commercial computational fluid dynamics (CFD) software AVL FIRE, AVL List GmbH Graz, Austria) to simulate particle fluid interactions.

The work is organized as follows: first, investigations on how the multi sphere model can be implemented into the existing XPS code were done, and the force and moment calculation will be tested. Second, models for the rotation were evaluated to see how it can be implemented. Third, test cases were performed using the new version of the DEM code. Finally, the created multi element particles were used in more sophisticated test runs and the results were compared to the EDEM one of the most used commercial DEM software.

2. Discrete Element Method Simulations

2.1. Application in Particulate Systems

A lot of the understanding of granular processes is superficial and the design of the processes consists mostly on assumptions based on macroscopic, phenomenological descriptions. These processes are often studied in lab-scale setups and are then extrapolated to the industrial scale. This process is error-prone and can only make limited statements on the real behavior in the industrial scale, be it for agriculture products, mining, or pharmaceutical ingredients. In the pharmaceutical industry, it is especially important to understand the behavior of particles like powders and tablets. Because the active pharmaceutical ingredients (API) has to be equally distributed and mixed with the bulk chemicals to get a homogenous mixture. This was previously done in experiment to get the ideal parameters. Nowadays, the ability arises touse a combination of experiment and DEM, resulting in fewer experiments, reduced cost, and better understanding of the processes. To improve granular processes, real understanding of the underlying laws on a micro- and mesoscale are important. Experiments with the granular material can give some insights of the processes, but often do not reveal the whole process. The Discrete Element Method (DEM) can be applied to get a better understanding on what is going on inside the bulk phase and during the whole process. The Discrete Element Method

Today, the DEM is used as a tool to model a lot of different processes involving particular matter. Modeled processes include conveyor belt transportation, silo filling storing and discharge. For pharmaceutical processes like tablet coating the DEM simulation is used to optimize process conditions and parameters.

2.2. The Discrete Element Method

The label “Discrete Element Method” includes a set of numerical methods, all of which are used to compute the motion and behavior of a large set of particles. It is closely related to molecular dynamics simulation.

The DEM was first introduced by Cundall and Strack, [4], building on an earlier work of Cundall on the behavior of soil. At first, it was developed for two dimensional simulation of hopper discharge and conveyor belt transport. To describe the particle, circles where used, mainly due to the low computational power available at that time and the simplicity of contact detection between circles. Their model had 197 two dimensional discs of different sizes to simulate the behavior of sand. The important aspects of the implementation were the incorporation of particle rotation during simulation. They use a linear force displacement and the inclusion of friction as well as global and local damping to dissipate energy.

With rising computational power, more complex models and more particles were simulated. Where the first DEM simulations described simple systems with a few hundred particles in two dimensions, later more sophisticated things, like complex particle shapes and more complicated processes with more particles, were modeled. Ellipse particles were first introduced by Ting et al [14]. Then the third dimension was introduced into the DEM simulation to model the process even more accurately. Spheres were first used by Cundall [12] later the ellipse were extended to ellipsoids Ng and Lin [15], and super quadrics Williams and Pentland [16] functions were applied model even more complex particle shapes.

A particle assembly is discretized in the DEM method as an assembly of discrete model particles. The shape of the modeled particle should match the shape of the actual particle as closely as possible. Every deviation from the real system will result in an error which will get bigger with time and size of the variation.

In the following chapters the process used in the DEM is shown, starting by the calculation cycle over the contact representation to the contact models and force calculation.

2.3. Calculation cycle

There are four main steps in the discrete element method which are run in every calculation cycle. Those are:

- Particle and environment generation

Here the particle position, velocity and other properties of the individual particles are retrieved from the previous step (or starting condition for the first step). This is shown in Figure 2-1

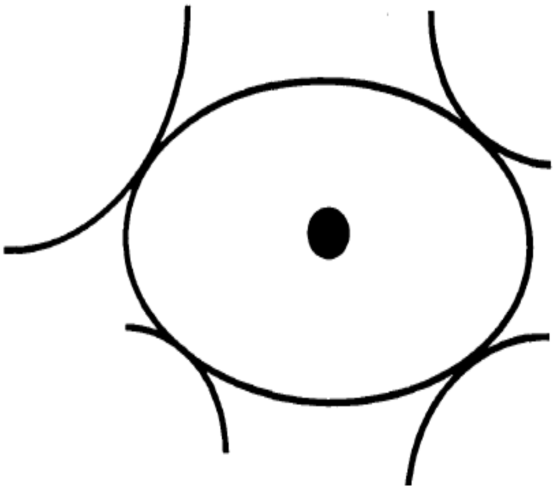

Figure 2-1: Enviroment generation[1]

- the contact detection

To reduce the contact search to a minimum, the global geometry is divided into cells, each two to three times the size of the typical particle. Then, a next neighbor sorting is done and the actual contact detection starts. This depends heavily on the geometry of the simulated particles. For spheres, it is simply a comparison of the distance between the centers and the radii of the involved spheres. More complex shapes need more elaborate contact algorithms. This step is shown in Figure 2-2.

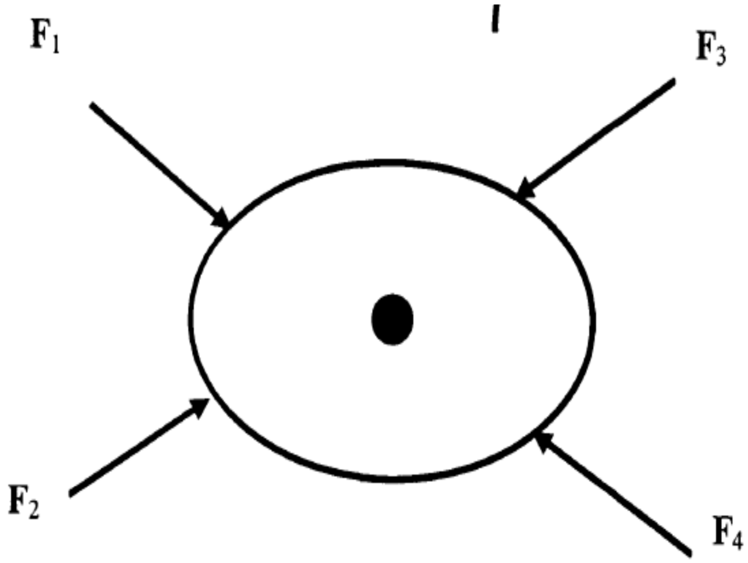

Figure 2-2: Force calculation for each contact[1]

- the force calculation

The force and momentum acting on the particles centroids for every particle is calculated using the models described in later chapters. These range from sophisticated force calculations to simple damper feather spring systems. The force calculation is shown in Figure 2-2 and the summation in Figure 2-3.

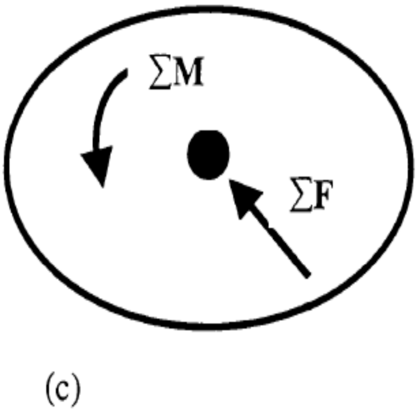

Figure 2-3: Momentum and force summation [1]

- the velocity/acceleration update

The gathered forces and momenta are used to calculated the new accelerations and with that the new velocity and position of the particles. After this step, the cycle is repeated until the final time step is reached. The velocity calculation is shown in Figure 2-4.

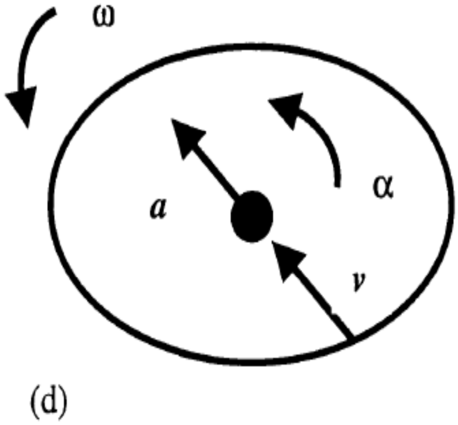

Figure 2-4: Resulting velocity and rotational velocity[1]

2.4. Contact representation

The contact between two particles was studied in detail by Hertz, who developed a theory for the interactions of frictionless convex elastic objects [25]. Hertzian objects are homogenous, perfectly smooth and elastic. Mindlin used Hertz' model and extended it to include contacts between frictionless elastic bodies in the absence of slip for spheres of different sizes. To this day, a commonly used contact model is the Hertz-Mindlin model; it is the standard contact model in the commercial software EDEM (DEM Solutions, Edinburgh, UK).

A number of parameters have to be considered to model the contact between particles. These parameters typically describe the particle properties and the contact problem. For the calculation, the contact point of one particle in an assembly with one or more of their neighboring particles at their boundaries has to be considered. In reality, the contact is over a finite area of the contacting bodies. In the DEM, this is simulated as a contact in a single point. This was shown to be sufficient by Hart, Lemos [26] and Cundall and Hart [12]. All forces are located at this contact point. The average number of contact points per particle is called the coordination number of the particle assembly. The contact normal n is defined as the outward normal vector of the tangential plane at the contact point between contacting particles.

For a particle of arbitrary irregular shape, the contact point is defined with reference to the centroid of the particle. For each contact point *i*, a contact vector $\boldsymbol{l_i}$ is defined from the centroid of the particle to that contact point. In the simplest and most used case of spherical particles, the length of the contact vector is always equal to the radius.

Contact between two particles will result in the displacement of the two particles relative to each other, and in deformation occurring at the contact point. The contact model is described in more detail in chapter 2.5. This is shown in Figure 2-5 where in the first time step no contact is detected and the particles move with a velocity to collide in the next time step. In the second time step the overlap is shown as δ in the figure. The third time step shows the resulting force of the contact between the two particles.

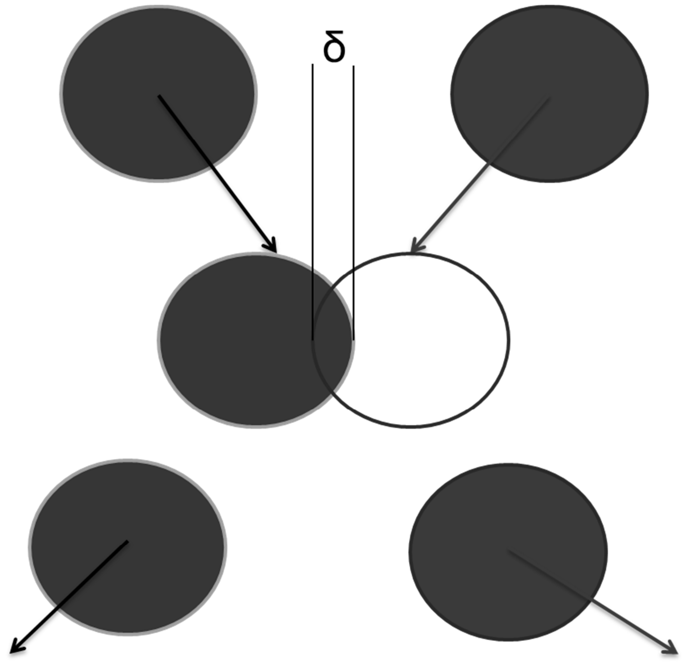

Figure 2-5: Soft sphere approach of two colliding particles, these are done in three time steps.

The tangential component of the force between two particles is limited by the amount of friction. If the tangential force reaches the maximum possible value (given by the coefficient of static friction), sliding occurs. In this case, the tangential force stays constant.

The soft sphere approach is described in the next sub-chapter and additional the hard sphere approach is explained. In this work we use the soft sphere approach

2.4.1. The soft sphere approach

In the soft sphere approach, the particles are assumed to deform at the contact point as a result of friction and stress. The normal stiffness is used to represent the relationship between the deformation and the magnitude of contact force. The Hertz theory is an example of this approach and is used to calculate the stiffness of the particle for two deformable elastic particles in contact. An arbitrary normal stiffness could also be used. In this approach, the overlap of the two particles is assumed to be small relative to the particle dimension. This approach is most suitable for handling particulate materials such as tablets and powders.

2.4.2. The hard sphere approach

The hard sphere approach assumes that no deformation is taking place, that is, an immediate inter particle collision occurs Walton [11]. The velocities and positions of particles after the collision are determined according to their initial condition and the rules governing two-body collision such as instant momentum exchange. Energy may or may not be conserved in the system. This approach is valid e.g. for molecular dynamic situations.

2.5. Contact model implementations

In the DEM, the deformation is not included directly. Instead, the particles are assumed to overlap. The resulting force is proportional then a function of the overlap of the particles. The contact force is split into a normal and tangential force component. From the tangential component, the moment resulting from the tangential part of the total force is calculated.

In an assembly of granular matter, energy can be dissipated to allow the system to reach a quasi – static equilibrium. During particle collision, the energy is dissipated through friction and/or deformation of the particles. Without energy dissipation, a particular system could never reach equilibrium, for example in the case of pure elastic behavior. In viscoelastic materials on the other hand, the energy is absorbed due to deformation at the contact point. Most particles show viscoelastic behavior and a contact model should take the energy dissipation into effect.

To calculate the contact force, the DEM model should consider the relationship between the contacting particles including contact stiffness coefficient and energy dissipation. The use of the right contact law and parameter is crucial for proper prediction of the DE model. The contact model consists normally of a spring and a dashpot in parallel and tangential direction and is known as Kelvin Voigt model. This model is shown in Figure 2-6.

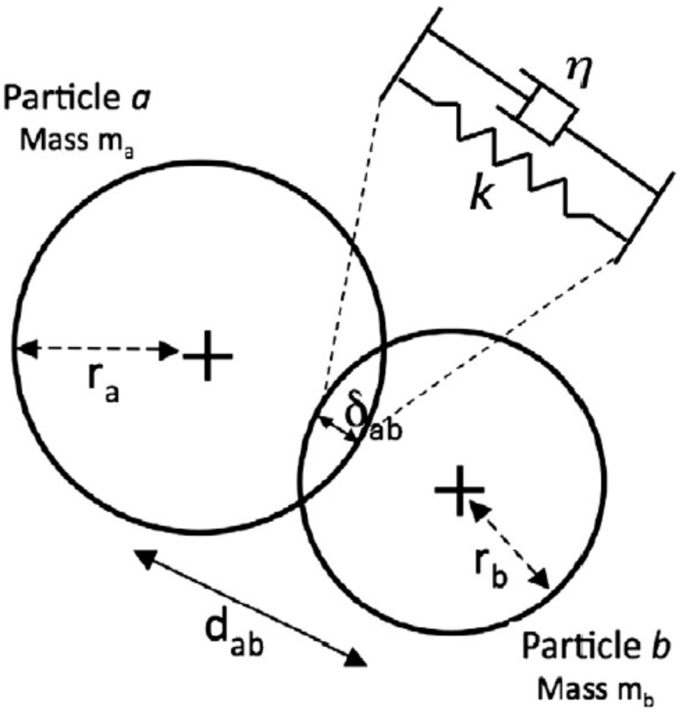

Figure 2-6: Kevin Voigt feather damper system of contacting spheres [27]

2.6. Contact Detection

To detect if two neighboring particles are in contact, at first a nearest neighboring sorting is used to find the particles next to the particle of interest. The whole simulation volume is divided in cells. This is shown in more detail in chapter 2.9. Without this, the amount of steps to check if two particles have contact for n particles would be ~ n^2; with the nearest-neighbor approach, the detection can be reduce to the number of next neighbor ($n{*}n_{NN}$) particles in the neighborhood which is ideally a lot smaller than n^2.

After the next neighbor sorting, the distance between the next neighbors is calculated. For spheres, the radii are added, and the sum is compared to the length of the real distance vector. From this, the overlap follows as well, and is used for the force calculation. The distance of the particles in a system is defined over their distances of centers:

$$d = c_a - c_b, \tag{1}$$

d is the distance and $c_{a/b}$ is the center of the respective spheres. The minimal contact distance is given by the addition of the two radii:

$$d_{contact} = R_a + R_b. \tag{2}$$

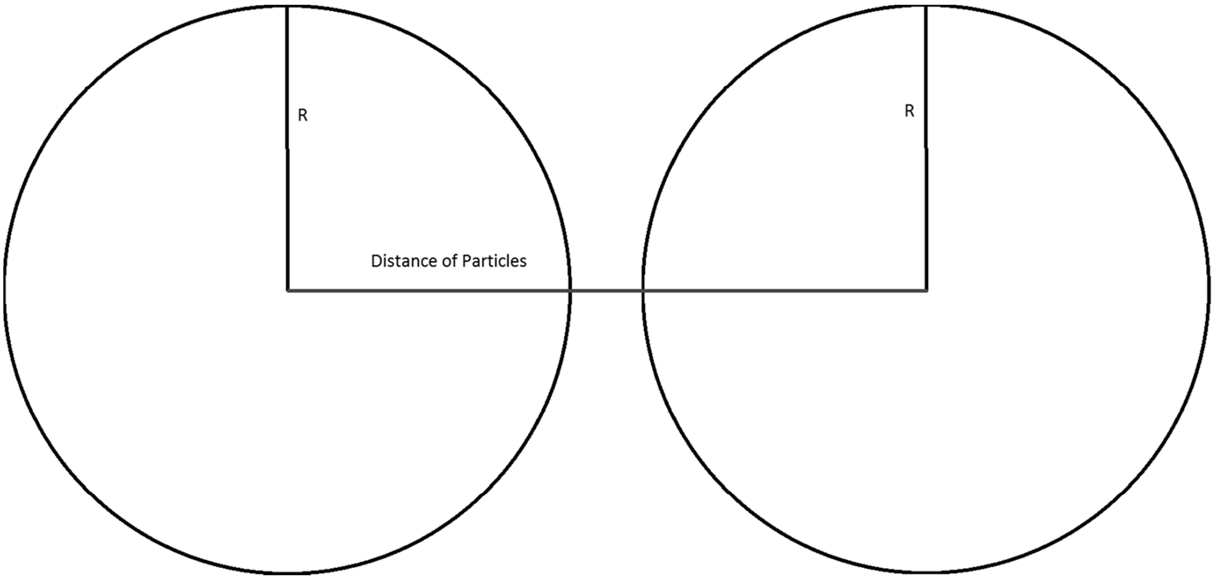

Figure 2-7: Contact detection of two discs.

For more sophisticated particle shapes, the contact detection gets more complicated in the sense that the orientation and extension in different dimensions has to be kept in mind. Different approaches for this kind of particles are used; a detailed description is out of the scope of this work.

2.7. Force Calculation

After the contact between particles is detected, the overlap is calculated and used to determine the resulting force. The force is divided into two components. The first is a force normal to the contact plane, going through the particle center and thus acting directly on the particle. The second is a tangential force (perpendicular to the normal force) which is used to calculate the momentum acting on the center of the particle. The total momentum is used to calculate the rotation of the particle.

For the calculation of the normal force in the soft sphere approach, normally a combination of springs and damps is used (see above):.

$$F = m \cdot d + m \cdot k. \tag{3}$$

Here *F* is the force, *m* is the mass of the particle, *d* is the damper constant and *k* is the spring constant. This is the simplified form of the spring damper equation. The models used for the normal and tangential force are described in chapter 4.3.

2.8. Time step

The time step in the DEM is the discrete time between two points in time for which all forces acting on the particle are calculated. The whole method is based on the assumption that the time step is small enough so that new contacts that take place in the current time step can be neglected in the sense that only those contacts that have already been recognized at the beginning of the time step are taken into account. Also the velocity and acceleration of the particles are assumed to stay during each time step. Thus, forces for a particle at the end of each time step are the result of the contact forces arising solely from the contacts particles shared in the current time step.

The time step should be as large as possible to increase the efficiency and speed of the simulation, but still be smaller than the critical time step to justify the assumption of constant acceleration within each time step and to ensure stability of the calculations Cundall and Strack [4].

The critical time step is calculated from the frequency of a single degree of freedom in the system of particles with the mass element connected to the ground. For linear contact models, the critical time step is based on the natural frequency of a linear spring system:

$$f = \sqrt{\frac{k}{m}} \qquad (4)$$

Here k is the spring constant and m is the mass of the individual particle. This shows that the critical time step depends on the smallest particle mass and constant contact stiffness and is calculated according to

$$t_{crit} = 2\pi \cdot \sqrt{\frac{k}{m}} \qquad (5)$$

This only counts for linear spring systems, for viscolelastic materials the damping has to be taken into account and the critical time step is

$$t_{crit} = \frac{\pi}{\sqrt{\frac{k}{m} \cdot \left(1 - \frac{\ln(e)^2}{\ln(e)^2 + \pi^2}\right)}} \qquad (6)$$

e is the coefficient of restitution. In this work a linear force system is used.

2.9. Grid cell generation

To save computational effort and wall clock time, the virtual space is divided into cubic cells. All cells have the same size; with a side length of two to three times the size of the particle diameter. The grid cell is used to reduce the amount of particles which have to be checked for contact: each particle can only be in contact with particles in the cells around the home cell of the particle. In two dimensional cases nine cells are to be searched for contact in three dimensional cases this leads to 27 cells. This can be applied manual but can also be automated for a given radius and particle position.

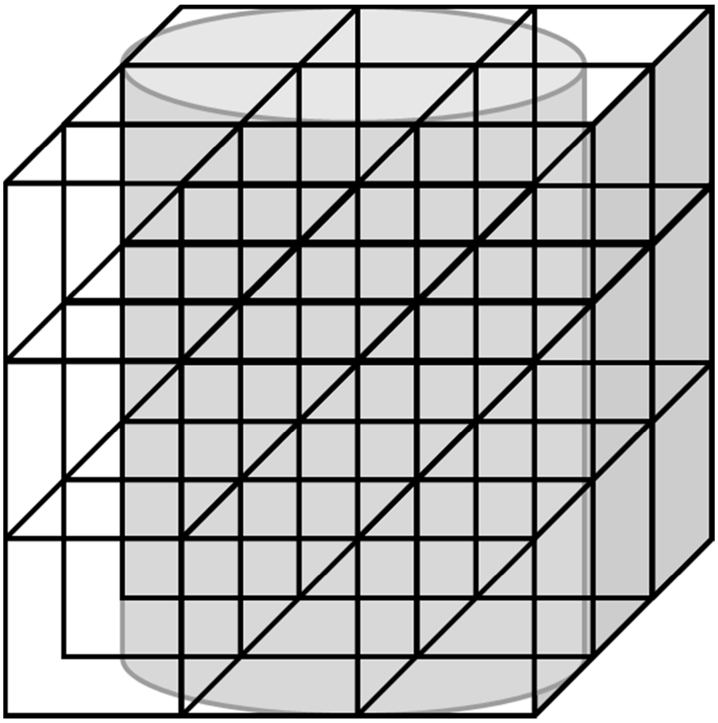

Figure 2-8: Cylinder in a DEM simulation divided in grid cells

It is important that the gird cells have the right size. When the grid cells are too small it is possible that contacts are overlooked or recognized to late and the resulting force is too big. When the grid cells are too big all contacts are detected but the simulations slows down due to the unnecessary contact detection steps.

3. The Multi-Sphere Approach

Most DEM simulations use spheres to simulate particle movement and interactions. This is due to the simplicity of contact detection and therefore low computational effort compared to other forms of shape representation and contact detection mechanisms. On the other hand, this simplicity and computational cheapness also neglects a lot of interactions, particle-particle or particle-geometry wise. These cannot simply be described by spherical particles. Examples are blocking and bulking while unloading/ loading of containers, or movement of tablets in a rotating drums.

Here, a method of particle shape representation using overlapping spheres to create composite non spherical particles is implemented. The positions of a number of spheres relative to each other can be defined to approximate almost every arbitrary shape Figure 3-1.

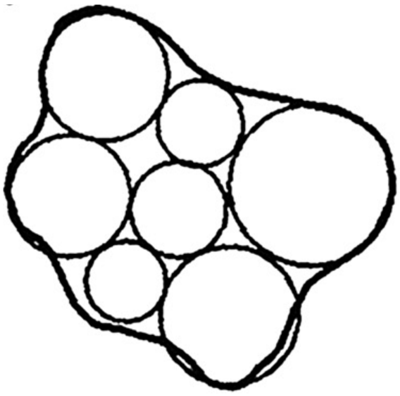

Figure 3-1: Shape approximations with circles in two dimensions [28]

The method used in this thesis is termed Multi Sphere Method (MSM), a means to model contacts between three dimensional particles constructed of overlapping spheres. For simplification, at first only axi-symmetrical particles, e.g. cylinders or biconvex tablets, are taken into account. The MSM is based on the constructive solid geometry technique for construction of complex solids by combining primitive shapes, in this case spheres. This technique is used in computer graphics to model complex objects as a combination of simpler shape. Applications are e.g. CAD/CAM software. Any number of elements can be used to change the shape of the particle or to refine the model. In theory every primitive shape (e.g. boxes, cuboids, cone), could be used to create such objects. But in the DEM, the goal is not only to represent the shape of the particle, but to detect contacts between different shapes. For each used primitive shape in the multi element particle, an own contact detection approach would be needed. Concentration on only one primitive shape, e.g. sphere, largely reduces this complexity. As it is the simplest shape, a good contact detection technique will reduce overall computation time compared to more sophisticated particles. However, it has the disadvantage that sharp corner cannot be represented with this method.

3.1. Multi sphere model particle construction

Particles consisting of multiple spheres, where the spheres may overlap and vary in diameter along the particle main axis, can be used to describe the surface of non-spherical particles. This can be done for example by combining spheres in such a way that the surface of the particle tangents the surface of each sphere or that the spheres describe the silhouette but are allowed to reach over the boarder of the particle shape. The position of each sphere is fixed relative to each other.

To what degree the surface of the original particle is captured depends on the number of element spheres used. Increasing the number of spheres per particle increases the computational time because the contact detection and force calculation has to be done for every sphere in the particle. For a multi element particle consisting of *n* spheres, the number of steps for contact detection is

$$N_{steps} = n_{spheres\ in\ the\ shape} * n_{\text{number of spheres in neighbouring particles}} * n_{nn}$$

This means that the number of spheres in a particle has to be a compromise between accuracy and speed of the simulation. This influences the error in the calculation of the contact forces for the particle. By using overlapping spheres, a pseudo friction is created when intending to describe a smooth surface. When the number of spheres is increased, this is reduced in comparison to fewer spheres.

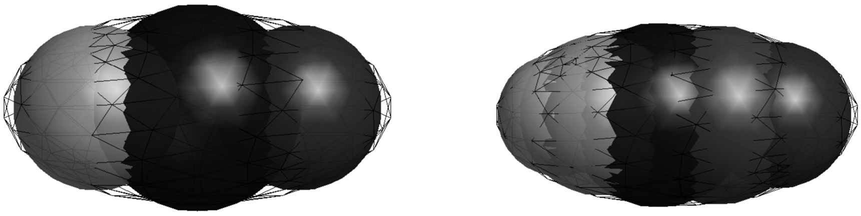

Figure 3-2: Two approaches of describing the ellipsoid, with three and five spheres[29]

This is shown in Figure 3-2 where the first approximations would create much more pseudo friction than the second shape.

3.2. Determination of the centroid

The particle centroid (center of mass) is required to update acceleration, velocity and displacement of the particle. For a geometrically simple shape such as spheres and ellipses, the position of the centroid is readily known. For axi-symmetrical irregular shaped particles created by the multi-sphere approach, the particle centroid and subsequently the relative position between the particle centroid and the centers of the element spheres needs to be determined in the first simulation step. As the body of a multisphere consists of a number of spherical segments, the principle of Varignone (principle of moments) can be used to calculate the position of the centroid:

$$\bar{X} = \frac{\sum_{i=1}^{n} V_i \cdot \bar{x}_i}{\sum_{i=1}^{n} V_i}, \bar{Y} = \frac{\sum_{i=1}^{n} V_i \cdot \bar{y}_i}{\sum_{i=1}^{n} V_i} \text{ and } \bar{Z} = \frac{\sum_{i=1}^{n} V_i \cdot \bar{z}_i}{\sum_{i=1}^{n} V_i} \qquad (7)$$

$\bar{X}, \bar{Y}, \bar{Z}$ are the coordinates of the particle centroid, with respect to the model particle axis. $\overline{x\,}, \bar{y}, \bar{z}$ are the known coordinates of the individual spheres with respect to the reference axis. V is the volume of each sphere and n is the number of sub - spheres in the particle.

3.3. Contact Mechanics and particle kinematics

Spheres in the model particles do not act independently on their own, but are rigidly connected to each other. Only the exposed part of a sphere can make contact with other particles, since non exposed sphere segments are positioned inside other element spheres and shielded from possible contact. The forces acting on each of the spheres are transferred to the center of mass of the particle. The contact detection and force calculation for a sub - spheres is the same as for a single sphere body. The forces and momenta generated at the contact points are then transferred to their center. In the following, it is described how the force is calculated and the particles behave as conglomerates.

3.3.1. Force and moment on a Multi Sphere Particle (MSP)

A model particle consisting of two spheres of the same size is considered, with the distance of the spheres to the centroid equal to the radius of each sphere. The model described is from Faviers paper for multi Element particles [17] and from Abbaspour PhD Thesis [1]. The position $\mathbf{r}_{\mathrm{pG}}$ of the center of mass (com) of each sphere is known, as are the distances $\mathbf{d}_{\mathrm{ps}}$ of the spheres from the com of the model particle. The position of the contact point **c** on each element sphere relative to the com is

$$\vec{\boldsymbol{I}}_{pc} = \vec{\mathbf{d}}_{ps} + \vec{\boldsymbol{r}}_{psc}. \quad (8)$$

$\mathbf{r}_{spc}$ defines the vector going from the contact point to the center of the element sphere of the particle.

Contact detection between multi sphere particles is the same as the contact detection of a single sphere, but done for every sub - sphere. A contact is detected if the distance between two spheres is smaller than the sum of their radii. Contacts between spheres of the same particle are ignored.

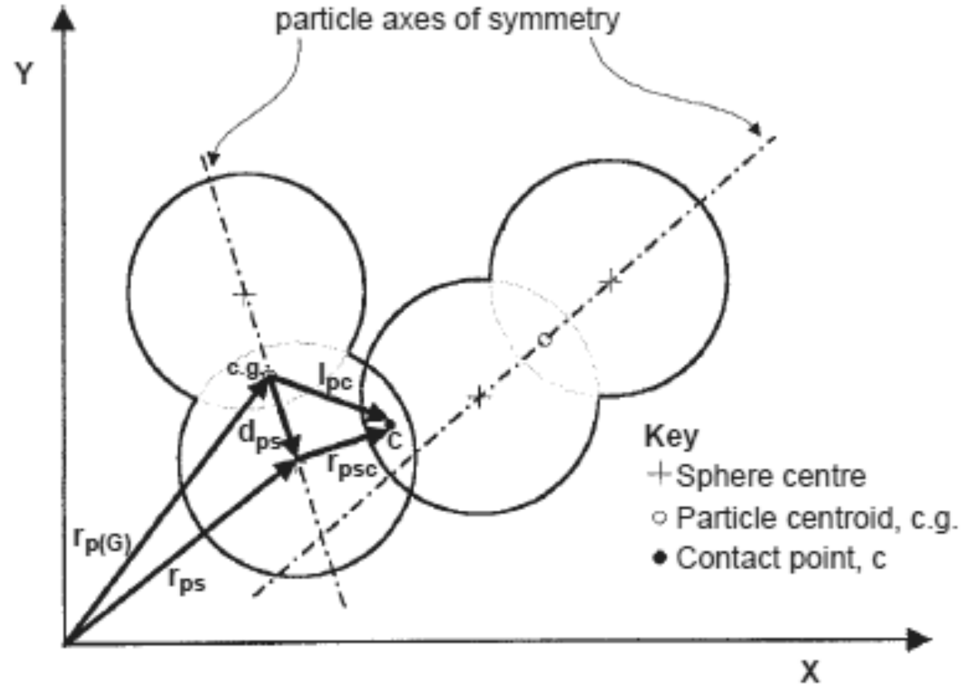

Figure 3-3: Two multi element particle colliding[17]

Contact forces are determined by the force displacement law for normal and tangential contacts from the normal and tangential force components:

$$\mathbf{f}_{psc} = \mathbf{f}_{nc} + \mathbf{f}_{t_{psc}} \quad (9)$$

With $\mathbf{f}_{psc}$ as the contact force, $\mathbf{f}_{nc}$ as the normal force and $\mathbf{f}_{t_{psc}}$ as the tangential force on the sub - sphere. The moment acting on the sub - sphere is calculated from the tangential force $\mathbf{f}_{t_{psc}}$ and the distance of the contact point $\mathbf{r}_{spc}$.

$$\mathbf{M}_{t_{ps}} = \sum_{c=1}^{C} \left(\mathbf{r}_{psc} \times \mathbf{f}_{t_{psc}} \right) \quad (10)$$

$\mathbf{f}_{t_{psc}}$ is the tangential force resulting from the c-th contact on a sub - sphere. $\mathbf{r}_{scpc}$ is the distance of the contact. $\mathbf{M}_{tps}$ is the tangential momentum of the sub - sphere. *C* is the total number of contacts acting on the element sphere. The total force acting on a –sub-sphere is the sum of the contact forces acting on it:

$$\mathbf{f}_{ps} = \sum_{c=1}^{C} \mathbf{f}_{psc} \tag{11}$$

The force is transferred to each element sphere at the current time step as shown in Figure 3-4. How the momenta and forces from the sub - spheres are transported to the center of the particle is shown in Figure 3-4.

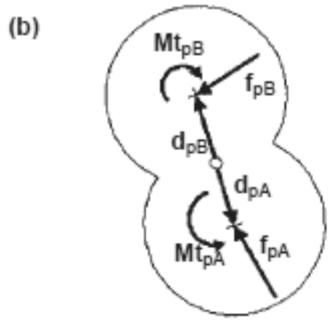

Figure 3-4: Moment and force transfer from the sub - spheres to the center of the particle [17]

The total moment generated by the forces is the sum of the total force acting on the sphere center and is the addition of moment of the single spheres and the total forces acting on the center of mass.

$$\mathbf{M}_{p} = \sum_{s=1}^{S} \left[\left(\mathbf{d}_{ps} \times \mathbf{f}_{ps} \right) + \sum_{c=1}^{C} \left(\mathbf{r}_{rspc} \times \mathbf{f}_{t_{psc}} \right) \right] \tag{12}$$

S is the total numbers of spheres in the particle. The total force acting on the particle is the sum of the contact forces acting on its element spheres

$$\mathbf{f}_{p} = \sum_{s=1}^{S} \mathbf{f}_{ps} \tag{13}$$

How the momenta and forces from the sub - spheres result in the total force is shown in Figure 3-5.

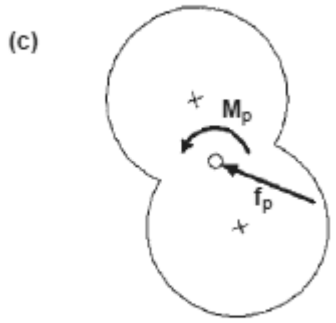

Figure 3-5: Total moment and force resulting from Figure 3-4 [17]

3.3.2. Translational and rotational velocity

When the total force and moment is known, the translational and rotational acceleration can be calculated according to Newton's second law.

$$\mathbf{a}_p = \frac{\mathbf{f}_p}{m_p} + \boldsymbol{g} \tag{14}$$

$$\boldsymbol{\alpha}_p = \frac{\mathbf{M}_p}{\mathbf{I}_p} \tag{15}$$

m_p is the particle mass, $\mathbf{a}_\mathrm{p}$ the particle acceleration and $\boldsymbol{\alpha}_\mathrm{p}$ is the rotational acceleration. Equation 16 applies only to spheres. To calculate the rotational motion of non-spherical particles, one has to calculate the mass moment of inertia with respect to the global axes. Ignoring the mass moment of inertia will lead to significant errors in calculating the rotational acceleration. This is true for all non-spherical particles.

The Euler equations of motion of a body in space describe the rigid body motion relative to the principle axes of inertia of the body (Hart, Lemos [26]) For this, the principle axes of the particle, the principle mass moments of inertia and the exact position and orientation of the particle has to be known for each time step in relation to the global axes.

Favier et al. [17] presented a solution for updating the rotational acceleration of a multi sphere particle. First, the total applied moments on a particle are transferred from global coordinates to local coordinates in each time step using the rotation Matrix ***R***. This matrix was determined by comparing the current position of three arbitrary particle reference point **A1**, **A2**, and **A3** in the local coordinate frame of a particle to the global coordinate frame. The vectors have to be linearly independent, so that their determinant is not zero. The matrix ***R*** is computed each time step from the rotated reference points compared to their origins:

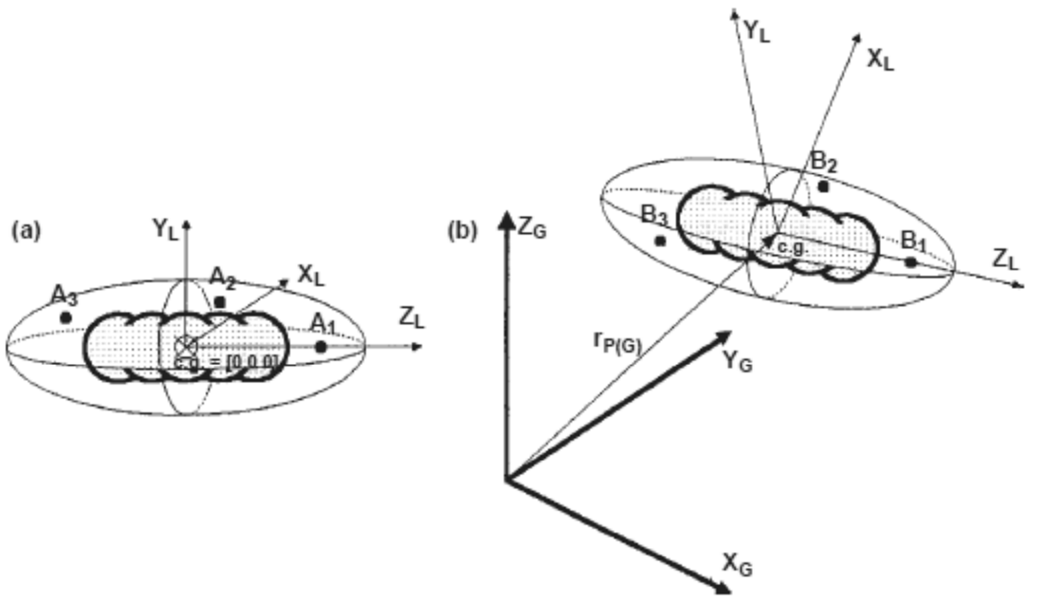

Figure 3-6: Reference points A1, A2 and A3 in the particle coordinate system (left) and the same in the global coordinate system (right) [17]

$$[\boldsymbol{R}] = [\boldsymbol{A}][\boldsymbol{B} - \mathbf{r}_{P(G)}] \tag{16}$$

$[\boldsymbol{B} - \mathbf{r}_{P(G)}]$ is the matrix of the rotated reference frame. The local moment is

$$[\mathbf{M}_L] = [\boldsymbol{R}][\mathbf{M}_G] \tag{17}$$

$\mathbf{M}_G$ is the total momentum acting on the particle.

With the Euler equations, the local rotational accelerations $\boldsymbol{\alpha}_{p(L)}$ can be calculated.

$$\begin{aligned} M_{X(L)} &= I_{XX} \cdot \alpha_{pX(L)} + (I_{ZZ} - I_{YY}) \cdot \alpha_{pY(L)} \cdot \alpha_{pZ(L)} \cdot (\Delta t)^2 \\ M_{Y(L)} &= I_{YY} \cdot \alpha_{pY(L)} + (I_{XX} - I_{ZZ}) \cdot \alpha_{pX(L)} \cdot \alpha_{pZ(L)} \cdot (\Delta t)^2 \\ M_{Z(L)} &= I_{ZZ} \cdot \alpha_{pZ(L)} + (I_{YY} - I_{XX}) \cdot \alpha_{pY(L)} \cdot \alpha_{pX(L)} \cdot (\Delta t)^2 \end{aligned} \tag{18}$$

$M_{X(L)}$, $M_{Y(L)}$ and $\mathrm{M}_{Z(L)}$ are the applied moments on the particle in the local coordinate system. I_{XX}, I_{YY}, I_{ZZ} are the principle moments of inertia of the particle and Δt is the time step. The acceleration $\boldsymbol{\alpha}_{p(G)}$ is then determined by a transformation with the inverted matrix $\boldsymbol{R}^{-1}$, i.e. $\boldsymbol{\alpha}_{p(G)} = \boldsymbol{\alpha}_{p(L)} * \boldsymbol{R}^{-1}$.

Equation 18 is solved according to Kremmer [30]

$$\alpha_{X(L)}^{N+\frac{1}{2}} = \frac{M_{X(L)}^{N} - (I_{ZZ}^{*} - I_{YY}^{*}) \cdot \omega_{Y(L)}^{N-\frac{1}{2}} \cdot \omega_{Z(L)}^{N-\frac{1}{2}}}{I_{XX}^{*}}$$

$$\alpha_{Y(L)}^{N+\frac{1}{2}} = \frac{M_{Y(L)}^{N} - (I_{XX}^{*} - I_{ZZ}^{*}) \cdot \omega_{X(L)}^{N-\frac{1}{2}} \cdot \omega_{Z(L)}^{N-\frac{1}{2}}}{I_{YY}^{*}} \tag{19}$$

$$\alpha_{Z(L)}^{N+\frac{1}{2}} = \frac{M_{Z(L)}^{N} - (I_{YY}^{*} - I_{XX}^{*}) \cdot \omega_{Y(L)}^{N-\frac{1}{2}} \cdot \omega_{X(L)}^{N-\frac{1}{2}}}{I_{ZZ}^{*}}$$

And the new acceleration is calculated from the new and old acceleration $\boldsymbol{\alpha}_{XYZ(L)}^{N-\frac{1}{2}}$ is the rotation velocity from the time step before the current time step.

$$\boldsymbol{\alpha}_{\mathrm{L}}^{\mathrm{N}} = \frac{\boldsymbol{\alpha}_{\mathrm{L}}^{N+\frac{1}{2}} + \boldsymbol{\alpha}_{\mathrm{L}}^{\mathrm{N}-\frac{1}{2}}}{2} \tag{20}$$

The local rotation acceleration is then transferred to the global rotation acceleration using the orientation matrix

$$\boldsymbol{\alpha}_{\mathrm{G}} = \boldsymbol{R} \cdot \boldsymbol{\alpha}_{\mathrm{L}} \tag{21}$$

From the global translational and rotational acceleration of the particles (which stay constant during a time step), the new velocities can be calculated using

$$\boldsymbol{v}_{\mathrm{p}}^{\left(\mathrm{N}+\frac{2}{1}\right)} = \boldsymbol{v}_{\mathrm{p}}^{\left(\mathrm{N}-\frac{2}{1}\right)} + \boldsymbol{a}_{\mathrm{p}}^{\mathrm{N}} \cdot \Delta t \tag{22}$$

And the rotational velocity can be computed as

$$\boldsymbol{\omega}_{\mathrm{p}}^{\left(\mathrm{N}+\frac{2}{1}\right)} = \boldsymbol{\omega}_{\mathrm{p}}^{\left(\mathrm{N}-\frac{2}{1}\right)} + \boldsymbol{\alpha}_{\mathrm{p(G)}}^{\mathrm{N}} \cdot \Delta t \tag{23}$$

The new position is then

$$\boldsymbol{r}_{\mathrm{p(G)}}^{(\mathrm{N}+1)} = \boldsymbol{r}_{\mathrm{p(G)}}^{(\mathrm{N})} + \boldsymbol{v}_{\mathrm{p}}^{\left(\mathrm{N}+\frac{1}{2}\right)} \cdot \Delta t \tag{24}$$

3.4. Velocity and position of element spheres

With respect to the global axes, the absolute velocity of the center of an element sphere is calculated as

$$\boldsymbol{v}_{\mathrm{ps}} = \boldsymbol{v}_{\mathrm{p}} + \left(\boldsymbol{\omega}_{\mathrm{p}} \times \boldsymbol{d}_{\mathrm{ps}}\right) + \boldsymbol{v}_{\mathrm{rel}} \tag{25}$$

$\mathbf{v}_{\mathrm{rel}}$ is the relative translational velocity of the element sphere regarding the particle center of mass. Because we defined the distance of each sphere to be constant, this simplifies to

$$\boldsymbol{v}_{\mathrm{ps}} = \boldsymbol{v}_{\mathrm{p}} + \left(\boldsymbol{\omega}_{\mathrm{p}} \times \boldsymbol{d}_{\mathrm{ps}}\right) \tag{26}$$

The magnitude of $\mathbf{d}_{\mathrm{ps}}$ is fixed but actual location of the sub - sphere can change with the rotation of the particle. It can be expressed as

$$\boldsymbol{d}_{\mathrm{ps}} = |\boldsymbol{d}_{\mathrm{ps}}| \cdot \boldsymbol{n} \tag{27}$$

n is the unit vector, in terms of its directions cosines with respect to the global axis it is expressed as

$$\mathbf{n} = \cos\theta_x \cdot \mathbf{i} + \cos\theta_y \cdot \mathbf{j} + \cos\theta_z \cdot \mathbf{k} \tag{28}$$

Θ_x, Θ_y and Θ_z are the directional angles of the vector regarding the global axes**.** **i**, **j**, **k** are the unit vectors for the x, y and z direction. The magnitude remains constant if

$$cos\theta_x^2 + cos\theta_y^2 + cos\theta_z^2 = 1 \tag{29}$$

The change of the unit vector due to particle rotation is expressed as

$$\dot{\mathbf{n}} = \boldsymbol{\omega}_p \times \mathbf{n} \tag{30}$$

This is only true for infinitesimal rotations, that is, when the time step Δt approaches zero. The actual time step in a simulation is always greater than zero and will therefore produce an error in the new unit vector. These errors get smaller by normalizing the error difference using the following procedure (and vanish as the time step approaches zero). The difference in **n**,

$$\Delta\mathbf{n} = \dot{\mathbf{n}} \cdot \Delta t \tag{31}$$

is added to the new unit vector

$$\boldsymbol{n}_{new} = \mathbf{n} + \Delta\mathbf{n} \tag{32}$$

$\mathbf{n}_{\mathrm{new}}$ has to be a unit vector and needs to be normalized:

$$\mathbf{n}_{\mathrm{new}} = \cos\theta_{x_{new}} \cdot \mathbf{i} + \cos\theta_{y_{new}} \cdot \mathbf{j} + \cos\theta_{z_{new}} \cdot \mathbf{k} \tag{33}$$

$$|\mathbf{n}_{\mathbf{new}}| = \sqrt{\cos\theta_{x_{new}}^2 + \cos\theta_{y_{new}}^2 + \cos\theta_{Z_{new}}^2} \tag{34}$$

With the normalized direction cosines

$$\cos\theta_{x_N} = \frac{\cos\theta_{x_{new}}}{|\mathbf{n}_{\mathbf{new}}|}, \cos\theta_{y_N} = \frac{\cos\theta_{y_{new}}}{|\mathbf{n}_{\mathbf{new}}|}, \cos\theta_{z_N} = \frac{\cos\theta_{z_{new}}}{|\mathbf{n}_{\mathbf{new}}|} \tag{35}$$

The new unit vector for each time-step is

$$\mathbf{n}_n = \cos\theta_{x_N} \cdot \mathbf{i} + \cos\theta_{y_N} \cdot \mathbf{j} + \cos\theta_{z_N} \cdot \mathbf{k} \tag{36}$$

And the position of the element sphere is

$$\mathbf{d}_{ps_{new}} = |\mathbf{d}_{ps}| \cdot \mathbf{n}_n \tag{37}$$

The global position of the sphere is then

$$\mathbf{r}_{ps_{new}} = \mathbf{r}_{p(G)} + \mathbf{d}_{ps_{new}} \tag{38}$$

This applies to each element sphere in a particle consisting of any number of spheres. The calculation cycle is completed when the position of every particle and its element spheres are updated. The distance of the element spheres from the center of mass is fixed and only the rotation for the particle is updated.

4. eXtented Particle System (XPS)

The “eXtended Particle System” (XPS) is a discrete element software developed at the Research Center Pharmaceutical engineering (RCPE). It builds heavily on massive parallelization on nVidia graphic cards (GPUs). The nVidia GPUS of the latest generations, since 2007, are designed for direct programming using the CUDA language GPUs [31] and are intrinsically designed for massive parallelization.

This is especially interesting for cases where a lot of relatively simple tasks have to be performed independently from each other, as it is the case when calculating the movement of many independent particles. Recently, the XPS code was coupled to the Computational fluid dynamics software package AVL FIRE, allowing e.g. the simulation of the behavior of particles inside a fluid.

4.1. Parallelization

Nearly all physical problems can be seen as a series of task and handled sequentially. Some can be divided into independent tasks; these tasks can be handled simultaneously on several processors. This is called the granularity of a numerical problem. In the latter category falls the contact detection, the inter particle force calculation and the time integration. Current common multi core microprocessors consist of up to six cores. Each core is a full CPU coupled on chip and connected to the memory, and designed to speed up the execution of sequential programs. For further parallelization, each core can handle two executions threads. In comparison, a standard GPU has a few hundred single cores and each core can be handled by a different thread.

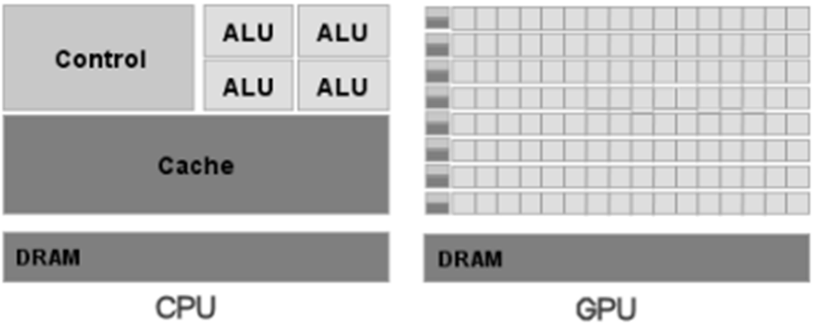

Figure 4-1: architecture of a GPU vs architecture of a CPU [32]

The difference of CPU and GPU architecture is easy to see in Figure 4-1. The GPU has more single cores but less cache and control memory for each core. Because of the smaller cache memory, most GPUs compute only single precision float numbers. Single Precision is enough for most graphic interfaces. Only specialized GPUS designed for crunching numbers can handle double precision numbers. The difference from single to double precision is seen in the Figure 4-2 and Figure 4-3.

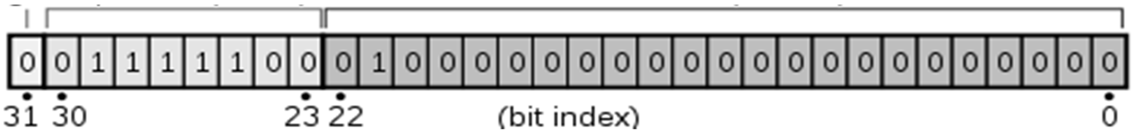

Figure 4-2: Single precision number [33]

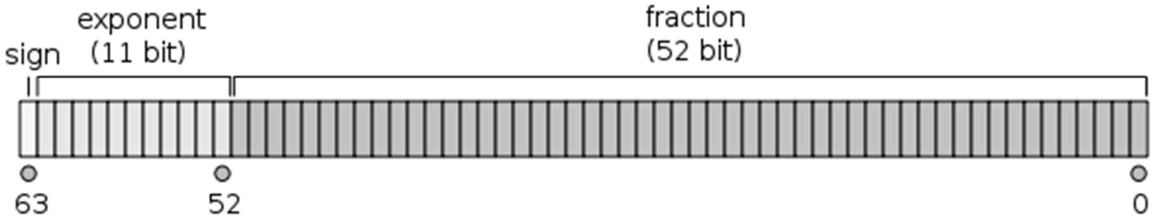

Figure 4-3: Double precision number [34]

Because of the different architecture of GPUs and CPUs, the computational speed of both has risen different over the last years. With more single cores and direct access to the DRAM, the speed of the GPUS grew six times higher than that of comparable CPUS. Because of that and the easier parallelization, the GPUS are more and more common for simple fast and cheap computational tasks.

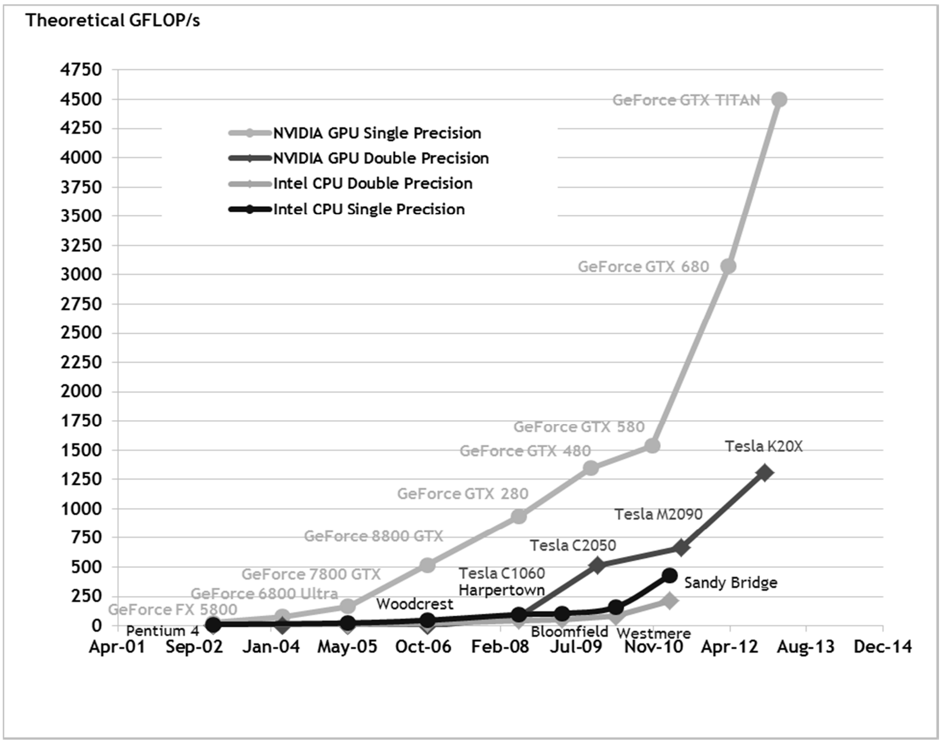

Figure 4-4: Memory bandwith GPU vs CPU [32]

4.2. Parallelization of the DEM code

To use the mentioned parallelization possibility in the DEM there are different approaches possible. One approach could be to assign one thread to one grid cell with particles existing in the cell. Then, all possible contacts for the particle in this cell are checked and the forces are calculated.

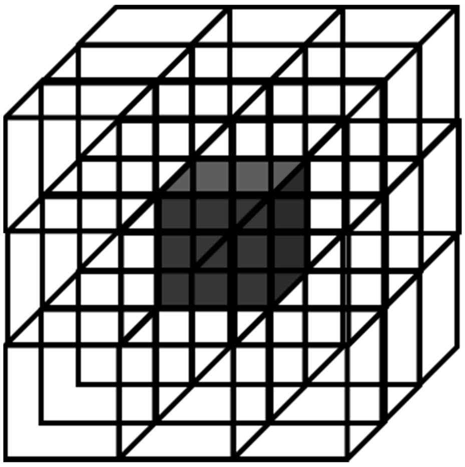

Figure 4-5: Cell grid division one block means one computing thread

This has some disadvantages as that empty cell spaces are still treated even if only to dismiss them. But this approach blocks unnecessary threads in each time step. And the contact and force calculation has still to be done for every particle.

In the XPS code, the approach is to have one thread per particle and the particle has to find the position and number of its home grid cell and the surrounding cells. Then, the possible contact and resulting forces are calculated. This has the advantage that no empty cells appear that still have to be treated, only particles with no contacts. However, these have to be considered in any case, but here they can be dismissed faster which further speeds up the code.

4.3. Force Calculation in the XPS code

In the XPS code, the contact detection and force calculations works as described in chapter 2.3. It uses a soft sphere approach; the force is calculated according to P. Pepiot and Desjardins [27]. The normal force is calculated according to :

$$\mathbf{f}_{b\to a}^{col} = (k \cdot \boldsymbol{\delta} - \eta \cdot (\mathbf{v}_a - \mathbf{v}_b) \cdot n) \cdot n \quad if\ \ d_{ab} < (r_a + r_b)\ , else\ 0 \tag{39}$$

Here $r_{a/b}$ is the radius of the particle a/b, d_{ab} is the distance between the centers of the particles a and b, $\boldsymbol{\delta}$ is the overlap between the particles, $\mathbf{v}_{a/b}$ is the velocity of particle a/b and **n** is the unit vector from the center of particle b to that of particle a, *k* is the spring constant as shown in Figure 2-6

The damping coefficient is calculated according to

$$\eta = -2 * \ln(r_C)\frac{\sqrt{m_{ab} \cdot k}}{\sqrt{\ln(r_C)^2 + \pi^2}} \tag{40}$$

r_C is the restitution coefficient which should have values between 0 (no restitution) and 1 (full restitution). Further, the effective mass is given as

$$m_{ab} = \frac{m_a \cdot m_b}{m_a + m_b} \tag{41}$$

$m_{a/b}$ is the mass of particle a/b Because of symmetry reasons, $\boldsymbol{f}_{b\to a}^{col} = \boldsymbol{f}_{a\to b}^{col}$. Therefore, when the collision forces for one particle are calculated, the forces for the second particle in the collision are also known. However, most likely it would be more resource intensive to find the contacting particles and assign them the already calculated forces, instead of simply calculating the force for every particle independently. Whether this assumption is true or not was not tested in this work, but due to the known properties of the GPU architecture it was deemed very likely.

. For the tangential force, another approach is used, which also takes the friction between particles into account. It is calculated according to Fries ([35])

$$\eta_t = 2 \cdot \alpha \cdot \sqrt{m \cdot k_t} \tag{42}$$

With α a measure of the viscoelastic material properties depending on the normal coefficient of restitution e:

$$\alpha = \begin{cases} \dfrac{\ln(e)}{\sqrt{\pi^2 + ln^2(e)}}\, if\ 0 < e \leq 1 \\ 1\ if\ e =\ 0 \end{cases} \tag{43}$$

The relative tangential velocity is

$$\mathbf{v}_{ab} = (\mathbf{v}_a - \mathbf{v}_b) - (R_a \cdot \boldsymbol{\omega}_a + R_b \cdot \boldsymbol{\omega}_b) \tag{44}$$

$$\mathbf{v}_{ab,t} = (\mathbf{v}_a - \mathbf{v}_b) - (R_a \cdot \boldsymbol{\omega}_a + R_b \cdot \boldsymbol{\omega}_b) \times \mathbf{n}_{ab} - (\mathbf{v}_{ab} \cdot \mathbf{n}_{ab}) \cdot \mathbf{n}_{ab} \tag{45}$$

The tangential stiffness is

$$k_t = \frac{2}{7} \cdot k \tag{46}$$

And the tangential contact force is calculated to

$$\mathbf{f}_{c,t} = \begin{cases} -k_t \cdot \delta_t - \eta_t \cdot \mathbf{v}_{ab,t} & if\ |\mathbf{F}_{c,t}| \leq \mu \cdot |\mathbf{F}_{c,n}|, \\ -\mu \cdot |\mathbf{F}_{c,n}| \cdot \mathbf{v}_{ab,t} \cdot \Delta t & if\ |\mathbf{F}_{c,t}| > \mu \cdot |\mathbf{F}_{c,n}|\,, \end{cases} \tag{47}$$

µ is the friction coefficient. This is done for every contact and the forces are summed up.

Then the tangential and normal force are added to give the resulting force:

$$\mathbf{f} =\ \mathbf{f}_t + \mathbf{f}_n. \tag{48}$$

Then the tangential moment is calculated to

$$\mathbf{M}_t =\ \mathbf{n}\ \times \mathbf{f}_t. \tag{49}$$

When the relative angle velocity is not zero,

$$\boldsymbol{\omega}_{rel} = \boldsymbol{\omega}_b - \boldsymbol{\omega}_a \neq\ 0\ , \tag{50}$$

The relative velocity is normalized to calculate the moment of rolling friction:

$$\boldsymbol{\omega}_n = \frac{(\boldsymbol{\omega}_{rel})}{|\boldsymbol{\omega}_{rel}|}, \tag{51}$$

$$\mathbf{M}_{roll} = \mathbf{R}_f \cdot k\ \cdot r\ \cdot \boldsymbol{\delta}\ \boldsymbol{\omega}_n. \tag{52}$$

Then the total moment is

$$\mathbf{M}_{tot} =\ \mathbf{M}_t + \mathbf{M}_{roll} \tag{53}$$

For multi sphere elements, the force and momenta calculation is extended according to 3.3.1. The total moment is then used to calculate the rotational acceleration of the clump particle, as laid down in chapter 3.3.2. The new position of the sub-spheres are then calculated as shown in chapter 3.4.

5. Implementation

5.1. Program structure

The methods described above were implemented in XPS. As mentioned, the DEM program code part is designed for heavy parallelization. The code is written in C++ and in the CUDA programming language where it relates to GPUs.

Building upon the already existing XPS code, a function to define and recognize multi sphere particles was implemented. The particles are always defined such that the center of mass is at the origin of the local coordinate system (x = y = z = 0). Most of the time in this thesis, particles consisting of only two identical spheres, which are located exactly one radius away from the center were chosen. This was due to simplicity, but the principle works for any axi-symmetrical particle and in principle for every particle form consisting of spheres.

At first, the particles are initiated at a given position of the simulation. Here it is important that no particles overlap at the beginning. Else, the simulation would start with a massive amount of energy input which would not be physical. It is important that during the simulation no energy is produced or lost without reason.

5.2. Particle initialization

To give the particles the position of the sub – spheres, these sub – positions are initialized in the CUDA rotation set of the DEM simulation. These rotation sets include the angle, angular velocity, torque, quaternions, and (where extended) the sub - positions and angular acceleration. This was necessary because the positions of the sub - spheres are needed not only for particle - particle contact but also for particle wall and particle boundary contact. At the initialization step, the sub - position data is loaded into a helper vector.

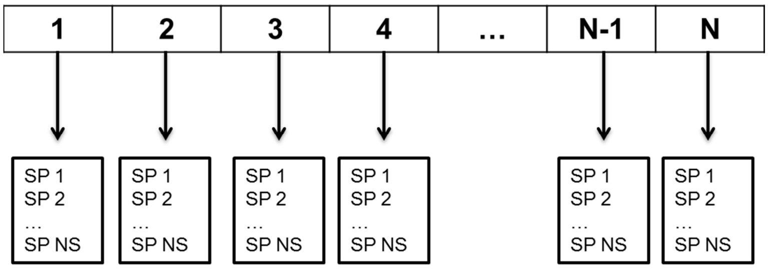

Figure 5-1: Design of the sub - position matrix, N is the number of particles, NS the number of sub – particles for each particle

The sub - position vector is created at the beginning of the calculation cycle.

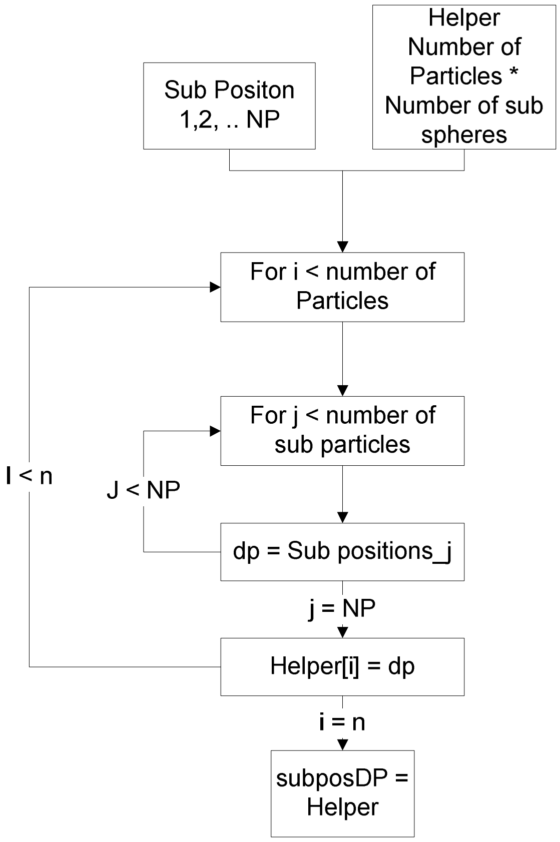

Figure 5-2: Initialization cycle for the sub - sphere positions

5.3. Calculation cycle

The calculation cycle used by the XPS software is shown in Figure 5-3. This cycle does not include the use of particles consisting of more than one sphere. First the particles are placed into the simulation environment. This happens in defined starting positions, which can have different shapes in the XPS code which have to be called in the configuration file before the simulation starts. These shapes include cubic and spherical order and a randomized configuration. The configuration file also defines the global gravitational force, the time step size and the material properties. Those properties have to be defined prior to the simulation; it is not possible to change the properties during run-time. When the simulation is started, all algorithms concerning the particles, like contact search or force calculation, is done as good as simultaneously as every particle is treated in its own thread. Strictly speaking, they are treated nearly simultaneously, because even high-end GPUs can handle “only” a few hundred threads simultaneously. This is still very efficient especially for a lot of particles, especially compared to DEM approaches limited to regular CPUs.

For each particle, it is checked if there are neighboring particles in the surrounding cells. If there are, it is checked if the particles are in contact. If the particle is in contact with one or more particles, the resulting forces and momenta are calculated. From these, the rotational and velocity are calculated and the position in the next time step is updated. This calculation circle is shown in Figure 5-3.

In Figure 5-4, the adapted XPS code is shown. The initialization is the same as in the original code. The difference is that the grid cells are bigger here. This is due to the fact that a particle consisting of two equally sized spheres is bigger than a particle consisting of just one of these spheres. The next neighbor sorting is the same for the multi element particles. However, the contact detection has to be done for every sub - particle separately. Then, the forces acting on every sub - particle and the momenta generated by these forces are transferred to the center of mass, were also the momenta are expanded to forces acting on each particle according to the distance of the center of mass.

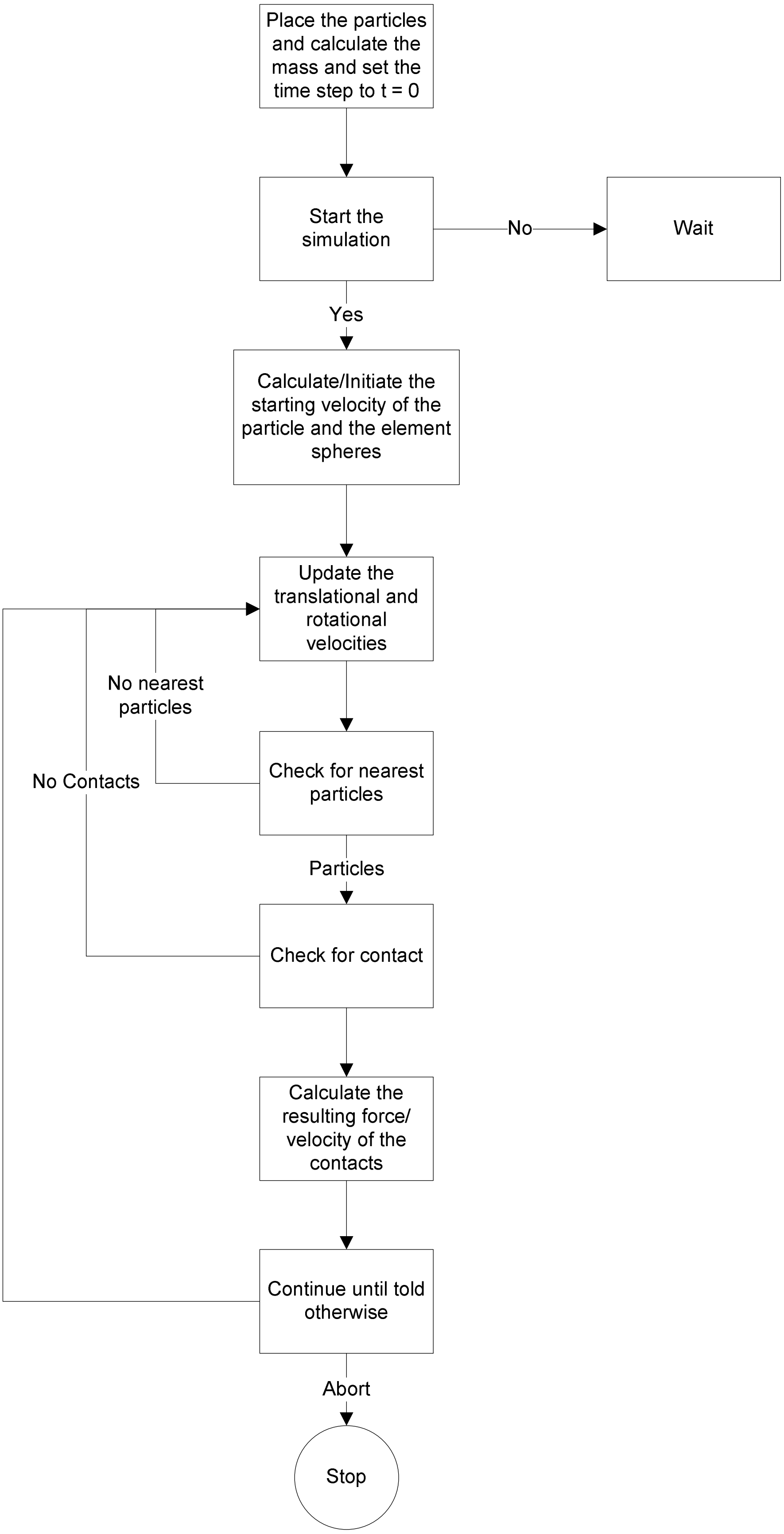
Place the particles and calculate the mass and set the time step to t = 0
Start the simulation
No
Wait
Yes
Calculate/Initiate the starting velocity of the particle and the element spheres
Update the translational and rotational velocities
No nearest particles
No Contacts
Check for nearest particles
Particles
Check for contact
Calculate the resulting force/ velocity of the contacts
Continue until told otherwise
Abort
Stop

Figure 5-3: Calculation cycle of the original XPS code

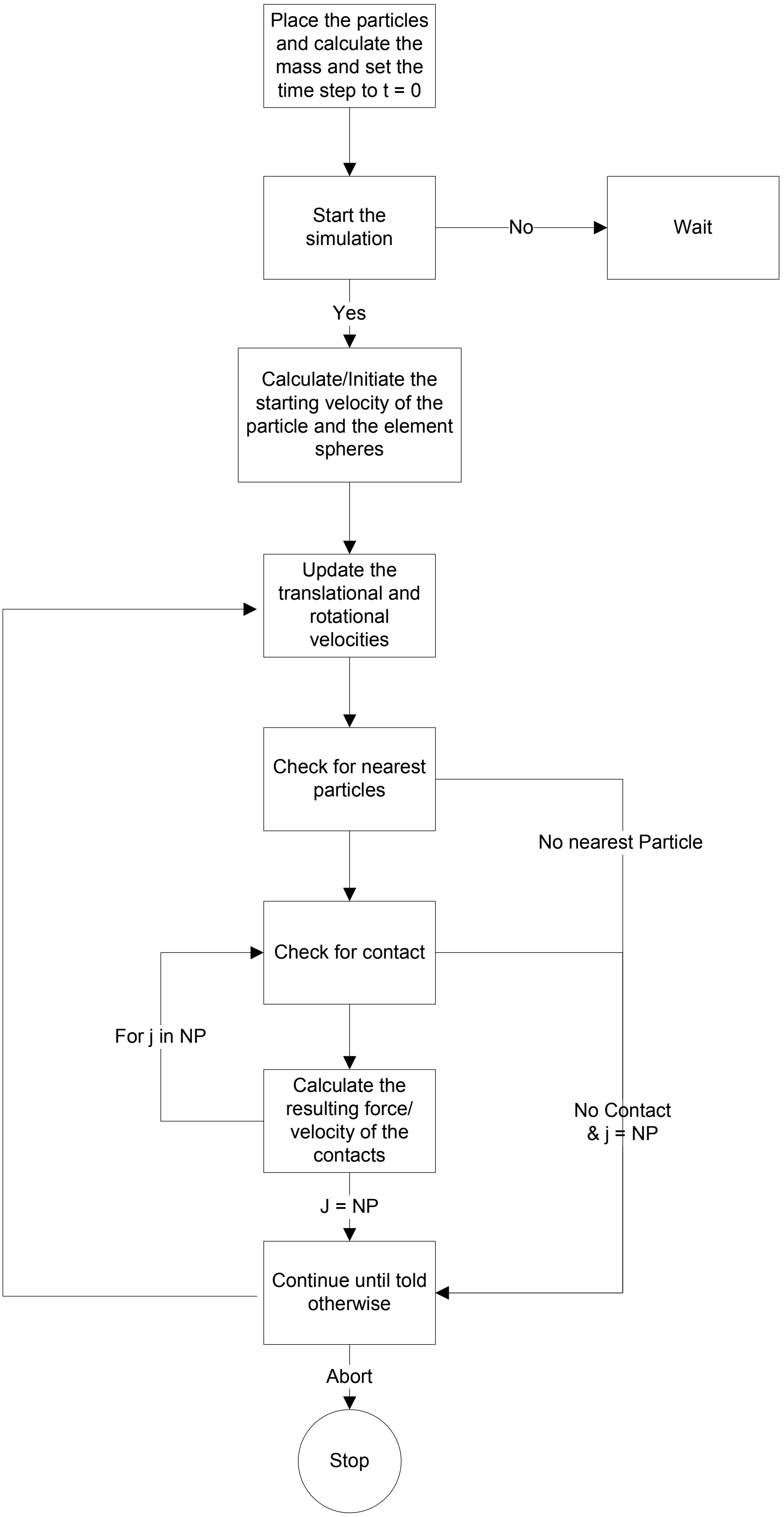
Place the particles and calculate the mass and set the time step to t = 0
Start the simulation
No
Wait
Yes
Calculate/Initiate the starting velocity of the particle and the element spheres
Update the translational and rotational velocities
Check for nearest particles
No nearest Particle
Check for contact
For j in NP
Calculate the resulting force/ velocity of the contacts
No Contact & j = NP
J = NP
Continue until told otherwise
Abort
Stop

Figure 5-4: Expanded calculation cycle of the XPS code

The forces acting on the particles are described by a Kevin Voigt model chapter 2.5. The velocities of the particles are calculated according to the models shown in the previous chapters. For a spherical particle, the velocity and rotational velocity can directly be calculated from the forces acting on the particle. From theses velocities the new position of a particle can simply be calculated. The rotational velocity of a spherical particle is not needed for the new position and only plays a role when calculating the next contact forces. This is visualized in Figure 5-5.

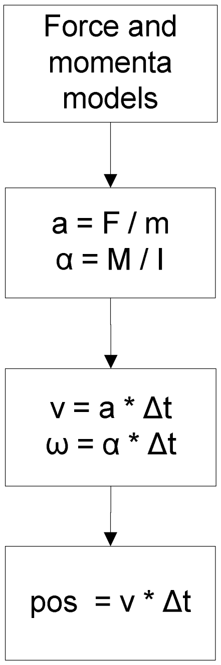

Figure 5-5: Old rotation and position calculation

The velocity calculation is the same for the multi element particle. The only difference is that the forces acting on each sub - sphere have to be summed up and brought to the center of mass of the model particle.

For non-spherical particles and especially multi element models, the rotational velocity is more important than for spherical particles. Through the rotational velocity the positions of the sub – spheres are defined. This begins with the adapted calculation of the momenta to the more complex algorithms of the rotational acceleration and ends in the calculation of the new positions of the sub - spheres, shown schematically in Figure 5-6.

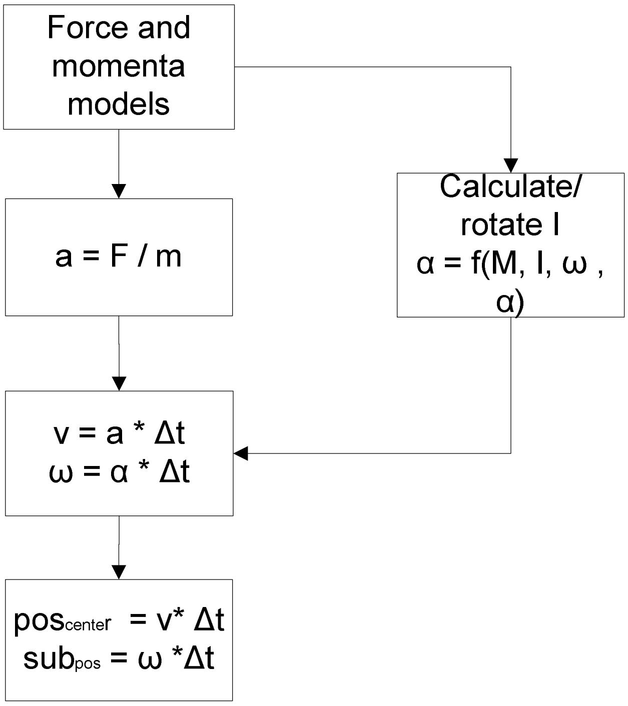

Figure 5-6: New force and position calculation models for multi element models

6. Material and Methods

In addition to XPS another DEM program was used to test the validity of the multi-element model.

6.1. EDEM

EDEM has wide-spread application in different fields and is known to deliver reliable results when given the right material parameters. It further provides a multi element model.

From the official DEM-Solution website [36] the detailed description of EDEM is given here.

Discrete Element Method (DEM) simulation is transforming the business of designing and optimizing equipment for the handling and processing of bulk materials.

When used properly, DEM simulation gives you key design information on bulk solid material flow behavior that is very difficult, or even impossible to get using standard test methods or other methods of engineering simulation. EDEM(R) is high-performance DEM simulation software-the only commercially available software that is capable of generating the powerful DEM simulations and analysis required to solve complex problems in the design, prototyping, and optimization of equipment that handles and processes bulk solid materials-across a wide range of industry sectors. First introduced to industry nearly a decade ago, EDEM is powered by state-of-the-art Discrete Element modeling technology and uniquely provides engineers with the capability to quickly and accurately simulate and analyze the behavior of their granular solids systems. EDEM has an easy-to-use GUI that speeds simulation set-up time with tools to quickly create a particle-scale parameterized model of a bulk granular solids system.

7. Testing

After implementing the code into XPS, a series of tests was performed. First, simple particle – particle contacts were simulated to verify the contact model between particles works as intended. After that, particle fall test were done in XPS and the commercial DEM software EDEM. Finally, a test based on the behavior of a large sample of particles colliding into each other was performed. Here a lot of particles start in a dense grid position, and fall into a heap under the influence of gravity. The height and form of the heaps forms were correlated to the particle form. The form of the particle heaps were compared to particle heaps in EDEM using similar starting conditions.

7.1. Particle - Particle Contact

For the particle-particle collision test, two particles were initialized at a distance of two times the sub-sphere radius. The gravitational forces were set to zero. One particle (termed particle 1) was at rest, the other (termed particle 0) had a starting velocity of the magnitude of 1 m/s. Two variations of the test were done. In the first variation (“circle run”), the position is changed along a half circle around the other particle, while the velocity vector is always pointing to the center of particle 1, see Figure 7-1. In the second variation (“line run”), the position of particle 0 is shifted along a line in x direction as seen in Figure 7-6, while the velocity vector points in negative y direction. The according positions of particle 0 are stated in Table 7-1 and Table 7-2 respectively. Then the simulation was started and after the two particles collided and were separate and the values of the particles where saved. The results for both runs are shown in Figure 7-2, Figure 7-3, Figure 7-4 and Figure 7-5 for the first run, and Figure 7-7, Figure 7-8, Figure 7-9 and Figure 7-10 for the second run. For both runs the first simulation is a central collision. Due to the constant starting velocity magnitude, the kinetic energy at the start of the simulation was always the same.

Note that the restitution coefficient was set to 1.0 so that no energy is lost in the collision (save for computational rounding errors).

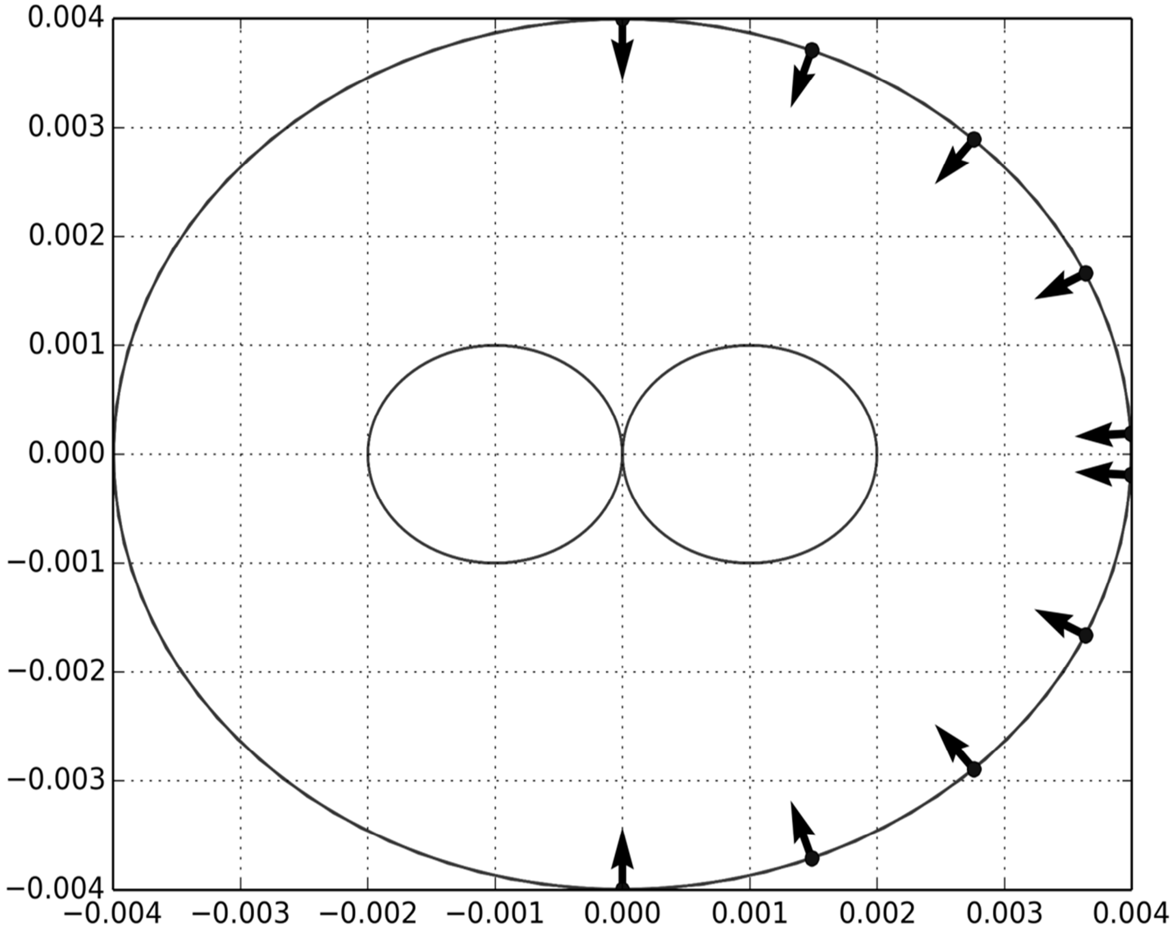

Figure 7-1: Position and velocity vector of the "circle run" particle-particle contact tests.

Table 7-1: Start position angle and velocity of particle 0 for the "circle run" cases.

p_x in m	p_y in m	angle in radiant	v_X in m/s	v_Y in m/s
0,0000	0,0040	1,5708	0,0000	-1,0000
0,0015	0,0037	1,1900	-0,3717	-0,9284
0,0028	0,0029	0,8092	-0,6901	-0,7237
0,0036	0,0017	0,4284	-0,9096	-0,4154
0,0040	0,0002	0,0476	-0,9989	-0,0476
0,0000	-0,0040	-1,5708	0,0000	1,0000
0,0015	-0,0037	-1,1900	-0,3717	0,9284
0,0028	-0,0029	-0,8092	-0,6901	0,7237
0,0036	-0,0017	-0,4284	-0,9096	0,4154
0,0040	-0,0002	-0,0476	-0,9989	0,0476

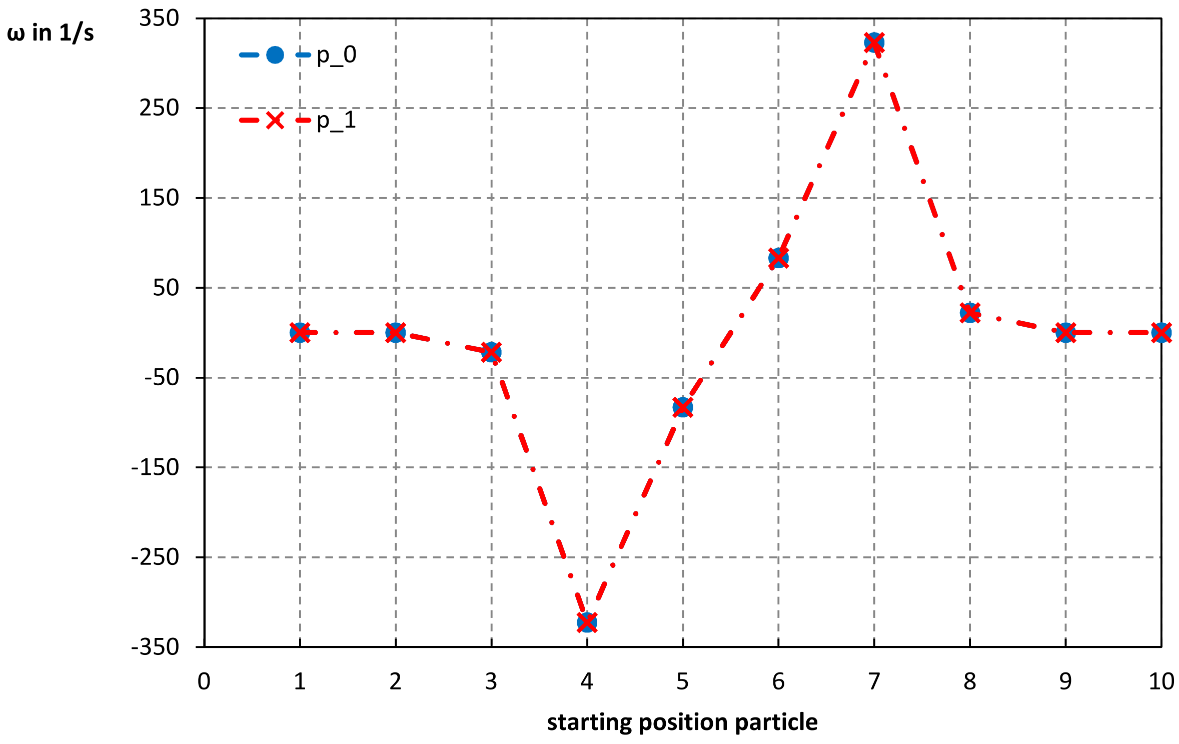

Figure 7-2: Rotation of velocity particle 0 and 1 in z direction

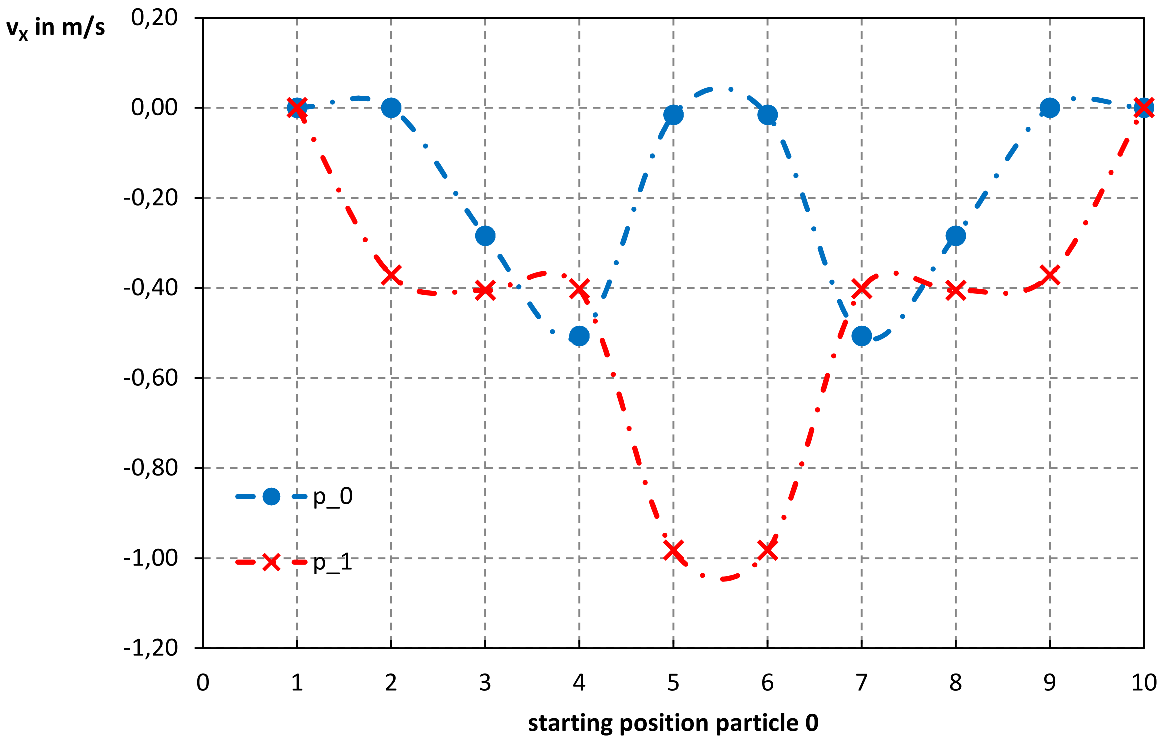

Figure 7-3: Velocity of particle 0 and 1 in x direction

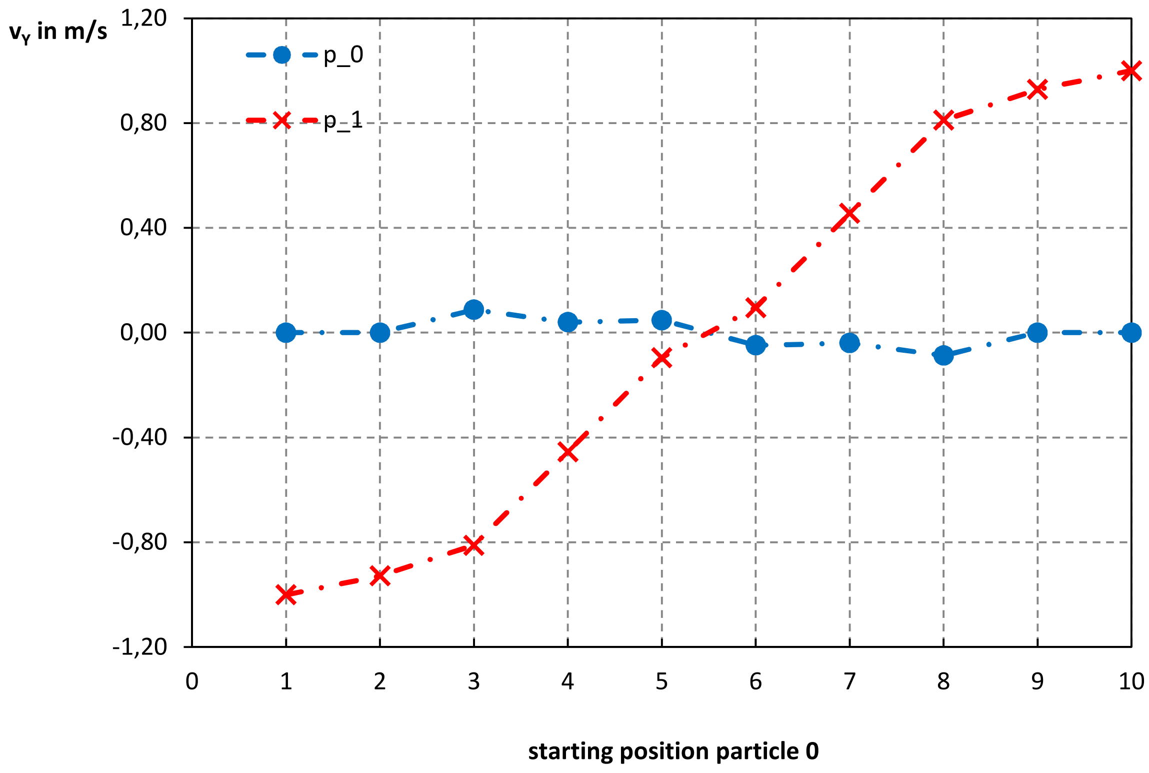

Figure 7-4: Velocity of particle 0 and 1 in y direction

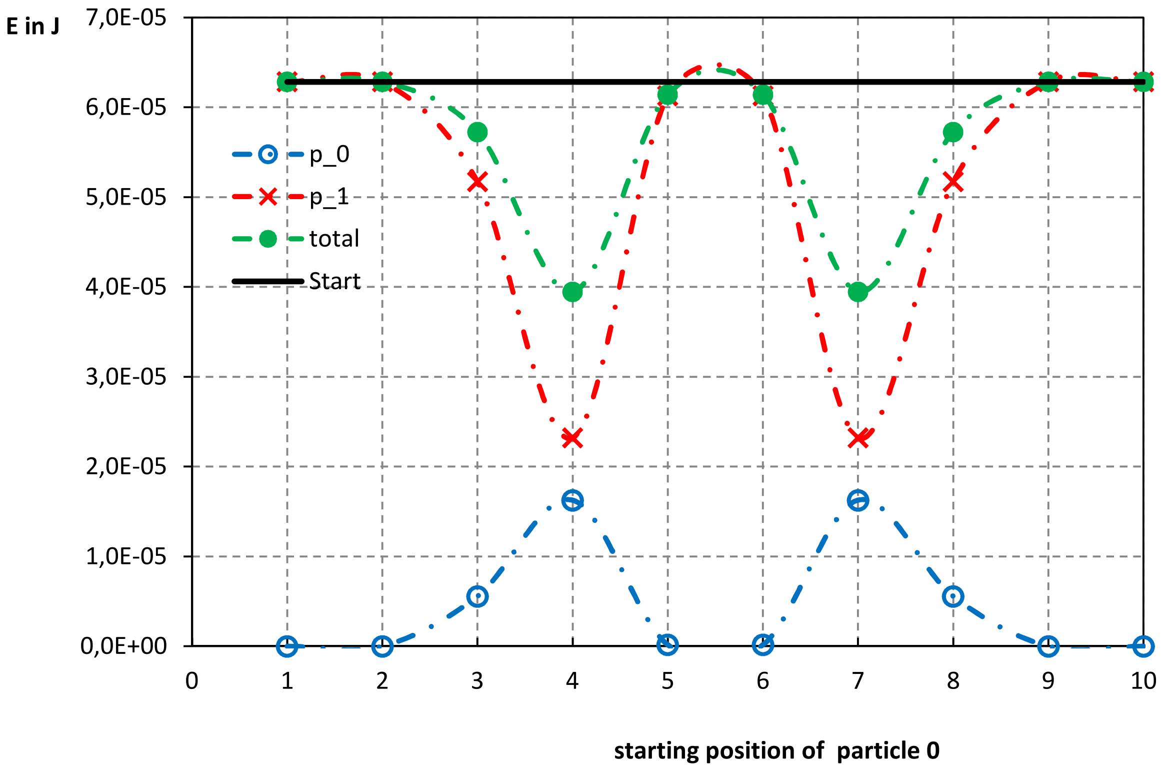

Figure 7-5: Energy in the system and particle 0, 1 and total energy, the black line shows the energy put into the system

Table 7-2: Starting position of particle 0 for the "line run" cases.

p_x in m	p_y in m	v_X in m/s	v_Y in m/s
0,000	0,03	0	1
0,004	0,03	0	1
0,009	0,03	0	1
0,013	0,03	0	1
0,018	0,03	0	1
0,022	0,03	0	1
0,027	0,03	0	1
0,031	0,03	0	1
0,036	0,03	0	1
0,040	0,03	0	1
0,000	0,03	0	1

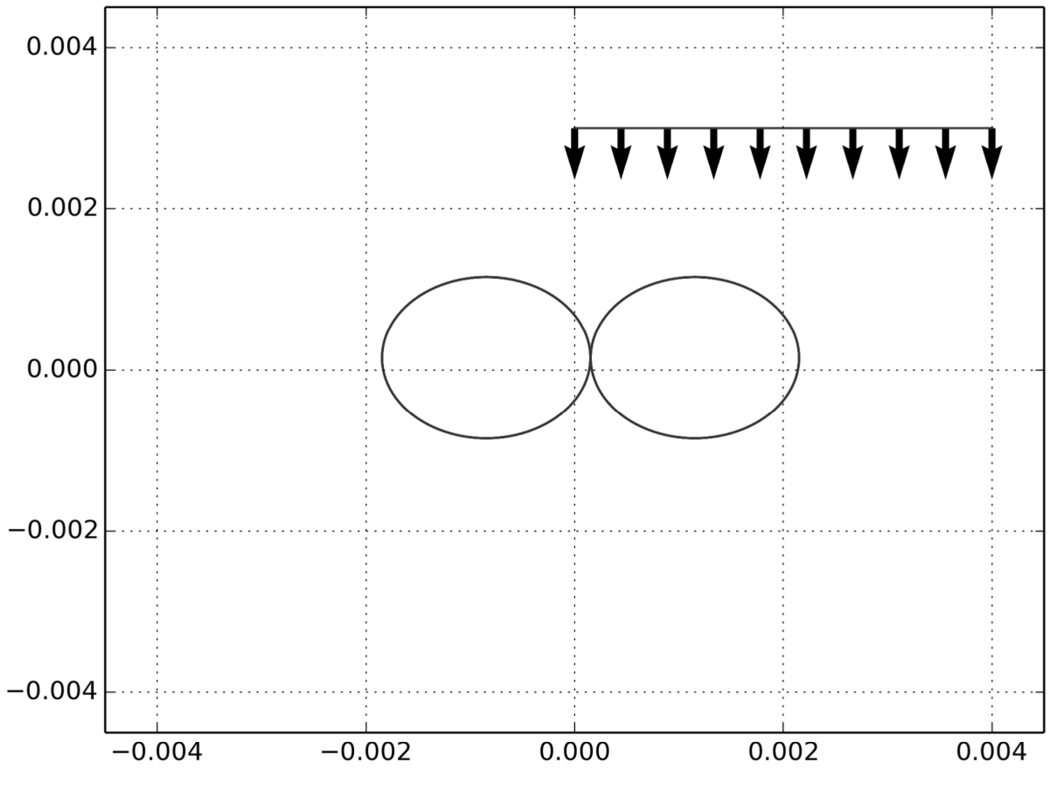

Figure 7-6: Position and velocity vector of the particle - particle contact tests line order

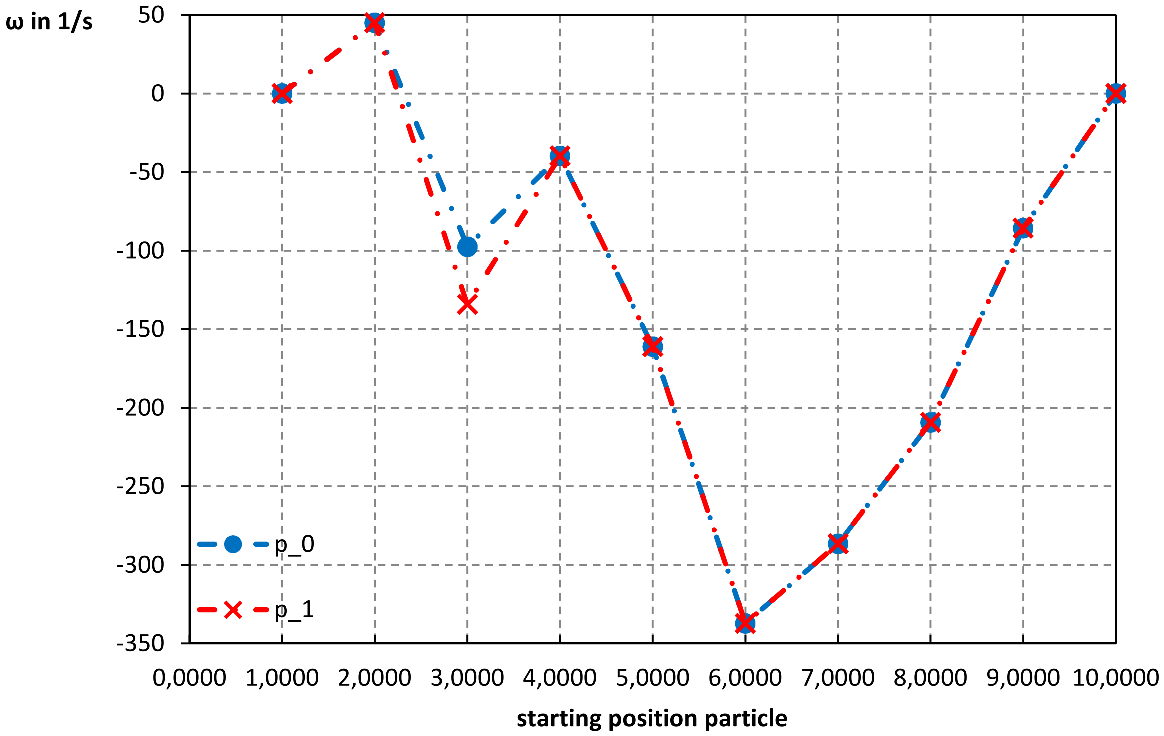

Figure 7-7: Rotation velocity of particle 0 and 1 in z direction

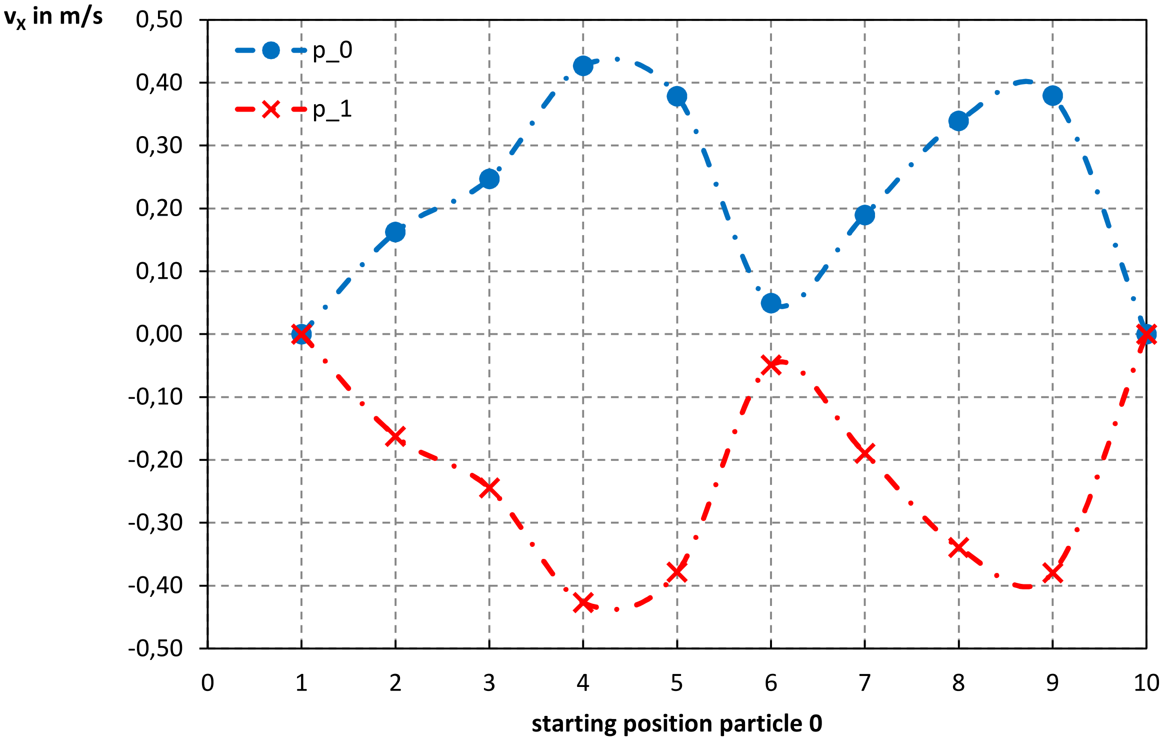

Figure 7-8: Velocity particle 0 and 1 in x direction

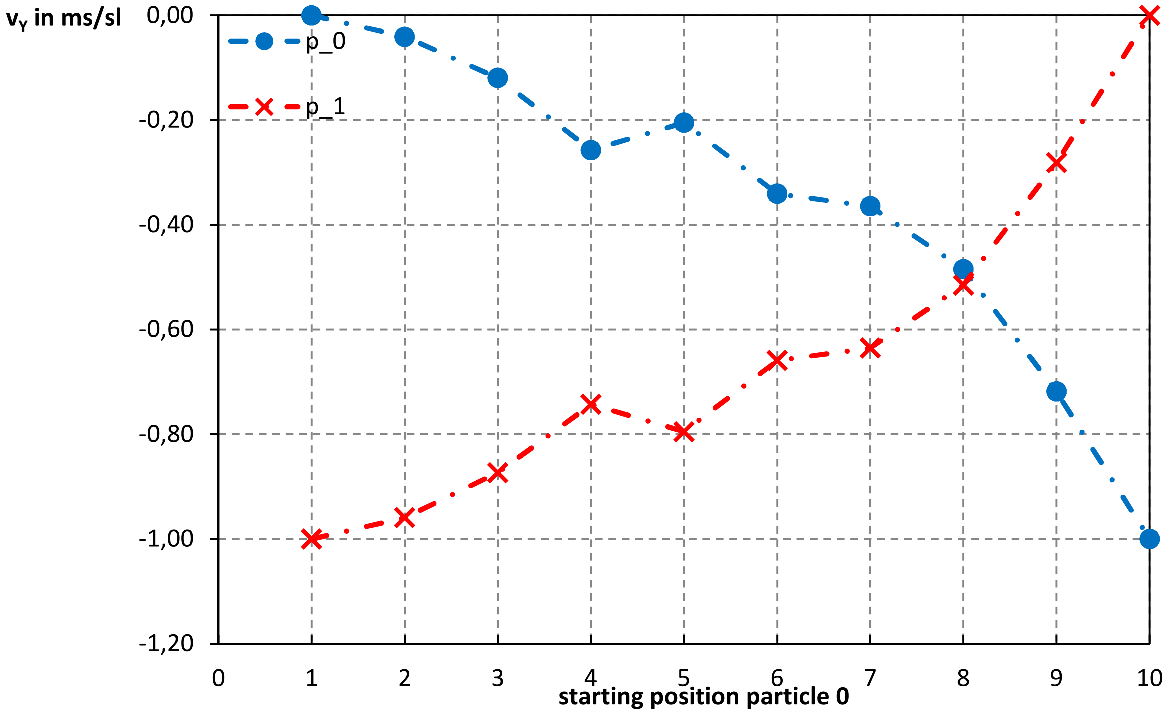

Figure 7-9: Velocity of particle 0 and 1 in y direction

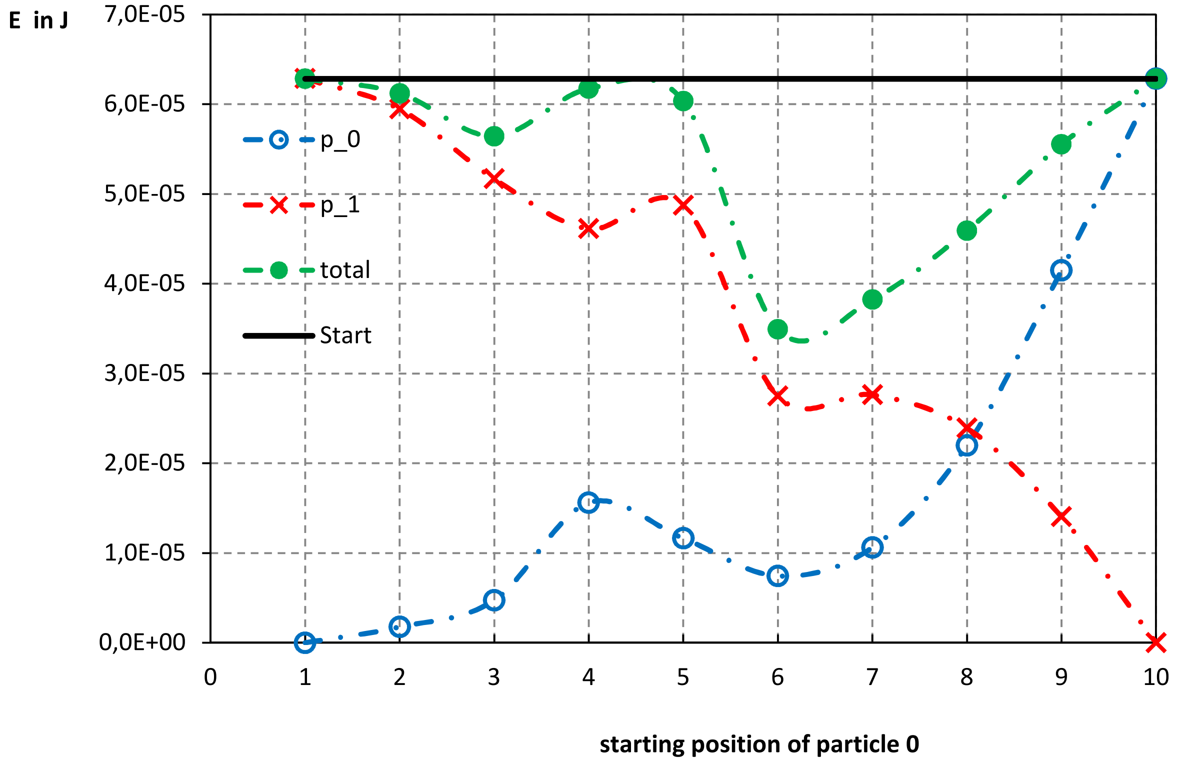

Figure 7-10: Energy in the system and particle 0, 1 and total energy, the black line shows the energy put into the system

In Figure 7-2 and Figure 7-7 it can be seen that the rotation velocity of the two contacting particles are (almost) identical to each other after the collision, as it should be when two identical particles in zero gravity collide. There is no energy generated in the process which is an important property in particle simulations. That the rotation is positive for small angles can be explained by the particular design of the particles, where a glancing hit can lead to this counter-intuitive (and relatively small) rotation.

The results for the velocities after the collision are shown in Figure 7-3 and Figure 7-4 for the circle test and in Figure 7-8 and Figure 7-9 for the line test run. In all cases, the velocity in the y direction is reduced by direct hits for the first particle, as it increased for the particle which has taken the hit. The velocity in the x direction is in both cases opposed to each other, as it should be. This is clearly visible in Figure 7-3 and Figure 7-8 were the velocity curves of the particle 0 and 1 in the x direction are exactly mirrored around the x axis. Similarly, the velocities in the y direction for the circle run after the half of the experiments are mirrored as seen in Figure 7-4. Showing that the total energy in the system is constant is shown for the circle run in Figure 7-5 and in Figure 7-10 for the line test run.

This test has shown that the general multi sphere model implemented in the XPS code works as intended, and that the particle rotation and translation follows the expectation. This was one of the main goals in this thesis. To show that it works also for more particle agglomerates is done in the next chapters.

7.2. Angle of repose

To test the interaction of a larger number of particles, a simple test case was set up. The particles start in a sorted grid in the middle of the simulation box, shown in Figure 7-11, and settle under the influence of gravity.

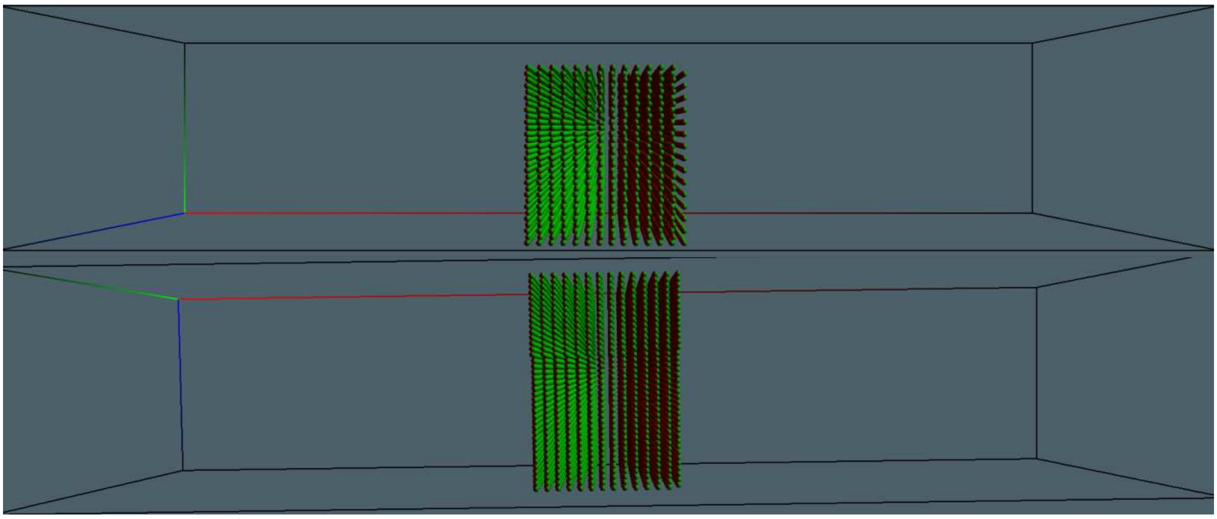

Figure 7-11: XPS particle initialization for the angle of repose simulation. Top: view from the side (along x direction), bottom: view from above (z direction).

The gravitational force is set to -9.81 m/s^2 in the direction of the y axis. The simulation is done using particles consisting of one sphere (simple spherical particle), two spheres, and four spheres arranged in a quadratic formation The number of particles was set according to Table 7-3. The radius of the sub-spheres is chosen such that the total volume occupied by the particle was always the same. For a given radius, the overall number of sub-spheres is the same for all shapes. The grid size was then varied according to Table 7-4 for spherical particles and Table 7-5 for particles consisting of two and four spheres. The particle parameters are shown in Table 7-6. The particles fall under the influence of gravity, taking into account their interaction with each other and the surrounding walls. The resulting particle piles are compared to each other to see the influence of the multi sphere model.

The resulting heaps are shown in Figure 7-14 for spherical particles, Figure 7-15for two connected spheres and in Figure 7-16 for four connected spheres

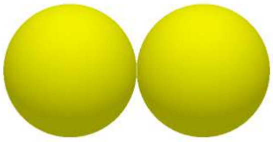

Figure 7-12: Two connected spheres particle design

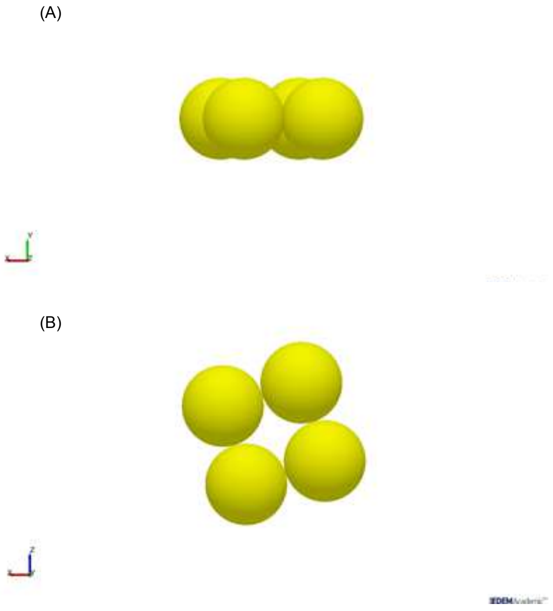

Figure 7-13: Four connected particles design seen from the z (A) and y (B) direction

Table 7-3: Size of spheres, number of sub - spheres and number of total particles

Nr of sub - spheres	1	2	4
Radius	Number of	Number of	Number of
in m	Particles	Particles	Particles
0,01	4000	2000	1000
0,005	32000	16000	8000
0,0025	256000	128000	64000

Table 7-4: Number of grid points and total cells for 1 sub - sphere particle

Grid number			Number of
X	Y	Z	Cells
100	20	20	40000
200	40	40	320000
400	80	80	2560000

Table 7-5: Number of grid points and total cells for 2 and 4 sub - sphere particle

Grid number			Number of
X	Y	Z	Cells
50	10	10	5000
100	20	20	40000
200	40	40	320000

Table 7-6: XPS particle parameters

density	in kg/m3	1200		
rc		0,4		
shear rate		1,00E-08		
attraction		0		
boundarydamping		-1		
globaldamping		1		
spring		500		
muwall		0,9		
gravity		0	-9,81	0

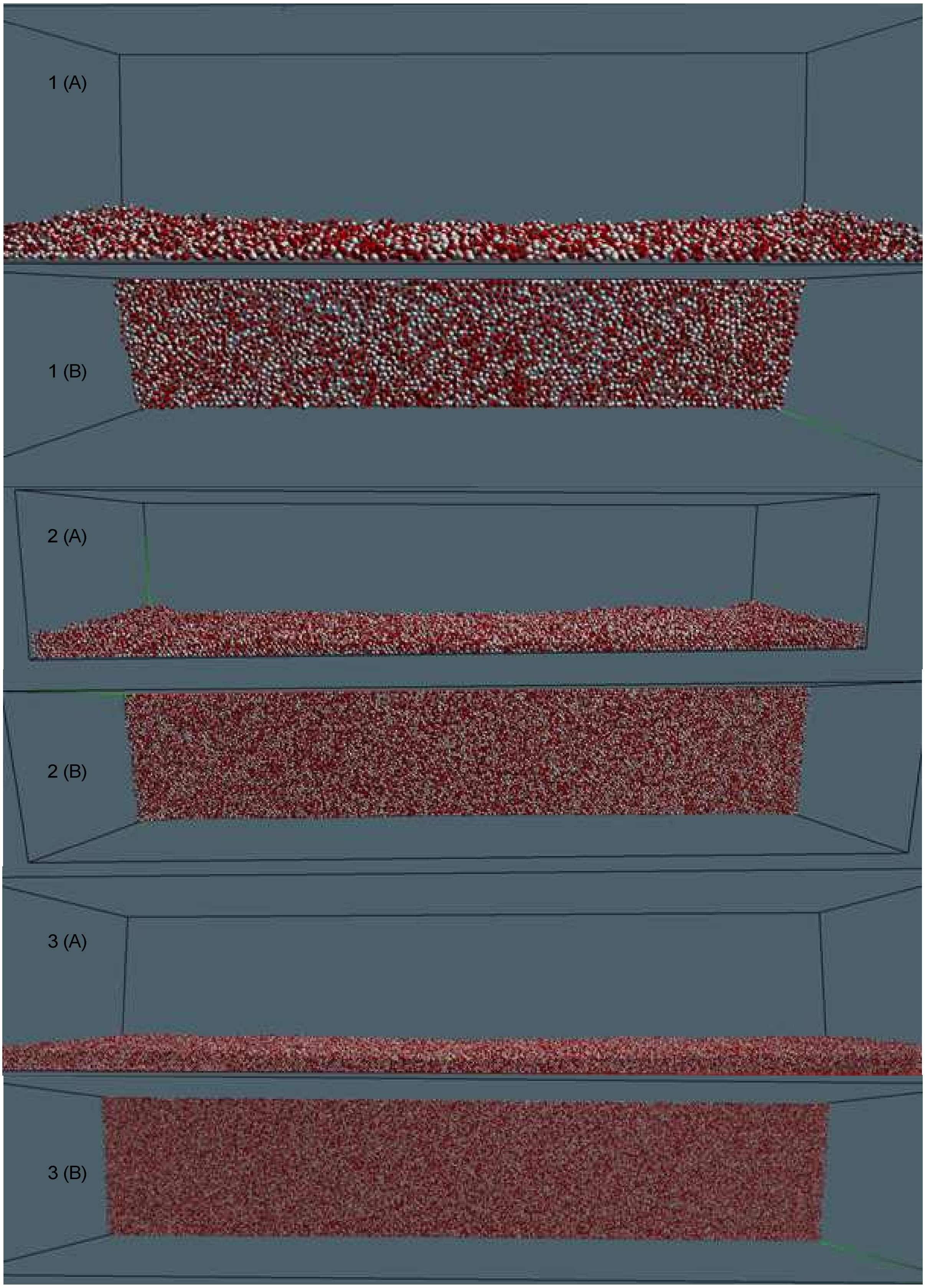

Figure 7-14: 1 sphere particle cases, with 4000 (1), 32000 (2), 256000 (3) particles seen from the side (A) and above (B)

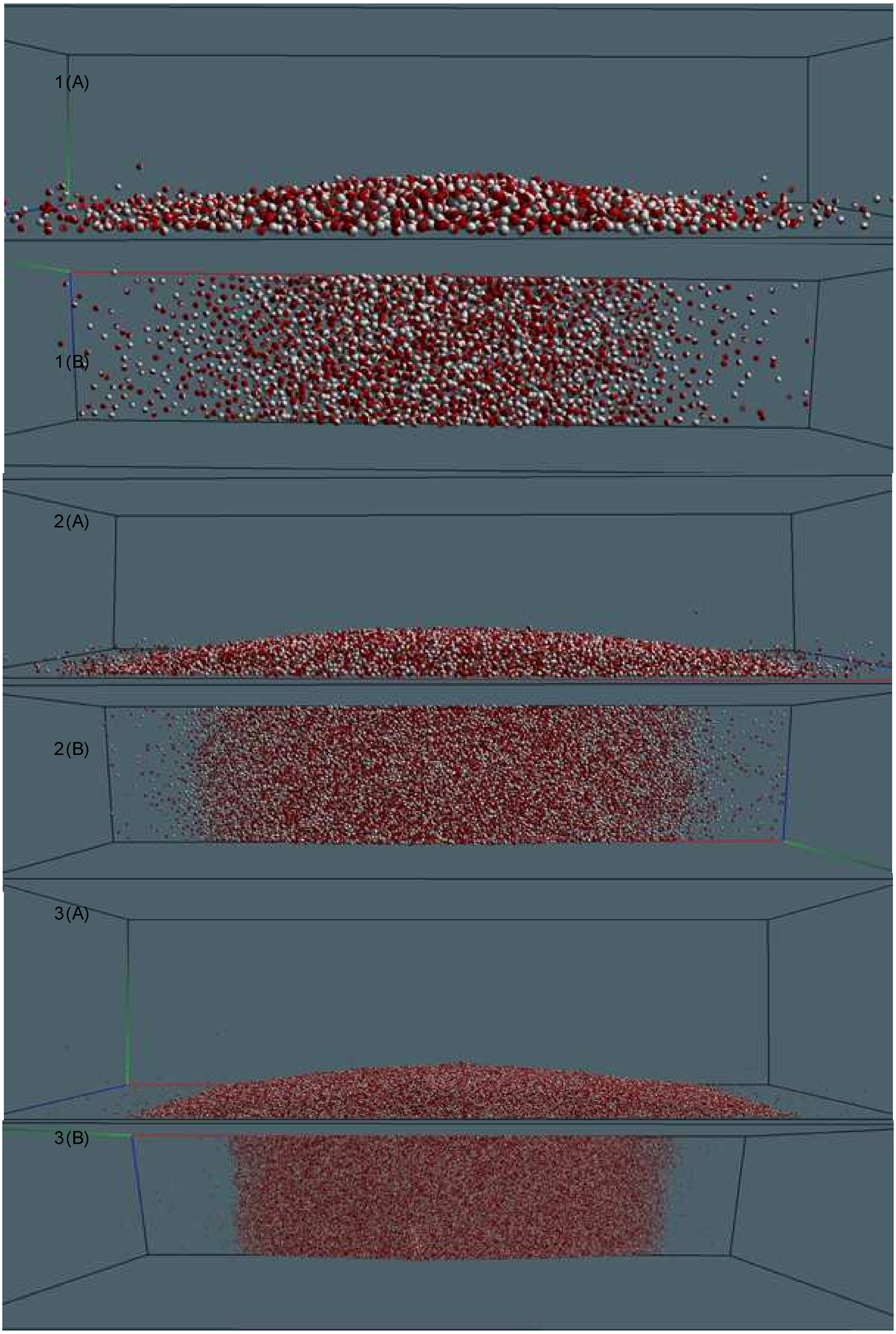

Figure 7-15: 2 sub - spheres particle 2000 (1), 16000 (2) and 128000 (3) particles seen from the side (A) and above (B)

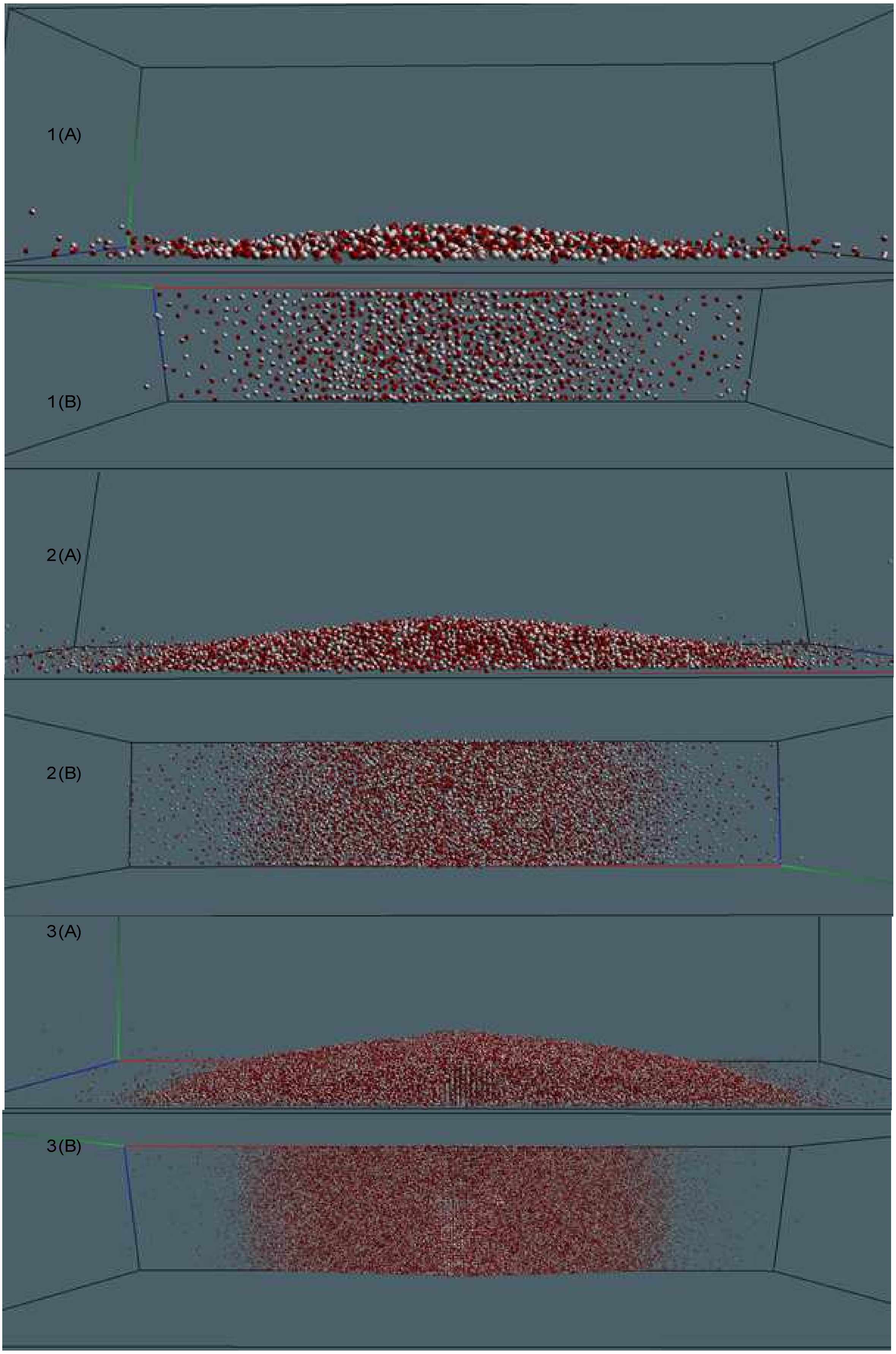

Figure 7-16: 4 sub - spheres particle 1000(1), 8000(2), 64000(3) particles seen from the side (A) and above (B)

The particle piles in Figure 7-14, Figure 7-15 and Figure 7-16 show that the particle shape has a significant impact on heap formation. The difference between spherical particles and multi sphere particles is immense. Instead of forming a pile, the spheres cover the whole available floor area. The multi sphere elements form a clearly defined pile due to interlocking with each other. The heaps for a size of r = 0.0025 (Figure 7-16 - 1) are a little higher than for r = 0.01 in (Figure 7-16 – 3) That is, smaller particle sizes form higher heaps than bigger particles, even if the total volume of the particle stays the same. This is due to the increased interlocking potential between the smaller particles.

7.3. Comparing EDEM with XPS multi element model

7.3.1. Resulting velocity

The results of a number of falling particles, each consisting of 10 sub – spheres, was simulated in EDEM and XPS, and were compared to each other. For this the gravity was set to -9.81 m/s in the y direction. The developments of the positions and velocities over time are used for comparison. The properties of the particle are shown in Table 7-7 for the EDEM simulation and Table 7-8 for the XPS simulation. The used moment of inertia was gathered from the EDEM software. It can be seen in Table 7-9. Three particles were initialized in the EDEM simulation; the same starting positions were put into XPS. After initialization, the gravitational force was activated in EDEM and in XPS. The shape of the particle is shown in Figure 7-17.

Table 7-7: Particle parameter and interaction parameters

Particle Paramters		
Poission ratio		0,5
shear modulus	in GPa	100
Density	in kg/m^3	1200
Particle Interactions		
Coefficient of Restitution		0,5
Coefficient of Static Friction		0,1
Coefficient of Rolling Friction		0,01

Table 7-8: Particle properties and interaction parameters XPS

ρ	in kg/m^3	2500
C_r		0,5
$d_{boundary}$	in	-0,55
k	in N/m	1500

Table 7-9: Moment of inertia for the tablet from EDEM

inertia direction		
X	in kg m^2	$2{,}11*10^{-9}$
Y	in kg m^2	$1{,}89*10^{-9}$
Z	in kg m^2	$3{,}42*10^{-9}$

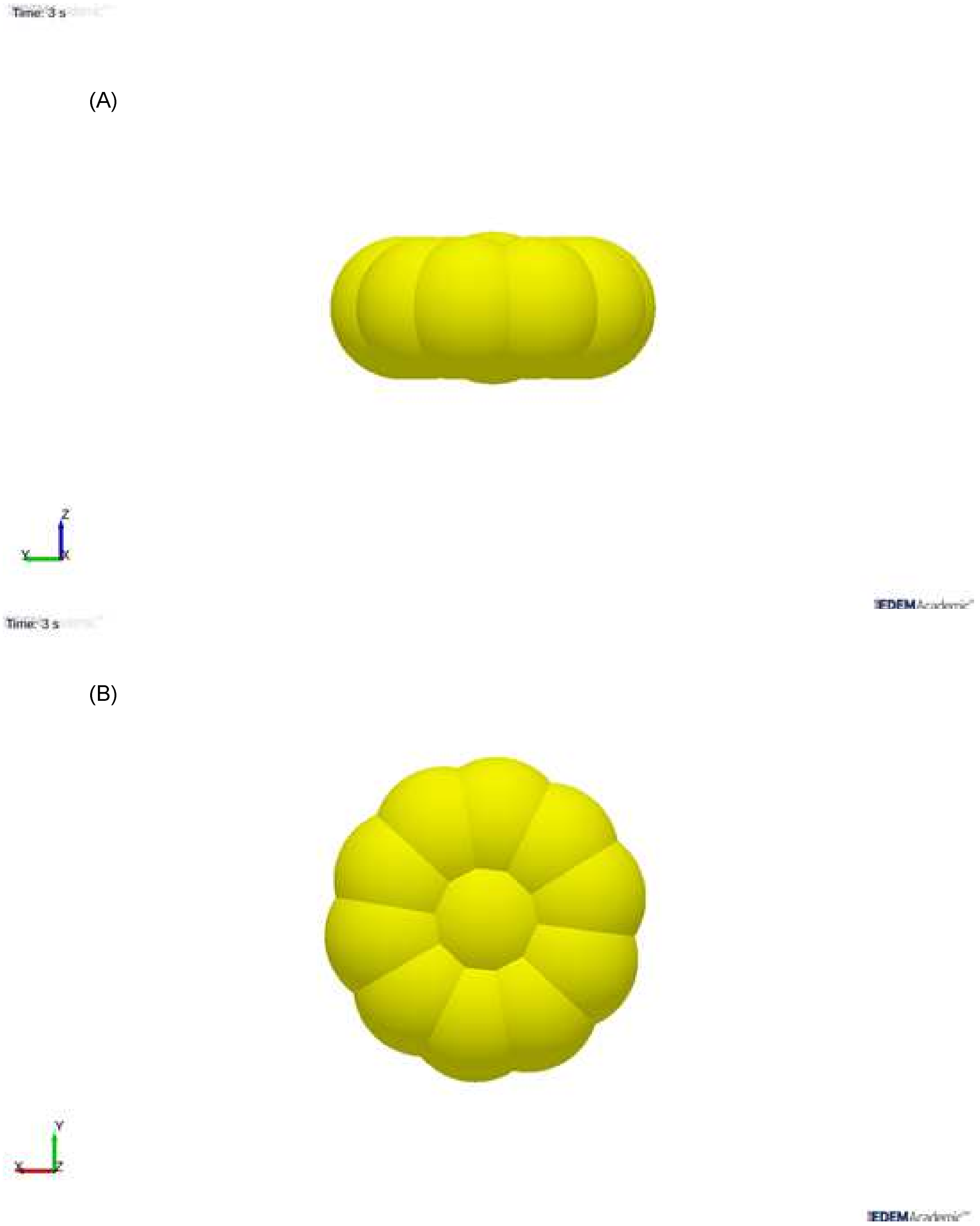

Figure 7-17: Tablet shape and positions of the sub – spheres from the X (A) and Z (B) perspective

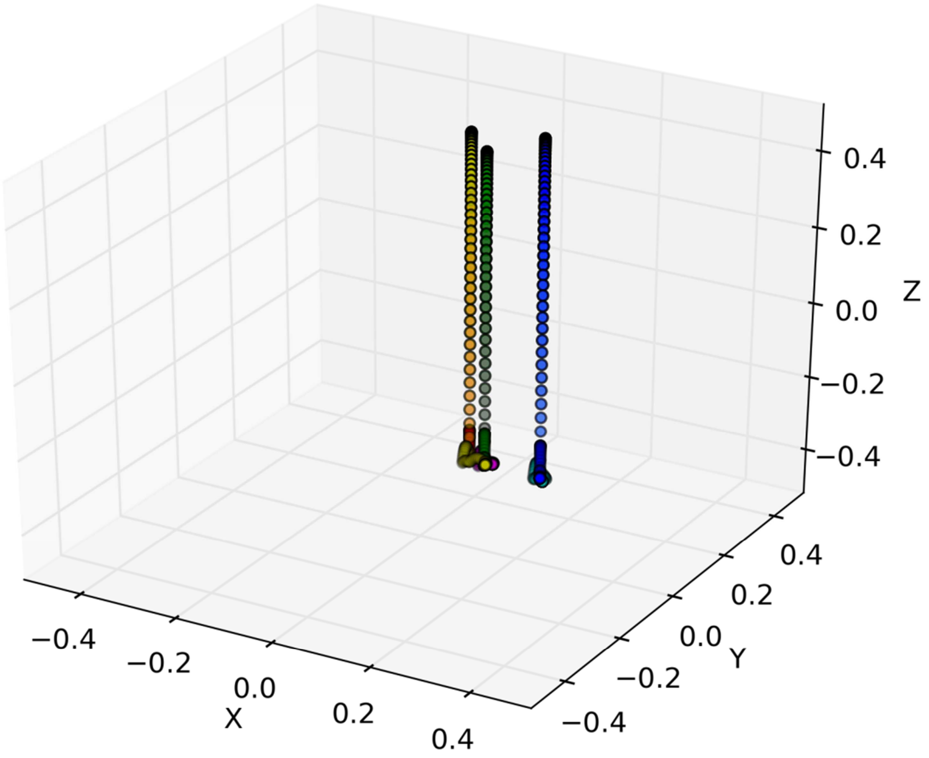

Figure 7-18: Position of the particles at subsequent timesteps in both EDEM and XPS.

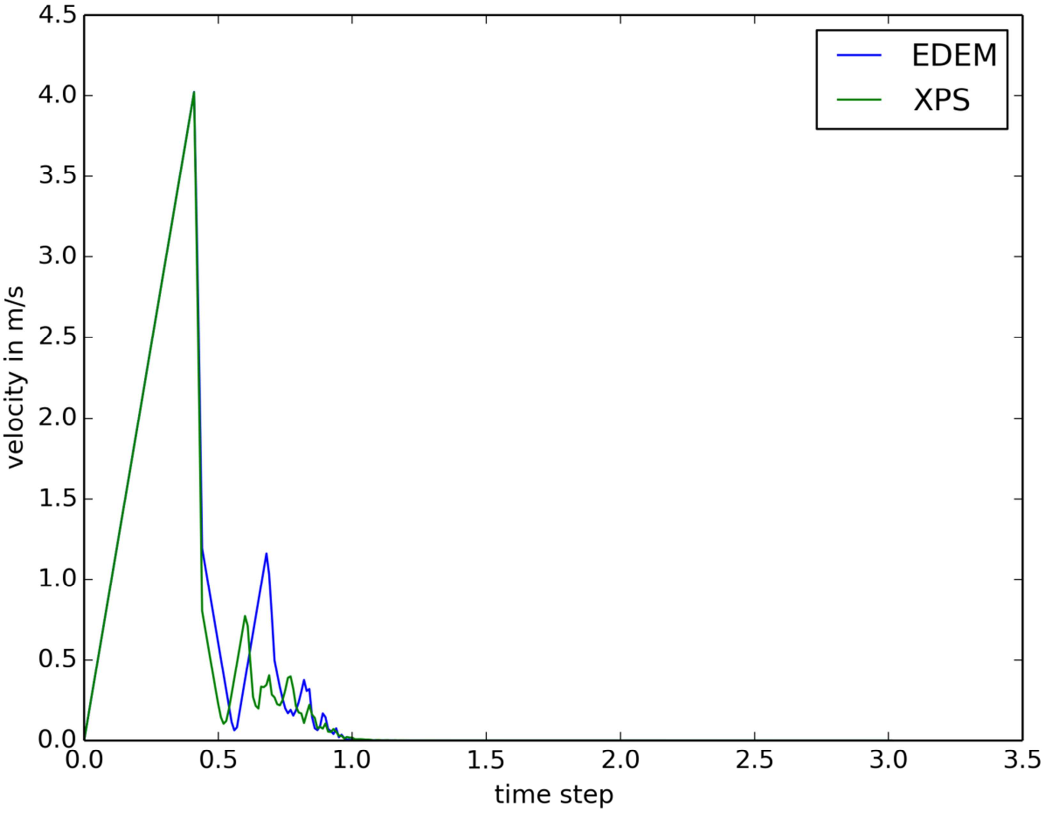

Figure 7-19: Resulting velocities of the XPS and EDEM simulation

Figure 7-18 shows the positions of the tablets during the experiment. Every circle represents a discrete time step. This shows that the positions during the simulation in the EDEM and XPS runs are identical to each other, especially the end positions when the particle is at rest. The magnitude of velocity from the three particles for the XPS and EDEM run are shown in Figure 7-19. It shows that the velocities are not the same but similar to each other in both simulations. The differences can be explained by the different methods of force calculation. XPS uses a linear damper spring system where EDEM uses a nonlinear Hertz – Mindlin model.

7.3.2. Angle of repose

The data retrieved for the repose angle were compared to the results of commercial DEM software (EDEM 2.5).

One of the difficulties is that EDEM uses a different set of properties for their force models. Where XPS uses direct parameters as input (e.g ,spring constant), EDEM uses parameter that describes the material and its interaction with each other (e.g. shear modulus), from which the spring constant is then calculated. The particle parameters and interaction parameters are shown in the Table 7-10. The particle number was set to 16000 and 8000 particles for two and four sub – spheres, respectively and had the same dimension as in the XPS simulations.

The resulting heaps are shown in Figure 7-20 and Figure 7-21.

Table 7-10: Particle parameters and particle interaction parameters

Particle Paramters		
Poission ratio		0,5
shear modulus	in GPa	100
Density	in kg/m^3	1200
Particle Interactions		
Coefficient of Restitution		0,5
Coefficient of Static Friction		0,1
Coefficient of Rolling Friction		0,01

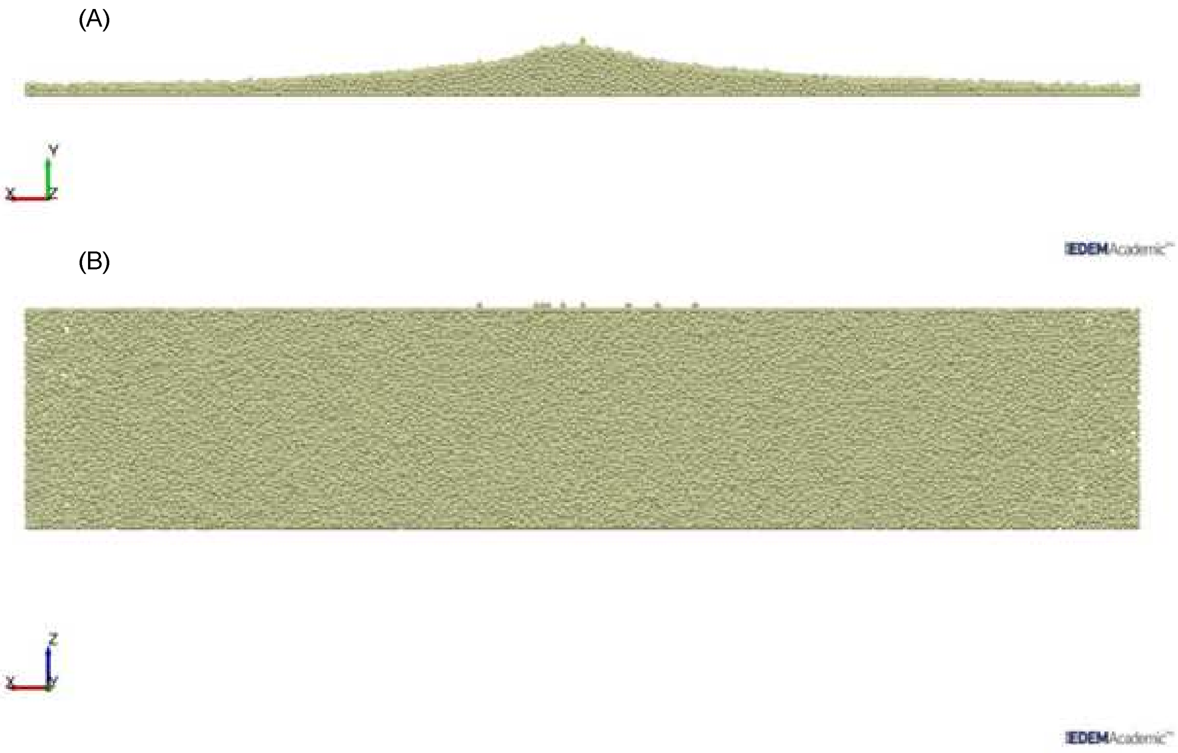

Figure 7-20: EDEM results for two connected spheres seen from the side (A) and above (B)

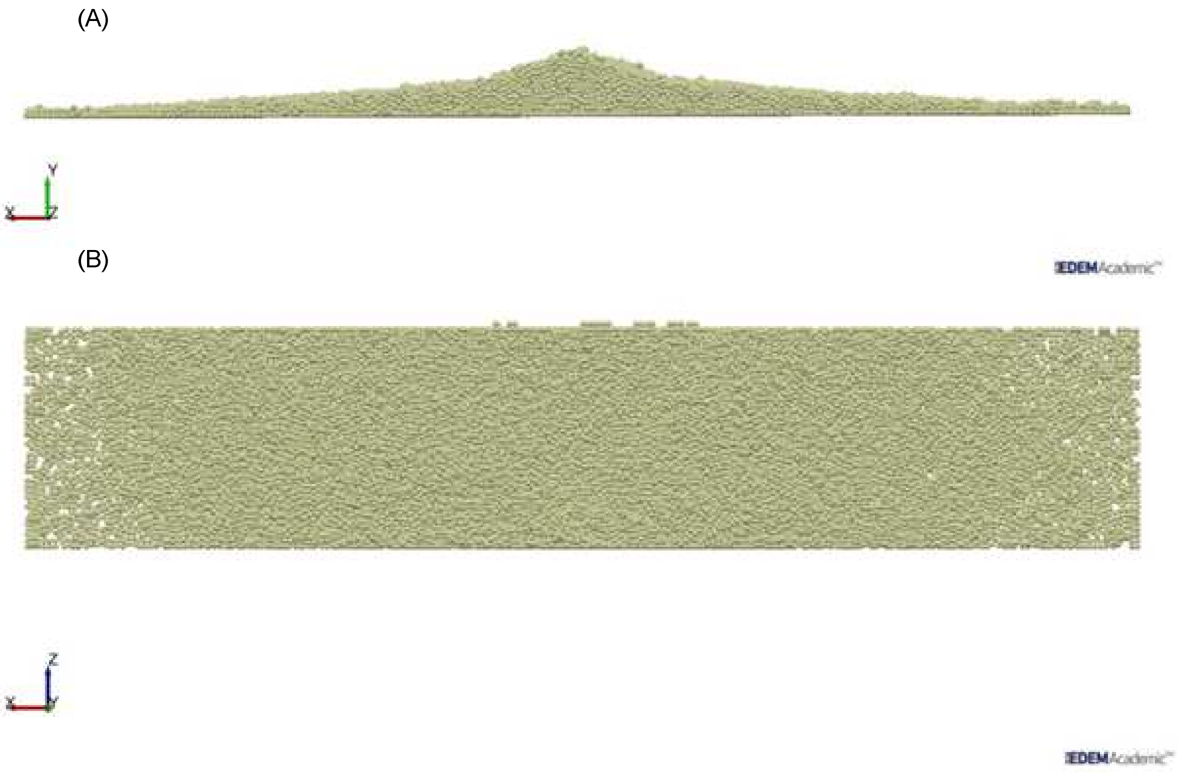

Figure 7-21: EDEM results for four connected spheres seen from the side (A) and above (B)

The EDEM particle parameters were chosen to give equivalent results to the XPS results as shown in chapter 6.2. It was found that similar results can be achieved in the simulation in EDEM and XPS. This can be seen especially by comparing the angle of repose and shape seen in Figure 7-20 compared to Figure 7-15 – 2 for 16000 particles with two sub – spheres and Figure 7-21 compared to Figure 7-16 for 8000 particles and four sub - spheres. Although the results are not completely identical, it shows that at this point a comparison is not a case of whether there are errors in the program itself, but a question of how far the particle interaction parameters can be related to each other. This is a research topic in itself, and not in the scope of this work.

8. Conclusion

The” eXtended Particle System” is still in development. A lot of features are waiting to be implemented. Besides multi element particles these features include polyhedron particles, multi GPU support, a GUI and much more. It is continuously improved, in form of the contact detection or force calculation and other related logarithm.

The aim of this work was to implement a multi element model into the existing “eXtended Particle System” code. The main objective is the possible implementations and the models that can be used in the existing code were reviewed. The secondary objective was to implement the code in such a way that the use of multi element particles does not slow the code down too much e.g. that not too much overhead is produced. The third objective was to implement the model and test it.

The third objective was done in three phases, first testing particle – particle interaction and seeing that no energy is produced, second that the software still can handle a lot of particle - particle interactions and third that it is equivalent to an established commercial DEM software. All these tests were successful and show that the developed models work.

After implementing the code in the main working tree of the development structure, XPS will be better suited to simulate industrial and research processes. With the implemented multi element model, it is possible to simulate full-scale processes with millions of particles with an increased level of accuracy.

9. References

[1] M. H. Abbaspour-fard, “discrete element modelling of the dynamic behaviour of non-spherical particulate materials,” no. September, 2000.

[2] D. Suzzi, S. Radl, and J. G. Khinast, “Local Analysis of the Tablet Coating Process: Impact of Operation Conditions on Film Quality,” *Chem. Eng. Sci.*, vol. 65, no. 21, pp. 5699–5715, Nov. 2009.

[3] I. Holubec and E. D’Appolonia, “Effect of particle slope on the engineering properties of granular soil,” *ASTM Special Technical Publications*, no. 523.

[4] C. P. A. and O. D. L. Strack, “A discrete numerical model for granular assemblies,” *Géotechnique*, vol. 29, no. 1, pp. 47–65, Jan. 1979.

[5] Y. C. Chung, H. H. Liao, and S. S. Hsiau, “Convection behavior of non-spherical particles in a vibrating bed: Discrete element modeling and experimental validation,” *Powder Technology*, vol. 237, pp. 53–66, Mar. 2013.

[6] S. Adam, D. Suzzi, C. Radeke, and J. G. Khinast, “An integrated Quality by Design (QbD) approach towards design space definition of a blending unit operation by Discrete Element Method (DEM) simulation.,” *European journal of pharmaceutical sciences : official journal of the European Federation for Pharmaceutical Sciences*, vol. 42, no. 1–2, pp. 106–15, Jan. 2011.

[7] C. a. Radeke, B. J. Glasser, and J. G. Khinast, “Large-scale powder mixer simulations using massively parallel GPUarchitectures,” *Chemical Engineering Science*, vol. 65, no. 24, pp. 6435–6442, Dec. 2010.

[8] P. W. Cleary, “Industrial particle flow modelling using discrete element method,” *Engineering Computations*, vol. 26, no. 6, pp. 698–743, 2009.

[9] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, and A. B. Yu, “Discrete particle simulation of particulate systems: A review of major applications and findings,” *Chemical Engineering Science*, vol. 63, no. 23, pp. 5728–5770, 2008.

[10] B. Freireich, J. Litster, and C. Wassgren, “Using the discrete element method to predict collision-scale behavior: A sensitivity analysis,” *Chemical Engineering Science*, vol. 64, no. 15, pp. 3407–3416, 2009.

[11] O. R. Walton, “Particle-dynamics calculations of shear flow,” *Presented at US-Japan Seminar on New Models and Constitutive Relations in the Mech. of Granular Mater., Ithaca, N.Y., 22 Aug. 1982*, vol. -1, Aug. 1982.

[12] P. A. Cundall, “Formulation of a Three-dimensional Distinct Element M odel Part I. A Scheme to Detect and Represent Contacts in a System Composed of Many Polyhedral Blocks,” vol. 25, no. 3, pp. 107–116, 1988.

[13] G. HOCKING, “THE DISCRETE ELEMENT METHOD FOR ANALYSIS OF FRAGMENTATION OF DISCONTINUA,” *Engineering Computations*, vol. 9, no. 2, pp. 145–155, Dec. 1992.

[14] J. M. Ting, M. Khwaja, L. R. Meachum, and J. D. Rowell, “An ellipse-based discrete element model for granular materials,” *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 17, no. 9, pp. 603–623, Sep. 1993.

[15] T.-T. Ng and X. Lin, “A three-dimensional discrete element model using arrays of ellipsoids,” *Géotechnique*, vol. 47, no. 2, pp. 319–329, Apr. 1997.

[16] J. R. WILLIAMS and A. P. PENTLAND, “SUPERQUADRICS AND MODAL DYNAMICS FOR DISCRETE ELEMENTS IN INTERACTIVE DESIGN,” *Engineering Computations*, vol. 9, no. 2, pp. 115–127, Dec. 1992.

[17] A. O. Favier, J.F., Abbaspour Fard M.H. , Kremmer, M. and Raji, “Shape representation of axi-symmetrical, non-spherical particles in discrete element simulation using multi element model particles,” *Engineering Computations*, vol. 16, no. 4, pp. 467–480, 1999.

[18] D. Höhner, S. Wirtz, H. Kruggel-Emden, and 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*, vol. 208, no. 3, pp. 643–656, Apr. 2011.

[19] D. Markauskas, R. Kačianauskas, a. Džiugys, and R. Navakas, “Investigation of adequacy of multi-sphere approximation of elliptical particles for DEM simulations,” *Granular Matter*, vol. 12, no. 1, pp. 107–123, Dec. 2009.

[20] G. Toschkoff, S. Just, A. Funke, D. Djuric, K. Knop, P. Kleinebudde, G. Scharrer, and J. G. Khinast, “Spray Models for Discrete Element Simulations of Particle Coating Processes,” *submitted to Chem. Eng. Sci.*, 2013.

[21] D. Suzzi, G. Toschkoff, S. Radl, D. Machold, S. D. Fraser, B. J. Glasser, and J. G. Khinast, “DEM simulation of continuous tablet coating: Effects of tablet shape and fill level on inter-tablet coating variability,” *Chemical Engineering Science*, vol. 69, no. 1, pp. 107–121, 2012.

[22] M. H. Abbaspour-Fard, “Theoretical Validation of a Multi-sphere, Discrete Element Model Suitable for Biomaterials Handling Simulation,” *Biosystems Engineering*, vol. 88, no. 2, pp. 153–161, Jun. 2004.

[23] H. Kruggel-Emden, E. Simsek, S. Rickelt, S. Wirtz, and V. Scherer, “Review and extension of normal force models for the Discrete Element Method,” *Powder Technology*, vol. 171, pp. 157–173, 2007.

[24] Y. Song, R. Turton, and F. Kayihan, “Contact detection algorithms for DEM simulations of tablet-shaped particles,” *Powder Technology*, vol. 161, pp. 32–40, 2006.

[25] H. Hertz, “Ueber die Berührung fester elastischer Körper .,” *Journal fuer reine und angewandte Mathemathik*, pp. 156–171, 1826.

[26] R. Hart, J. Lemos, and M. Engineering, “U.s.a.,” vol. 25, no. 3, pp. 117–125, 1988.

[27] P. Pepiot and O. Desjardins, “Numerical analysis of the dynamics of two- and three-dimensional fluidized bed reactors using an Euler–Lagrange approach,” *Powder Technology*, vol. 220, pp. 104–121, Apr. 2012.

[28] S. Abedi and A. A. Mirghasemi, “Particle shape consideration in numerical simulation of assemblies of irregularly shaped particles,” *Particuology*, vol. 9, pp. 387–397, 2011.

[29] M. Price, “Sphere clump generation and trajectory comparison for real particles,” pp. 1–8, 2007.

[30] A. O. Favier, J.F., Abbaspour Fard M.H. , Kremmer, M. and Raji, “Modelling nonspherical particles using multisphere discrete elements,” *Eng. Mech.*, no. October, pp. 971–977, 2001.

[31] W. Hwu and D. Kirk, “Programming massively parallel processors,” *Special Edition*, 2009.

[32] “CUDA C Programming Guide.” .

[33] “Float example.svg - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/File:Float_example.svg.

[34] “IEEE 754 Double Floating Point Format.svg - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/File:IEEE_754_Double_Floating_Point_Format.svg.

[35] L. Fries, S. Antonyuk, S. Heinrich, and S. Palzer, “DEM–CFD modeling of a fluidized bed spray granulator,” *Chemical Engineering Science*, vol. 66, no. 11, pp. 2340–2355, Jun. 2011.

[36] “DEM Solutions - The leader in bulk materials engineering simulation solutions.” [Online]. Available: http://www.dem-solutions.com/. [Accessed: 14-Feb-2014].

10. Appendix

Particle 0	vx	vy	vz		ωx	ωy	ωz
1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
2,0000	0,1623	-0,0409	0,0000	0,0000	0,0000	0,0000	45,0416
3,0000	0,2470	-0,1196	0,0000	0,0000	0,0000	0,0000	-97,4522
4,0000	0,4269	-0,2573	0,0000	0,0000	0,0000	0,0000	-39,6611
5,0000	0,3784	-0,2050	0,0000	0,0000	0,0000	0,0000	-160,9750
6,0000	0,0492	-0,3407	0,0000	0,0000	0,0000	0,0000	-337,1450
7,0000	0,1895	-0,3646	0,0000	0,0000	0,0000	0,0000	-286,5670
8,0000	0,3391	-0,4846	0,0000	0,0000	0,0000	0,0000	-209,3810
9,0000	0,3796	-0,7185	0,0000	0,0000	0,0000	0,0000	-85,6872
10,0000	0,0000	-1,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Particle 1	vx	vy	vz		ωx	ωy	ωz
1,0000	0,0000	-1,0000	0,0000	0,0000	0	0	0
2,0000	-0,1623	-0,9591	0,0000	0,0000	0	0	45,0416
3,0000	-0,2446	-0,8733	0,0000	0,0000	0	0	-134,234
4,0000	-0,4269	-0,7427	0,0000	0,0000	0	0	-39,6611
5,0000	-0,3784	-0,7950	0,0000	0,0000	0	0	-160,975
6,0000	-0,0492	-0,6593	0,0000	0,0000	0	0	-337,145
7,0000	-0,1895	-0,6354	0,0000	0,0000	0	0	-286,567
8,0000	-0,3391	-0,5154	0,0000	0,0000	0	0	-209,381
9,0000	-0,3796	-0,2815	0,0000	0,0000	0	0	-85,6872
10,0000	0,0000	0,0000	0,0000	0,0000	0	0	1,89E-07

Particle 0	vx	vy	vz		ωx	ωy	ωz
1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
2,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	-0,0128
3,0000	-0,2843	0,0875	0,0000	0,0000	0,0000	0,0000	-21,9039
4,0000	-0,5071	0,0398	0,0000	0,0000	0,0000	0,0000	-322,8130
5,0000	-0,0161	0,0479	0,0000	0,0000	0,0000	0,0000	-83,1734
6,0000	-0,0161	-0,0479	0,0000	0,0000	0,0000	0,0000	83,1734
7,0000	-0,5071	-0,0398	0,0000	0,0000	0,0000	0,0000	322,8130
8,0000	-0,2843	-0,0875	0,0000	0,0000	0,0000	0,0000	21,9039
9,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0128
10,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Particle 1	vx	vy	vz		ωx	ωy	ωz
1,0000	0,0000	-1,0000	0,0000	0,0000	0,0000	0,0000	0,0000
2,0000	-0,3716	-0,9284	0,0000	0,0000	0,0000	0,0000	-0,0128
3,0000	-0,4058	-0,8113	0,0000	0,0000	0,0000	0,0000	-21,9036
4,0000	-0,4025	-0,4552	0,0000	0,0000	0,0000	0,0000	-322,8130
5,0000	-0,9828	-0,0954	0,0000	0,0000	0,0000	0,0000	-83,1734
6,0000	-0,9828	0,0954	0,0000	0,0000	0,0000	0,0000	83,1734
7,0000	-0,4025	0,4552	0,0000	0,0000	0,0000	0,0000	322,8130
8,0000	-0,4058	0,8113	0,0000	0,0000	0,0000	0,0000	21,9036
9,0000	-0,3716	0,9284	0,0000	0,0000	0,0000	0,0000	0,0128
10,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000