Chair of Mining Engineering and Mineral Economics

Master's Thesis

A DEM Model for Complex-Shaped Particles to Simulate Breakage in Large-Scale Simulations

Felix Platzer, BSc.

November 2022

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

AFFIDAVIT

I declare on oath that I wrote this thesis independently, did not use other than the specified sources and aids, and did not otherwise use any unauthorized aids.

I declare that I have read, understood, and complied with the guidelines of the senate of the Montanuniversität Leoben for "Good Scientific Practice".

Furthermore, I declare that the electronic and printed version of the submitted thesis are identical, both, formally and with regard to content.

Date 07.11.2022

Signature Author
Felix Platzer

Title:	A DEM Model for Complex-Shaped Particles to Simulate Breakage in Large-Scale Simulations
Author:	Felix Platzer
Year / Place:	2022 / Leoben, Austria
DOI:	https://doi.org/10.34901/mul.pub.2022.06

Chair of Mining Engineering and Mineral Economics – Conveying Technology and Design Methods at the University of Leoben (Montanuniversität Leoben).

Submitted by
Felix Platzer, BSc.

Supervisors:

Nikolaus August Sifferlinger (*Montanuniversität Leoben*)
Eric Fimbinger (*Montanuniversität Leoben*)

Copyright declaration

Acknowledgements

At this point I would like to thank everyone who helped and supported me during my work on this thesis.

I would first like to thank my supervisors, Univ.-Prof. Dipl.-Ing. Dr.mont. Nikolaus August Sifferlinger and Dipl.-Ing. Dr.mont. Eric Fimbinger, whose helpful suggestions and insightful feedback were invaluable in the preparation of this work. Thank you for the support and the motivation to publish my work and thus promote me in my career.

In addition, I would like to acknowledge the entire team at the Chair of Mining Engineering – Conveying Technology and Design Methods, around the head of the institute, Univ.-Prof. Dipl.-Ing. Dr.mont. Nikolaus August Sifferlinger, who, thanks to their high level of commitment, encourage students to write scientific work and who also supported the publication of my work.

I would like to express my deepest gratitude to my parents, who supported me from an early age and shaped me into who I am today.

Last but not least, I would also like to thank FLSmidth, who financed the project that made my work possible and approved the publication of project data.

Abstract

In Discrete Element Method (DEM) simulations, the breakage behavior of particles can be simulated based on different principles. In the case of large, complex-shaped particles that show various breakage patterns depending on the scenario leading to the failure and often only break locally instead of fracturing completely, some of these principles do not lead to realistic results. The reason for this is that in said cases, the methods in question, such as the Particle Replacement Method (PRM) or Voronoi Fracture, replace the initial particle (that is intended to break), also called the parent-particle, into several sub-particles when certain breakage criteria are reached, such as exceeding the fracture energy. That is why those methods are commonly used for the simulation of materials that fracture completely instead of breaking locally.

When simulating local particle failure, it is advisable to pre-build the initial particle from sub-particles that are bonded together. The dimensions of these sub-particles consequently define the minimum size of the fracture results. This structure of bonded sub-particles enables the initial particle to break at the locations of the highest local loads – due to the failure of the bonds in those areas – with several sub-particle clusters being the result of the fracture, which can again also break locally.

In this thesis, different methods for the generation and calibration of complex-shaped particle conglomerates using bonded-particle modeling (BPM) to enable the ability to depict more realistic fracture behavior are evaluated based on the example of rock-like materials, in this context specifically filter cake. The detailed method that proved suitable for this purpose and which furthermore allows efficient and realistic simulation of breakage behavior of complex-shaped particles is presented in this thesis.

In order to simulate industrial-sized processes with several differently shaped parent-particles, following major aspects, among others, are taken into account. A correlating mass and volume flow during the breakage process is maintained, enabled by a approach using large overlaps in combination with a relaxation model after breakage of the bondings. Dynamic behavior of the parent-particles especially regarding bonding-structure-wide damping is considered. Also, the detection of different

sub-particle clusters resulting from the breakage of bonded-particle structures is of major interest and is thus covered respectively. Optimization in regard to computational efficiency is further achieved by introducing the "bonded-particle replacement method" (BPRM), combining the advantages of the BPM with those of the PRM.

By developing virtual prototypes instead of physical ones, and furthermore reducing the required efforts in context with these virtual prototypes and their successful simulation, the required resources are reduced and a step towards sustainability is taken.

Kurzfassung

Bei Diskreten-Elemente-Methode (DEM) Simulationen kann das Bruchverhalten von Partikeln mit verschiedenen Bruchmodellen simuliert werden. Bei großen, komplex geformten Partikeln, welche je nach Szenario das zum Versagen der Partikel führt unterschiedliche Bruchmuster aufweisen und oft nur lokal anstatt vollständig zerbrechen, führt die Anwendung einige dieser Modelle nicht zu realistischen Ergebnissen. Der Grund dafür ist, dass in diesen Szenarien die infrage kommenden Methoden, wie die „Particle Replacement Method“ (PRM) oder „Voronoi Fracture“, bei dem Erreichen eines bestimmten Bruchkriteriums, wie zum Beispiel das Überschreiten der Bruchenergie, den bruchfähigen Partikel, auch Elternpartikel genannt, durch mehrere Subpartikel ersetzen. Deshalb werden diese Methoden üblicherweise für die Simulation von Materialien verwendet, welche vollständig zerbrechen, anstatt lokal zu versagen.

Bei der Simulation von lokalem Versagen ist es daher ratsam, dass der bruchfähige Ausgangspartikel bereits im Vorhinein aus mittels „Bonds“ verbundenen Subpartikeln generiert wird. Die Abmessungen dieser Subpartikel bestimmen folglich die kleinstmögliche Größe der Bruchfragmente. Diese Struktur aus miteinander verbundenen Subpartikeln ermöglicht es, dass der Ausgangspartikel an den Stellen der höchsten lokalen Belastung bricht – aufgrund des Versagens der „Bonds“ in diesen Bereichen – wobei mehrere Subpartikelpartikel-Cluster das Ergebnis des Bruchs sind, welche folglich ebenso lokal brechen können.

In dieser Arbeit wurden Anhand des Beispiels von Filterkuchen verschiedene Methoden zur Erzeugung und Kalibrierung komplex geformter Partikelkonglomerate mittels „Bonded-Particle Modeling“ (BPM) verglichen, um ein realistischeres Bruchverhalten abbilden zu können. Jene Methode, welche sich für diesen Zweck als geeignet erweist und darüber hinaus eine effiziente und realistische Simulation des Bruchverhaltens komplex geformter Partikel ermöglicht, wird anschließend näher erläutert.

Um Prozesse im industriellen Maßstab mit mehreren unterschiedlich geformten Elternpartikel simulieren zu können, werden folgende wesentliche Aspekte, unter anderen, berücksichtigt. Es wird ein korrelierender Massen- und Volumenstrom während des Bruchvorgangs aufrechterhalten, was durch

einen Ansatz mit großen Überlappungen in Kombination mit einem Entspannungsmodell nach dem Bruch der „Bonds“ ermöglicht wird. Das dynamische Verhalten der Elternteilchen wird insbesondere hinsichtlich der Dämpfung berücksichtigt. Die Möglichkeit der automatischen Erkennung von einzelnen Subpartikelclustern, die aus dem Aufbrechen von gebundener Partikelstrukturen resultieren, ist von großem Interesse und wird entsprechend behandelt. Optimierung im Hinblick auf die Recheneffizienz wird durch die Einführung der Bonded-Particle Replacement Method (BPRM) erreicht, die die Vorteile der BPM mit denen der PRM kombiniert.

Durch die Entwicklung virtueller anstelle von physischen Prototypen, und eine zusätzliche Reduzierung des Aufwandes im Zusammenhang mit diesen virtuellen Prototypen und deren erfolgreicher Simulation, werden die benötigten Ressourcen reduziert und ein Schritt in Richtung Nachhaltigkeit getan.

Contents

1 Introduction

From today's perspective, it is impossible to imagine hardly any industry sector without numerical simulation. Whether in the development of new types of processes or the optimization of existing ones, a wide variety of calculation methods can be applied. These simulation methods have been further developed and refined over the years to increase the accuracy of the simulation results and computational efficiency. In this thesis, the breakage behavior of filter cakes or similar brittle materials in arbitrary shapes are simulated using state-of-the-art Discrete Element Method (DEM) models. The application is particularly but not exclusively aimed at the sectors of bulk material handling and process engineering.

1.1 Motivation and project

During many industrial processes, the particles of bulk solids with various material properties and shapes are broken down into smaller fragments, which can further break again. In many cases, this breakage is undesirable, such as the degradation of sinter during conveyance of this type of bulk material to the blast furnace since too fine-grained particles impede a sufficient gas flow in the furnace [1], when exceeding the maximum load-bearing capacity of building materials, such as concrete, in the construction industry [2] [3], or the damage of supporting rock structures in mining engineering [4] [5]. However, there are also many areas in which material breakage is desired, such as relating to crushing, grinding, or drilling rock in the areas of mining engineering and mineral processing [4] [6] [7], handling cemented sands or heavily overconsolidated soils [8], or the cutting and threshing of agriculture products during harvesting [9] [10], only to name a few.

The work of this thesis was done in the course of a project with the tailings management department of FLSmidth USA Inc. which, among other things, deals with the storage of residuals from the extraction of the target mineral from an ore, called tailings. These fine- grained residues in the form of a slurry are most of the time stored within reservoirs contained by a tailings dam. However, failure of these tailings management facilities is one of the worst environmental liabilities that can occur during a mining project. For this reason, many mining companies are looking for lower-grade deposits, which require higher throughputs to ensure profitability [11].

With water becoming more difficult to access per year at many mining sites, the recovery of a significant amount of reusable water from the tailings during the fast-filtering process is crucial. FLSmidth, in cooperation with GoldCorp, has spent the past several years developing a process for blending concentrator tailings, called $EcoTails^{TM}$, with mine waste in a continuous process to produce a stable deposit, called $GeoWaste^{TM}$. In doing so, $GeoWaste^{TM}$ is expected to be cost-competitive with conventional tailings disposal methods [11]. In this case, the continuous process of mixing occurs while the materials are being transported from the processing plant to the disposal site. For this purpose, the development of a chute optimized for blending the transported materials, which is used at transfer points of two belt conveyors, is essential. The continuous process, in which the concentrator tailings are mixed with the waste rock to $GeoWaste^{TM}$ within the transfer chutes during the conveying process is shown in figure 1.

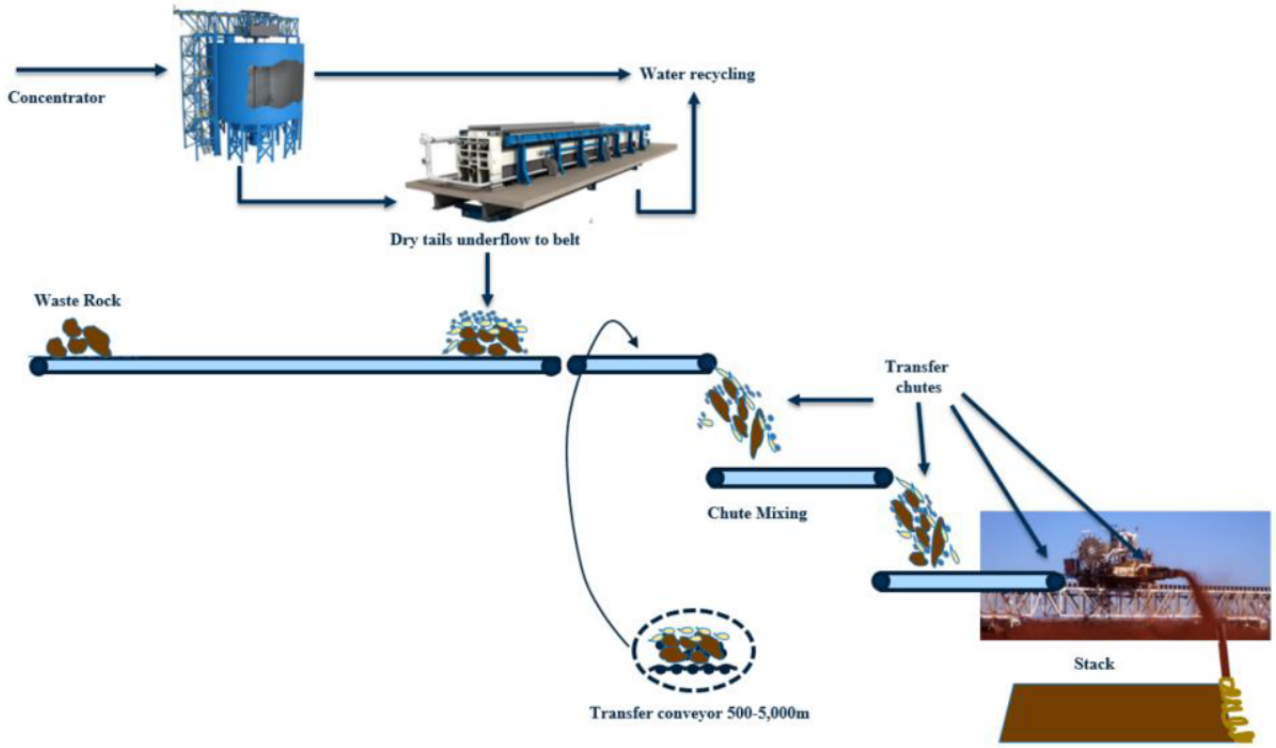

Figure 1: Conveying process of the concentrated tailings from the processing plant to the disposal site during which the $EcoTails^{TM}$ are mixed with mine waste, producing $GeoWaste^{TM}$ [11]

Nowadays, numerical simulations are indispensable for the optimization of processes, as they allow a fast and accurate insight into said processes. For this reason, in this project, the conveying process is to be simulated in order to numericaly represent the breakage behavior of the concentrator tailings as well as the continuous blending of the fracture results with mine waste. In the field of conveying technology, the Discrete Element Method (DEM) is ideally suited for the simulation of bulk materials.

Since in this project the mixing of a complex-shaped, breakable particle with waste rock was to be simulated, several possibilities had to be considered to represent breakable particles in a simulation environment on an industrial scale.

1.2 Scope and objectives

This thesis aims to introduce a guideline for the simulation of complex-shaped breakable particles applicable in industrial-scale simulations using the DEM. This covers several objectives:

Objective 1:
The main objective of this thesis is to depict the macroscopic breakage behavior of filter cake material (and the like) in a sufficient way, providing a way to numerically predict if and where such filter cake plates fracture in process-like situations. It is furthermore said that this approach is thus not intended to replicate the exact microscopic crack propagation during breakage, as the focus is set on depicting effects from a bulk-oriented perspective. The material considered in this thesis is defined as rock-like due to the low moisture content of the filter cake, whereby rock-like materials are understood to be materials that exhibit the same failure criteria as rock and, above all, brittle material behavior after a relatively small initial elastic deformation.

Objective 2:
Implementation of an efficient and simple – in terms of operation and computing time – generation process of arbitrarily-shaped (bonded) particle structures specified by a geometry given by a 3D-CAD file.

Objective 3:
The measurement of data required for the calibration of the simulation in laboratory tests by selecting a test procedure suitable for both: the material considered and the final process to be simulated.

Objective 4:
Calibration of the applied breakage model, as well as the determination of the simulation parameters with the greatest influence on the simulation results, thus enabling an efficient calibration of future simulations via this thesis also being a guideline for the calibration process.

Objective 5:

Consideration of several aspects to enable the simulation of not only lab-scale, but also industrial-sized processes.

The following section 2 describes the different approaches to simulate particle breakage and describes the DEM model used in this thesis in more detail. In section 3, the methodology of generating complex-shaped particles, acquiring the data required for the calibration of the simulation model, as well as some other aspects, are discussed. The results of this thesis, regarding the implementation of the concepts within the simulation software, obtained data from the laboratory test, the calibration of the simulation model, as well as optimization efforts, are presented in section 4. The final section 5 provides a summary of the results, the conclusions drawn from this thesis, and an outlook on further potentials.

1.3 Prior publications

Several contents of this thesis have previously been published, particularly at the 16^{th} International Conference on Discrete Multiphysics, Modelling Complex Systems with Particle Methods and Discrete Element Method, Virtual Conference of the World Academy of Science [12]

2 General approaches of simulating particle breakage

To depict filter cake breakage, the Discrete Element Method (DEM) [13], also called Distinct Element Method [14], is ideal since in many other methods the simulated material is considered as a homogeneous continuum, whereas in the DEM it is represented as discrete and inhomogeneous [15], which is required for the case in question.

There are several other simulation methods capable of simulating the deformation and fracture behavior of complex-shaped particles, such as FEM or XFEM [16] [17]. Even though these methods can depict crack propagation in a very detailed way, they are of no interest since the behavior of thousands of breakable particles and their interactions can be depicted in a much more suitable way with a particle-based method, and only the macroscopic material failure is of relevance. Coupled FEM-DEM approaches, also called FDEM, are an option in which the continuum, meaning the particle before fraction, is simulated using FEA, while the interaction between the fracture results is analyzed using the DEM [18] [19]. This method requires software that does include not only standard DEM and FEM solvers but also additional algorithms to transition the simulated material from continua to discontinua [19]. FEM software often uses an implicit solver, in contrast to DEM software which usually uses an explicit solver, bringing up another aspect to consider if coupling is required. Since the DEM is sufficient for the scope of this thesis, these additional requirements are not needed.

In DEM software, the change of motion and position of particles between discrete timesteps is computed based on the forces and torques acting on said particles using the laws of motion. These forces can be divided into general forces ($F_{General}$), forces due to gravity or force fields, and contact forces ($F_{Contact}$) resulting from interactions of a particle with other particles or system components.

$$F_{Particle} = F_{Contact} + F_{General} \tag{1}$$

Furthermore, the contact force is divided into master contact force (F_{Master}) and slave force (F_{Slave}), which will be superimposed by means of superposition.

$$F_{Contact} = F_{Master} + F_{Slave} \tag{2}$$

The master contact force corresponds to the sum of forces resulting from the basic contact model, or master contact model, in the normal and tangential direction of particles in contact (cf. [20]). The most commonly used Master contact model in DEM is the Hertz-Mindlin model, which is also used in this thesis. It is based on the Hertzian Contact [21], for the interaction of particles with other particles or system components in the normal direction, and on the interaction effects regarding tangential loading as according to Mindlin [22] [23]. This non-linear elastic contact model computes the forces in the normal and the tangential direction according to two spring-dashpot models in the corresponding orientation in combination with Coulomb friction and rolling friction. A visualization of this contact model is shown in figure 2.

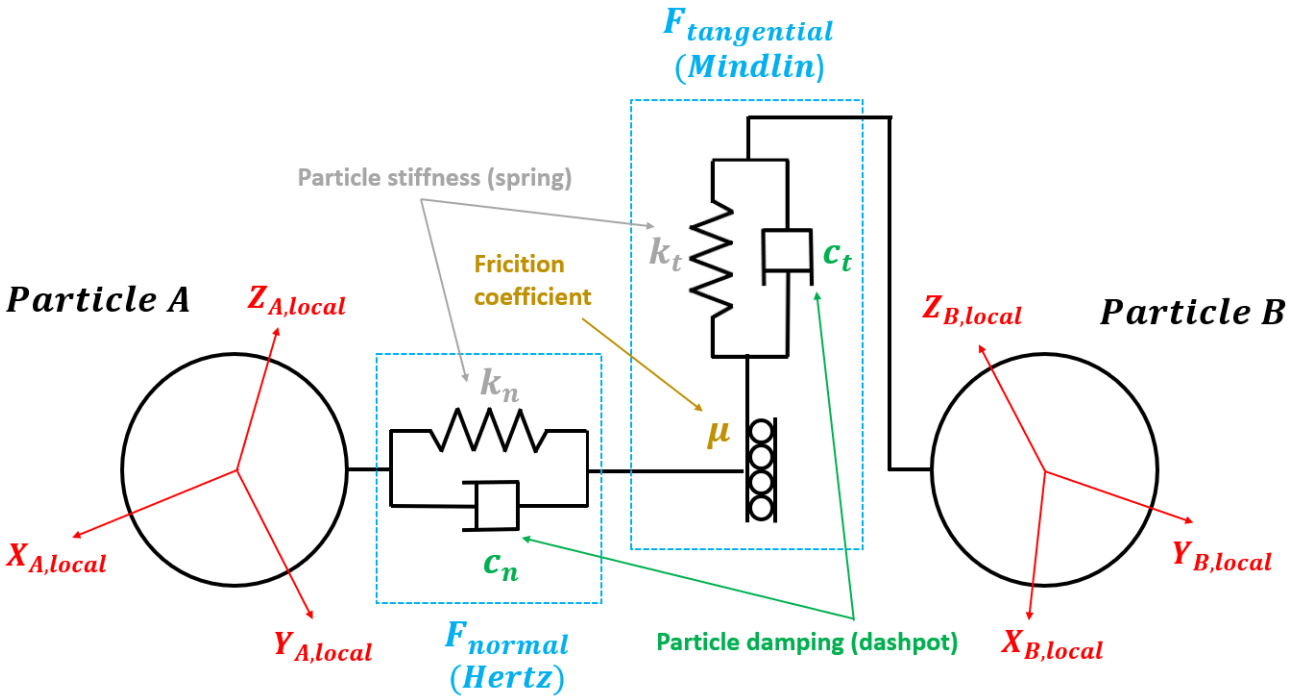

Figure 2: Schematic visualization of the Hertz-Mindlin contact model showing the spring-dashpot model in normal and tangential direction, as well as the friction coefficient

Since the calibration of this Master contact model by means of standardized tests is deemed as trivial, the Equations of the contact forces are not listed, except for the spring-back force in normal direction, as this is required later on, which is calculated from

$$F_{normal} = \frac{4}{3} E^* \sqrt{R^* \delta_n^{\frac{3}{2}}} \tag{3}$$

with the overlap in normal direction δ_n and the Equivalent Young's Modulus E^* and equivalent radius R^* according to equation 4 to equation 5

$$\frac{1}{E^*} = \frac{1-\nu_i^2}{E_i} + \frac{1-\nu_j^2}{E_j} \tag{4}$$

$$\frac{1}{E^*} = \frac{1}{R_i} + \frac{1}{R_j} \tag{5}$$

with the Poisson's ratio ν and the indices i and j corresponding to the two particles in contact. Additionally slave contact models can be added to represent various physical phenomena, such as cohesion acting between moist particles, or in this case: a physical connection of particles making up a continuum. In this thesis, the DEM-based Multiphysics simulation software *ThreeParticle/CAE* by *BECKER3D* [24] was used for simulation.

2.1 Existing particle breakage models in the DEM

The two most common models for simulating particle breakage using the DEM are addressed below, also already outlining the suitability of the bonded-particle modeling (BPM; section 2.1.2) as the base approach for the following developments.

2.1.1 Particle Replacement Model (PRM)

The PRM, also referred to as fragment replacement method, instantaneously replaces "as-broken-defined" particles with a set of unconnected sub-particles when a certain criterion for failure is met, e.g., exceeding a specific breakage energy limit [25]. In addition to the particle breakage criterion, the size of the sub-particles, depending on the scenario leading to the failure, as well as the packing configuration of the fragments, are to be defined [26]. By initially simulating with a small number of breakable particles and only increasing the particle number if a material failure occurs, the PRM requires less computational effort than the BPM (as discussed in the following section 2.1.2), since in the DEM the computation time significantly increases with the number of particles, requiring more RAM and faster CPU's or GPU's. Depending on the shape of the sub-particles, a further sub-categorization of this method is possible.

2.1.1.1 Spherical sub-particles

To ensure mass conservation, the spherical sub-particles replacing the particle after reaching the breakage criterion must be initialized with an initial superposition. High repulsive forces resulting from this initial overlap leading to an unstable simulation are avoided by adding a relaxation factor until the particles are separated. This replacement process is visualized in figure 3.

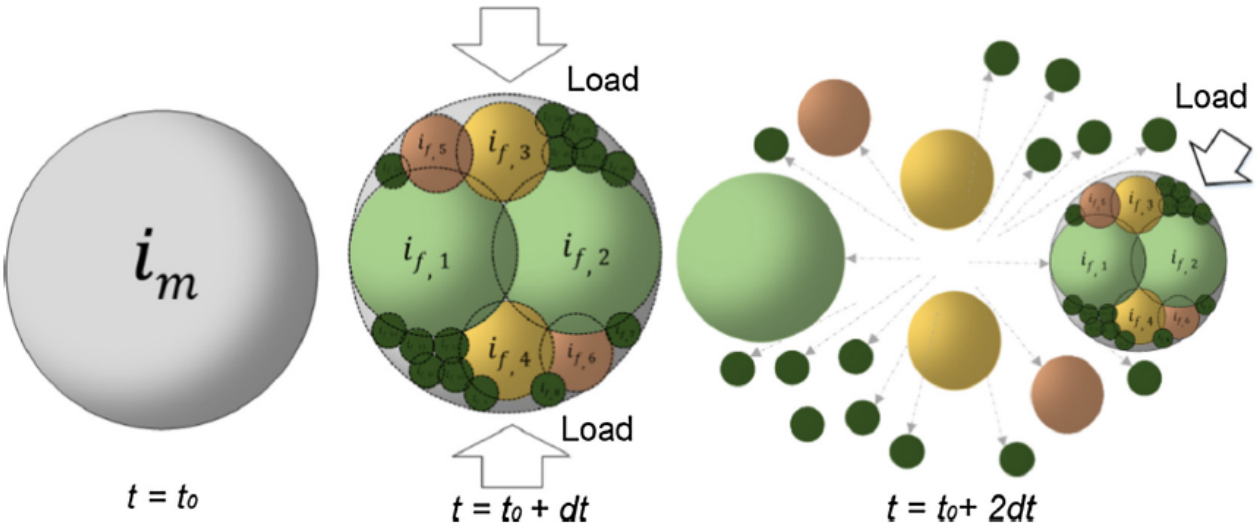

Figure 3: Schematic depiction of the PRM [25]

2.1.1.2 Complex shaped sub-particles

The packing configuration in this method is most commonly defined by a Voronoi tessellation [25] [27], segmenting the breaking particle into polygonal sub-particles, as shown in figure 4. This results in an indefinite conservation of mass and volume after the breakage process, compared to using spherical sub-particles. Depending on the breakage energy, a different tessellation, resulting in various size classes of the fragments, is calculated. However, these complex particle shapes come at the price of a reduction of computational efficiency by increasing contact detection time and requiring smaller time steps.

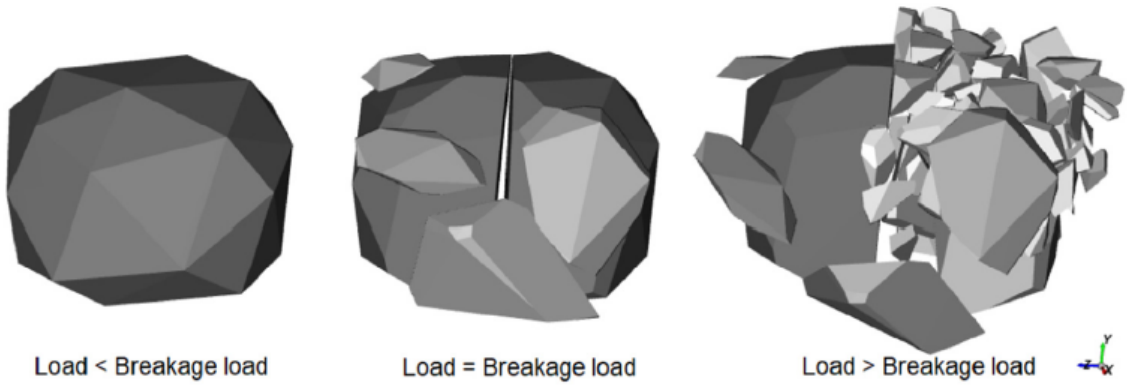

Figure 4: PRM with polygonal sub-particles resulting from a Voronoi tessellation; Figure adapted from Fig. 8 in [25]

2.1.2 Bonded-Particle Model (BPM)

In this method, two discrete particles are joined together with a virtual connection called a bond, also known as a joint or bonding. This element has neither mass nor volume, meaning that the bond does not have any inertia and cannot interact with other bonds or particles, except the two particles connected with said bond. Bonds exert loads (forces/torques) on the two particles joined by the bond as they deviate from their original relative positions. When a certain criterion is reached, the bond breaks and is deleted from the simulation, resulting in local material failure of the parent-particle in the area the bond was located. A bond connection in its initial state and under deformation is depicted in figure 5.

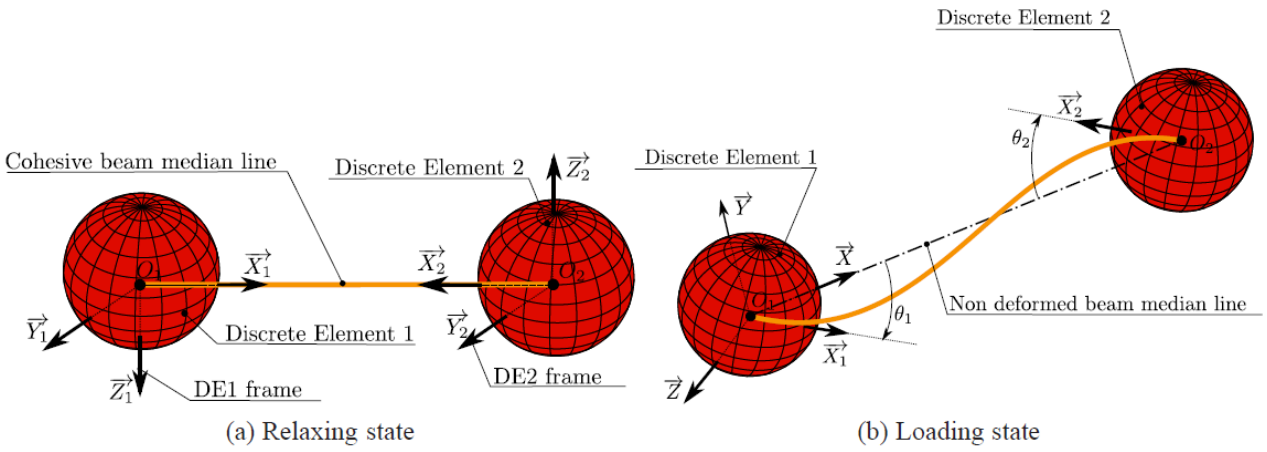

Figure 5: Bond element connecting two particles in its initial state (left) and in the loaded state (right) [28]

If several sub-particles are bonded together to form a cluster, a complex particle shape can be represented as such a bonded-particle network, as shown in figure 6 on the example of an infinity knot.

In figure 6, as well as in the following figures concerning bondings, the particles are depicted in grey, with the shade of grey visualizing the particle size distribution (PSD), and the bonds depicted as black lines.

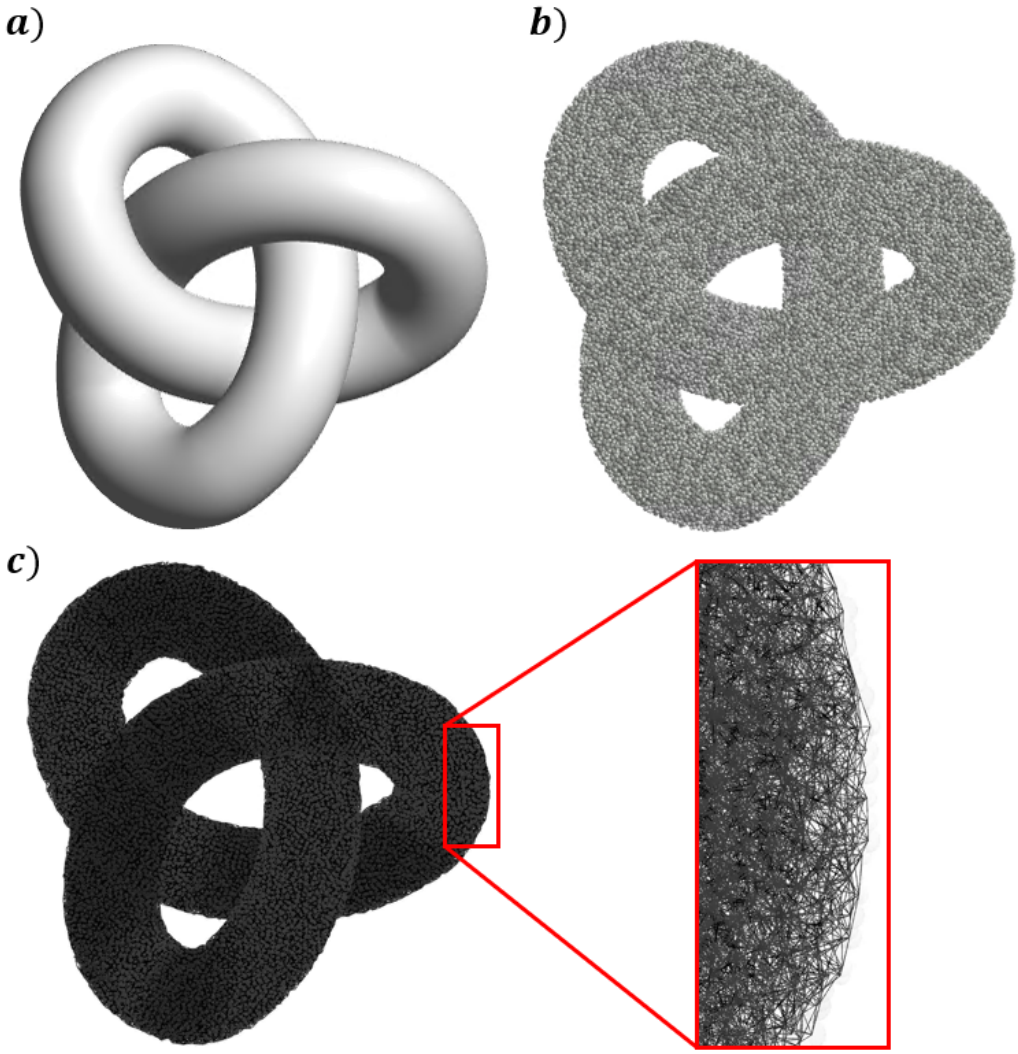

Figure 6: Visualization of an arbitrary geometry (a) represented as a cluster of sub-particles (b) interconnected with bonds, forming a bonded-particle network (c)

The main disadvantage of this method, when simulating a breakable particle, is that the parent-particle must be pre-build from sub-particles corresponding to the smallest achievable fracture fragments, which increases the number of overall particles in the simulation and thus also the computation time.

2.1.3 Conclusive Assessment

Even though of the two main methods that can be used in the DEM to simulate particle breakage, the PRM is more computationally efficient, only the BPM is suited to represent local material failure under preceding deformation [8] [15] [29], as required, which is why this method was selected for this thesis.

2.2 The BPM approach

To depict the macroscopic behavior of the material correctly, the influence of the bonds on the bonded-particle cluster and how they are affected by their model parameters must be studied. Therefor the BPM used in this thesis is described in more detail.

To depict the macroscopic behavior of the material correctly, the influence of the bonds on the bonded-particle cluster and how they are affected by their model parameters must be studied. Therefor the BPM used in thesis is described in more detail.

In ThreeParticle, an extended model of the classic Timoshenko beam theory [30] [31] based on cemented sand [8] is available as a slave contact model. The bonds of this model can transmit tensile, compressive, and shear forces, as well as torque and bending moments. The forces are calculated in a local bond coordinate system with the x-axis corresponding to the bond axis, as seen in figure 5. For this reason, all quantities required for the calculation of the bond forces and moments, such as the strain Γ, the curvature of the bonds κ, the translational as well as the relative rotational velocity v_{12} and Ω_{rel}, are expressed in local bond coordinates.

The reaction forces $F_{s,ax}$ are calculated from the Youngs Modulus E_b, the cross-section A_b, the shear coefficient α_s, and the shear modulus of the bond G_b, according to equation 6

$$F_{s,ax} = \begin{bmatrix} E_b A_b & & \\ & \alpha_s G_b\, A_b & \\ & & \alpha_s G_b\, A_b \end{bmatrix} \Gamma \tag{6}$$

The viscous damping force F_d is then

$$F_d = d\ v_{12} \tag{7}$$

with the damping coefficient d. The torque acting between the bonded particles $T_{b,t}$ is calculated from

$$T_{b,t} = \begin{bmatrix} 2G_b\ J_b & & \\ & E_b J_b & \\ & & E_b J_b \end{bmatrix} \kappa \tag{8}$$

where J_b corresponds to the second moment of inertia of the beam cross-section. The damping torque T_d corresponds to

$$T_d = \begin{bmatrix} \sqrt{\frac{2G_b\ J_b}{l_0} I_r} & & \\ & \sqrt{\frac{E_b J_b}{l_0} I_r} & \\ & & \sqrt{\frac{E_b J_b}{l_0} I_r} \end{bmatrix} \Omega_{rel} \tag{9}$$

with the initial bond length l_0 and the reduced moment of inertia I_r, which is calculated from the individual moments of inertia of the particles connected with the bond according to

$$I_r = \frac{I_1 I_2}{I_1 + I_2} \tag{10}$$

For the calculation of all the above-mentioned quantities, a circular bond cross-section is assumed. According to the beam theory, the stresses can be calculated on the basis of the strain and curvature of the bond element

$$\sigma_{s,ax} = \begin{bmatrix} E_b & & \\ & G_b & \\ & & G_b \end{bmatrix} \Gamma \tag{11}$$

$$\sigma_{t,b} = \begin{bmatrix} G_b & & \\ & E_b & \\ & & E_b \end{bmatrix} \kappa r_b \tag{12}$$

where r_b corresponds to the radius of the bond.

The equivalent stress is calculated using the von Mises yield criterion (cf. equation 13) [32]. It should be noted that due to the material being much more sensitive to tensile stress, the normal stress component is only considered under tension and not under compression.

$$\sigma_v = \sqrt{{\sigma_{axial}}^2 + 3{\tau_{shear}}^2} \tag{13}$$

When a critical equivalent stress $\sigma_{v,krit}$ is reached, the bond is deleted, and the bonded particles experience no additional reaction forces based on this slave contact model henceforth. Due to the forces and moments being expressed in a local bond coordinate system, a final transformation into global coordinates is required.

2.2.1 Advanced BPM models

More detailed bond behavior, especially on a microscopic scale, requiring more calibration effort and computational power, is not called for in this thesis. Several existing bond models considering the most common advanced requirements are listed below, as a more detailed depiction of the bonds may be required in future work based on this thesis:

2.2.1.1 Bond damage

The macroscopic properties of a solid body are given to a bonded-particle network by its individual bonds. An increase in the bond's stiffness results in an increase in the parent-particles overall stiffness, and a deleted bond acts like a crack within the material. Stress concentration in the immediate vicinity of a deleted bond, due to the absence of a bond that acts as a notch, often leads to an immediate failure of the neighboring bonds. However, this abrupt crack growth through the entire parent particle often does not correspond to the behavior of the physical material, where crack propagation often occurs much slower. One reason for this is that when a bond is deleted, a crack with a length dependent on the distance to the next bond occurs abruptly instead of growing steadily. One way to overcome this issue is to use an accumulated damage model, commonly known as a cohesive zone model [33] [34] [35]. This accounts for damage to a bond after a specified load is exceeded, resulting in bond softening and therefore a steady local weakening of the complex-shaped parent-particle. The process of bond damaging is illustrated in figure 7.

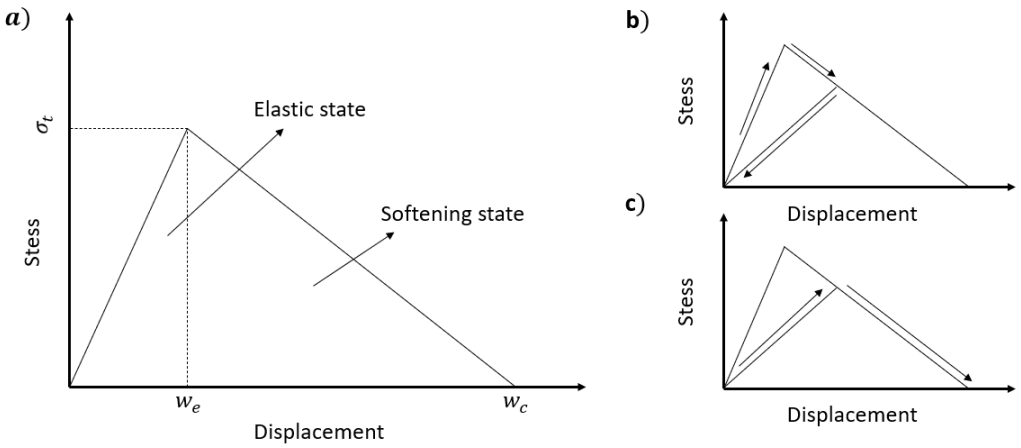

Figure 7: Cohesive zone model, considering bond softening after exceeding the tensile stress σ_t at the opening displacement w_e and bond failure at the critical displacement w_c (a); the damage remains in the bond after unloading (b), resulting in a different stress-displacement, and also force-displacement, curve in future loading cases(c), cf. [33]

2.2.1.2 Flat-joint model

Developed for application in rock mechanics, the flat-joint model [36] discretizes the connection, also called the interface, between the joined particles into several elements, each acting as a single bond capable of breaking. This allows partial damage of the interface, acting as a sub-level crack within the connection, as shown in figure 8.

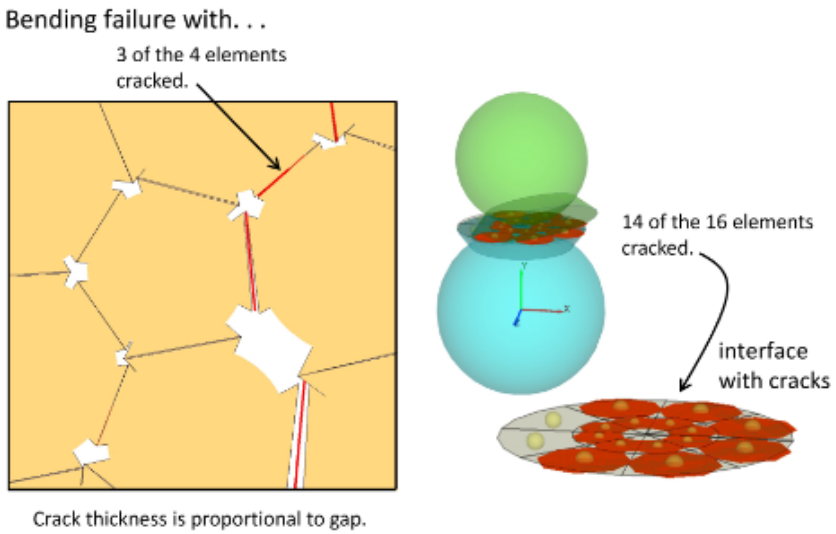

Figure 8: Partial damage of the flat-joint model due to the failure of several bonding elements, resulting in a crack within the interface [37]

2.2.1.3 Elasto-plastic deformation

Elasto-plastic deformation changes the shape of an object permanently in addition to elastic deformation (reversible deformation). This allows the material to absorb large quantities of energy but has little to no effect on very brittle, rock-like materials. The application of this model is aimed at the simulation of ductile material undergoing large deformation, e.g., before buckling of ductile material structures [38].

3 Methodology

This chapter discusses the general approaches for achieving a constant mass and volume flow as well as the requirements for a successful implementation of these approaches. Furthermore, it is elaborated how arbitrarily shaped bonded-particle networks are efficiently generated. Subsequently, test setups depicting static and dynamic material behavior are compared regarding their suitability to calibrate both, the material considered and the depiction of the same prevailing load case as in the final process to be simulated.

3.1 Mass and volume conservation

The mass of a (homogeneously considered) object is linked to its shape, i.e., its volume, by the solid density of the material. In bulk solids handling, where the mass and volume flow of a material is of importance, the characterization of the bulk material by its solid density is not recommended since the bulk material usually contains air-filled cavities. To link the mass of a given bulk volume to its total mass, a bulk density is specified, which takes the cavities into account and is therefore lower than the solid density. A comparison between solid density and bulk density is shown in figure 9. If an object fractures in the course of a process, this can be considered as a transition from a solid material to a bulk material; regarding this single object, the mass is conserved resulting in an increase of volume (cf. bulking factor).

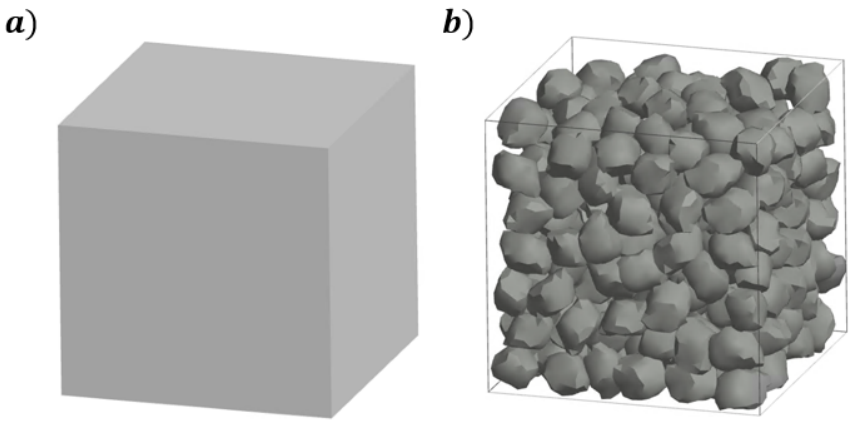

Figure 9: Comparison of solid material (a) to bulk material (b) filling the same volume

In DEM simulation, particles are often approximated as spheres for computational reasons. The solid density of the particles (also called particle density) is calibrated to depict the bulk density of the physical material [39]. However, if a breakable particle consisting of several non-overlapping spherical sub- particles is simulated, the particle density of the sub-particles can either be calibrated using the solid density or the bulk density of the broken-down material. Correspondingly, this leads to a correct density consideration of either the parent-particle or the bulk of the fractured fragments but not of both, resulting in a correct depiction of the mass though not of the volume. A comparison between the volume of a bonded parent-particle with non-overlapping sub-particles to its smallest possible fracture fragments is visualized in figure 10. Since the volume of the bulk material remains constant after fracturing instead of increasing, as shown, the density erroneously stays the same as well. These addressed effects can be described by voids already being present within the parent-particle depicting the solid.

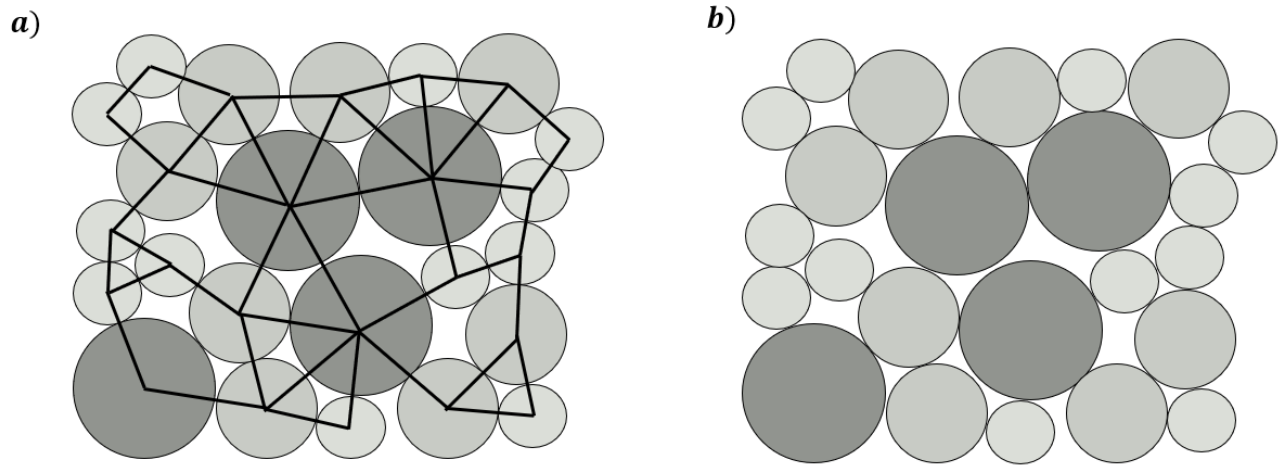

Figure 10: Representation of a bonded-particle cluster depicting a solid (a) and the same particle arrangement after every bond has broken, representing the fracture fragments (b)

Incorrect depiction of the bulk material's total volume is of no concern if the simulation stops after the breakage, e.g., when evaluating the maximum load capacity of a solid object. In that case, however, other simulation methods, such as the FEM are recommended either way. When simulating an industrial-sized process with several hundreds of breakable particles, e.g., in the field of conveying technology, this incorrect depiction of the volume flow is not acceptable (also, as the process typically does not end after a particle breakage occurs). To overcome this problem, two methods can be applied, as stated in the following.

3.1.1 Overlapping spherical sub-particles

This method is similar to the one being used in the PRM (see also section 2.1.1), where the fractured particle is replaced by spherical fragments with an initial overlap [25] [40]. Applied to a bonded-particle network, repulsive forces resulting from this overlap must be avoided, posing the following challenges for this method:

- Forces due to the master contact model must be disabled between particles of the same cluster to avoid an unstable simulation or crooked parent-particle shapes. This is demonstrated in figure 11 on the example of two particles (grey) bonded with an initial overlap (orange). In the initial state, the bond (blue) does not exert any forces onto the particles, but due to the overlap, the Hetz-Mindlin contact model does, as shown in figure 11 a, point 1. The resulting repulsive force leads to a deformation of the bond, which in turn exerts an opposing bond force on the particles. These counteracting forces lead to an oscillation of the bonded particles (figure 11 a, point 2), which, even if an equilibrium state (figure 11 a, point 3) is reached, depending on the parameters of the contact model such as stiffness or damping coefficient, results in a distortion of the initial parent-particle shape (compare figure 11 b to figure 11 c).

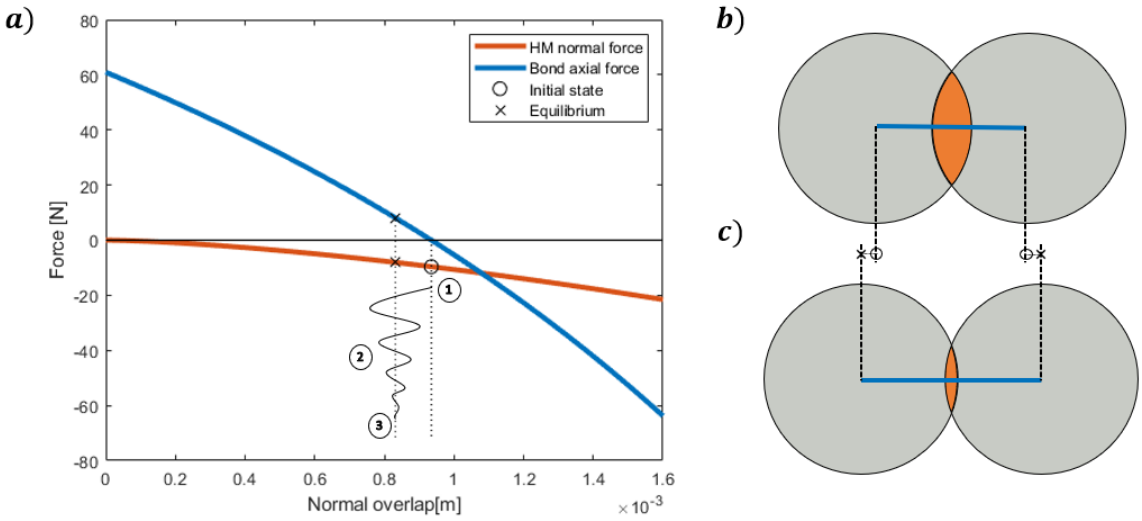

Figure 11: Oscillation of the bonded particles due to forces resulting from the BPM and HM contact model (a); the initial particle positions (b) deviates from the state of equilibrium (c)

- Since the reaction forces of the master contact model are reactivated after two particles are no longer part of the same cluster, an additional damping model must be used to prevent high reaction forces due to the remaining large overlap after bond failure, resulting in an unstable simulation.

3.1.2 Complex shaped sub-particles

Voids within the solid can be prevented by using complex-shaped sub-particles that fill up the entire volume (similar to a Voronoi setup, cf. paragraph 2.1.1.2). A common method is to tesselate the parent-particle and thereby modeling the sub-particles as polygons [27]. However, the use of this method also leads to several disadvantages:

- The simulation of non-spherical particles leads to a significant increase in contact detection, which already takes up much of the simulation time in particle-based simulation methods such as DEM [41]. In Molecular Dynamics (MD), for example, it takes up over 90 % of the computation time [42].
- A smaller timestep is required compared to simulations in which spherical particles of the same size are used. This can be justified by the fact that the timestep must be set small enough in order for the force changes, and therefore changes in motion, of a particle being small enough to only affect its immediate neighbors [41]. When it comes to the overlapping of corners and edges of contacting polygons, this can only be achieved with a small timestep.

3.1.3 Conclusive Assessment

Due to the aim of this thesis being to develop a method that is applicable to industrial-scale simulations in addition to allowing a combination of suitable simulation results with high computational efficiency, spherical sub-particles are chosen as suitable. However, this requires additional efforts as introduced above; the results of which are presented in section 4.

3.2 Generation of particle clusters

Pre-generation of parent-particles is the first step before any simulation can be carried out. The requirements for these generation methods are:

- Capable of generating complex parent-particle shapes
- Low generation time
- Possibility to generate sub-particles with an initial overlap, addressing the solid/bulk density effect (described in section 3.1)

Different ways of generating complex-shaped parent-particles, which can later be imported into calibration simulations as well as into large-scale simulations, are described in this chapter. They are illustrated by means of the generation of cuboidal calibration samples described/used in section 4.4.1.

3.2.1 Simulation of the generation process

The simulation of the generation process follows a simple scheme: A three-dimensional volume is filled with sub-particles, which are then bonded together. This procedure can easily be simulated in DEM software, especially if the required shapes of the parent particles are simple, as do the calibration samples as well as the filter cake plates generated for future simulations. Although this method of generating breakable parent particles may appear to be the easiest at first glance, it quickly reaches its limits if more complex geometries are required. Furthermore, the computational efforts for simulating filling processes increase exponentially with increasing particle numbers, as is common for DEM simulations. The individual steps of the simulation are shown in detail in figure 12.

Before the simulation is started, the desired shape of the parent-particle must be defined by means of a mold. Simple geometries already available in the simulation environment, like a cuboid, or more complex ones, imported from an STL file may be used. Like in a casting process, the mold requires an opening through which the particles are filled. This already shows a major disadvantage of this method, as not every mold shape has an optimal filling point, or filling area, from which the particles can reach every location within the geometry, sometimes requiring multiple filling points,

further increasing the complexity of the simulation. Once the mold has been defined, it is filled with particles, as shown in figure 12 a, with the number of particles generated calculated via the volume of the parent-particle and the solid density of the material. Since the sub-particles within the parent-particle must have an overlap, as described in section 3.1.1, the particles are pressed into the mold by moving a plane over a distance Δh, depicted in figure 12 b. Due to this compression to a very large overlap, resulting in high reaction forces, the timestep must be lowered from this point on to avoid oscillation of the particles (which could otherwise result in an unstable simulation).

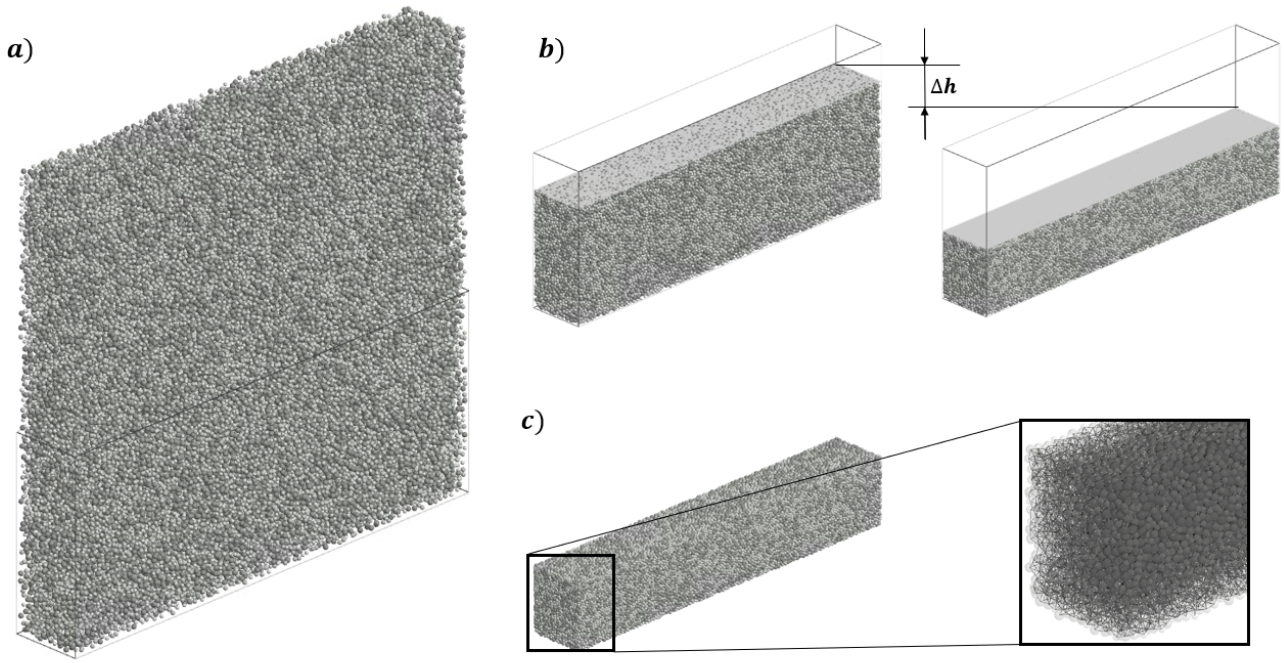

Figure 12: Visualization of the sample generation by simulating the filling of the mold (a), compressing the particles to achieve the desired overlap (b) and bonding the particles, creating the finished parent-particle (c)

Particles being pushed through the walls of the mold due to high reaction forces, as depicted in figure 13, can additionally be avoided by lowering the particle's shear modulus and increasing the shear modulus of the mold during the filling simulation. Since a regular arrangement of the particles is not desired, as this would introduce preferred crack paths into the material, which is to be avoided [43], a particle size distribution (PSD) is used. After the sub-particles are compressed to the desired density, they are bonded to their immediate neighbors, forming the finished bonded-particle network.

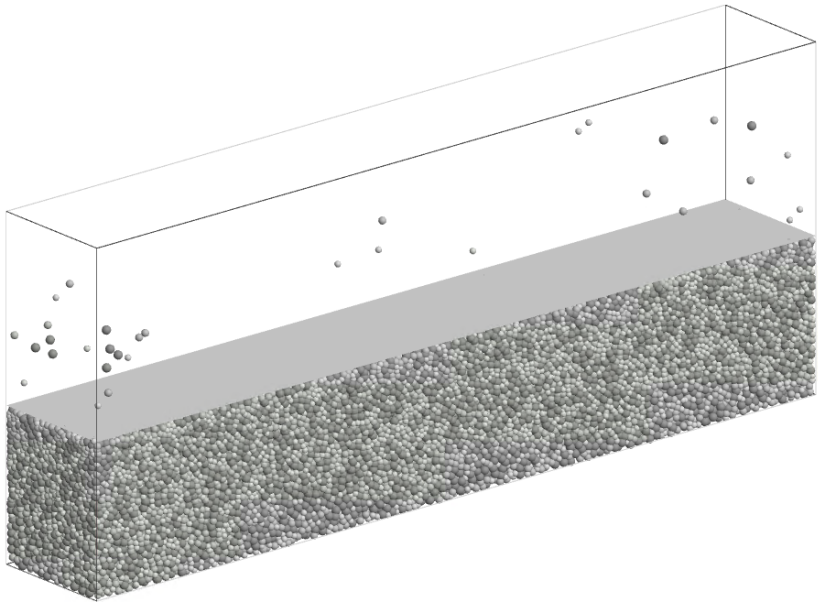

Figure 13: Particles being pressed trough the walls of the mold during the simulation of the filling process due to poor choice of simulation parameters

A uniform overlap between particles and with the container walls results in homogeneous material behavior and to the desired geometry of the generated parent-particle; since a local increase of the overlap between particles an the walls of the mold result in a deviation from the desired geometry in this region. This is especially the case for tall containers, since the pressure increases over the height due to gravity (cf. Jansen's theory [44]), and results in density deviations within the generated particle, which is not the case with the following method.

3.2.2 Filling algorithm

Another generation method, which requires a one-time preparation effort, is the use of a filling algorithm, of which several already exist, both for arbitrarily shaped sub-particles [45] as well as for spherical ones [46], whereas the latter is much more computationally efficient. Generation algorithms, in general, follow the same principles: Placing an initial sub-particle within the required shape of the parent-particle, either at a seed within the shape [46] or at the boundary [47], , followed by additional particles placed adjacent or with overlap next to the existing ones to achieve a high packing density. During the filling process, each newly placed particle is directly bonded to its immediate neighbor resulting in a bonding network shown in figure 14 e.

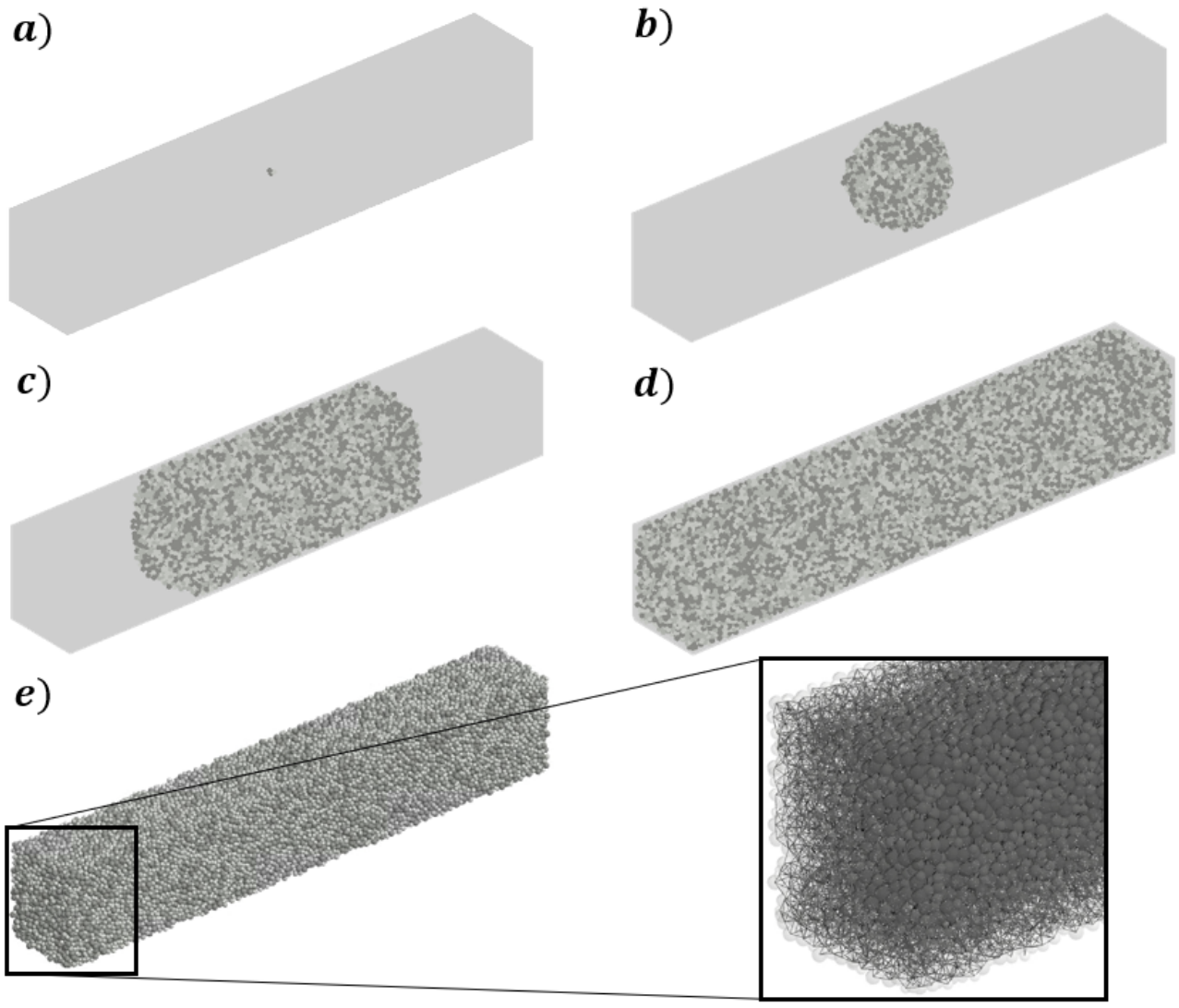

Figure 14: Visualization of the sample generation using a filling algorithm starting with a seed within the desired geometry specified by an STL file (a), the filling process (b-d) and the finished parent particle within the DEM software including a closeup of the bonds connecting the sub-particles (e)

In addition to a much shorter generation time of the parent-particle by means of a filling algorithm, compared with the simulation of the filling process, an algorithm is also characterized by the fact that the computation time increases significantly slower with increasing particle numbers. As with the simulation of the generation process, care must be taken not to introduce any preferred crack paths into the parent particle during generation, which can be controlled relatively easily when using a filling algorithm.

3.2.3 Conclusive Assessment

Once a filling algorithm is set up, it outperforms the simulation of the generation (filling) process in every single requirement. The generation of the most complex shapes by means of an algorithm does not require additional preparation time, and furthermore, any pre-defined overlap can be set. The main advantage, however, is the efficiency of the algorithm, illustrated with the shown example, requiring 2 min and 12 s to generate the calibration sample, whereas the filling simulation with the same number of particles and bonds (77 487 sub-particles and 377 052 bonds) took 780 min, which is why the use of a filling algorithm for the generation of breakable complex-shaped particles is indispensable.

3.3 Test setup

To obtain results from the simulation of a process in a suitable form, meaning the depiction of the correct macroscopic behavior, the parameters of the bonding model must be sufficiently calibrated. For this purpose, suitable calibration tests, i.e., reflecting the loads prevailing in the final process, must be selected. The calibration of typical simulation parameters, such as particle density or friction coefficients between particles, is not further discussed in this thesis, since values in these contexts can be determined with commonly known, standardized tests, such as an angle of repose test (cf. [39]).

For the calibration of rock and rock-like materials, whereby rock-like materials are understood to be materials that exhibit the same failure criteria as rock and, above all, brittle material behavior [48], numerous standardized tests already exist. These tests are used to determine characteristic material parameters, such as tensile or compressive strength. Due to the tensile strength of the filter cake being significantly lower than the compressive strength, as is common for rock-like materials, and the parent-particles being initialized in thin plates in the final simulation, resulting the parent-particle to most likely fail due to bending, i.e., due to exceeding tensile stress limits in the outer layers of the cake-structure, standardized tests that measure tensile strength are chosen to calibrate the bond parameters.

Depending on the deformation rate, different tests are recommended to measure various material properties. Especially in dynamic processes, the consideration of dynamic material behavior is of great importance. Different loading techniques for rock testing, classified by the strain rate, are shown in figure 15.

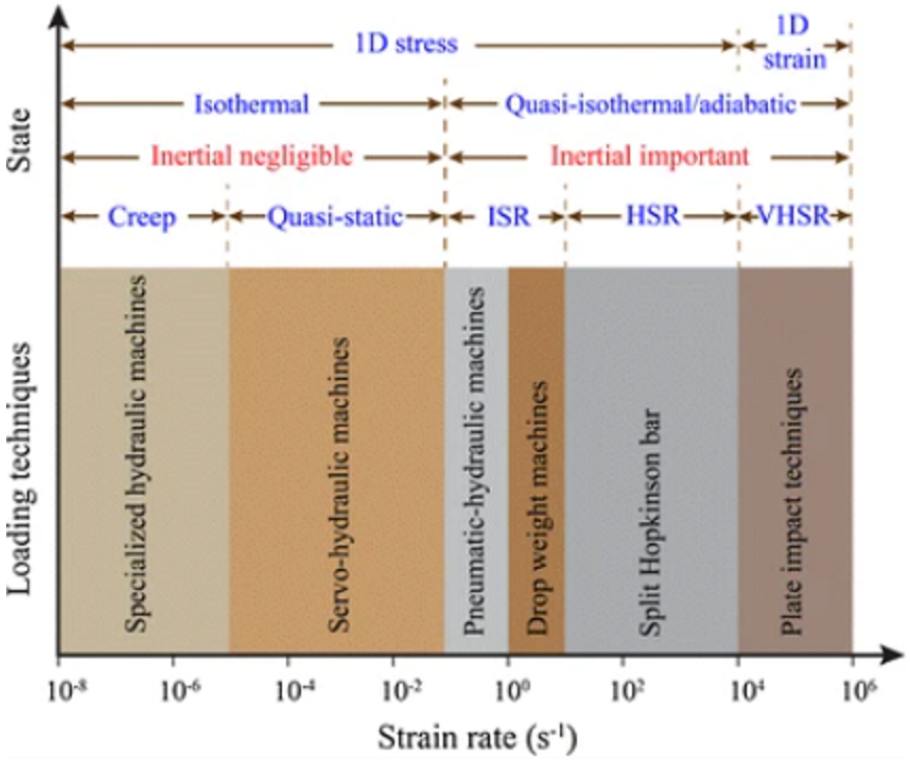

Figure 15: Visualization of the strain rate dependent loading techniques [49]

3.3.1 Static tests

Tests with very low strain rates are considered static or quasi-static. A slow change of the system state enables a measurement of the state variables, such as force and deformation, with simple and relativey cheap measuring devices. Even though, as according to figure 15, hydraulic machines are recommended to generate the forces used to break the material sample, small electric motors can also be used in this case, due to filter cake breaking under much smaller loads than rock. The most commonly used static tests are as follows:

3.3.1.1 Bending tests

Bending deformation results in tensile stress in the edge fibers/layers of the object, leading to material failure due to tensile stress. Depending on the number of loading points, bending tests can be further subdivided.

3.3.1.1.1 Three-point bending test

As the name implies, the material sample is supported on three points in the three-point bending test (TBPT), also called the three-point flexural test. It can be carried out in any universal testing machine equipped with a three-point bending fixture. In this laboratory test, the sample is loaded with a force F in the middle of the specimen. While measuring the deformation as well as the applied force, the load is increased steadily to ensure a constant deformation rate until material failure occurs under the maximum force F_{max} at a deformation of w_{max}, as shown in figure 16. The procedure for testing rock by means of an FPBT is standardized according to EN 12372 [50].

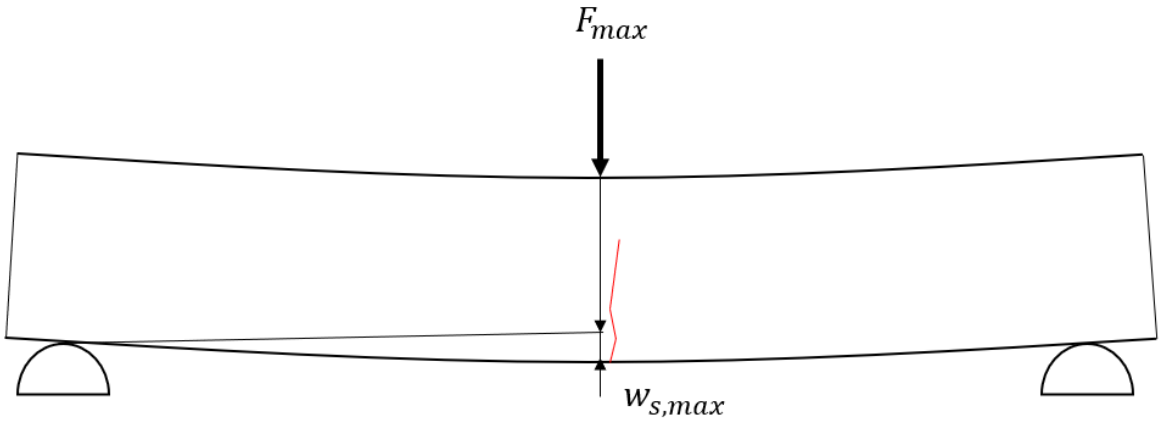

Figure 16: Schematic depiction of the three-point bending test

The dependency of the test results on the dimensions of the test sample is the main disadvantage of the TPBT [51].

3.3.1.1.2 Four-point bending test

The procedure of the four-point bending test (FPBT), also called the four-point flexural test, specified in various international standards for testing rock or cement-based products (cf. EN 13161 [52], ASTM C880 [53]), only slightly differs from the TPBT, since the force F is divided between two loading points, as can be seen in figure 17.

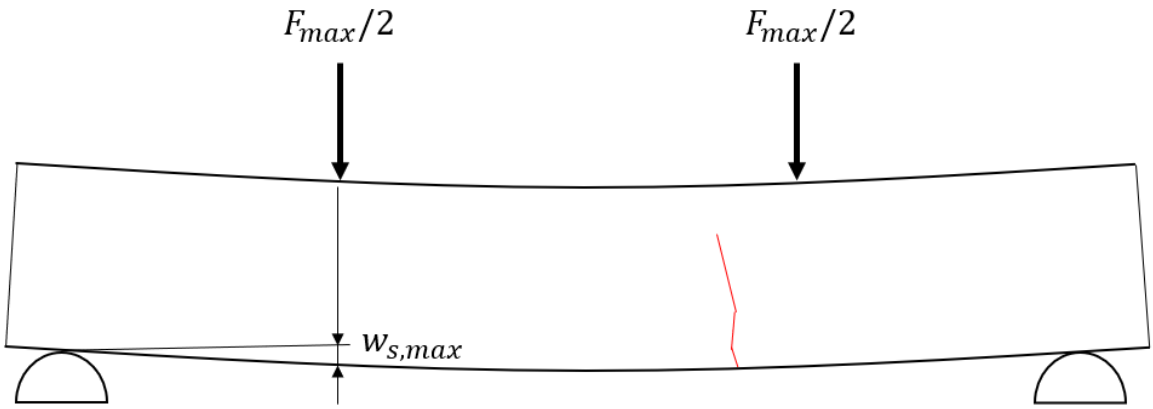

Figure 17: Schematic depiction of the four-point bending test

In addition to the results not depending on the dimensions of the specimen, the internal reaction forces of the FPBT allow a much easier determination of the material's tensile strength. As shown in figure 18, the test sample is subjected to a constant bending moment and no shear forces between the two loading pins. Material failure therefore occurs only due to tensile stresses - as the crack is enabled to be initiated between the loading pins, instead of failing due to stress near the center of the specimen only, as in the case of the TPBT.

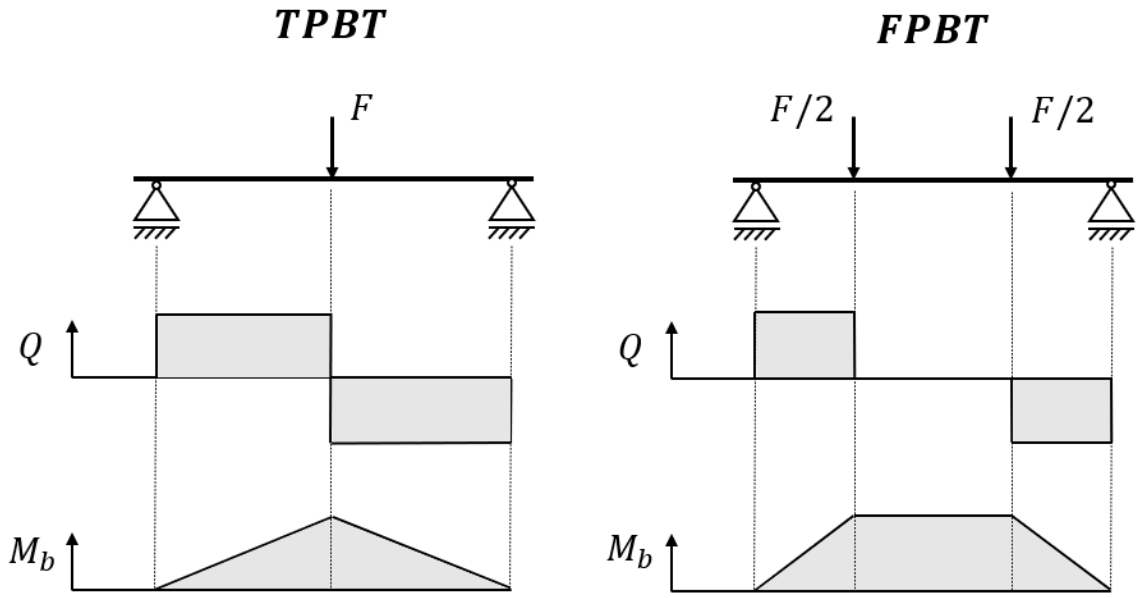

Figure 18: Comparison of the internal forces of the TPBT to the FPBT

3.3.1.2 Brazilian Test

One of the most popular tests, due to the ease of specimen preparation, resulting in a low dispersion of experimental data, is the Brazilian Test (BT), or Brazilian Disc Test [51]. The procedure of the BT, as well as the cylinder dimensions, are specified in various international standards for testing rock or concrete (cf. ASTM C496 / C496M-17 [54], ASTM D3967-16 [55], ISRM [56]). After the cylindrical material sample is prepared, it is placed between two jaws, as seen in figure 19, which are then applied with a constant loading rate until material failure occurs.

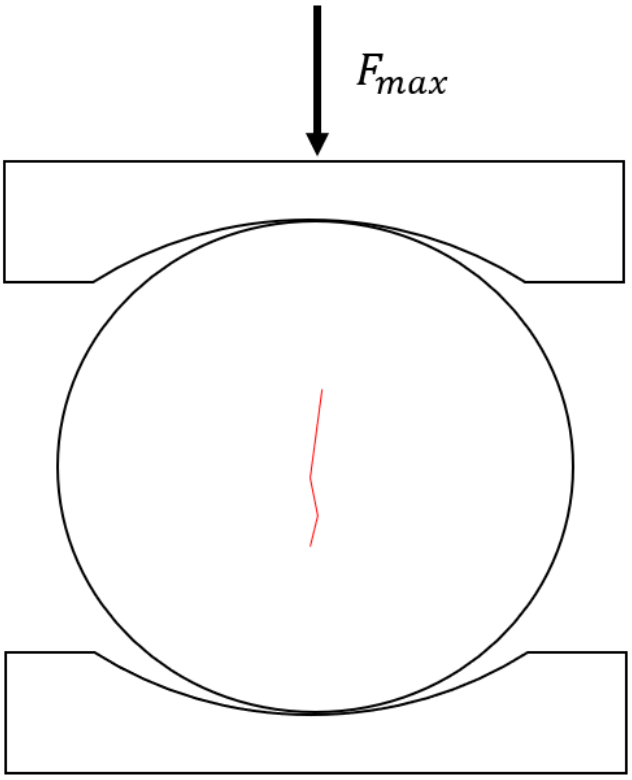

Figure 19: Schematic depiction of the Brazilian Test

Due to the compression of the disc, normal stress perpendicular to the diameter is introduced, with nearly constant tensile stress in the middle of the specimen, as shown in figure 20. If the tensile strength of the material is calculated from the BT with the assumption that material failure occurs due to tensile stress, only samples that crack in the central tensile stress zone are deemed viable [57].

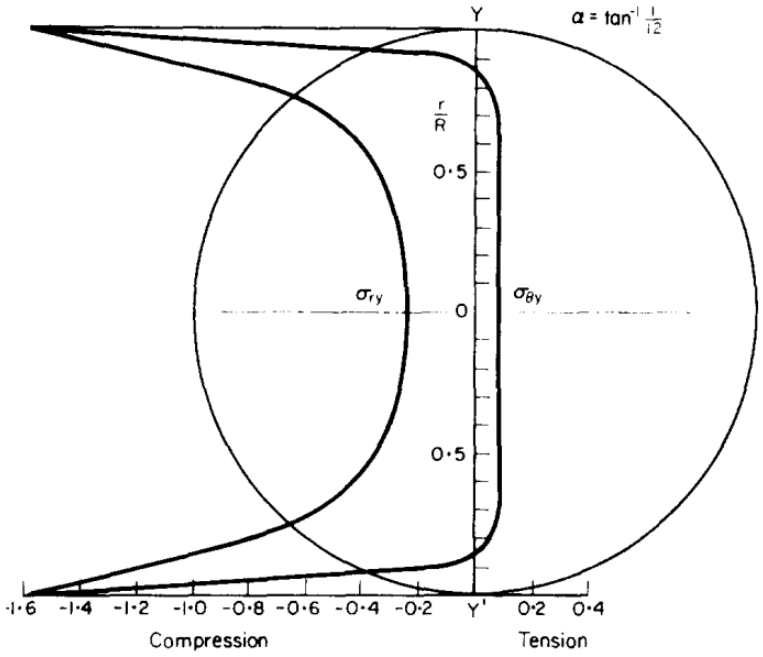

Figure 20: Stress distribution in a Brazilian disc [57]

3.3.1.3 Direct tensile test

As with testing the tensile strength of steels, the Direct Tensile Test, also known as the Direct Pull Test (DPT), is the best method for determining the tensile strength of rock specimens since the material sample fails under pure, uniaxial tensile stress. Different versions of the DPT are distinguished depending on the shape of the specimen and the coupling mechanism, examples shown in figure 21, described in various standards (cf. ASTM D2936-08 [58], ASTM D2936-20 [59], ISRM [56]). Samples in a dumbbell or dog bone- shape are gripped by a metal holder, resulting in 3D stress concentrations near the connection points. In this case, samples are only accepted as valid if the specimen fails in the middle area under uniaxial tensile strength. Cylindrical samples are glued to metal connection plates, resulting in a uniform stress distribution across the whole sample. The reason why this test is not very common, despite its accurate results, is the difficult and expensive preparation of the samples and their coupling to the connectors [51].

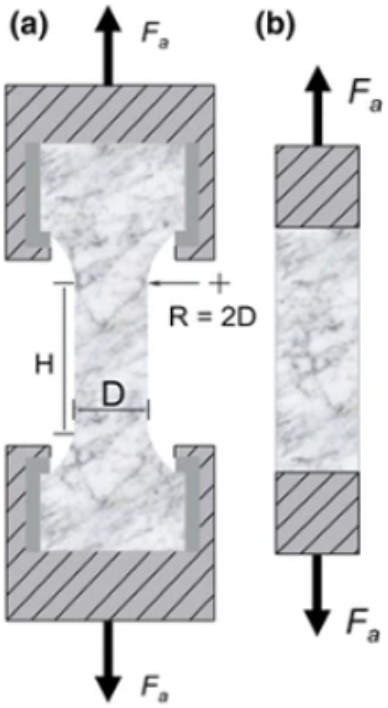

Figure 21: Direct Pull Test gripping devices using a dog bone- shaped sample (a) [60] and cylindrical specimen with glued end caps [56]; Figure adapted from Fig. 1 in [61]

3.3.1.4 Additional static tests

For an exact calibration of the BPM, not only Young's Modulus and tensile strength are of importance, but also other characteristic values, such as the material's compressive strength or shear strength. These material properties are determined using established test methods, such as the Uniaxial Compression Test and the Triaxial Compression Test (cf. ASTM D7012-14e1 [62]) for compression strength, or the Direct Shear Test (cf. ASTM D5607-16 [63]) for shear strength. Since the properties of rock-like materials in combination with the geometry of the filter cake plates mainly result in material failure due to tensile stress, these tests are of secondary interest for this application and thesis and are therefore neglected.

3.3.2 Dynamic tests

Since material properties, such as tensile strength or fracture toughness, depend on the deformation rate of rock- like materials [49] [61], dynamic tests are required. For cases with material failure occurring under a fixed strain rate, such as hard rock mining with explosives where the stress waves

propagate at a certain speed through the rock body, dynamic tests with the same, or similar, strain rates must be used. When simulating processes in which material fails at different strain rates, as it is the case in this thesis, it is advisable to determine the dependency of the dynamic material parameters on the strain rate. The dependency of the tensile stress on the strain rate is described by means of the Dynamic Increase Factor (DIF).

3.3.2.1 Drop weight test

In this test method, the gravitational-potential energy of a hammer is used to load the specimen. The amount of energy introduced into the material sample by the hammer is set by its mass and the drop height, as well as by additional acceleration of the weight by means of a gas gun. A reduced strain rate of the specimen can also be achieved by using a rubber buffer to prolong the loading duration. During the impact, the momentum impulse acting on the falling weight is measured to calculate the impact velocity, from which strain rate and energy input are calculated. However, this testing method is not commonly applied due to several disadvantages, ranging from the loading rate not being that easy to control to difficult interpretation of the measured data due to coupling effects between machine vibration and wave propagation [49].

3.3.2.2 Split Hopkins Bar

A materials dynamic stress-strain response at high strain-rates, cf. figure 15, is most commonly determined using the Split Hopkins Bar (SHB). The main common version of this test is called the Split Hopkins Pressure Bar (SHPB) or Kolsky Bar (a modification of the Hopkinson pressure bar [64] by Kolsky [65]) used for determining a material's compressive strength, as seen in figure 22.

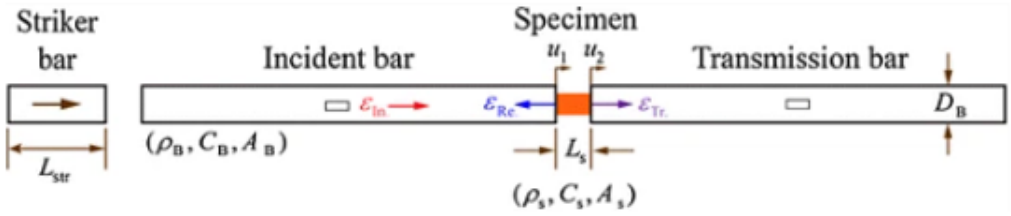

Figure 22: Schematic depiction of the components making up a SHPB or Kolsky bar [49]

In this test, the specimen is sandwiched between two bars, called the incident bar and transmission bar. A third bar, the striker bar, is used to introduce a stress wave into the end of the incident bar propagating towards the specimen. Upon reaching the connection between the incident bar and the specimen, the wave is split into two smaller ones, one traveling through the specimen into the transmission bar and the other one being reflected back into the incident bar. Strain gauges, applied in the middle of the two bars are used two measure the stress pulses, allowing the calculation of the specimen's stresses and strains from the amplitudes of the two stress waves.

Dynamic tensile strain-strain behavior of a material is determined by a variation of this test setup, called the Split Hopkins Tensile Bar (SHTB), requiring a more complex specimen coupling mechanism and loading techniques, as seen in figure 23.

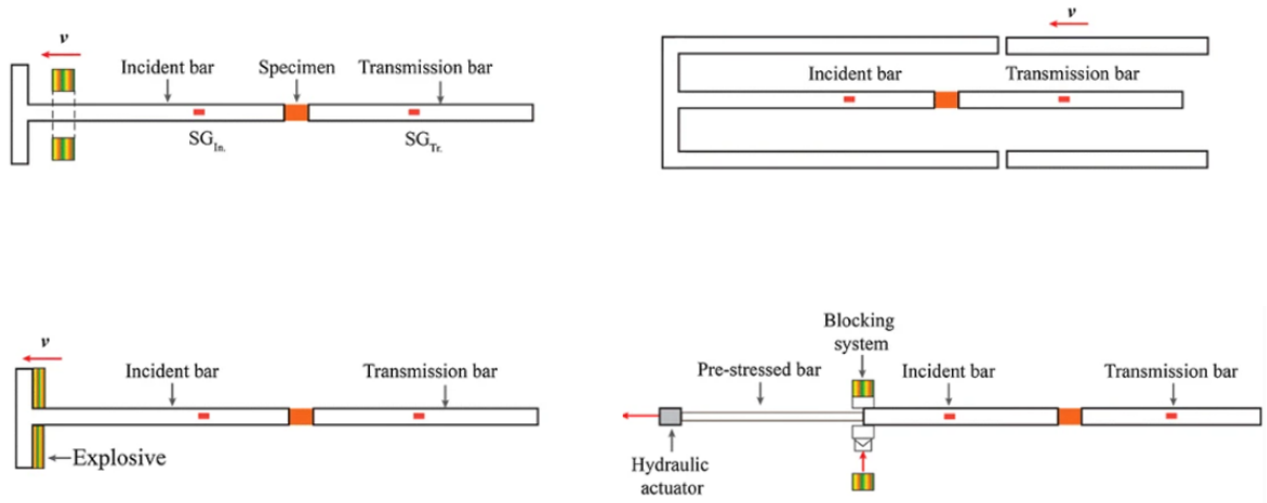

Figure 23: Variations of the SHTB, showing different loading techniques [49]

However, this test setup encounters the same problems regarding sample preparation and coupling mechanisms as the DPT (dumbbell-shaped specimen or connections with epoxy resin). To avoid this problem, the same setup as the SHPB can be used with Brazilian-Disc-shaped specimen, resulting in material failure due to tensile stress, depicted in figure 24.

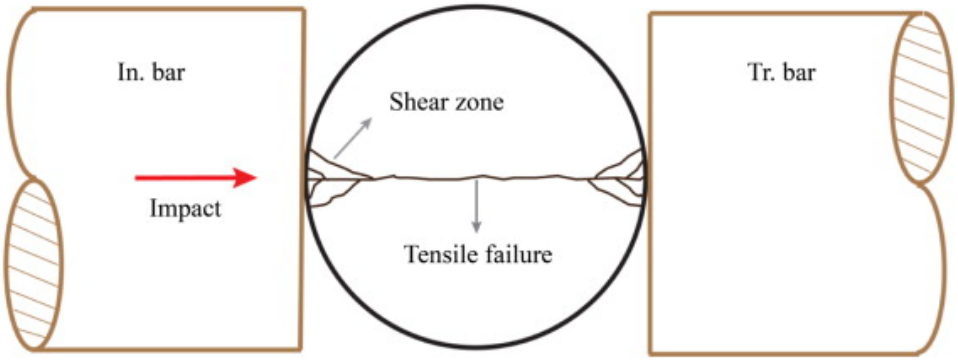

Figure 24: Failure pattern of a Brazilian Disc under dynamic load in a SHPB; Figure adapted from Fig. 12 in [66]

3.3.2.3 Drop test

This method, in contrast to the other tests described in this chapter, is neither standardized nor an established method for testing rock-like materials in this given context . The course of the experiment is as follows: a material sample is dropped from a set height onto a horizontal or inclined plane (with the sample's impact velocity correspondingly resulting from the drop height). During the impact of the specimen, the deformation of the sample is observed, and potential damage, up to material failure, is analyzed. Only visual observations are made, instead of measuring the energy, velocity, strain-rate, etc. Due to the resulting lack of experimental data, this method is relatively inaccurate compared to standardized test methods but provides a simple way to observe the dynamic material behavior for various and also application-related impact scenarios.

3.3.3 Conclusive Assessment

The four-point bending test leads to material failure under the same prevailing load case as in the final simulation and additionally does not dependent on the shape of the test sample. With a quite straightforward design and the forces and deformation being easy to measure, it is the ideal test setup for the measurement of the static material properties for this user case.

With most dynamic test setups being more complex than static ones and requiring more expensive measuring equipment, the DIF is derived from empirical data. Additionally, a simple drop test is carried out to visually compare the dynamic material behavior with the simulation results.

3.3.4 Parameter calibration

To calibrate the simulation parameters of the bond model, the laboratory tests are replicated under the same conditions in the simulation environment. Subsequently, the deformation paths of the simulated sample and the required force are compared with the values measured in the laboratory tests as well with analytical values. After achieving an accurate depiction of the simulated material on a small scale, additional optimization steps are taken to ensure a computationally efficient simulation of processes on an industrial scale.

4 Results

The results in this chapter show the implementation of the methods assessed in section 3, as well as the steps required to achieve them.

4.1 Particle generation

Some adaptions have been made to the generation algorithm template developed by Lozano [46], as it either does not meet certain objectives or requires sub-algorithms, which are only faster than the consequently described alternatives, if professionally optimized.

4.1.1 Seed position

To start the generation process, three adjacent starting particles must be defined, which are placed in the origin point, also called the seed. This point can be defined manually via its Cartesian coordinates or automatically placed in the center (average of the geometries maximum and minimum points to the coordinate origin in all spatial directions) of the parent-particle geometry to be generated. However, neither a manually picked seed nor it being placed at the center of the geometry guarantees that the starting particles are within the boundaries of the parent-particle, requiring the particles to be checked after the seed is set. If an arbitrarily placed point is moved from inside of an object in any spatial direction towards infinity, it will intersect the object's surface an odd number of times. However, if the point is already outside of the object, the number of intersections is even. Thus, casting a ray from the three seed particles center in any spatial direction and counting the number of intersections of this ray with the triangles of the tessellated surface, given by an STL file, it can be said with certainty whether the seed is located within the boundaries of the parent-particles geometry. This was realized by implementing the Möller-Trumbore intersection algorithm [67] and is illustrated in figure 25.

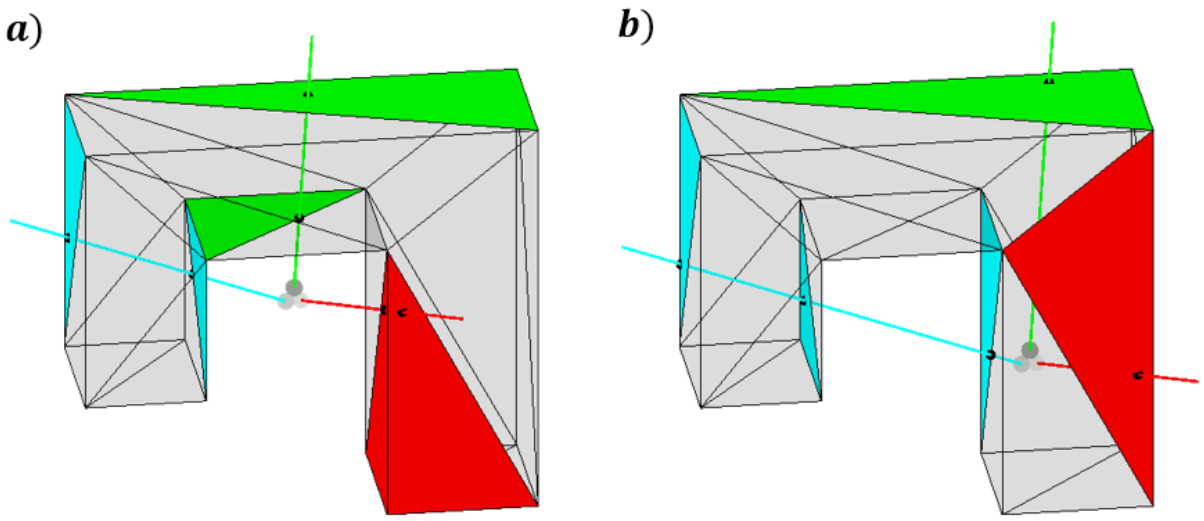

Figure 25: Visualization of the Möller-Trumbore algorithm to calculate the number of intersection points from a casted ray, originating from the seed particles centers, into a random spatial direction; The seed placed into the center of the geometry, which in this case lies not within the object, results in an even number of intersections between each ray with the tessellated surface (a) while a manually defined seed within the geometry leads to the number of intersections being odd (b)

4.1.2 Generation of overlapping particles

Since the original algorithm is intended to generate non-overlapping particles, the method for calculating the position of to be generated particles was adapted. Every seed particle as well as all newly generated particles are stored in a list, the so-called FrontSpheresQueue. The first particle of this queue is used as the initial particle for the generation process and is removed from the list if there are no further possible positions for new particles adjacent to it, resulting in the generation algorithm to stop if there are no particles left in the queue.

A newly generated particle, whose radius is chosen according to the given PSD, is always placed adjacent to three particles to obtain the densest packing of spheres; therefore, the direct neighbors of the initial particle must be found. This is done in a computationally efficient way by subdividing the entire generation space into uniform grid cells. Each particle is assigned to a cell according to its position and can thus be easily identified as a potential neighbor for the initial particle.

The calculation of candidate points for new particles is carried out by using the concept of a “Halo”, meaning an expansion of the particles’ radii by the radius of the to-be-generated particle. Intersection points of the initial particles halo with the halos of neighboring ones, thus indicating possible positions where the newly added particle would be generated adjacent to three existing ones. However, since overlapping of the sub-particles is required, the halos must be reduced by a percentage of their respective particle’s radii. This guarantees the same relative overlap, in regard to the radii of the overlapping particles, for each possible particle combination according to the PSD. This introduced reduction factor must be calibrated to ensure a correct solid density of the parent-particle with the particle density calibrated from the materials bulk density. The resulting calculation process of candidate points can be seen in figure 26.

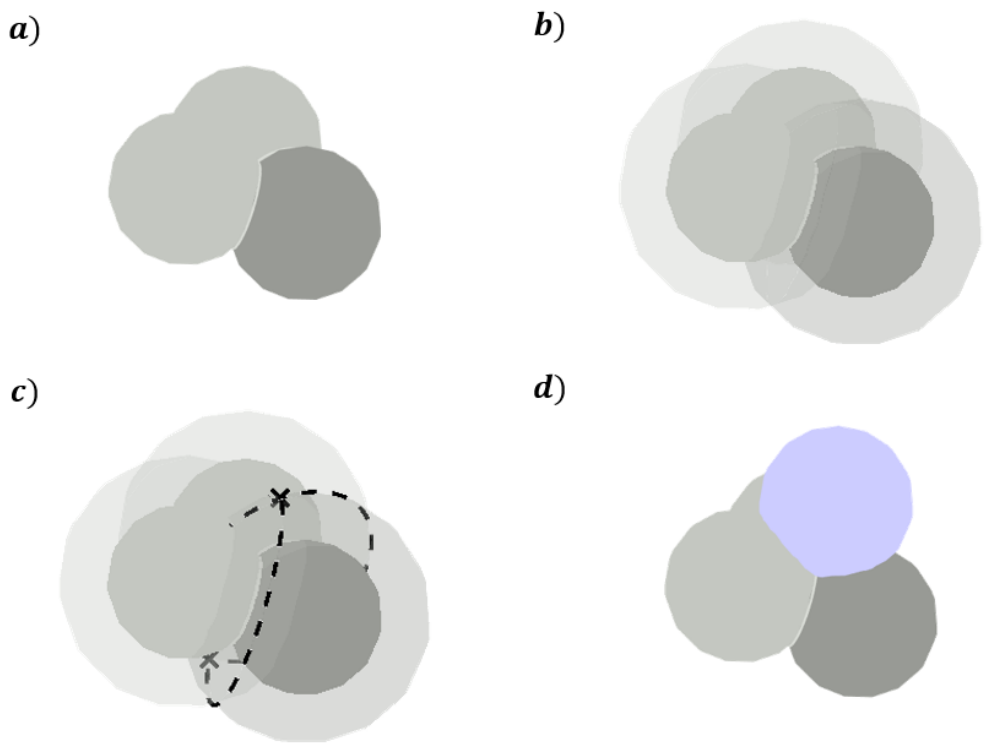

Figure 26: Process of generating overlapping particles by finding the initial particles (dark gray) direct neighbors (light gray) (a), equipping these particles with a halo (b), determining the two candidate points for a newly added particle by calculating the halo intersections (c) and adding the new particle (light blue) (d)

Subsequent to a collision detection, with other existing particles and the walls of the geometry to be filled, the new particle is added. A newly generated particle is immediately bonded to its neighbors, within a pre-defined distance from it, discussed in detail in paragraph 4.5.1.6.

4.1.3 Particle-to-boundary contact detection

The contact detection of newly added particles with the surface of the geometry was changed as well. This process is calculated in the adapted algorithm by means of a distance field (cf. [68]). In this method, the shortest distance of the tessellated surface to every grid cell vertex is calculated. If the considered vertex lies beyond a certain distance to the surface, the distance is defined as negative or positive infinity (depending on whether the point is inside or outside the geometry). Otherwise, the exact distance value is stored in the vertex, resulting in both an enlarged and downscaled shell around and, respectively, within the object's surface. The calculation of the distance field is shown in figure 27.

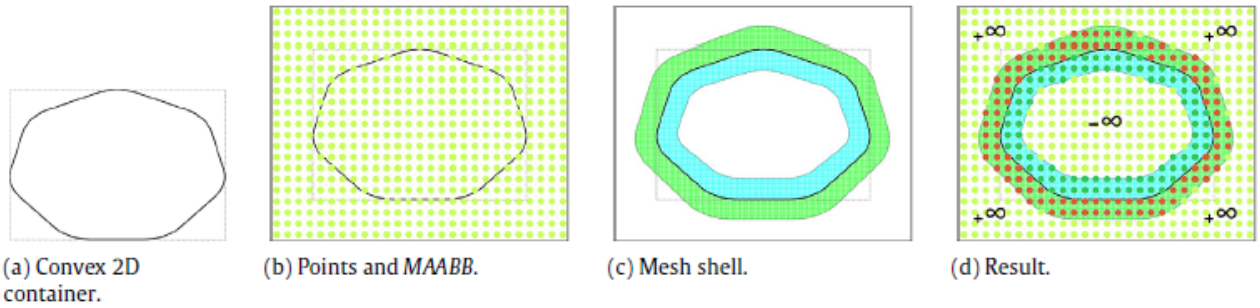

Figure 27: 2D distance field calculation of a convex container [46]

Before a new particle is generated at a candidate point, its distance to the geometry's surface is calculated. Since the candidate point lies within a grid cell, this is achieved by interpolating the previously determined distances of the grid cell's vertices. If the assessed distance is smaller than the particle's radius, the candidate point is deemed unfit. In summary, this method is based on calculating the distance to the surface for each vertex of the grid cells. After implementation, this has turned out to be rather time-consuming and thus computationally not that efficient as potential other methods, which is why another contact detection method is presented.

After reading the STL file, the faces of the triangles (including their normal vectors) are assigned to every grid cell that they penetrate. This way, for every calculated candidate point located within a grid cell, the triangles in its immediate vicinity are determined efficiently, as shown in figure 28 on the example of a bunny.

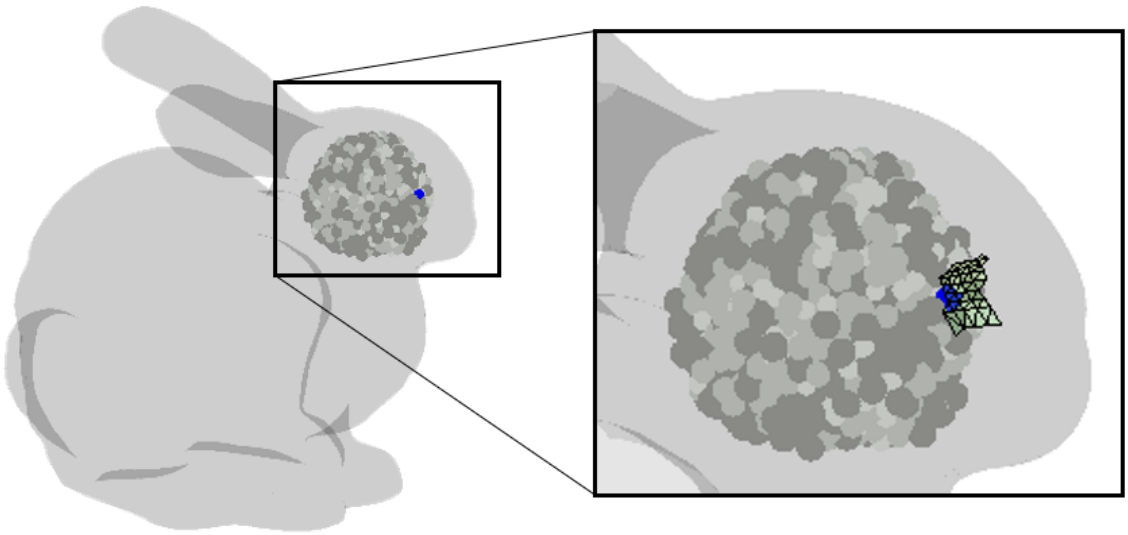

Figure 28: Determination of triangles in the immediate vicinity of a particle (blue) in a candidate point

The shortest distance from the candidate point to these triangle surfaces is calculated in several steps with increasing computational effort. Every step checks for a case scenario that, if fulfilled, indicates that a particle- boundary contact has been detected or that a contact is not possible. Aborting the algorithm in certain cases saves computational time, thus enhancing the overall efficiency.

Calculation of the shortest distance:

1. 1. Determination of triangles in the immediate vicinity of the candidate point::
   Calculation according to the description in the previous paragraph and as shown in figure 28.

   Case: If there are no triangles $\rightarrow$ abort calculation since no particle- boundary contact is possible

2. Calculation of the normal distances d_n:
   This is achieved by projecting the distance of the candidate point to any vertex $\overrightarrow{\mathrm{d}}$ onto the normal vector $\overrightarrow{\mathrm{n}}$ of the triangle by means of the dot product, as shown in figure 29.

   Case: If all calculated unsigned distances are greater than the particle's radius $\overrightarrow{\mathrm{d}}$ abort the calculation since no particle-boundary contact is possible. Otherwise, only triangles that don't fulfill this condition are considered in the following steps.

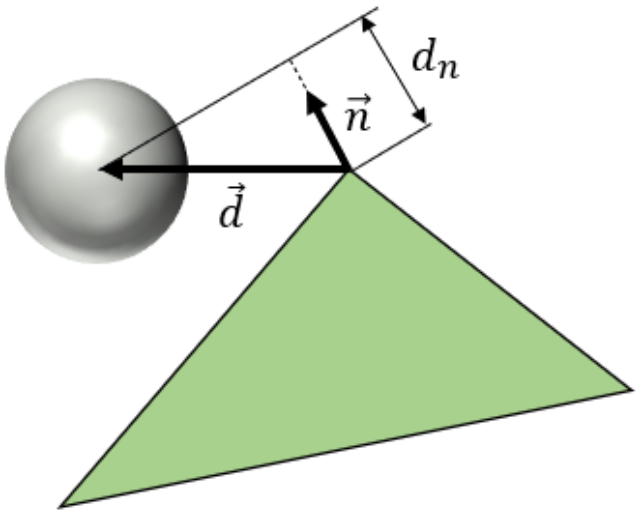

Figure 29: Calculation of the normal distance from a particle to a triangular surface element

3. Determining the candidate point-to-vertex distances:

   An unsigned normal distance from the candidate point to a boundary element, calculated in bullet point 2, being less than the particle radius does not automatically result in an overlap. This is illustrated in figure 30 using a 2D example, resulting in the boundary layer to consist of lines instead of triangles, of a corner protruding into the geometry.

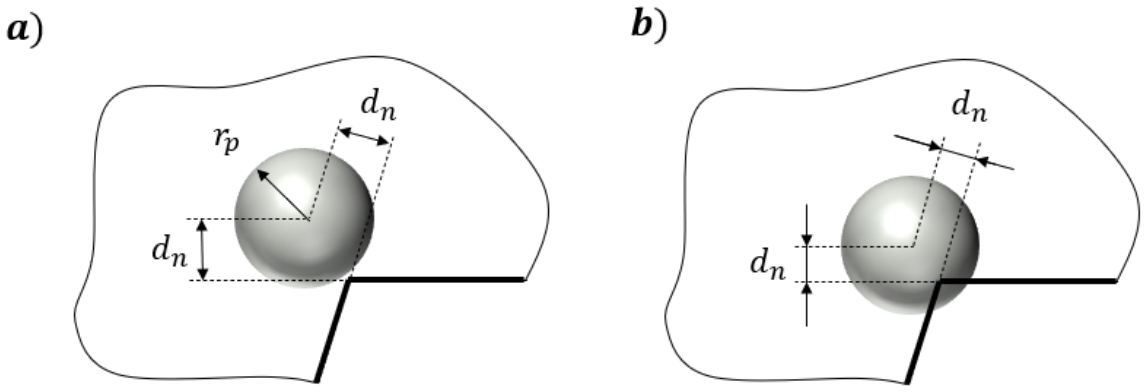

Figure 30: 2D representation of a particle with unsigned normal distances to the boundary being less than the particle radius resulting in no particle-boundary contact in one case (a) but in another very much so (b)

Case: Termination of the calculation if the distance to one of the considered vertices is smaller than the particle radius r_p, since contact is detected:

4. Comparison of the candidate point-vertex distance signs in all spatial directions

   This can be used to determine if the candidate point lies within the extruded volume of the triangle, as shown in figure 31, resulting in the normal distance being the shortest distance between the candidate point and the triangle surface.

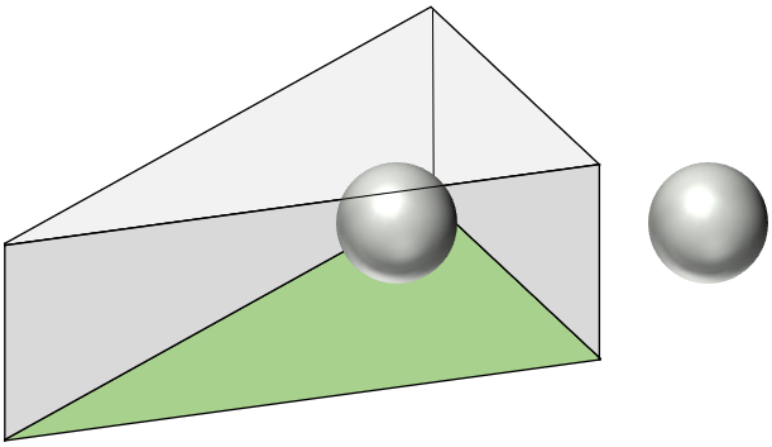

Figure 31: Extrusion of a surface triangle element into its normal direction to check if the candidate point lies within the volume

   Case: If this is the case, the unsigned normal distance to the triangle, calculated in bullet point 2, is compared to the particle radius to determine a particle-boundary contact.

5. Calculation of the smallest edge distance:

   With the candidate point being outside the projected volume, the only possibility for the new particle to be in contact with the surface is for it to intersect the triangle's edge.

   Case: The particle radius must be less than the distance to the nearest edge of a triangle to avoid contact with the surface

This contact detection method proved to be faster due to it only being executed for candidate points in the immediate vicinity of the surface and not for every point of the grid cell. Especially for objects with a small surface compared to their volume, a considerable saving of computation time is noticeable.

4.1.4 Adapted algorithm

The adapted algorithm was programmed in MATLAB [69] and is illustrated in the following pseudocode (algorithm 1): (with the enclosed if-loops stated in the given form on purpose to represent the description as given in the bullet points of the previous chapter)

Algorithm 1 Filling Algorithm

Input: PSD, particle density, STL file, *seedPosition*, desired overlap, bondingRadius
Output: Solid density, Packing density, INP - file, TXT - files

```
Create a grid from the imported STL file and assign surfaces of the triangulated mesh to the
according grid cell                                                          ▷ cf. section 4.1.3
Create three seed spheres in seedPosition, add them in an empty frontSpheresQueue,
assemblyList and bondingList, check if they are located within the STL and assign
them to a grid cell                                                          ▷ cf. section 4.1.2
while frontSpheresQueue.size() != 0 do
    initialParticle ← frontSpheresQueue[1]
    r_new ← getNewParticle(PSD)
    neighbouringParticles ← getNeighbours(initialParticle, r_new, grid)
    for i = 0; i < neighbouringParticles.size(); i++ do
        intersectionCurve ← getHaloIntersection(initialParticle, neighbouringParticles[i])   ▷ cf. section 4.1.2
        for j = 0; j < neighbouringParticles.size(); j++ do
            candidatePoints ← getCandidatePoint(intersectionCurve, neighbouringParticles[j])  ▷ cf. section 4.1.2
            if candidatePoints.size() != 0 then
                if noParticleIntersection(candidatePoints, neighbouringParticles) then
                    if noBoundaryIntersection(candidatePoints, grid) then                ▷ cf. section 4.1.3
                        newParticle ← particle(candidatePoints, r_new)
                        bondingList ← addBonds(newParticle, neighbouringParticles, bondingRadius)
                        add newParticle to frontSpheresQueue, assemblyList, neighbouringParticles and grid
                        r_new ← getNewParticle(PSD)
                    end if
                end if
            end if
        end for
        if i == neighbouringParticles.size() then
            remove initialParticle from frontSpheresQueue
        end if
    end for
end while
Calculate the Solid density and Packig density
Convert assemblyList and bondingList into an INP-file and TXT file
```

The generated parent-particle can be directly imported into the simulation software by reading the resulting INP file calculated by the algorithm, which, however, proved to be relatively slow. A faster method is to import the generated parent-particle via the API's flexible inlet, discussed further in section 4.6. For this purpose, the generation algorithm additionally writes the sub-particles' types and

positions, as well as the bond information, into separate TXT files, which are easier to read via the DEM software's API.

In this thesis, a filling algorithm was implemented in which three adjacent spherical sub-particles are initially placed in a seed within an arbitrary geometry given by a triangulated surface mesh, following the placement of additional sub-particles with a pre-defined overlap to the already generated particles following a desired particle size distribution (PSD) [46]. As a result, the volume of the parent-particle is filled with adjacent particles starting from the seed until the to-be-added sub-particles collide with the surface mesh, as can be exemplarily seen in figure 32. During the filling process, each newly placed particle is directly bonded to its immediate neighbor(s) resulting in a complex and dense-packed bonded-particle network.

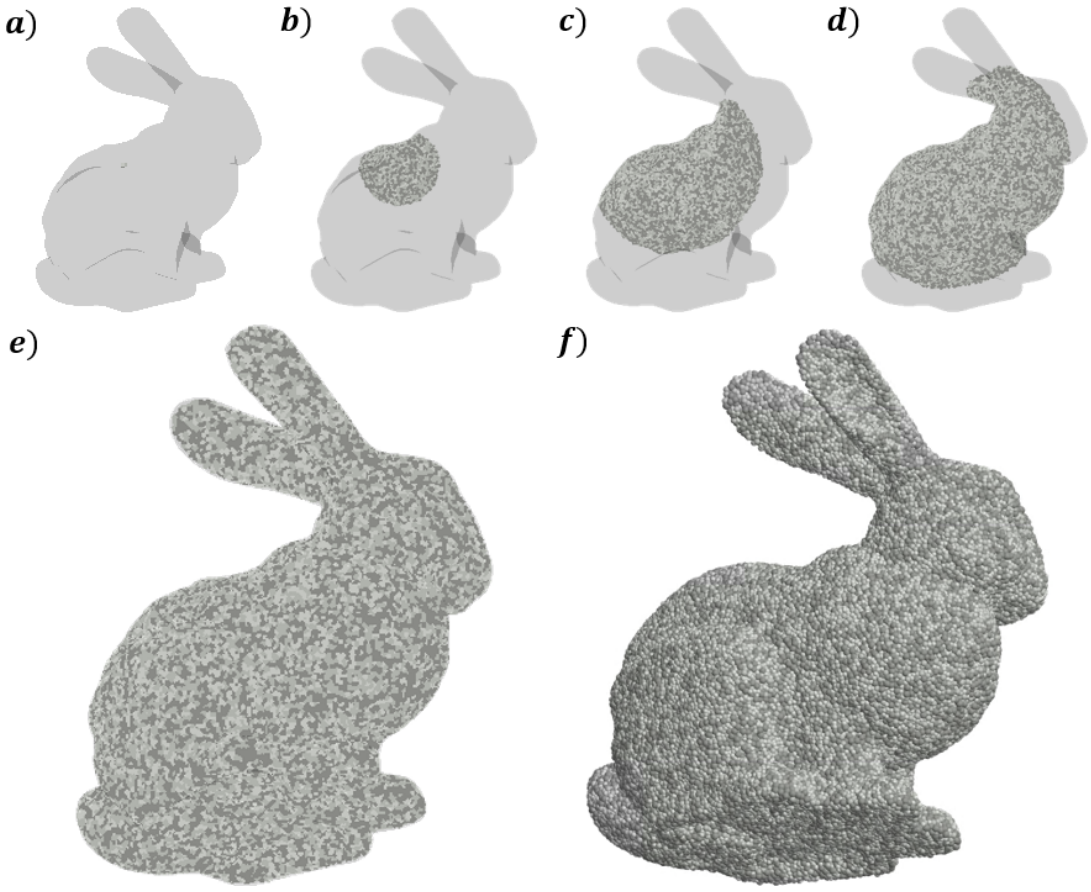

Figure 32: Visualization of the generation process by importing an STL file and generating seed particles (a), placing additional particles with a predefined overlap adjacent to the existing ones (b-d), and, after completion of the generation process (e), importing the generated parent-particle into the simulation software (f)

4.2 Cluster detection

As explained in section 3.1.2, the forces of the master contact model must be deactivated to avoid oscillation of the overlapping bonded particles or distortion of the parent- particle. In the simulation software used, it is possible to assign particles a bonding ID. If two overlapping particles with the same bonding ID are detected, neither the master contact model nor the slave contact model is calculated (except for the bonding model). Thus, only the bonds provide the parent particle with the properties of a solid. However, if so many bonds fail, by exceeding the local critical equivalent stress $\sigma_{v,krit}$, that the pre-generated parent-particle breaks into several non-connected pieces, the particles of the resulting sub- clusters must now be able to interact with each other, otherwise the fracture fragments could pass through each other without resistance, as can be seen in figure 33.

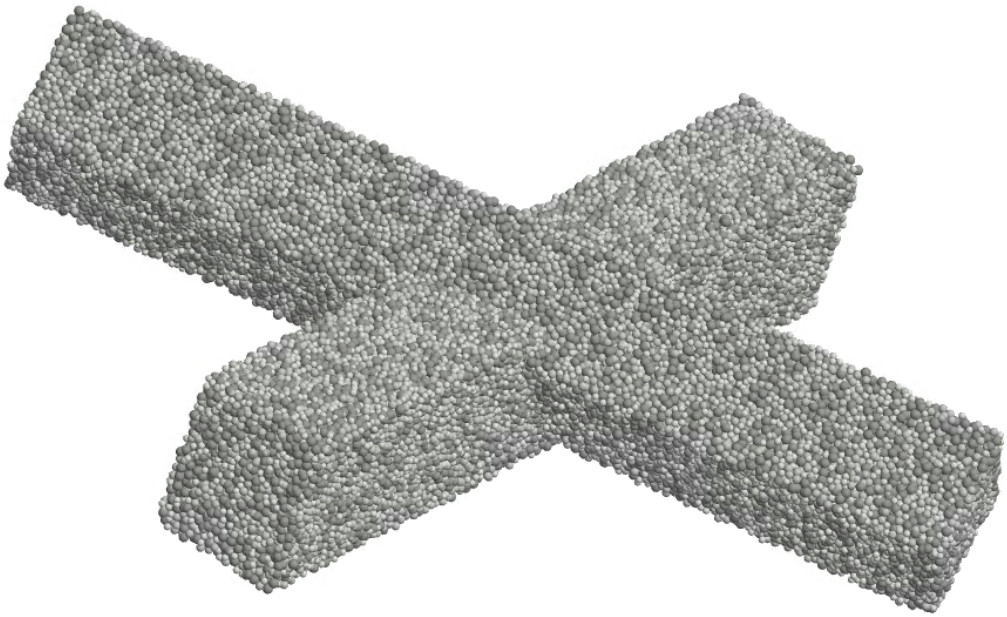

Figure 33: Two cuboidal parent-particles passing through each other without interaction due to the clusters having the same bonding ID

To avoid this scenario, the bonding ID of all particles of the original parent-particle must be adjusted according to their affiliation to the newly formed clusters, for which a cluster detection algorithm is required. However, the assignment of bonding IDs also brings another advantage; it is possible to sum up the properties of all sub-particles within a cluster and thus obtain the total mass of a particle

cluster, its center of gravity, the cumulative force acting on it, as well as the translational and rotational velocity of the cluster. This is especially useful for data evaluation since it allows a simple analysis of the fracture fragment sizes, as well as a good visualization. In figure 34, the comparison between the mass analyzation with and without the cluster detection algorithm is showcased. While usually only the mass of the individual sub-particles can be accessed, the total mass of the cluster is calculated by the algorithm implemented into the software's API.

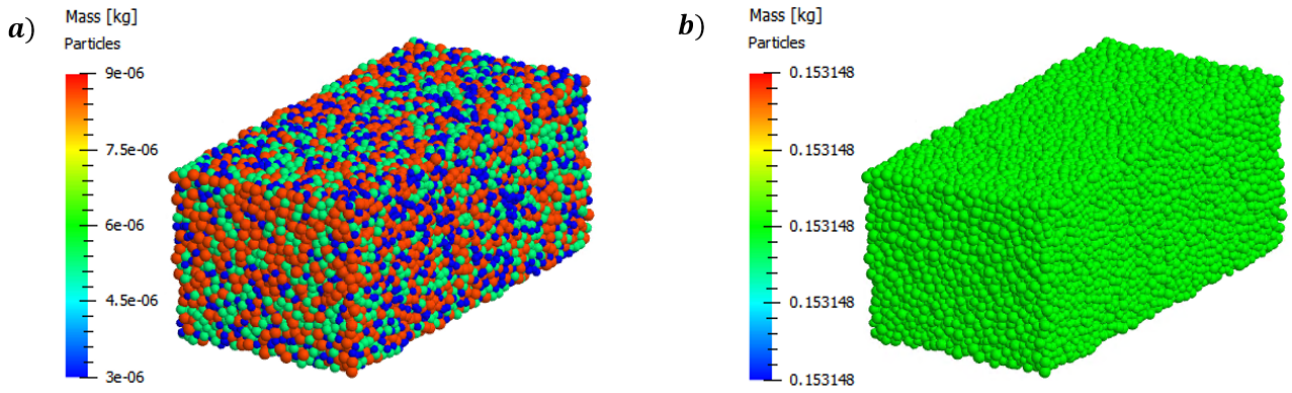

Figure 34: Difference between the mass analyzation without (a) and with (b) the cluster detection algorithm

4.2.1 Existing algorithm

Variations of cluster detection for determining the size distribution of fracture fragments [70], which are based on the Breadth First algorithm [71], already exist. In this user case, the determination of the size distribution is only performed after the simulation for data evaluation; however, the basic approach for contact detection is similar to the one used in this thesis, with the main difference compared to a Breadth First algorithm being that not every connection on a certain depth of the Breadth First Tree needs to be evaluated before moving to the next level, as seen in figure 35, meaning that not every immediate neighbor of the starting particle must be found first before searching for the neighbor's neighbors.

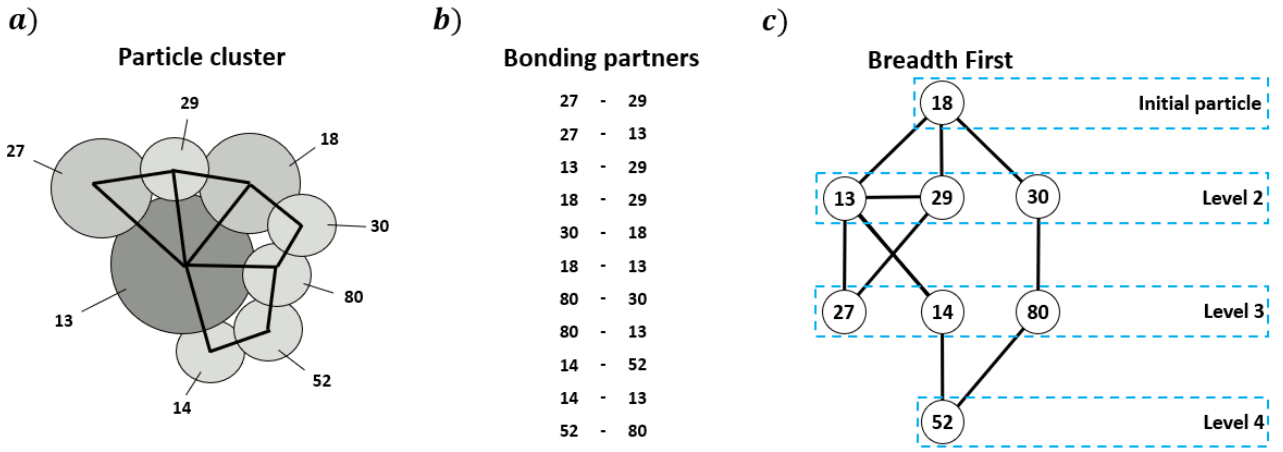

Figure 35: Schematic depiction of the cluster detection using a Breath First algorithm starting with particle ID 18 (random) of the particle cluster (a), identifying the direct neighbors by analyzing the bonding partners (b) on level 2 of the Breadth First tree, before moving on to the next level, until the whole cluster is identified (c)

4.2.2 Detection process

The cluster detection method is called in a wide variety of scenarios, but the basic procedure for finding particle affiliation is always the same. Particle affinity is determined based on detected contacts given by the bonds. The first two bonded particles are assumed to form the initial cluster, and all further particles are added to this cluster, or form a new one, depending on whether contact to the particles in the existing clusters was detected or not. If a contact is detected between two clusters, they are joined. That way, several clusters may be identified during the detection algorithm that are joined later on, which is not the case in the Breadth First Algorithm. Since in this work, it is indispensable that the cluster detection is carried out during the simulation, a way must be found to execute it as efficiently as possible, meaning that the execution is triggered only when it is necessary, and the computation time of the algorithm is minimized by forward-thinking data management.

Clusters identified by this algorithm are stored in several multidimensional vectors. The first of these vectors, from now on called *IDCombinations*, contains several sub-vectors in which the unique IDs of all sub-particles located within a cluster are stored. Furthermore, two additional vectors are needed, called *IDA* and *IDB*, which contain the raw data of all bonding combinations (the raw data will be needed in further steps).

To simplify data association, the position of data belonging to the same cluster must be the same in all vectors, meaning that the IDs of particles belonging to a cluster are located at the same index in *IDCombinations* as the corresponding raw data of bonding partners in *IDA* and *IDB*, as shown in figure 36.

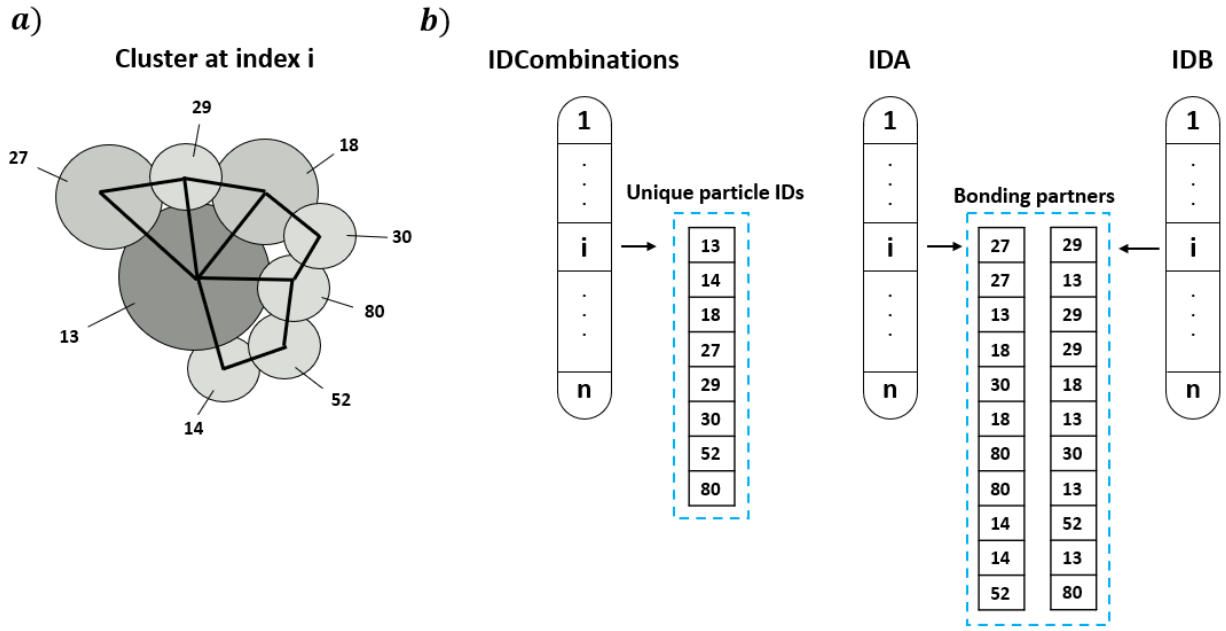

Figure 36: Schematic depiction of how the sub-particle IDs belonging to a particle cluster (a) are stored in several multidimensional vectors of the size n, the number of clusters present in the simulation, at the same index i (b)

As the input of this algorithm, illustrated in the pseudocode (algorithm 2) on the following page, only two vectors of bonding partners are required. The output of the algorithm is similar to the three multidimensional vectors described above. They are called *IDCombinationsCalculated*, *IDACalculated*, and *IDBCalculated*. They have the same structure but only contain data of the clusters calculated by the algorithm, whereas *IDCombinations*, *IDA*, and *IDB* store the data of every cluster present in the simulation. At the heart of the algorithm, a subroutine scours the already detected clusters entered in *IDCombinationsCalculated* for IDs of the currently accessed bonding partners. Depending on whether one of the two or none of the IDs are found, a new cluster is then created, existing ones are extended or connected. If the cumulative force, the total mass, the translational velocity, or any other further information of a cluster is required, the algorithm can be extended by additional subroutines. However, this prolongs the simulation time and is therefore recommended to be carried out by post-processing.

Algorithm 2 Cluster Detection Algorithm

Input: *BondingPartnerA*, *BondingPartnerB*
Output: *IDCombinationsCalculated*, *IDACalculated*, *IDBCalculated*

```
Initialize IDCombinationsCalculated, IDACalculated and IDBCalculated as multidimensional vectors
Write the first entries of both the bonding partner vectors as the first entries of the first sub- vector of
 IDCombinationsCalculated, IDACalculated and IDBCalculated
for i = 0; i < BondingPartnerA.size(); i++ do
    IDA, IDB ← BondingPartnerA[i], BondingPartnerA[i]
    ClusterIndexA, ClusterIndexB ← findInIDCombinationsCalculated(IDA, IDB)      ▷ Index -1 if ID was not found
    if ClusterIndexA == ClusterIndexB then
        Bonded Particles are in the same Cluster
        joinClusters(ClusterIndexA, ClusterIndexB)
    else if ClusterIndexA == −1&ClusterIndexB == −1 then
        Both Particles are not present in any cluster detected yet
        newClusters(IDA, IDB)
    else if ClusterIndexB == −1 then
        First particle of bonding partner was located in a cluster. ID of other partner is added
        addID(ClusterIndexA, IDB)
    else
        Second particle of bonding partner was located in a cluster. ID of other partner is added
        addID(ClusterIndexB, IDA)
    end if
end for
for i = 0; i < IDCombinationsCalculated.size; i++ do
    for j = 0; j < IDCombinationsCalculated[i].size; j++ do
        newBondingID(IDCombinationsCalculated[i][j])      ▷ Assign new bonding ID to Particles in Cluster
    end for
end for
Further sub-routines if additional infromation e.g. cluster mass is of interest
```

4.2.3 Algorithm triggers

The way the input of the algorithm is determined and how the vectors, which are obtained as output of the cluster detection algorithm, are further used depends on different triggers for the algorithm's execution. Two scenarios come into question that are basically similar. At the start of the simulation and then after each call of the flexible inlet API, the bonding partners used as the algorithm's input are known (at the start of the simulation, all created bonds are taken as input, and after a generation of bonds, these newly added bonds are correspondingly known). In the simulation software, all particles, as well as all bonds, are defined as objects stored in a respective particle or bonding vector. Therefore, only an index range of the bonding vector is required, from which the clusters are to be calculated (the

bonding partners are an attribute of a bonding object). The only difference in these scenarios is that after a call to the flexible inlet API, the global *IDCombinations*, *IDA*, and *IDB* vectors are extended by the algorithm's output, whereas at the start of the simulation, they are overwritten. In these cases, a large amount of bonding partners is used as input, resulting in the execution of the algorithm taking considerably longer compared to the following scenario. However, the longest computation time, at the start of the simulation, is performed only once. With a reasonable choice of generation steps for the flexible inlet, the cluster detection is also called rarely for this scenario.

The most common scenario leading to a required execution of the algorithm is due to the breakage of a particle cluster. When a bond is marked as broken, the bonding partners are stored to identify the clusters to be recomputed. After the clusters are determined, the corresponding sub-vectors of *IDA* and *IDB* are sequentially fed as input to the cluster detection algorithm. This guarantees that only one old cluster is recalculated at a time, and, thus, the amount of input data is minimized since multiple executions of the algorithm with small amounts of data take less computing time than a single execution with the entire amount of data (due to memory management when vectors are combined or deleted in the detection algorithm).

Depending on the algorithm's output, a distinction can be made as to whether the cluster was only damaged or if it broke into several subclusters, whereas the first case does not require a new assignment of the bonding IDs.

Additionally, a distinction can be made between further triggers for this scenario:

1. **Every time step**

   At each time step, bonds are checked if broken up (deleted) – if so, the clusters that have been damaged are identified, and finally, the cluster detection is executed. Because of very small time steps, in the range of $10^{-5}s$ to $10^{-6}s$, only a few, if any, bonds fail every trigger event. Therefore, clusters breaking completely apart at every time step is highly unrealistic. This frequent execution of the algorithm in which mostly no breakage of a cluster occurs is very computationally expensive; however, it should be noted that this method produces the most accurate results at the expense of computational efficiency.

2. **Depending on the number of broken bonds**

   As previously mentioned, the number of bonds that broke since the last cluster update is counted and evaluated at every time step. If this number exceeds a critical value, the cluster detection is initiated. The selection of this number of broken bonds, triggering the cluster detection, can be based on, e.g., the minimum number of bonds in the smallest cross-section of the parent-particles. However, since the probability of particle clusters getting smaller increases with simulation time, this choice should be considered carefully.

3. **Depending on the time step**

   When only executing the cluster detection algorithm at a defined interval of multiple time steps (instead of checking at each time step, such as in bullet point 2), a lot of computational time can be saved. The disadvantage of this trigger event is that within the time period passed from one to the next interval of time steps, a very large amount of bonds may fail, thus likely breaking up clusters into several smaller ones without detecting them at their actual occurrence of complete cluster breakage. However, this scenario is extremely unrealistic due to the very small time step.

4. **Combined trigger event**

   A trigger event dependent on a combination of the time step and number of broken bonds was ultimately chosen as suitable. This saves computational effort by only checking the need for an update every interval of multiple time steps and in addition not immediately carries out the cluster detection but checks how many bonds were broken since the last update. Furthermore, all damaged clusters are updated every multiple of the time step trigger event, regardless of whether the number of damaged bonds has been reached. While reducing simulation time provides a great advantage, choosing a trigger event that is too large can lead to instability and erroneous effects consequently, as it can result in an update of the bonding ID when two fracture fragments are already largely overlapping. However, due to the relaxation after breakage, addressed in the following chapter, and the small time step, an execution every, e.g., 50 time steps is still plausible (this should still be checked for plausibility since the amount of time steps over which a particle cluster breaks apart strongly depends on its velocity in the simulation, thus

on the actual simulation setup). The number of bonds that trigger the detection should be based on the minimum expected fragment size in the simulation.

This trigger event of a combination of the time step and number of broken bonds was ultimately chosen as suitable for the given simulation case of filter cakes.

The following pseudocode (algorithm 3) is an illustration of the algorithm that is used to trigger the cluster detection. It takes two boolean parameters, *simulationStart* and $flexibleInlet$, as input to check if the simulation has just started or if newly added clusters need to be evaluated. The other input parameters, $numberOftimesteps$, and $numberOfBrokenBonds$, specify the number of times after which the need for an update is checked and the number of broken bonds that triggers the actual cluster detection algorithm. *brokenBonds* contains the actual bonding partners of bonds that broke since the last execution of the algorithm.

Algorithm 3 Cluster Detection Algorithm Triggers

Input: *simulationStart*, *flexibleInlet*, *numberOftime steps*, *numberOfBrokenBonds*, *brokenBonds*

if *simulationStart* **then** ▷ Calculate clusters from all bondings
  $bondingIDmin = 0$
  $bondingIDmax = bonding.size()$
  $BondingPartnerA, BondingPartnerB \leftarrow$ getBondingPartners$(bondingIDmin, bondingIDmax)$
  $IDCombinations, IDA, IDB \leftarrow$ clusterDetection$(BondingPartnerA, BondingPartnerB)$
else if $flexibleInlet$ **then** ▷ Calculate clusters from newly added bonds
  $bondingIDmin = getNumberOfBondsBeforeInlet()$
  $bondingIDmax = bonding.size()$
  $BondingPartnerA, BondingPartnerB \leftarrow$ getBondingPartners$(bondingIDmin, bondingIDmax)$
  $IDCombinations, IDA, IDB \leftarrow$ appendClusters(clusterDetection$(BondingPartnerA, BondingPartnerB)$)
else if modulo$(calculationTime, numberOfTimesteps) == 0$ **then** ▷ Check every $numberOfTimesteps$
  if $brokenBonds.size() > numberOfBrokenBonds$ **then**
    $ClusterIDs \leftarrow$ getDamagedClustedIDs$(brokenBonds)$
    $BondingPartnerA, BondingPartnerB \leftarrow$ getBondingPartnersFromClusters$(ClusterIDs)$
    $IDCombinations, IDA, IDB \leftarrow$ updateClusters(clusterDetection$(BondingPartnerA, BondingPartnerB)$)
  end if
end if

After the last bond connected to a particle breaks, the bonding ID is automatically reset by the software to -1, which corresponds to the default value and automatically allows all interactions by the master and slave contact models again.

4.3 Relaxation after breakage

As described in the previous chapter, after the breakage of a cluster, particles are assigned new bonding IDs according to their membership to the resulting sub-clusters. These particles, now belonging to different clusters and able to interact with each other, may still have relatively large overlaps. The abrupt forces resulting from these overlaps lead to a strong repulsion of the particles, which not only pushes the newly formed clusters apart but can also lead to the failure of further bonds. An extreme case, in which the parent-particle "explodes" after impact is shown in figure 37. This phenomenon is quite common for simulations for which parameters have not been set correctly, especially regarding the time step set too high; such cases must be avoided at all costs.

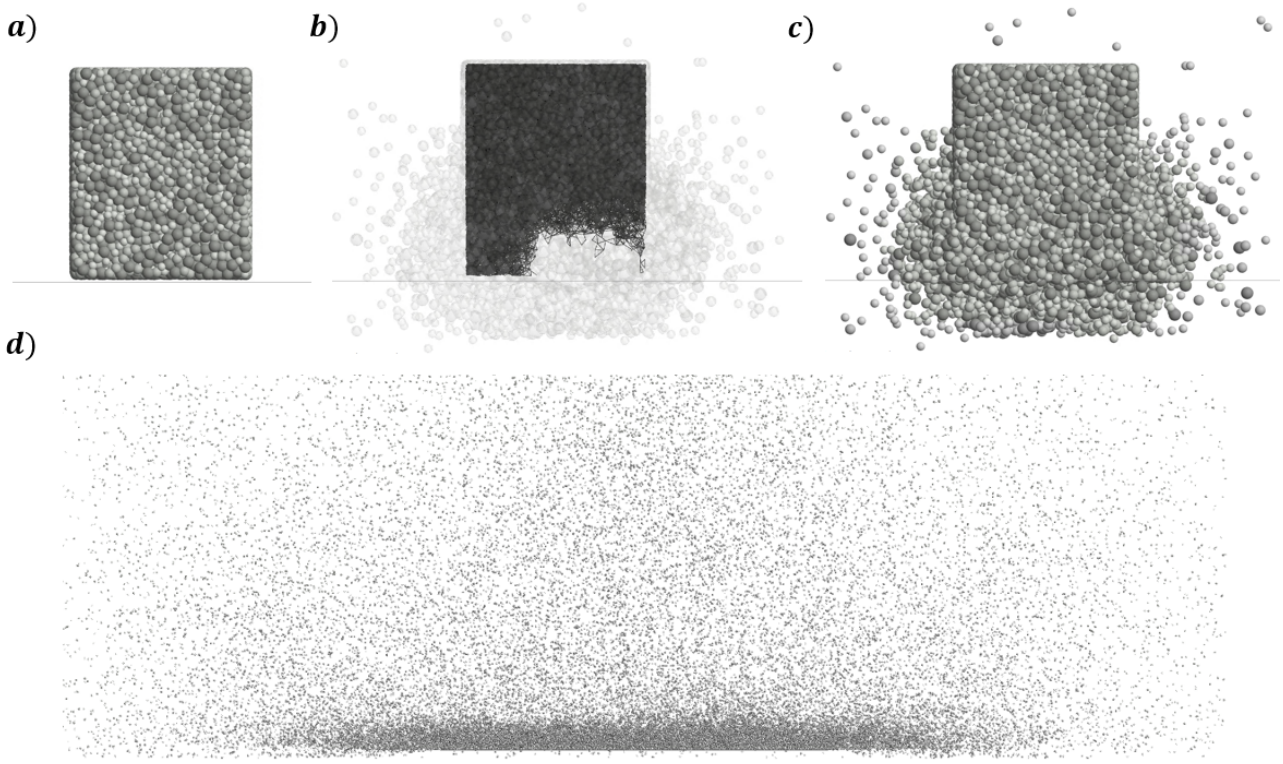

Figure 37: Impact of a parent-particle without relaxation after breakage (a), leading to strong repulsive forces due to the initial overlaps of the sub-particles, in turn, resulting in the breakage of further bonds (b) and the repulsion of more sub-particles (c) and therefore to a chain reaction that causes the entire parent-particle to explode (d)

To avoid this scenario, these abruptly occurring, high repulsive forces must be disabled. This is accomplished in similar cases (such as the PRM) with a relaxation factor [25]. However, a constant reduction factor, which neither depends on the particle size nor the overlap, either repels the interacting particles too fast or allows them to be further compressed without much resistance. Either way, the residual forces after reduction produce local elastic overstresses [27] in the resulting sub-clusters. If a filter cake plate is cut in the middle, the resulting halves are neither pushed apart nor can they pass through each other. For this reason, this method will not be discussed any further.
The method introduced in this thesis aims to negate repulsive forces after bond breakage completely until the particles are separated, while also preventing the particles from being pressed further into each other, which would be the case if the contact force would be reduced by a constant factor or completely suspended; the cut surfaces are therefore still able to interact with each other.

For this reason, the initial overlap of two particles, at the time step they are able to interact again for the first time after the breakage of the bond, is used as the point of origin for a Hertz-Mindlin (HM) contact model. Thus, if the particles are further compressed, a contact force corresponding to equation 3 is the result. However, if the overlap of the particles is decreased, the contact force is negated, and the current overlap is set as the new point of origin. This procedure continues until the particles are separated, completing the relaxation process, and therefore ensuring the ability of cut surfaces to interact while preventing the fracture fragments from being repelled, as shown in figure 38. This displacement of the HM contact models origin causes the interacting fracture fragments to behave as if they had smaller, temporary, radii, which are continuously increased during the relaxation process [72].

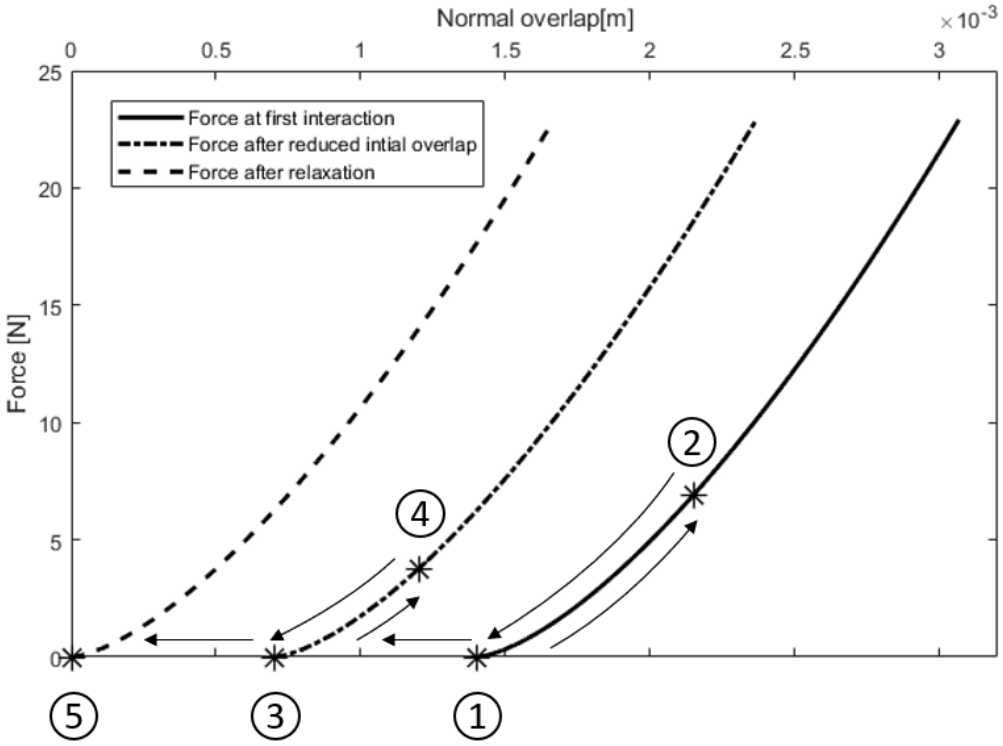

Figure 38: Relaxation process starting at the overlap of the first interaction after the bond connecting two particles broke (1), resulting in a repulsive force according to the HM contact model with a displaced origin after further compression of the particles (2), no contact forces at overlaps smaller than the initial one (3), which in turn sets a new origin for the HM contact model in the case of compression following the relaxation (4) and finally completing the relaxation process and returning to the original HM contact model (5)

The relaxation process is carried out by an algorithm implemented as a slave contact model in the API of the simulation software. It is only called if two overlapping particle pairs have been marked as candidates for relaxation, which is done by means of the Co-Simulation API for every newly generated parent-particle. The mark for relaxation is removed as soon as the overlap falls below a minimum distance relative to the effective radius of the particle pair. The following pseudocode (algorithm 4) is an illustration of the algorithm implemented into the slave API.

Algorithm 4 Relaxation Algorithm

Input: *MarkForRelaxation*, *Overlap*, *minDistance*
Output: *ContactForce*

```
if MarkForRelaxation then
    if Overlap < minDistance then                          ▷ Relaxation process is finished
        unmarkParticles()
    else
        InitialOverlap ← getInitialOverlap()               ▷ Current origin for HM contact model
        if InitialOverlap == −1 then                       ▷ InitialOverlap is set to -1 at first interaction
            ContactForce ← setContactForcesToZero()
            newInitialOverlap(InitialOverlap)
        else if InitialOverlap < Overlap then              ▷ Relaxation in process
            ContactForce ← setContactForcesToZero()
            newInitialOverlap(Overlap)
        else                                               ▷ Compression of particles past the origin point
            ContactForce ← setNewContactForce(InitialOverlap, Overlap)
        end if
    end if
end if
```

The functionality of the relaxation model is determined by dropping the parent-particle shown in figure 39 again under the same conditions, whereby the critical equivalent stress of the bonds is reduced to guarantee the failure of all bonds at impact. If the relaxation model works as intended, the parent-particle should now break into its smallest fracture fragments, which instead of ”exploding” subsequently form an angle of repose, as after the breakage of all bonds a bulk solids consisting of the sub-particles results. As can be seen in figure 39, that is the case, thus proving the model’s functionality as suitable.

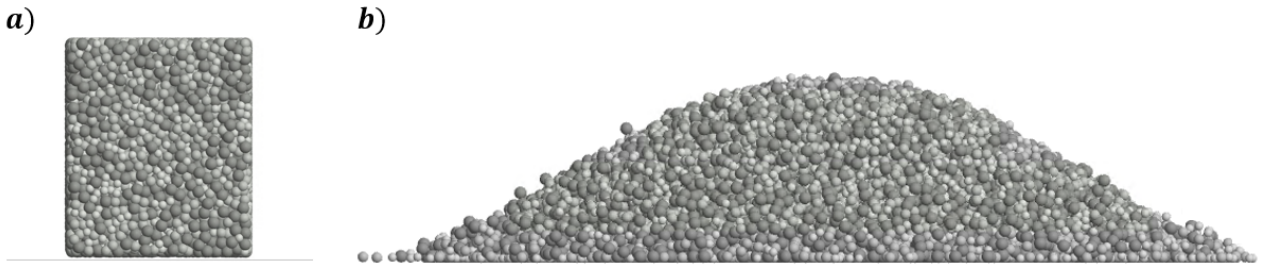

Figure 39: Impact of a parent- particle with relaxation after breakage (a) resulting in the fracture fragments forming a bulk solid pile with the corresponding angle of repose

4.4 Calibration test

This chapter covers the setups for the calibration tests selected in section 3.3, as well as the preparation and evaluation of the gathered data.

4.4.1 Four-point bending test

The procedure of the four-point bending flexural test, as well as the sample dimensions, are specified in various international standards, as already addressed in subparagraph 3.3.1.1.2. However, since in this specific application, the height of the specimen is given by the filter press, a slightly modified sample geometry and position of the force application are selected, as shown in figure 40.

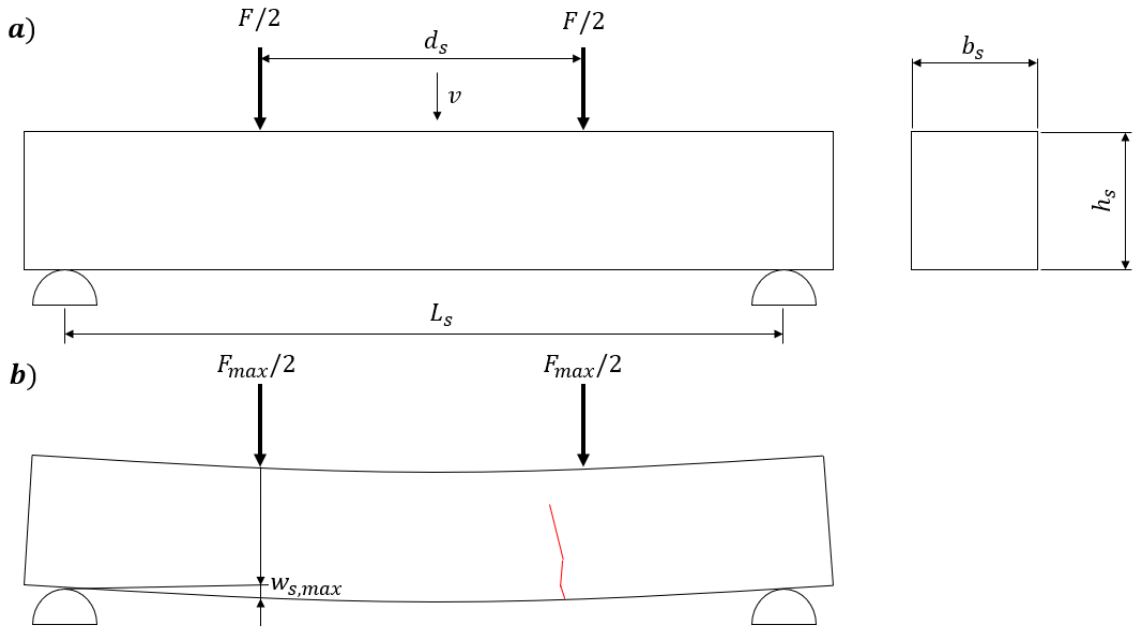

Figure 40: Schematics of the four-point bending test, showing the beginning of the experiment (a) and the breaking of the test sample (b)

The length L_S of the test sample was set at 180 mm, the width b_S at 30 mm, the sample thickness h_S given by the filter press, is 35 mm, and the distance between the loading points d_S is 80 mm.

4.4.1.1 Setup

To characterize the material behavior under bending load, the deformation of the material sample up to the point of failure must be determined in addition to measuring the force required for the specimen's distortion. For this purpose, as mentioned in subparagraph 3.3.1.1.1, any universal testing machine equipped with a four-point bending fixture can be used. However, since universal testing machines are usually used for much larger force ranges, a test rig, shown in figure 41, was designed specifically for this type of filter cake material. It is controlled by a microcontroller (Arduino Mega 2560) and records the force over the traveled path of the loading pins.

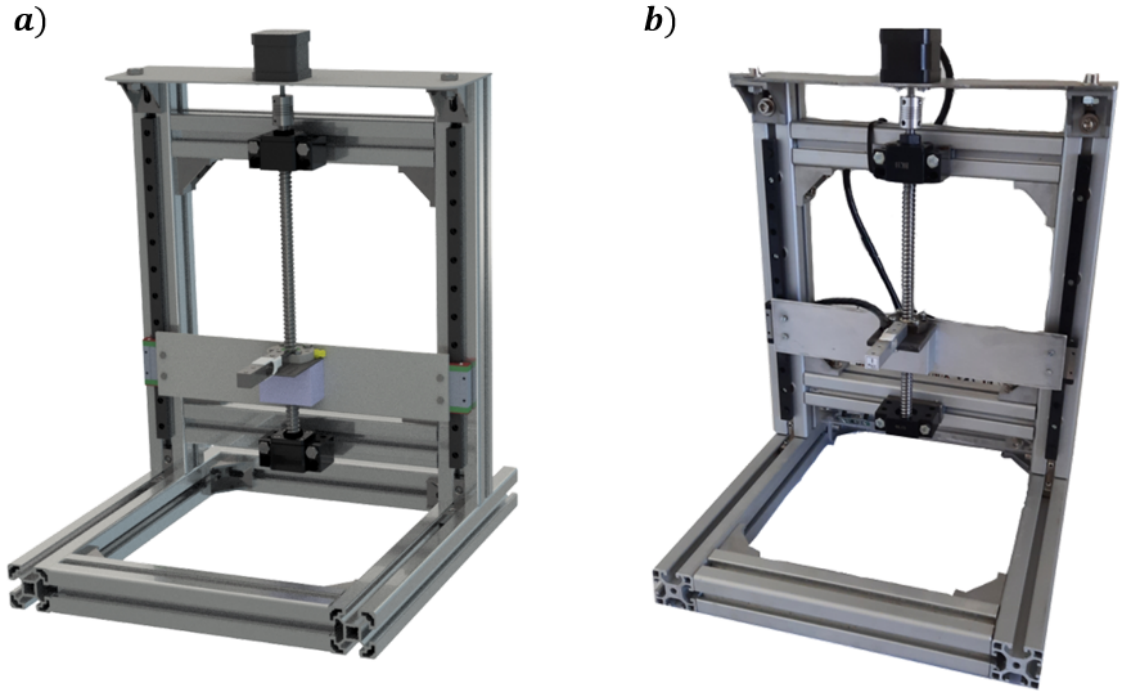

Figure 41: CAD design of the test rig (a) and its physical setup in the laboratory (b)

Before carrying out the bending tests, the measuring unit of the test rig must be calibrated, for which a calibration program is used. Following the calibration, the test rig is equipped with the FPBT adapter, which, before each test cycle, must be checked to ensure that the inserted specimen is level and is loaded by both pins at the same time; slight deviations due to an inconsistent thickness of the filter cake sample are tolerated.

Afterwards, the samples are cut from the filter cake plate and are measured and checked for any existing cracks before being placed in the test rig; specimens that are already damaged are considered invalid and discarded. After the loading pins are positioned slightly above the material sample, via the test rigs fast travel mode, the test procedure is started (by executing the main program) whereby the specimen is loaded at a deformation rate v of $0.0002\frac{m}{s}$ to guarantee quasi-static material behavior until failure occurs while recording the force over the travelled path of the loading pins. Only specimens that fail under pure tensile stress are considered valid, so specimens that fail due to a crack forming outside the central area between the two loading pins are discarded.

4.4.1.2 Data Evaluation

The forces measured by the test rig at a specific deformation are evaluated to determine characteristic material properties. Initially, the measured force is plotted over the deformation of the filter cake sample, resulting in a curve as shown in figure 42. This plotted curve can be divided into several areas: Since the loading pins are positioned slightly above the material sample before the start of the test, as described in paragraph 4.4.1.1, a distance without resistance is covered by the pins up to the initial contact with the specimen, see figure 42-1. As described in the previous subchapter, the thickness of the filter cake plate from which the samples are cut is not always consistent, leading to the specimen not lying perfectly level. This causes one of the loading pins to contact the sample's surface before the other and thus slightly twisting the specimen, resulting in the initially flatter force slope in figure 42-2. If this area is large in relation to the following one, the specimen is also discarded since it results in prestresses that falsifies the result. Then, the linear-elastic region of the material (that is of interest) starts, marked as region 3 in figure 42, and from which the material properties are derived in the following chapters. As soon as the first cracks on the sample's surface occur, the slope of the graph decreases until the specimen breaks abruptly upon reaching a critical crack length, marking the end of region 4. All subsequently measured forces, see figure 42-5, are discarded since they were recorded after the failure of the specimen. They originate, for example, from one half of the broken specimen getting wedged between the loading pin and the ground.

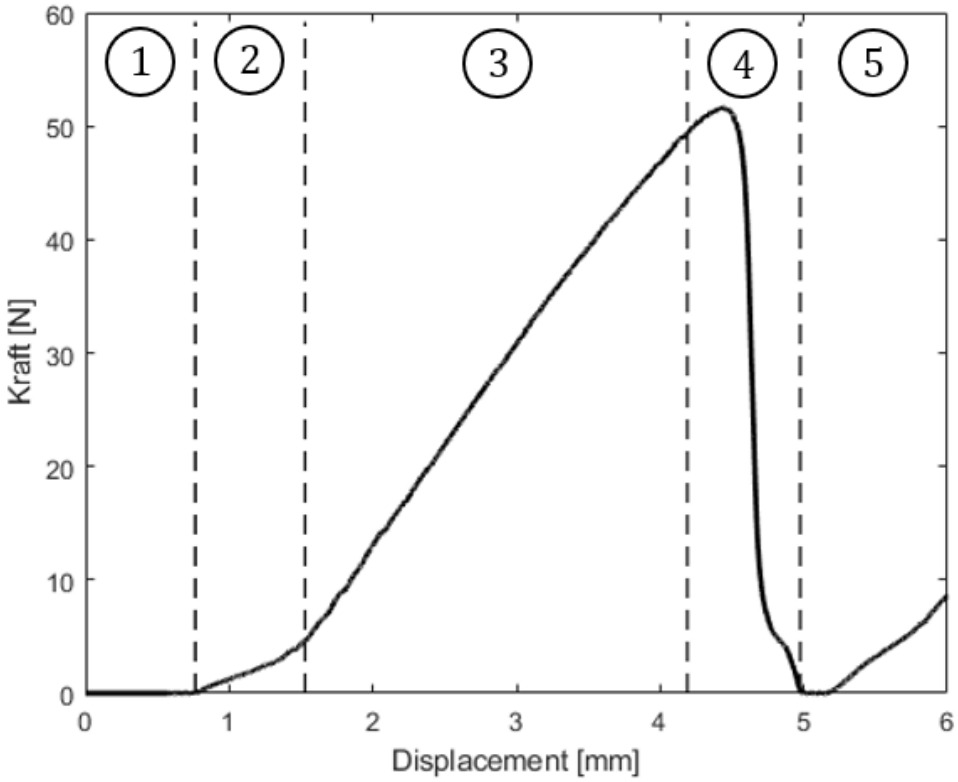

Figure 42: Raw data of the measured force-deformation relation of a material sample

To obtain data sets that are comparable with the simulation results as well as other samples, they must first be processed. This is done by discarding unnecessary data, i.e., area 1 and 5 from figure 42, and compensating force offsets if the load cell was not zeroed. Subsequently, the data is filtered from influences of an inconsistent sample thickness. This is done by scaling the force-deformation curve according to the sample's second moment of area and additionally shifting the curve to ensure that the linear-elastic region starts in the coordinate origin. This processing of the data is illustrated in figure 43.

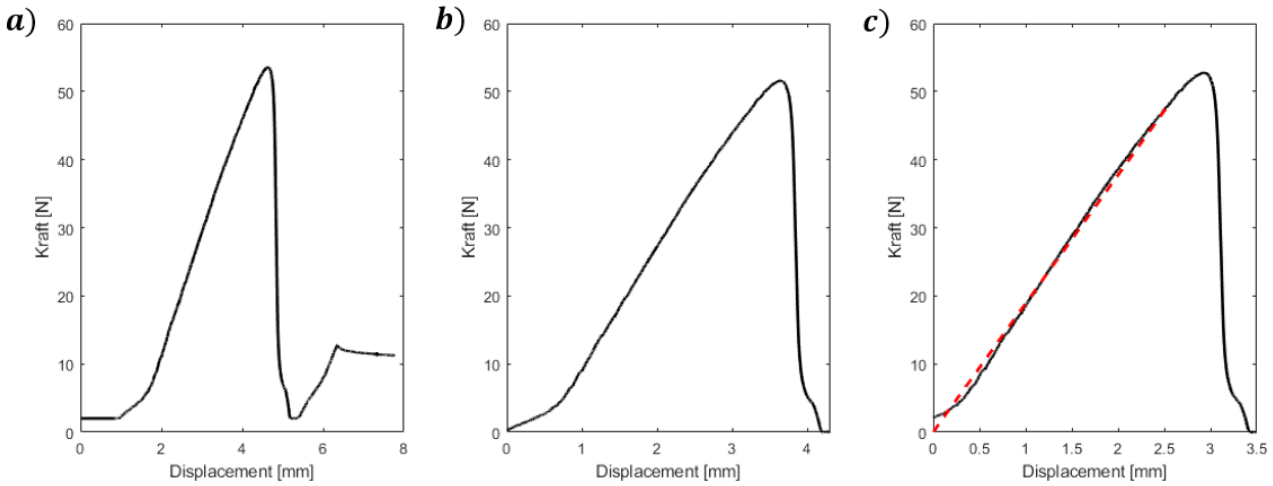

Figure 43: Preparation of the raw data gathered by the test rig (a) eliminating invalid data and unwanted offsets (b) and finally adjusting the graph to compensate for the samples second moment of area in addition to repositioning the graph for the linear material behavior to start in the origin of the coordinate system (c)

The maximum force and the corresponding deformation, as well as the limits of the linear elastic range, are averaged over all specimens. This averaged data is then used to calculate material data, such as the Youngs Modulus, as described in paragraph 4.5.1.2.

4.4.2 Drop test

To obtain a visual comparison of the dynamic behavior of the actual and the simulated material, a drop test is performed. The impact velocity of the sample should approximately correspond to the maximum impact velocity in the final to-be-simulated process, which in this case is about $4\frac{m}{s}$, and is set by dropping it from a set height. Based on the acceleration of a body in free fall, a drop height of $815mm$ is calculated, as corresponding to the $4\frac{m}{s}$ velocity resulting at impact. The rebound behavior of the specimen, investigated by a drop test onto a level surface, as well as the damage to the material during an impact on a surface inclined by various angles α ranging from 15° to 25° with different orientations of the specimen's area of particular interest . In this test, the geometry of the material sample is the same as in the four-point bending test. A schematic depiction of the drop tests is visualized in figure 44.

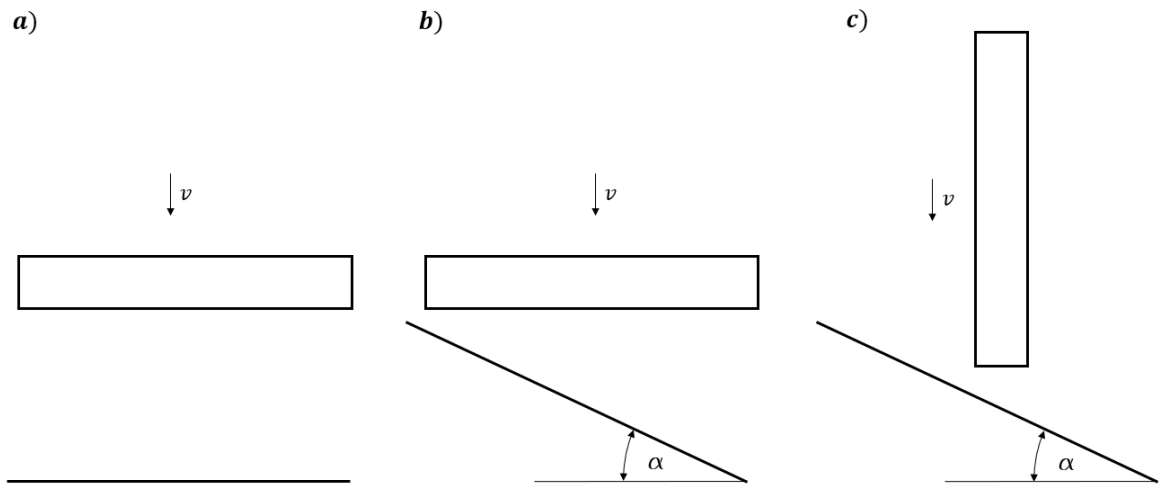

Figure 44: Schematic depiction of the drop test variations onto a level surface for the determination of the material's internal damping (a) and the drop of a horizontally (b) or vertically (c) aligned sample onto an inclined surface to visually analyze the material's dynamic behavior in those cases

The observed rebound behavior during a drop test of the specimens on a flat surface resulted in a rebound height of only a few percent of the drop height, which suggests relatively high internal damping. It should be noted, however, that a perfectly parallel alignment of the specimen to the impact surface is not possible in physical setups, resulting in some of the energy being transformed into rotational energy of the specimen, albeit a small one, and therefore further reducing the rebound height. However, as long as this is taken into account in the simulation, this influence is negligible in the laboratory test. The impact of the specimen on an inclined surface further shows that the material is not damaged by the collision, during which the sample is strongly deformed, and minor surface cracks are only formed after dropping the same specimen several times.

4.5 Calibration of model parameters

To ensure an efficient calibration of the simulation model, the parameters with the most significant influence on the simulation results are determined. These parameters are both general simulation parameters, such as the time step and particle size, as well as model-specific micro-parameters.

4.5.1 Conventional way

Before the simulation parameters are optimized to ensure an efficient simulation on an industrial scale, the conventional method for the calibration of the micro parameters is shown, which is mostly based on the analytical determination of the physical parameters.

4.5.1.1 Time step

The time step (Δt) has a great influence on a DEM simulation since its size affects the computational efficiency as well as the stability and accuracy of the simulation. For this reason, it should be chosen as large as possible but not exceeding a critical value at which the simulation tends to become unstable. Influenced by several different simulation parameters, this critical value also depends on the type of interaction.

For particle-particle interactions or particle interacting with system components, about 20% of the Rayleigh time step $t_{Rayleigh}$ [73] is usually used to determine a suitable time step, which corresponds to

$$t_{Rayleigh} = \frac{\pi \ R \ \sqrt{\frac{\rho}{G}}}{0.1631 \ \upsilon + 0.8766} \tag{14}$$

with the particle Radius R, density ρ, shear Modulus G, and Poisson's ratio υ, all being constant for all particles except the Radius, resulting in the critical time step being that of the smallest particle Radius.

For bonded structures, the critical value is calculated from the critical vibration frequency of the particles connected with massless bonds [74], following

$$t_{Bond,crit.} = 0.17\sqrt{\frac{m}{K}} \tag{15}$$

in three-dimensional space, with the particle mass m and the bond stiffness K. This value is determined for the smallest, and therefore the lightest, particles bonded together with the highest bond stiffness. The time step used in the simulation is the smallest critical time step, either from the bonded or the non-bonded contacts.

$$\Delta t = min(t_{Rayleigh}; t_{Bond,crit.}) \tag{16}$$

4.5.1.2 Bond stiffness

The stiffness of the bonds corresponds to the macroscopic Youngs Modulus, which can be measured by means of local instrumentation during the calibration tests. In this case, the material's Youngs Modulus E is calculated from the beam deformation w_s at a loading point under a load F according to the elastic beam theory. For this purpose, the material specimen is considered an elastic beam. According to the Euler-Bernoulli beam theory, the differential equation of the deflection curve of a beam is given with

$$M = -EI\frac{d^2w}{dx^2} \tag{17}$$

with the bending moment M, the bending stiffness EI, and the second derivative of the beam deflection w in respect to the coordinate x [75]. To obtain the deflection of the beam at the location of the loading pins, which is known, the bending moment of the beam is calculated, and the differential equation is solved by integrating two times. For this purpose, the beam is partitioned two times, and the bending moment is calculated for each region, as shown in figure 45 a-b.

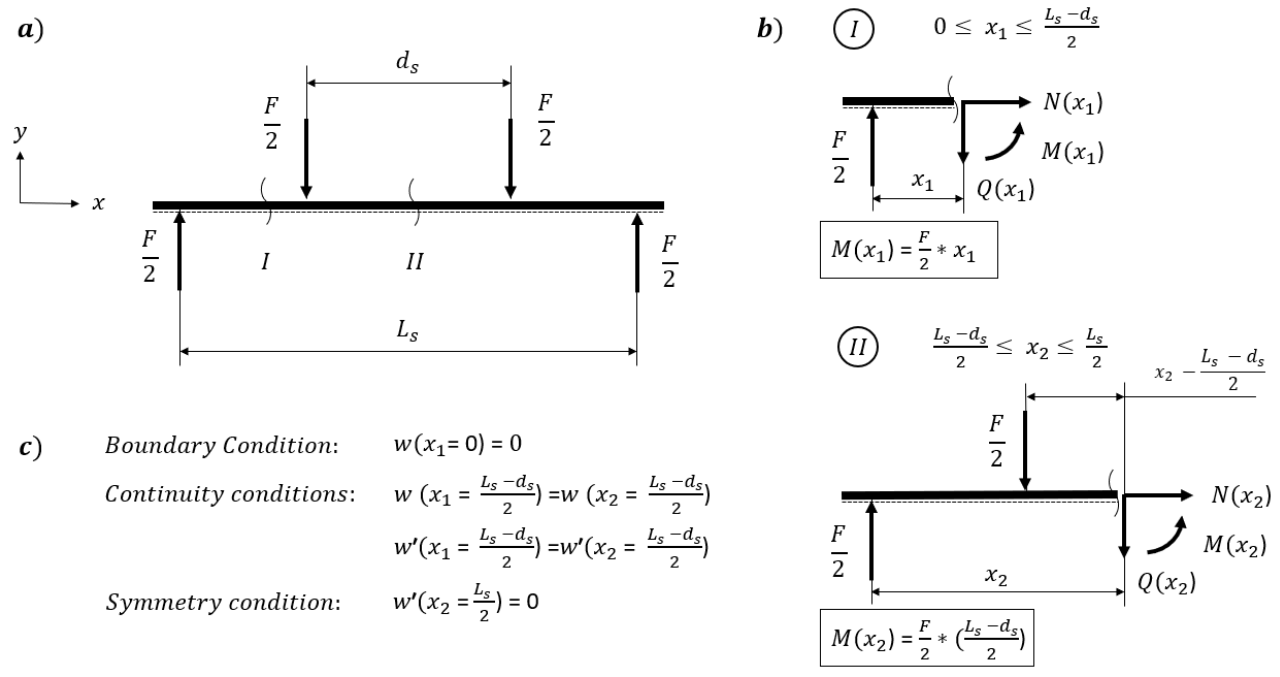

Figure 45: Partitioning of an elastic beam undergoing a FPBT (a), calculating the bending moment in each region (b) and evaluating the known conditions (c)

With the four known conditions, all integration variables of the two beam regions are calculated, and an equation for the deflection is established. The deflection of the beam at the location of the loading pins corresponds to the deflection of region I at the coordinate $x = \frac{L_s - d_s}{2}$, described by the following equation

$$w_s = \frac{F}{48\ E\ I}\ (L_S - d_S)^2\,(L_S + 2\ d_S) \tag{18}$$

with the second moment of area I of the sample cross-section. By solving this equation for E and substituting the average values of the average difference of force F and deformation w_s from the start and end of the linear elastic range, respectively, from all valid bending tests, the macroscopic Young's Modulus is calculated, resulting in $12,05\frac{N}{mm^2}$.

SSince the interaction of the sub-particles within a cluster is disabled, the bonds give the parent-particle its material properties in terms of elasticity, breakability, etc. (apart from mass and geometry). However, due to the bonds connecting the particles at their center, there are no bonds in the outermost edge fiber of the parent-particle, but at a minimum distance of the smallest radius. This amounts to a reduction in the effective cross-section, evident in figure 46, which results in a decrease in stiffness.

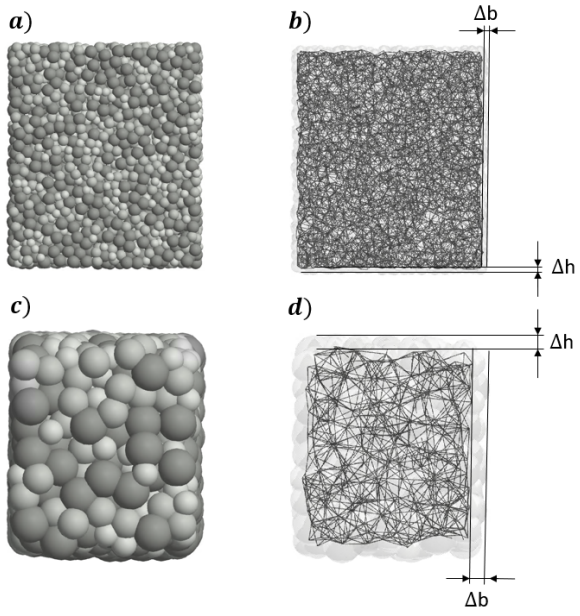

Figure 46: Visualization of the effective cross-section's dependency on the particle size, showcased by the example of a parent-particle composed of small (a) and large (b) sub-particles

Since the Young's Modulus depends on the second moment of area, which in turn depends on the height of the effective cross-section to the power of 3, this influence is negligible if the smallest particle radius is small in relation to the cross-section's height, resulting in only a small deviation of the Youngs's Modulus and can be compensated by increasing the number of bonds, see paragraph 4.5.1.6. However, if the radii of the particles are scaled, as suggested in section 4.5.3, this influence becomes correspondingly relevant.

4.5.1.3 Critical stress

In order to reproduce the results of the laboratory tests, the maximum bond stress at the desired fracture point is evaluated from the calibration simulation, which is then checked and fine-tuned, if necessary, in subsequent simulations. In addition to this, the results can be checked for plausibility by calculating the tensile stress due to the bending moment, according to equation 19.

$$\sigma_b = \frac{3F(L_S - d_S)}{2\, b_S {h_S}^2} \tag{19}$$

Since the compressive strength of rock-like materials is a multiple of the tensile strength, the normal stress component of the bond is only considered under tension and not under compression, as described in section 2.2. The critical stress corresponds to the tensile stress at the peak of the graph seen in figure 42.

4.5.1.4 Sub-particle size

Although the size of the sub-particles is not considered a classic simulation parameter, it has a considerable influence on the results of the calibration simulation. That is why the relation between the smallest distance within the sample geometry L to the average sub-particle diameter d must be taken into account when establishing the PSD used in the simulation. When testing the compressive strength of rock the American Society for Testing and Materials (ASTM) recommends a ratio of L to the maximum grain size d_{max} at a value of 10, while the International Society for Rock Mechanics (ISRM) suggests this ratio to be at least 20. When applied to DEM modeling, a ratio of L/d of 25 is recommended to keep the coefficient of variation of most model parameters under 2% [76].

4.5.1.5 Sub-particle overlap

To depict a particle cluster with the material's solid density, as well as the fracture fragments with its correct bulk density, the parent-particles are generated from overlapping sub-particles, as described in section 4.1.2. After calibrating the particle density from the bulk density, the sub-particles overlap is varied until the parent-particles solid density corresponds to the one of the to-be-simulated material. The overlap is set by the input parameter $desiredOverlap$ in algorithm 1, which reduces the radius of the sub-particles in the calculation; e.g., if the parameter is set as 0.1, the actual radius of the particles in the simulation is 10% larger than the one in the generation algorithm. Two sub-particle neighbors with respective radii of $1mm$ and $0.7mm$ show an initial overlap of $0.155mm$. The solid density of the generated parent-particle, being an output of the generation algorithm, makes the calibration of the overlap relatively easy. Due to the parent-particles being pre-generated, this is done before starting the simulation.

4.5.1.6 Number of bonded neighbors

Even though the filling algorithm generates a particle network consisting of overlapping sub-particles with a very high packing density, not every neighboring particle is in direct contact due to the PSD. That, in addition to a reduced effective cross-section due to no bonds being present in the edge fiber, results in complications with the common generation method for bonds (if the analytical method to determine the Bond Youngs' modulus as described above is used). Since the common method for generating bonds is to bond only overlapping sub-particles together, this results in the parent-particle appearing less stiff than the actual material. This can easily be avoided by bonding the sub-particles to their neighbors that are not only in direct contact but also to those with a slight offset, increasing the number of bonded neighbors and therefore increasing the stiffness of the parent-particle. This is implemented in the generation algorithm by adding an additional offset to a sub-particles radius defined by the input variable *bondingRadius* in algorithm 1.

4.5.1.7 Damping coefficient

Determining the damping factor of bonds within a cluster analytically is more difficult than with other parameters, which is why this parameter is determined iteratively by comparing the simulation results

of the drop test to the laboratory tests. However, since the bonding model has a viscous damping term (cf. equation 7), it is important to note that a higher damping factor requires a smaller time step to ensure a stable simulation. For a bond not being part of a cluster, the selection of the damping coefficient d from 0 to even greater than 1 results in the expected undampened, dampened, critically dampened, or overdamped oscillation, which can be seen in figure 47.

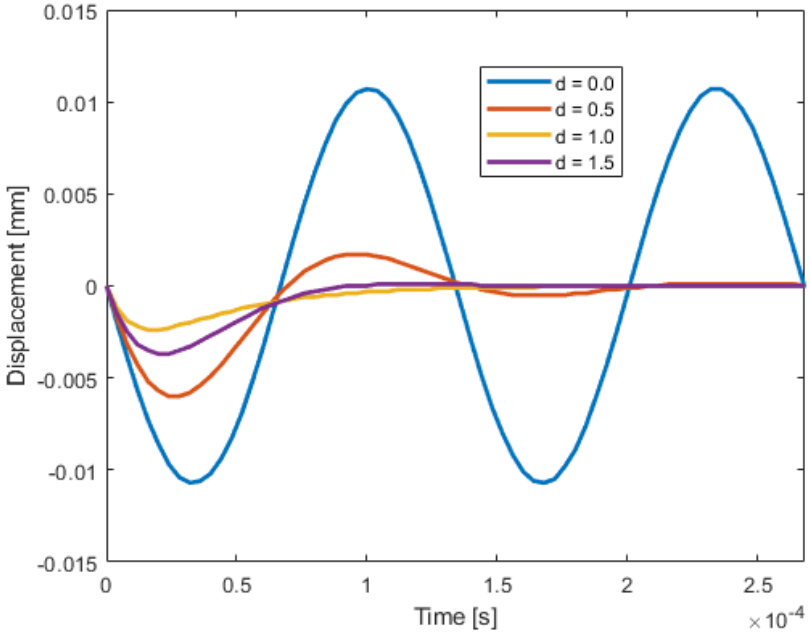

Figure 47: Oscillation of two bonded particles excited by a force

When comparing the simulation results with the drop test on a level surface, complications with the damping term become apparent since, in this case, the particle cluster consists of hundreds of thousands of springs and dampeners influencing one another. This leads to the rebound height not being significantly reduced by an increase of the damping factor, even if it is set greater than 1, which in theory should reduce the amplitude of the deflection, as seen in figure 47, and therefore dissipate more energy. Since, as already mentioned several times, a viscous damping term is used in the bond model, the time step must be decreased if the damping coefficient is increased to ensure a stable simulation. For this reason, to save computation time, it is recommended to set the damping coefficient to the lowest possible value, above which the rebound height is no longer significantly affected by a further increase of the coefficient. This problematic rebound behavior could be solved by using a numerical damping term to dissipate the energy stored in the bonds more effectively.

However, since in the final simulation particle clusters are dragged along with the material flow after impact, this result is tolerated in this thesis.

4.5.1.8 Dynamic increase Factor (DIF)

Rock and rock-like materials respectively experience a change in mechanical properties, such as the material's tensile strength, when exposed to dynamic loading conditions [49] [77], in some situations even exceeding the static properties by one order of magnitude [78]. As described in section 3.3.2, the measurement of these dynamic values turns out to be challenging, which is why in this thesis the dynamic behavior is based on empirical data published in available literature. The influence of the strain rate on the dynamic tensile stress is commonly described by the Dynamic Increase Factor (DIF), i.e., the ratio of the dynamic tensile stress to the static tensile stress. By determining the static tensile stress from the FPBT, the dynamic material property can be derived by means of the DIF, which is calculated after Zhou [79] from

$$DIF = \frac{\sigma_{td}}{\sigma_{ts}} = \begin{cases} 1 + 0.0225 * \lg(\dot{\varepsilon}_z) + 5.3333 & \dot{\varepsilon}_z \leq 0.1 s^{-1} \\ 1.6 + 0.7325 * \lg(\dot{\varepsilon}_z)^2 + 1.235 * lg(\dot{\varepsilon}_z) & 0.1 \leq \dot{\varepsilon}_z \leq 50 s^{-1} \end{cases} \tag{20}$$

where σ_{td} and σ_{ts} are the dynamic and static tensile stresses, with $\dot{\varepsilon}_z$ corresponding to the strain rate, which is calculated from the relative velocity v_r of the particles at the current time step and the initial bond length l_0 according to

$$\dot{\varepsilon}_z = \frac{v_r}{l_0} \tag{21}$$

With the DIF depending on the logarithm of the strain rate, it is usually also plotted over $\lg(\dot{\varepsilon}_z)$, resulting in the graph seen in figure 48.

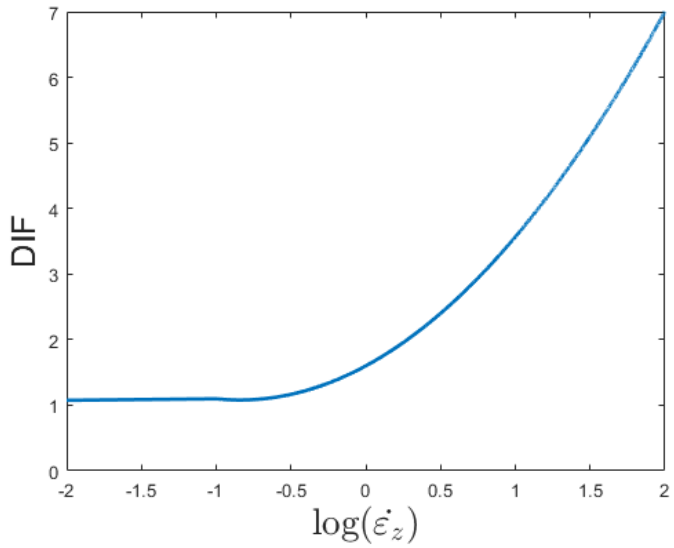

Figure 48: Strain rate dependency of the dynamic tensile strength after Zhou [79]

The course of the graph indicates a transitional strain rate between 10^0 and 10^1 s^{-1}, after which the influence of the strain rate on the DIF sharply increases [78]. Since, in the DEM, particles are subjected to slight oscillations even in quasi-static situations, such as the FPBT, the lower limit of the DIF was changed to $\dot{\varepsilon}_z \leq 1.1\ s^{-1}$, as the static tensile strength measured in the simulation is otherwise increased by a few percent.

4.5.2 Comparison to calibration tests

To calibrate the parameters of the bonds, the laboratory tests are replicated in the simulation environment with the parent-particle being deformed at the same rate until failure occurs; respectively, the impact during the drop test is replicated under the same conditions.

4.5.2.1 Four-point bending test

A comparison of the laboratory test with the simulation results of the FPBT is shown in figure 49, where in this case, the broken bonds are highlighted instead of deleted to better visualize the fracture of the simulated test sample.

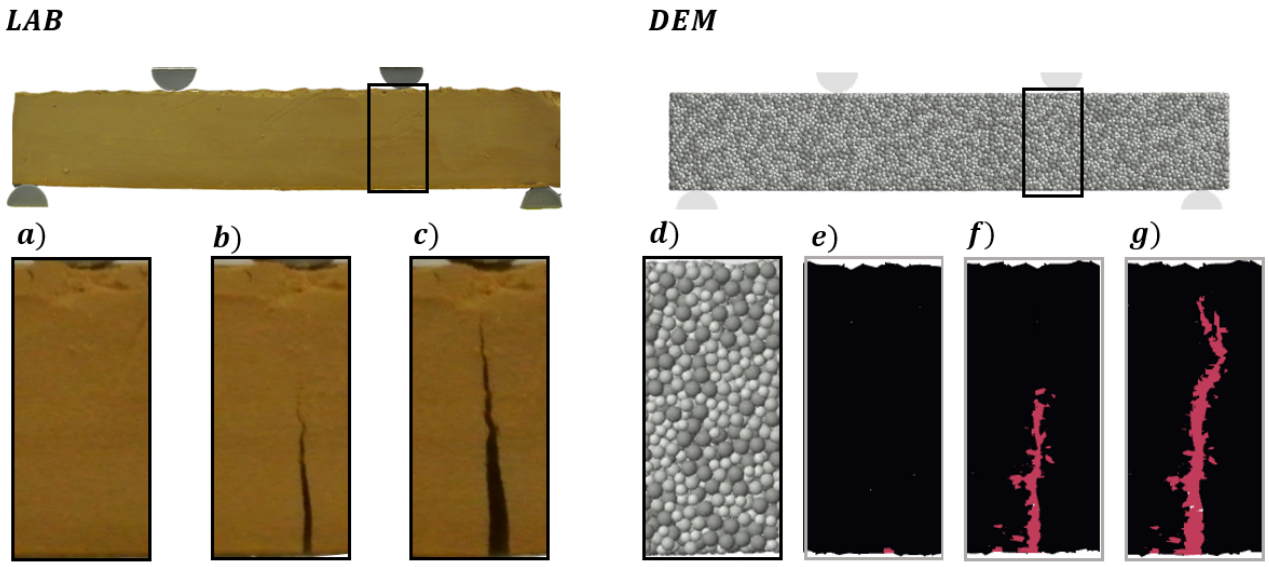

Figure 49: Visual comparison of the material failure in the calibration test (LAB) and the simulation (DEM) under the maximum force F_{max}. Closeup of test sample (a) and crack propagation (b and c), as well as a closeup of the simulated parent-particle with the sub-particles visible at first (d), then depicting a cross-section with only the bonds visualized (e) and finally the crack propagation (f and g) with the broken bonds highlighted.

To compare the results of the laboratory tests with the simulation results, the force-displacement curve of a tested specimen is scaled so that its maxima in terms of force and deformation correspond to the average values of all valid tested specimens. The simulation output, which consists of the deformation path of the loading pins in addition to their reaction forces, is processed in the same way as the data from the laboratory tests, as described in paragraph 4.4.1.2, and then overlaid with the graph of the average force-displacement described above, seen in figure 50.

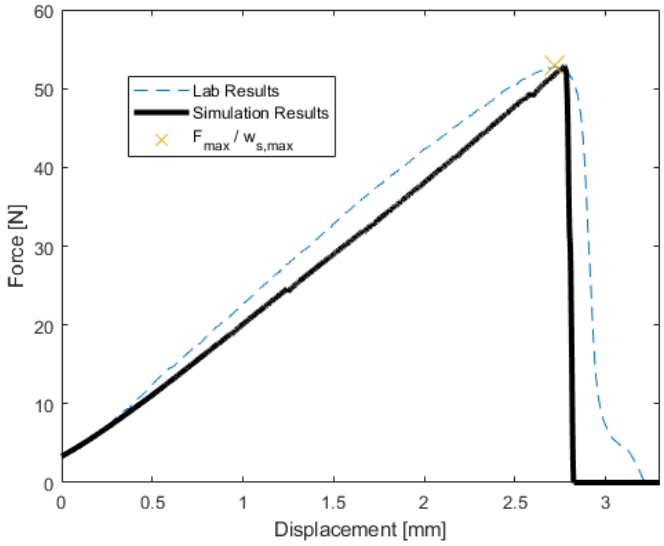

Figure 50: Comparison of the measured data from the laboratory tests and the DEM

The force-displacement curve resulting from the DEM simulation, calibrated with the analytical values from the previous chapters, corresponds clearly to the values measured in the laboratory. It slightly differs mainly by the abrupt drop in the force, which, however, as described in paragraph 2.2.1.1, results from the abruptly occurring notch effect after the failure of the first bonds.

4.5.2.2 Drop Test

The problems associated with the rebound behavior of bonded clusters have already been addressed in paragraph 4.5.1.7. While in the laboratory tests, the rebound height corresponds to only a few centimeters, it can be reduced to only about 50% of the drop height in the simulation, regardless of a further increase of the damping coefficient. Since a further increase of the coefficient does not result in a significant improvement of this problem, it is set to be the value at which the drop height starts to asymptotically approach its minimum and to keep the simulation stable at a larger time step. This issue is to be addressed in future work by means of a different damping approach regarding the the bond behaviour. To check the dynamic behavior of the particle clusters, the simulation results of the drop test onto an inclined surface are visually compared to the actual material behavior, which showed no damage to all samples after one drop, but a slight buildup of surface cracks after being dropped several times. To illustrate the need for the DIF, simulation results without it are shown where the parent-particle fails by impacting on a plane inclined by the minimum angle of 15°, which is shown in figure 51.

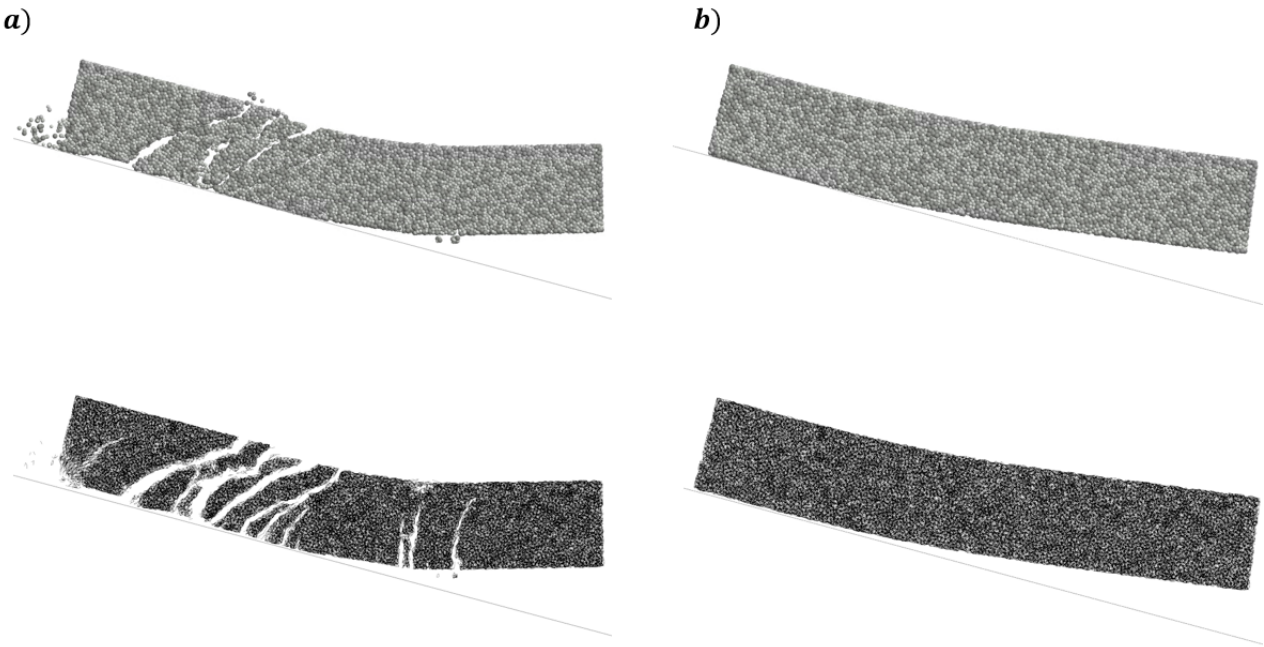

Figure 51: Comparison of the simulated drop test of a horizontally oriented material sample onto a plane inclined by 15°, where the DIF has not yet (a) and already (b) been implemented

This failure of all bonds under tensile load, whose tensile strength was not increased by the DIF, can also be observed if the sample is oriented vertically, as shown in figure 52.

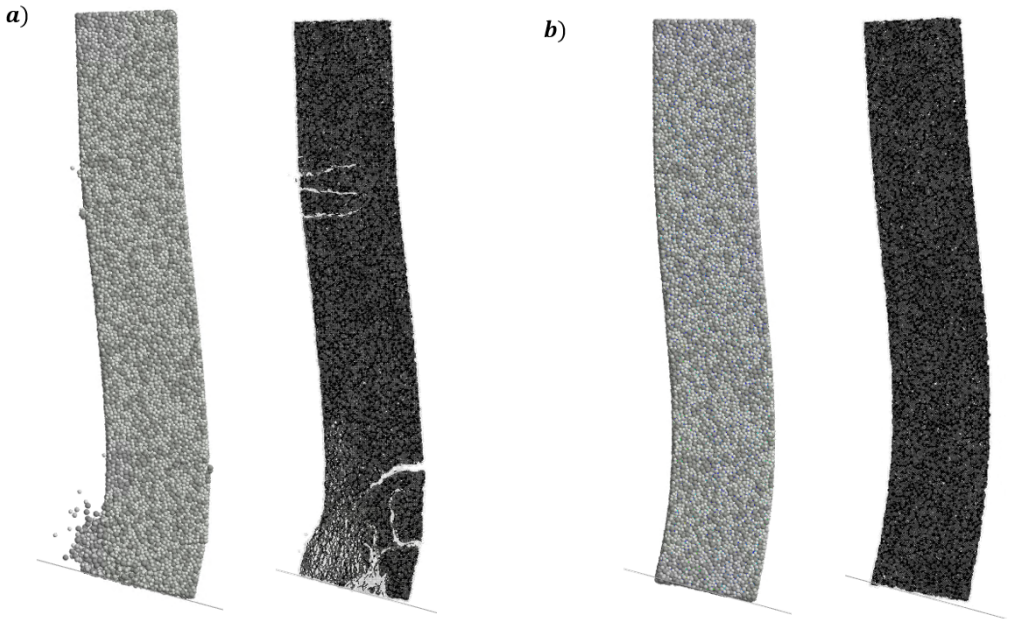

Figure 52: Comparison of the simulated drop test of a vertically oriented material sample onto a plane inclined by 15°, where the DIF has not yet (a) and already (b) been implemented

Since the bonds, whose tensile strength are calculated by means of the DIF, endure the impact without breaking, it is further tested, whether the specimen also, as in the laboratory, remains intact after an impact on a plane inclined by 25°, in which the bond loads are considerably higher. A visual comparison of the laboratory tests with the simulation results is shown in figure 53. Even though a camera with 400 fps was used to capture the images, the specimen still appears a bit blurry due to it moving at 4 $\frac{m}{s}$ or 10 $\frac{cm}{frame}$.

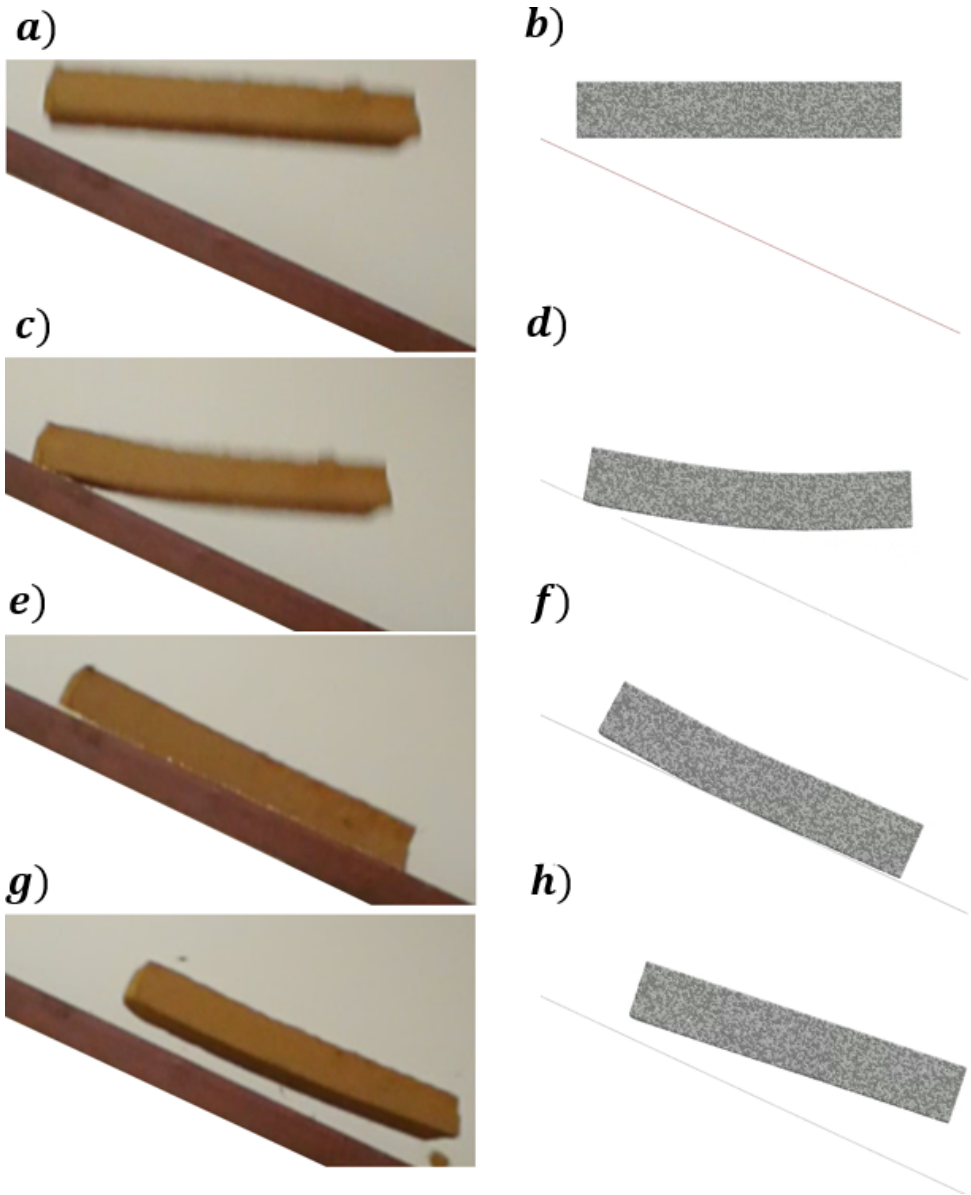

Figure 53: Visual comparison of the material behavior during the drop test of a horizontally oriented specimen onto a plane inclined by 25° in the laboratory (LAB) and the simulation (DEM), shortly before impact (a-b), deformation after first contact (c-d), specimen rolling off on the plane until it is no longer deformed (e-f) and the trajectory after impact (g-h)

In a visual comparison, the only difference that can be detected between the drop test carried out in the laboratory and the simulated one is the slight plastic deformation of the sample. The same dynamic behavior can be observed with a vertically oriented specimen, as shown in figure 54.

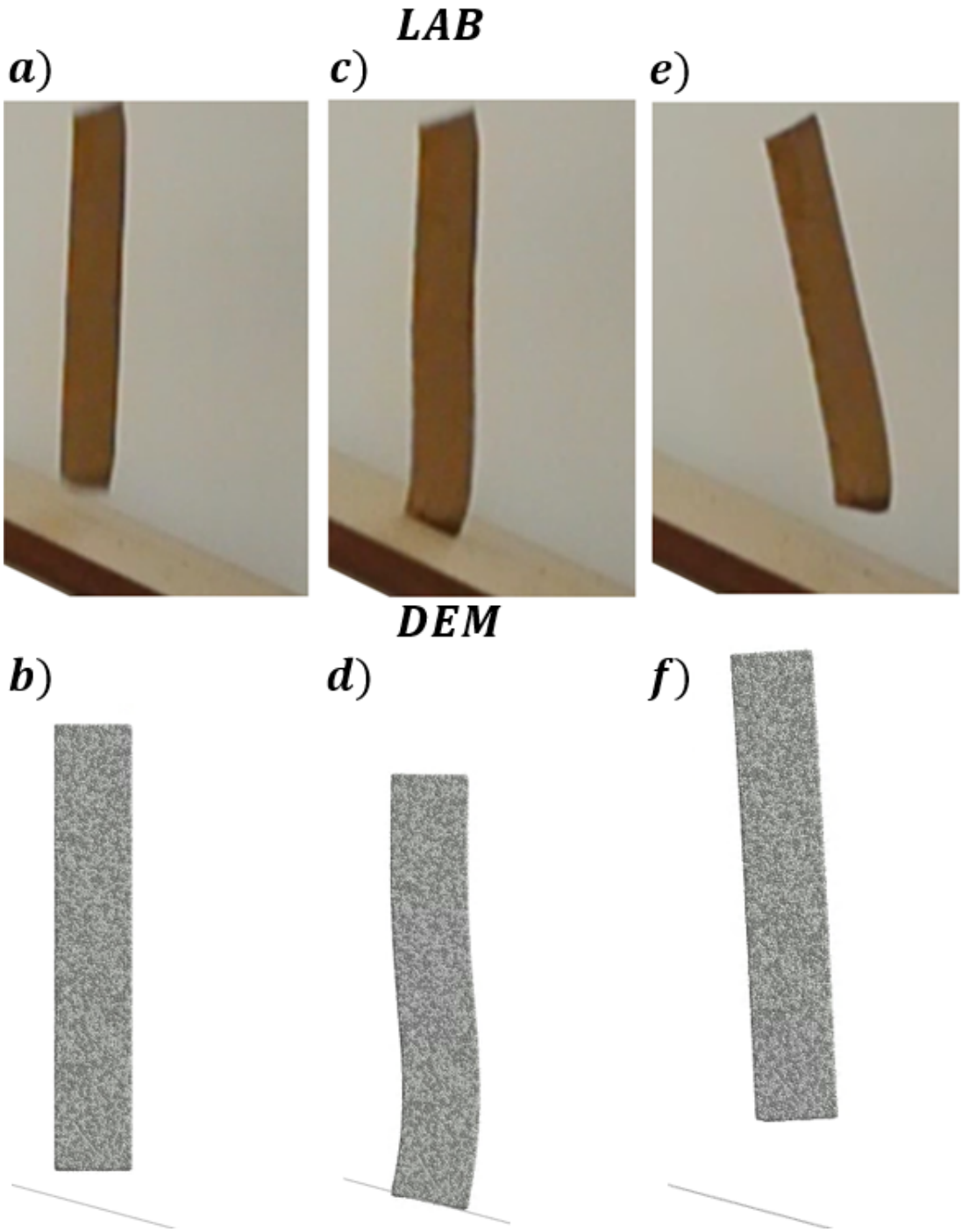

Figure 54: Visual comparison of the material behavior during the drop test of a vertically oriented specimen onto a plane inclined by 25° in the laboratory (LAB) and the simulation (DEM), shortly before impact (a-b), deformation at contact (c-d) and the trajectory after impact (e-f)

During the impact, the DIF, as well as the utilization of the bonds, i.e., the tensile stress σ acting on the bonds in relation to their dynamic tensile strength σ_{ts}, were observed in the simulation, as visualized in figure 55.

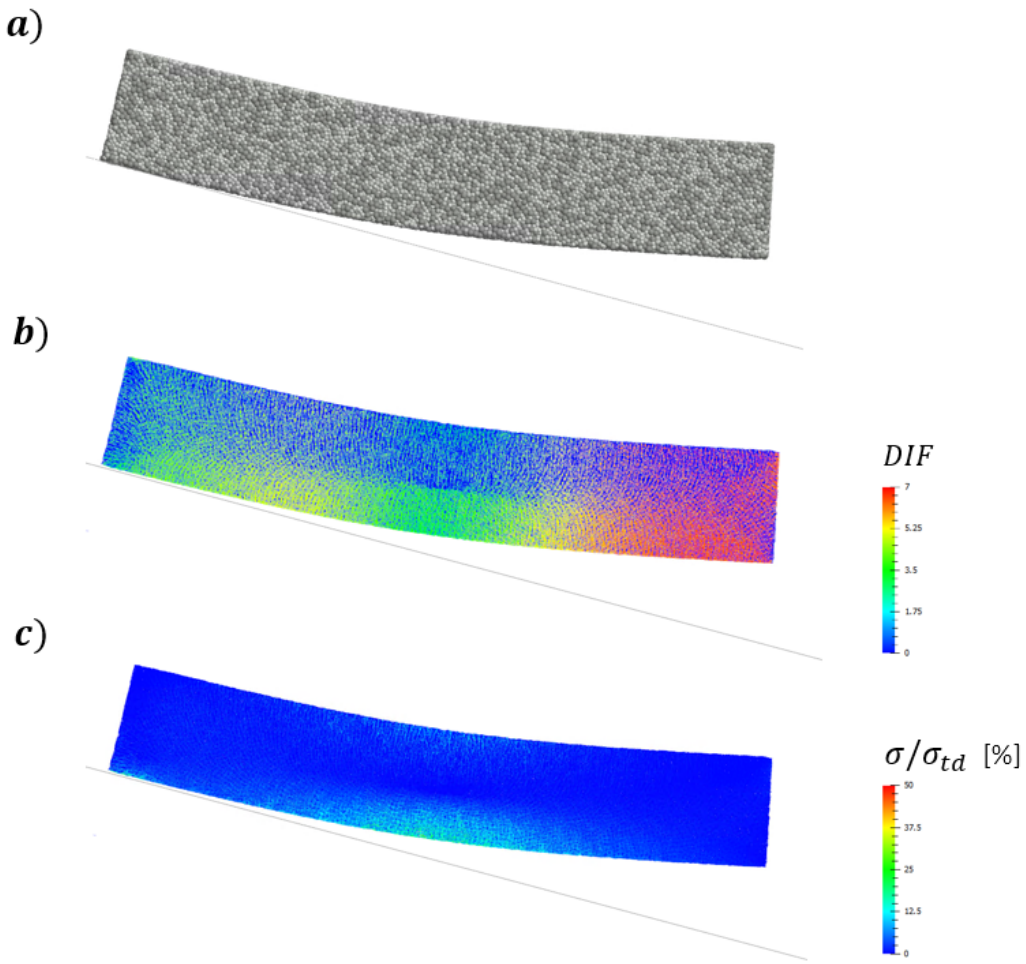

Figure 55: Monitoring of the horizontally oriented parent-particle during impact (a), the calculated DIF of the individual bonds (b) and the utilization of the bonds dynamic tensile strength (c)

The bond utilization, which for most areas maxes out at 50%, except for a local maximum of 71%, shows that breaking the bonds requires a larger amount of energy input, which is especially important for simulations in which particle breakage is desired. Furthermore, by means of the DIF, it can be easily determined in which region the particle currently undergoes the fastest change in deformation. The monitoring of these important bond properties can also be applied to the drop of the specimen in the vertical orientation, as shown in figure 56.

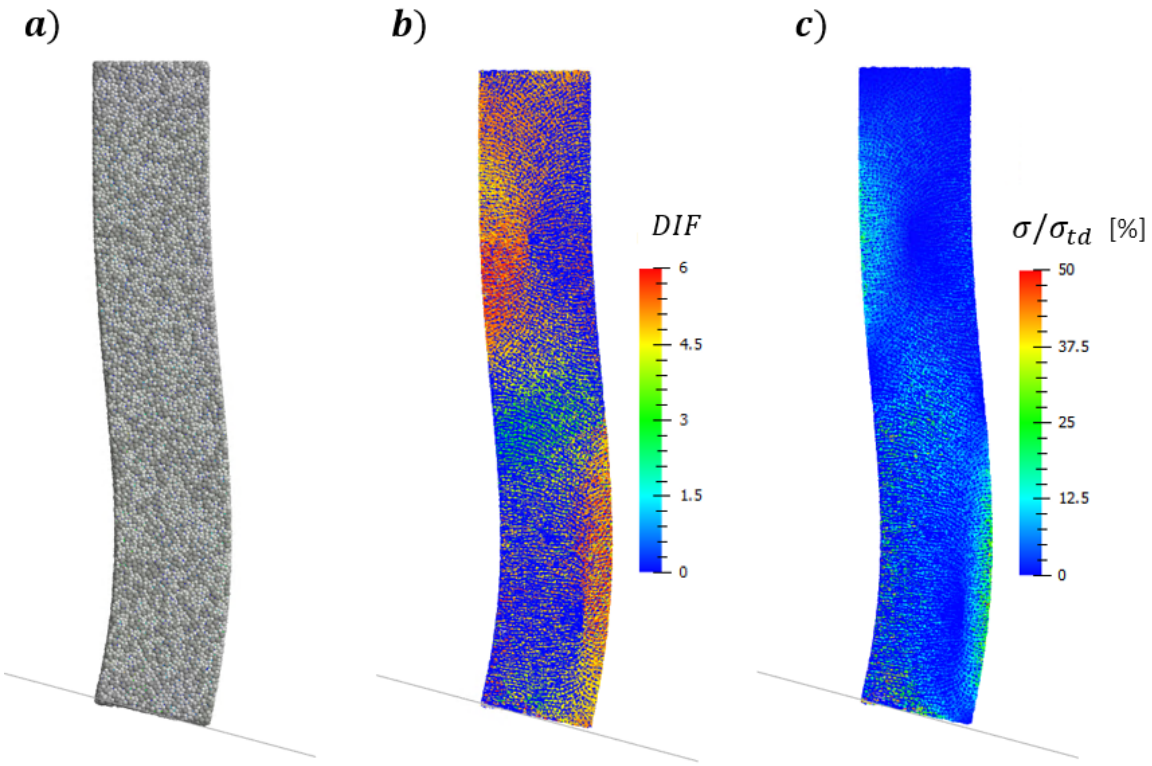

Figure 56: Monitoring of the vertically oriented parent-particle during impact (a), the calculated DIF of the individual bonds (b) and the utilization of the bonds dynamic tensile strength (c)

Small surface cracks occurring after the test's repetition with the same sample cannot be modeled as accumulating damage to individual bonds with the method presented in this thesis since the bond model applied does not take that specific kind of bond damage into account, resulting in the bonds to either fail or remain intact instead of getting gradually damaged; and which is fully sufficient for this thesis. If in future projects, the failure after repeated deformation, e.g., cyclic loading, becomes of interest, this work must be supplemented by a respective (accumulative) bond damage model.

The determined parameters used in the simulations showcased above are listed in table 1.

Property	Describtion	Value	
ρ	Particle density	2050	[kg/m³]
G	Particle Shear Modulus (reduced, as typicall; cf [80])	10	[MPa]
ν	Poisson's ratio	0.3	[-]
d_{min}	Minimum Particle diameter	0.7	[mm]
$d_{max}/\ d_{min}$	Ratio of maximum to minimum particle diameter	1.43	[-]
$desiredOverlap$	Initial sub-particle overlap after generation	0.25	[-]
$bondingRadius$	Responsible for the number of bonded neighbors	1.1	[-]
L/d	Ratio of characteristic length of the parent particle to median particle diameter	35	[-]
Eb	Bond Youngs Modulus	12.05	[MPa]
$\sigma_{v,krit}$	Critical equivalent bond stress	0.21	[MPa]
d	Damping coefficient	0.6	[-]
Δt	Time step	2e-6	[s]

Table 1: Summary of the DEM simulation parameter

4.5.3 Parameter optimization

In the DEM, it often makes sense not to define a simulation model with the actual physical parameters, e.g., as can be seen from table 1, in the selection of the shear Modulus of the particles. Of course, this is only acceptable if the simulation results nevertheless correspond to reality and is only reasonable if it increases the computational efficiency. So, all parameters with a significant influence on the computation time are considered more closely. In this case, the most influential parameters are as follows:

4.5.3.1 Sub-particle size

The smallest radius in the simulation not only influences the Rayleigh time step (see paragraph 4.5.1.1) but also determines how many sub-particles are needed to build up a parent-particle. Since a larger time step reduces the number of computational steps required for the entire simulation time and a smaller number of particles in the simulation significantly affects collision detection, this parameter has the most significant influence on computational efficiency. As mentioned in paragraph 4.5.1.4, however, the number of sub-particles over the smallest distance within a particle cluster should not be chosen too low to reduce the coefficient of variation of the simulation results. Furthermore, it should be noted that the selection of the particle size also determines the smallest possible fracture fragments,

which can have a significant influence on simulations where the intermixing of the fracture fragments with other particles is of great interest.

4.5.3.2 Number of bonded neighborse

By reducing the offset that is used to determine the neighbors with which a sub-particle is bonded at generation, the total number of bonds within the parent-particle is reduced as well. This reduction of bonds within each cluster can have a significant influence on the computational efficiency if there is a large number of bonds to be computed within the bonded particle networks present in the simulation.

4.5.3.3 Young's Modulus and critical stress

Due to a smaller effective cross-section resulting from the larger sub-particle size, as explained in paragraph 4.5.1.2, and the number of bonds being reduced as well, the parent-particle's stiffness decreases, which must be compensated for by increasing the Young's modulus correspondingly. However, since an increase of the Young's Modulus requires a lower time step, see equation 12, it must always be weighed whether the increase of computational speed per time step, resulting from lower numbers of bonds and particles, justifies this reduction. Also, a change of the Young's Modulus automatically results in the necessity to adapt the critical stress as well. To calibrate the bond's Young's Modulus, it is most convenient to initially deactivate the ability of the bonds to break. For a fast calibration, it is recommended to iteratively adapt the analytically determined Young's Modulus in the simulation of the four-point bending test until the average force-displacement curve, as in figure 50, is replicated in the simulation. The lower the stiffness of the bonds, the less force must be applied to deform the parent-particle to achieve the desired state of deformation. If a lower force at the deformation at which the fracture occurs in the laboratory test is measured in the simulation, the stiffness is continuously increased until the exact laboratory values are reproduced, respectively decreased if a higher force is measured. Once the desired force-displacement behavior has been achieved, the maximum bond stress is analyzed at the deformation at which the material failure should occur, the ability of the bonds to break is reactivated, and a final simulation is carried out, which is consequently able to replicate the laboratory test as accurate as possible. The calibration simulation with optimized parameters leads to nearly the same results achieved with the analytical values, as shown in the following figure 57, but with the great advantage that the computational efficiency has been improved by a factor of about 36 (due to aspects described in the previous two sections).

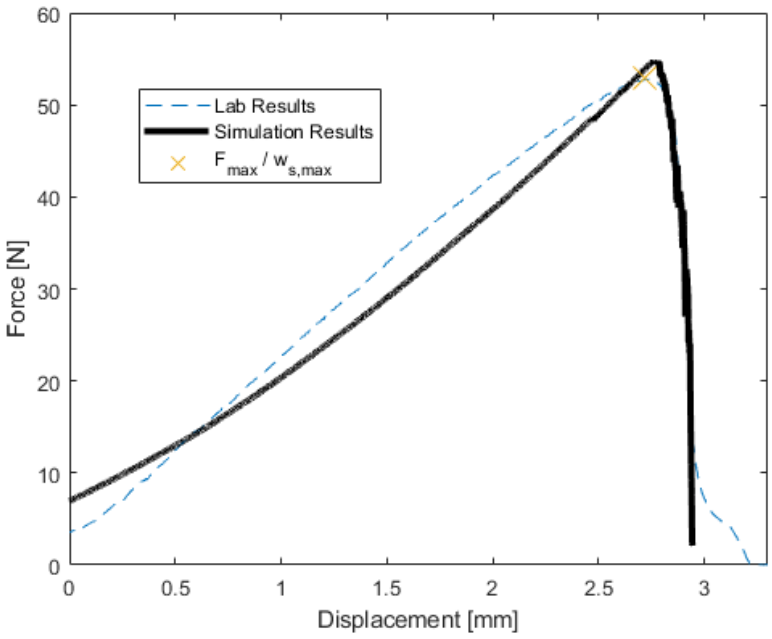

Figure 57: Comparison of the measured data from the laboratory tests and the optimized DEM simulation

The material failure occurs under the same deformation and applied force as in the laboratory test; furthermore, in a very similar area, cf. figure 58 with figure 49.

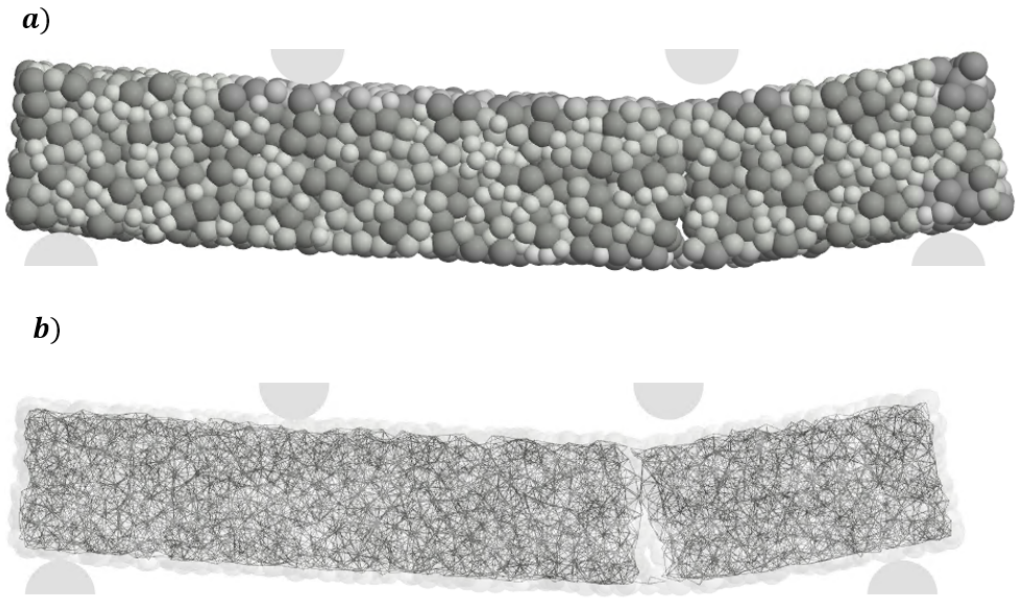

Figure 58: FPBT simulation results with optimized parameters (a), showing the broken bonds that resulted in the specimen's failure (b)

The optimized parameters are listed in table 2.

Property	Describtion	Value	
ρ	Particle density	2050	[kg/m³]
G	Particle Shear Modulus (reduced, as typicall; cf [80])	10	[MPa]
ν	Poisson's ratio	0.3	[-]
d_{min}	Minimum Particle diameter	2.0	[mm]
$d_{max}/\ d_{min}$	Ratio of maximum to minimum particle diameter	1.5	[-]
$desiredOverlap$	Initial sub-particle overlap after generation	0.296	[-]
$bondingRadius$	Responsible for the number of bonded neighbors	1.0	[-]
L/d	Ratio of characteristic length of the parent particle to median particle diameter	10	[-]
Eb	Bond Youngs Modulus	15.0	[MPa]
$\sigma_{v,krit}$	Critical equivalent bond stress	0.185	[MPa]
d	Damping coefficient	0.6	[-]
Δt	Time step	4e-6	[s]

Table 2: Summary of the optimized simulation parameters

4.5.4 Additional optimization

Industrial scale simulations can be optimized even further for computational efficiency without changing the simulation parameters. If particle clusters break exclusively in a certain region of the simulation space, e.g., inside a crusher or at the transfer chute during a conveying process, it makes sense to only simulate them as a bonded-particle network within this area, and as single, (correspondingly complex-shaped), rigid particles otherwise. This method, which in its principle corresponds to the PRM (but without the immediate brakage to be depicted), drastically reduces the number of particles and bonds in the overall simulation, which has a significant impact on the computation time, thus total efficiency.

Replacing the complex-shaped rigid particle with a bonded-particle cluster, which is furthermore able to break apart, is termed bonded-particle replacement method (BPRM), as it can be seen as a kind of a specific extension to the conventional PRM.

4.5.4.1 BPRM implementation

Instead of choosing typical trigger events for the PRM, like reaching a certain breakage energy, the procedure to replace the rigid (mother) particle in the BPRM is as follows. Breakable particles are at first modeled and simulated as single, rigid, complex-shaped particles whose positions are tracked by

means of the Co-simulation plugin. If their center of mass enters a predefined area (in which breakage is expected), their state information (position, velocity, spacial alignment, etc.) is stored, and the rigid particle is deleted. For the next time step, a pre-generated parent-particle (as a bonded-particle setup) gets generated by means of the flexible-inlet plugin matching the physical attributes of the previously deleted rigid particle, thus enabling the particle's deformation and ability to break further on. This procedure is depicted in figure 59. The trigger event can of course also be independent of the particle's position, but can instead depend on the contact force etc.

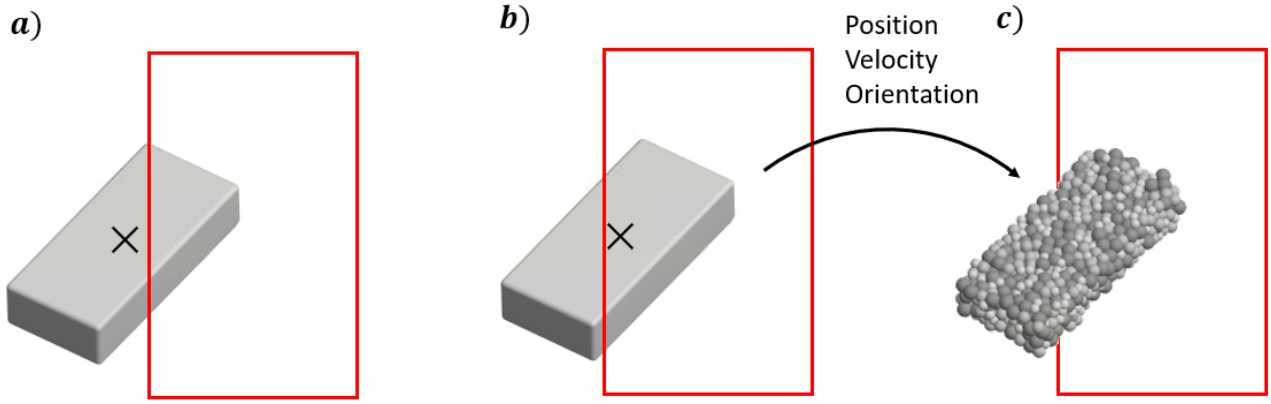

Figure 59: Schematic depiction of a complex-shaped, rigid particle moving into a critical area (a). As soon as its center of mass enters this region marked in red (b), it is replaced by a bonded-particle clusters with identical physical attributes (c)

It should be noted that a large rigid particle can have large absolute overlaps (relative to small particles). To avoid these unnaturally large overlaps of the sub-particles, which would consequently be present when the BPRM replaces the rigid particle with the bonded particle, the rigid particle to-be-replaced is recommended to be dimensioned slightly larger than the parent-particle. (Correspondingly considering mass-related aspects, i.e., the rigid particle's density.)

4.5.4.2 Example

The functionality and computational efficiency of the BPRM, in combination with all the parameter optimizations described in section 4.5.3, is showcased by means of an example. Since the objective of the project in the course of which this thesis was derived was the simulation of a conveying process, a scenario similar to breakable particles impacting in a chute was chosen. In order to not reveal specific design-related contents, the only parts used in this example are a simplified version of a conveyor belt moving the particles at a velocity v of 3 $\frac{m}{s}$ and an impact plate, see figure 60. The complex-shaped

rigid particles are generated with initial velocity (equivalent to the belt speed) at 2 m from the drop point.

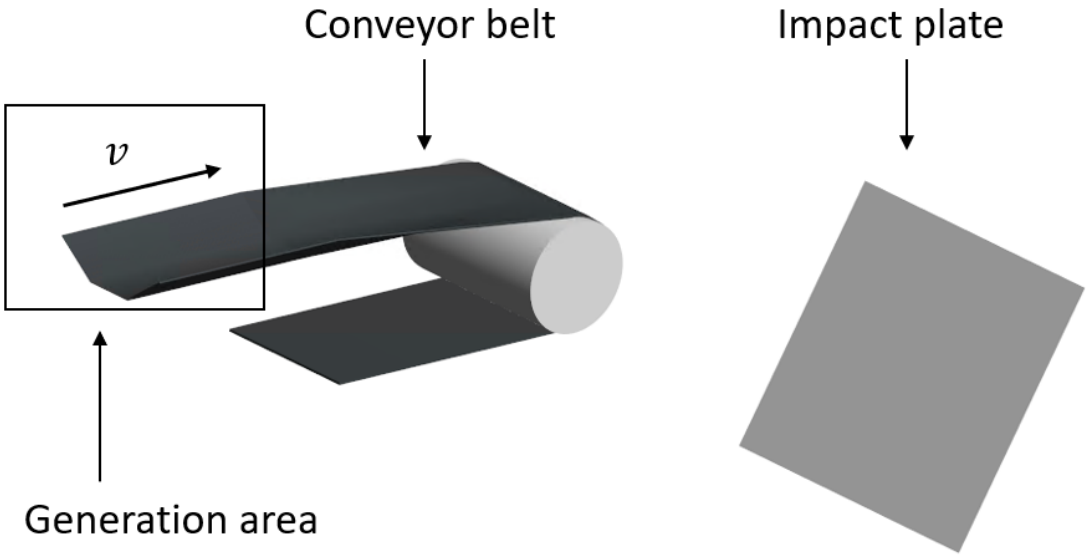

Figure 60: Simulation setup, depicting the system components and the generation area

In this example, the critical area begins shortly before the rigid particles collide with the impact plate, see figure 61, and is marked as a bordered box in the following figures.

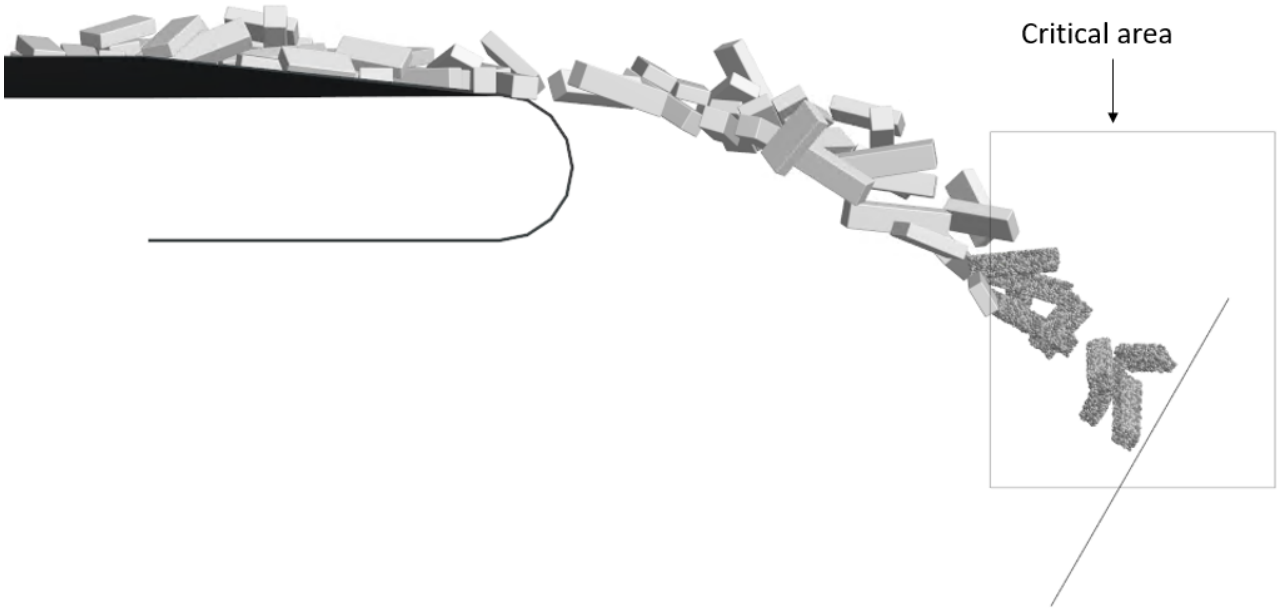

Figure 61: Side view of the simulation showing the replacement of rigid particles with bonded-particle clusters in the critical area

In this example, the loads related to the kinetic energy of the parent-particles at impact, combined with additional loads from collisions with further particles, are sufficient to break the bonded clusters. Furthermore, the functionalities of the approaches/methods/algorithms described in this thesis are demonstrated on this example. Rigid particles are replaced with deformable, breakable particle-clusters when entering a critical area, which, after breaking apart, are identified as separate sub-clusters, allowing the determination of the individual cluster masses, with the relaxation after breakage leading to a stable simulation. All this is illustrated in figure 62.

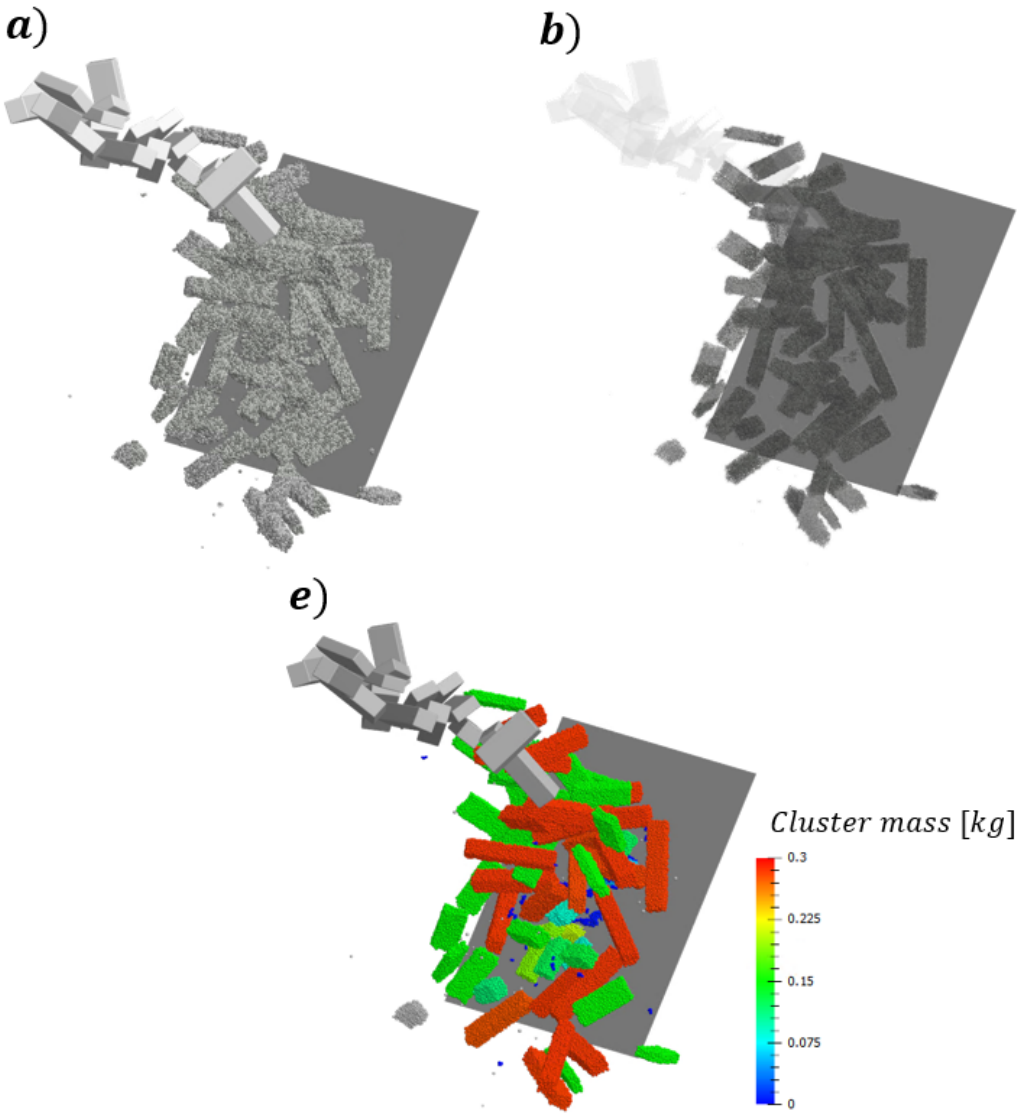

Figure 62: Simulation results of the introduced example (a), depicting the breakage of the particle clusters (b) as well as visualizing the masses of every individual resulting (sub-)cluster, allowing easier identification and evaluation of the fracture fragments (e)

The increase of computational efficiency by means of parameter optimization was already addressed in section 4.5.3, which only leaves the testing of the efficiency increase by means of the BPRM. This was accomplished by simulating the example again under the same conditions but without the BPRM, which is shown in figure 63.

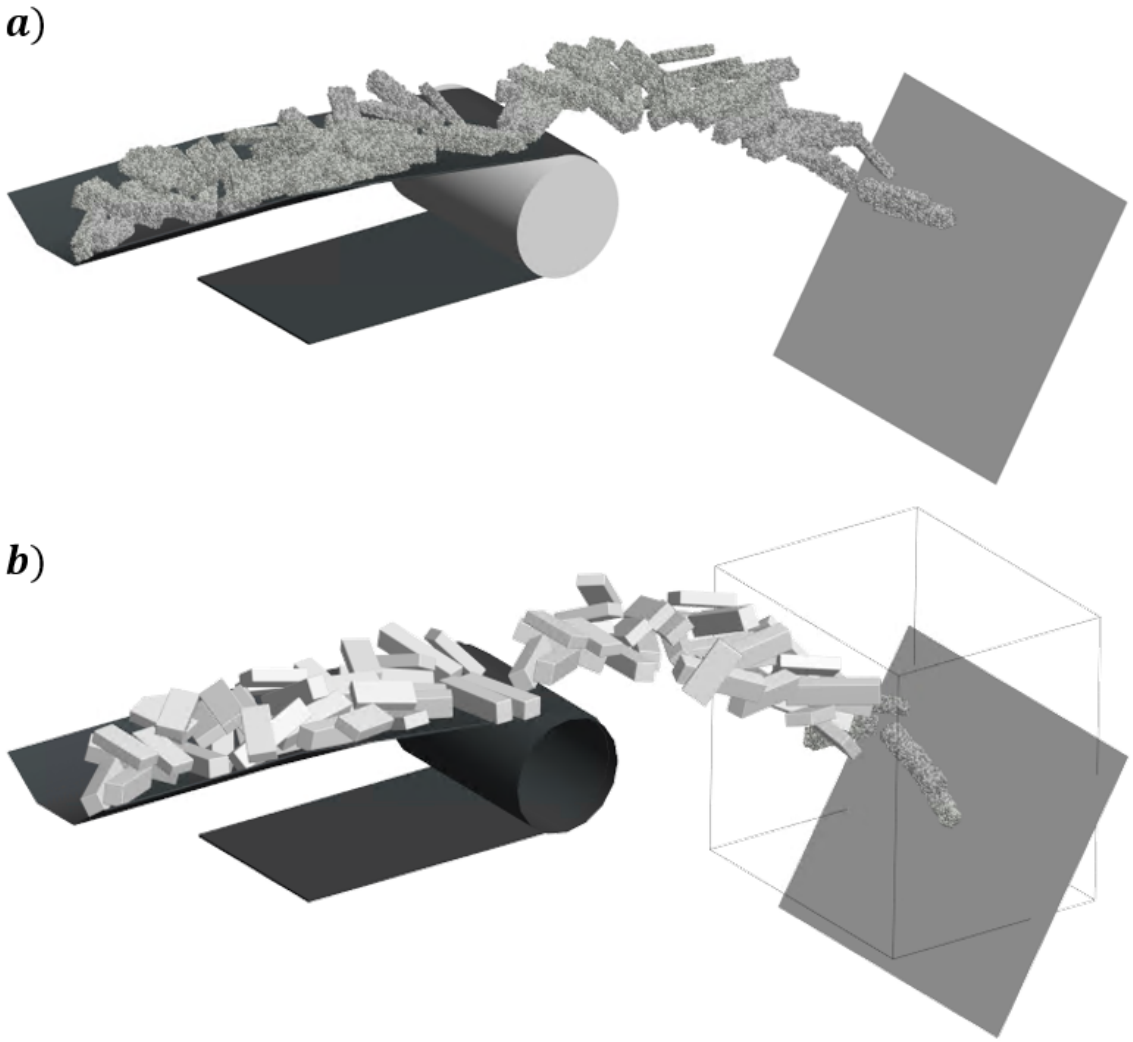

Figure 63: Comparison of the simulated example without (a) and with the implemented BPRM; respectively (b) showing a significant decrease of particles and bonds

By comparing the computational time of both examples, the implementation of the BPRM resulted in a further increase in computational efficiency (computation time required) by a factor of about 3. (Concerning the example given, and which can reach even higher values in cases with more sub-particles required in areas where breakage is actually not expected.)

4.6 Implementation of the concepts

Except for the generation algorithm, which was programmed with MATLAB, all other algorithms are implemented into the simulation software ThreeParticle/CAE via the API (in C++). At each discrete time step of the simulation, the DEM software runs through a computation cycle, seen in figure 64, in which adaption is allowed at different points with various API plugins. With each plugin, different simulation data can be accessed and modified, which is why for each task, the ideal plugin must be carefully considered.

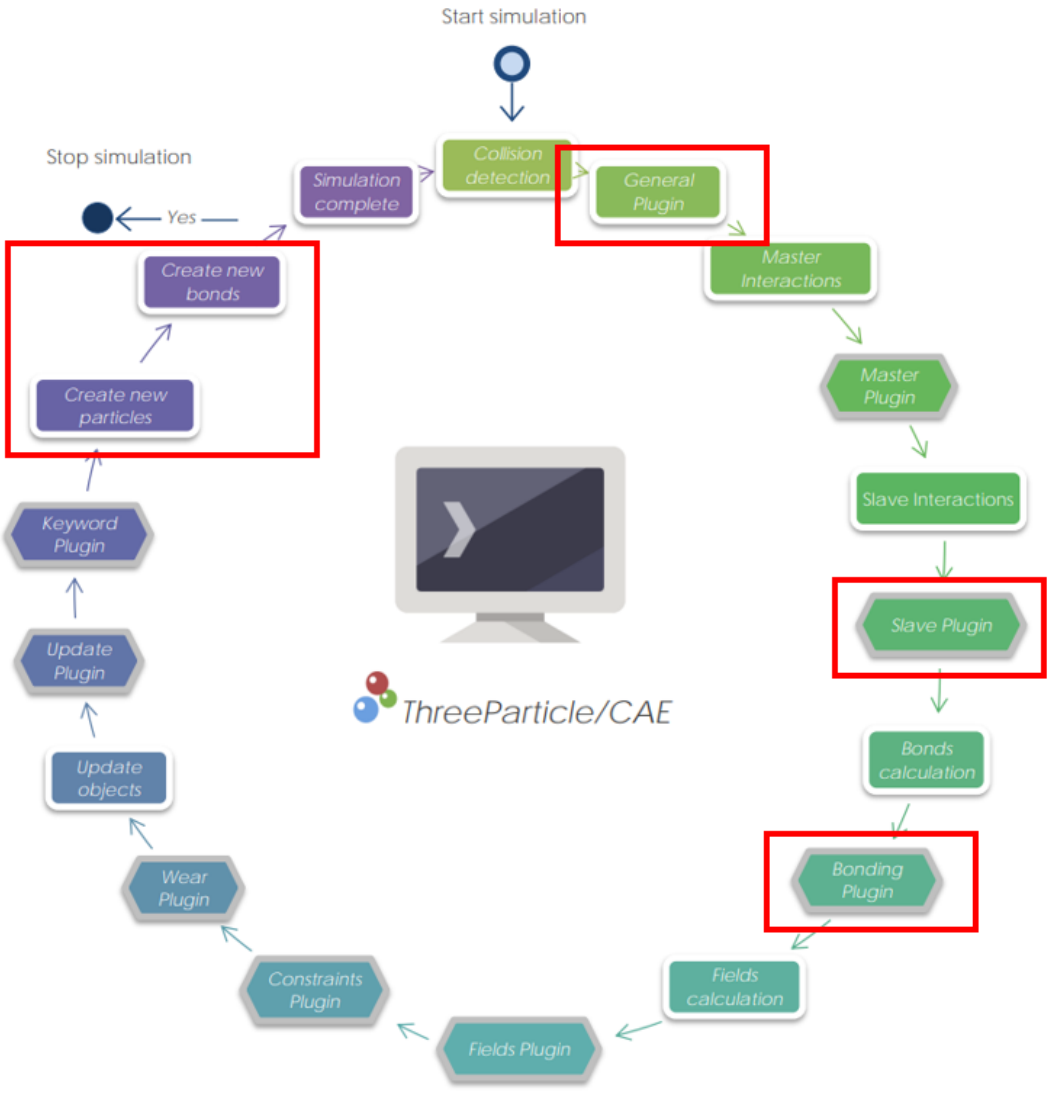

Figure 64: Simulation sequence of ThreeParticle/CAE with the used plugins highlighted; Figure adapted from Fig.1 in [81]

The plugins are coded in the programming language C++ and compiled into a dynamic library (.dll-file), which, loaded as such an extending dll-plugin, can interact with the simulation software correspondingly. Both the slave and the bonding plugin are executed with multiple threads in parallel, which is why attention must always be paid to thread save operations when accessing data available to multiple threads at once, such as the custom properties. To prevent unintended access in this multithread-context, a thread lock is used for reading and writing, ensuring that only one thread accesses a mutable variable, thus avoiding race conditions. The entire data exchange is visualized in figure 65.

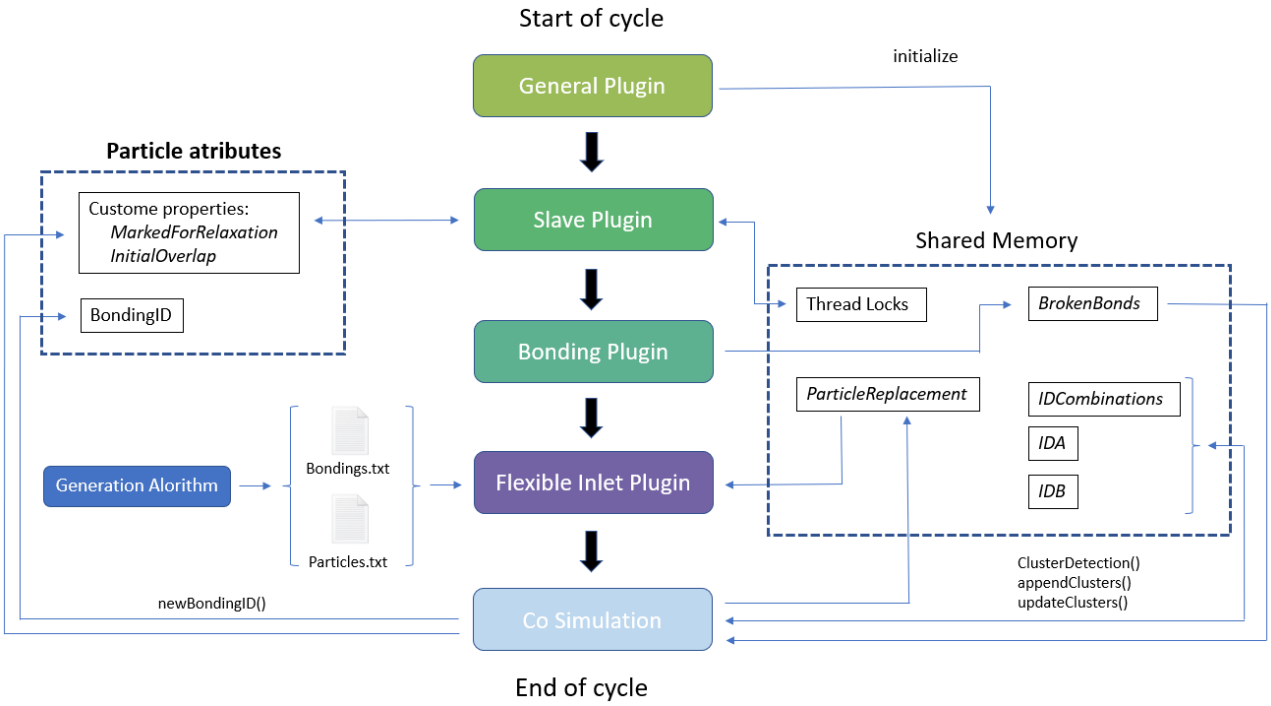

Figure 65: Data management between the API plugins in the simulation environment

The tasks executed by each plugin, as well as the data exchanged are as follows:

General Plugin

Executed by a single thread after the collision detection, even before the master contact model is calculated, this plugin initializes the shared memory used to exchange data between the individual plugins. The shared memory consists of a vector of void pointers, to which sub-vectors are assigned that contain double values, thread locks, strings, and all other required data types. This plugin is

executed only in the first time step after the program was started, and, therefore, an empty shared memory is detected. **If the simulation is paused, it must be ensured that the program is restarted** to reinitialize the shared memory. Otherwise, the data stored in the shared memory is not updated and still originates from the time the simulation was stopped. This can result in incorrect simulation results due to wrong data being accessed or even a crash of the simulation software.

Slave Plugin

With this plugin, the relaxation after breakage is achieved, executing algorithm X to recalculate the force of the master contact model acting between two particles that, until recently, were connected with a bond (i.e., particles that were connected via a bond, now broken but with the particles still contacting each other). Whether two interacting particles are a candidate for relaxation ($MarkedForRelaxation$) and if so, how large their $initialOverlap$ is, is stored in the custom properties of the individual particles.

Bonding Plugin

This plugin is executed directly after the calculation of the bond contact model and fulfills two tasks: At first, if a bond is marked as broken, the IDs of the bond partners are stored in the shared memory since they are required for the cluster detection.

The second task addresses the dynamic behavior of the bond; by monitoring the strain rate under tensile load, the DIF is calculated according to equation 20 if a certain strain rate is exceeded, increasing the temporary critical stress.

Flexible Inlet

As part of "Create new Particles" and "Create new Bonds" in the simulation sequence shown in figure 64, new flexible (and in this application: bonded) particle clusters are created in this part of the cycle. This, as explained in paragraph 4.5.4.1, corresponds to the implementation of the BPRM (thus, as already addressed, further increasing the computational efficiency in industrial scale simulations). However, only one type of parent-particle, in regard to their shape, can be generated per cycle, thus, it is required to use differently shaped clusters to generate multiple bonded particle clusters from rigid particles parallel and in successive timesteps . Information about the generated parent-particle, regarding their shape, position, spatial alignment, velocity, etc., is fed to the plugin via the shared

memory (*ParticleReplacement*). The positions of the sub-particles within the cluster and the bonding partners are imported from external TXT files, which are an output of the generation algorithm (algorithm 1).

Co-Simulation

Implemented in the latest update of the software and therefore not yet included in figure 64, the Co-Simulation plugin opens up a wide range of possibilities, as it can be used to access and modify almost all simulation data. In this plugin, all algorithms of the cluster detection (algorithm 2 and 3) are executed as well as the detection of particles, which are to be replaced by a cluster takes place.

Additionally, after the generation of a parent-particle, this plugin stores the IDs of each sub-particle's bonding partners into its custom properties , an attribute of the particle, thus marking them as *MarkedForRelaxation*.

4.7 Publications

The paper published in the course of this thesis can be found in the appendix.

5 Concluding remarks

5.1 Summary

This thesis presents a method to generate and calibrate complex-shaped, deformable, and furthermore breakable particles using the Discrete Element Method for application in industrial-scaled simulations. For this purpose, the bonded-particle model (BPM) was chosen, which uses elastically deformable bonding elements to connect individual discrete particles, also called sub-particles. Exceeding a critical stress results in the breakage of the respective bonding elements, allowing local failure of any arbitrarily shaped particle depicted as such a bonded-particle network, the so-called parent-particle, into sub-clusters, again consisting of several bonded particles.

To determine characteristic material properties required for the calibration of the simulation model, standardized tests, common for testing rock-like materials, were compared, and selectively adapted/ applied. For the measurement of the quasi-static physical parameters of the material to be calibrated, the four-point bending test was selected, which is carried out on a test rig designed specifically for the type of material belonging to this thesis (filter cake). The data measured during the test procedure was prepared and evaluated to obtain the information needed for the calibration of the bond elements. Furthermore, the dynamic behavior of the material was also analyzed visually by means of drop tests.

The generation of parent particles is carried out by a program coded in MATLAB, whose output file – containing the bonded particle network information of overlapping sub-particles for the correct depiction of the solid density of the intact material and the bulk density of the fracture fragments respectively – can be imported directly into the simulation software.

Calibrations of the simulation parameters were done by replicating the tests, as carried out in the laboratory, in the simulation environment. In this process, the parameters were first derived analytically, which resulted in a correct replication of the force-deformation correlation measured in the laboratory. Subsequently, to increase computational efficiency, these parameters were further optimized, resulting in simulation results that still matched the laboratory test data, in combination with a significant reduction of required computation time.

By implementing a cluster detection algorithm in combination with a relaxation-after-breakage technique, data useful for the evaluation of the simulation results, e.g., cluster masses, were determined, and a stable simulation could be guaranteed despite particles having relatively large overlaps in relation to their radii, which would usually result in unusable simulation results.

At last, a bonded-particle replacement method (BPRM), a combination of the Bonded-Particle Model with the Particle Replacement Method, was implemented. By applying this method, a further increase of the computational efficiency without additional adjustment of the simulation parameters was achieved. Therefore, the BPRM is based on the selective replacement of individual, complex-shaped, rigid particles with parent particles (as bonded-particle models, pre-generated by the aforementioned generation algorithm) in defined areas of the simulation, specifically in which particle breakage is expected to occur.

5.2 Conclusions

The objectives set for this thesis were met as follows:

- Both the depiction of the static and dynamic macroscopic breakage behavior of filter cake material was achieved sufficiently in process-like simulations.
- A computationally efficient, easy-to-use generation algorithm was programmed to generate arbitrarily shaped, bonded-particle structures provided via an STL file.
- Under consideration of the final process to be simulated, calibration tests were selected. The data measured during these laboratory tests are automatically prepared and evaluated to obtain the information needed for the calibration of the bondings using a MATLAB script.
- Analytical calculation of the simulation parameters was demonstrated ultimately allowing an efficient calibration process
- By optimizing the parameters with the most significant influence on the computation time, and by also introducing the BPRM, the combination of the advantages of the Bonded-Particle Model with the Particle Replacement Method, efficient simulation at industrial scale is enabled, as demonstrated on an exemplifying showcase.

5.3 Outlook

This thesis presents a viable modeling approach for complex-shaped, deformable, and furthermore breakable particles. To obtain even more accurate simulation results, the following points can be considered:

Calibration tests:
The BPM model can be calibrated even more precisely by determining further static material parameters using additional established test methods, such as the determination of compressive strength by means of a uniaxial or triaxial compression test or the shear strength by means of a direct shear test. Furthermore, by performing dynamic tests, such as the Split Hopkins Bar (addressed in paragraph 3.3.2.2), dynamic properties of the material can be measured instead of approximating them with empirical values as in this work.

Contact model:
By using advanced bond models that take, e.g., bond damage into account, would extended the scope of usage of this presented modelling technique even more, especially regarding crack propagation or failure under cyclic loads – although at the cost of increasing the calibration and also the computational effort consequently. Additionally, the implementation of numerical damping is recommended.

Validation:
To demonstrate the applicability of the developed model not only in laboratory tests, but also in large scale processes, it is recommended to perform validation tests by replicating such processes, e.g., a specific transfer chute, and subsequently comparing the results to measured data.

Further applications

Enabling simulation of filter cake behaviour with methods introduced in this thesis is a major outcome, thus applicable for successful simulation in future applications.

Applying the methods for simulating complex-shaped, breakable particles in large-scale simulations presented in this thesis is also possible in various other industrial fields. In general, other types of processes in which materials with similar physical properties are handled, such as tablets in the pharmaceutical industry, are potential scenarios for applying the developments or specific adaptations of it as presented. Individual parts of this work can also be applied independently in similar simulation scenarios. Such as relating to the bonded-particle replacement method introduced in this thesis, which combines the advantages of the bonded-particle model and the particle replacement method; e.g for simulating deformable bodies, like stalks during a harvesting process, where only particles in the immediate vicinity of the blade are represented by a bonded-particle network. By drastically reducing the number of particles in the simulation with this approach, a significant increase in computational efficiency can be achieved.

5.4 Closing words

The content of this thesis, which aims to reduce the modeling effort of highly complex tasks resulting in significantly increased efficiency, allows a step towards sustainability through the associated saving of required resources; on the one hand, using virtual instead of expensive physical prototypes, and on the other hand, furthermore reducing required efforts in context with these virtual prototypes and their successful simulation. Hereby, this thesis and the presented developments form a contribution to the future of numerical modeling in particle technology, which I look forward to with great anticipation.

Directories

Abbreviations

1/2/3D	one/two/three-dimensional
API	Application Programming Interface
BPM	bonded-particle model
CAD	computer-aided design
CAE	computer-aided engineering
cf.	compare (to) (confer)
CPU	central processing unit
DEM	Discrete Element Method (also Distinct Element Method)
DIF	Dynamic Increase Factor
e.g.	for example (exempli gratia)
Eq.	Equation(s)
FDEM	Finite-Discrete Element Method
FEA	Finite Element Analysis
FEM	Finite Element Method
Fig.	Figure(s)
GPU	Graphics processing unit
GUI	Graphical User Interface
HM	Hertz-Mindlin
i.e.	which means/in other words (id est)
INP	ThreeParticle/CAE input (file format)
ISRM	International Society for Rock Mechanics and Rock Engineering
MD	Molecular Dynamics
PRM	Particle Replacement Model
PSD	Particle Size Distribution
RAM	Random Access Memory
ref.	referring (to)
STL	Standard Triangle/Tessellation Language (file format)
XDEM	a specific extended form of the DEM

Appendix

Publications

Title

Simulation of Complex-Shaped Particle Breakage Using the Discrete Element Method

Paper published in the *International Journal of Chemical and Molecular Engineering, Vol:16, No:9, 2022*
Page 79–85

issued by
World Academy of Science, Engineering and Technology

Online ISSN : 1307-6892

List of figures

List of tables

List of algorithm

References

[1] Michael Denzel and Michael Prenner. “Minimierung des Sinterzerfalls mittels DEM”. In: *BHM Berg- und Hüttenmännische Monatshefte* 166.2 (2021), pp. 76–81. ISSN: 0005-8912. DOI: 10.1007/s00501-021-01081-7.

[2] Eugenio Oñate et al. “A local constitutive model for the discrete element method. Application to geomaterials and concrete”. In: *Computational Particle Mechanics* 2.2 (2015), pp. 139–160. ISSN: 2196-4378. DOI: 10.1007/s40571-015-0044-9.

[3] Peter Domone and Marios Soutsos, eds. *Construction materials: Their nature and behaviour*. 5. ed. Boca Raton: CRC Press, Taylor & Francis Group, 2018. ISBN: 1315164590. URL: https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1769439.

[4] Zong-Xian Zhang. *Rock Mechanics Related to Mining Engineering*. Helsinki, Finland, October 11–12, 2017.

[5] Peixian Li, Lili Yan, and Dehua Yao. “Study of Tunnel Damage Caused by Underground Mining Deformation: Calculation, Analysis, and Reinforcement”. In: *Advances in Civil Engineering* 2019 (2019), pp. 1–18. ISSN: 1687-8086. DOI: 10.1155/2019/4865161.

[6] Johannes Quist and Carl Magnus Evertsson. “Cone crusher modelling and simulation using DEM”. In: *Minerals Engineering* 85 (2016), pp. 92–105. ISSN: 08926875. DOI: 10.1016/j.mineng.2015.11.004.

[7] R. A. Bearman, C. A. Briggs, and T. Kojovic. “The applications of rock mechanics parameters to the prediction of comminution behaviour”. In: *Minerals Engineering* 10.3 (1997), pp. 255–264. ISSN: 08926875. DOI: 10.1016/S0892-6875(97)00002-2.

[8] Martin Obermayr et al. “A bonded-particle model for cemented sand”. In: *Computers and Geotechnics* 49 (2013), pp. 299–313. ISSN: 0266352X. DOI: 10.1016/j.compgeo.2012.09.001.

[9] Petre Miu. *Combine Harvesters: Theory, modeling, and design*. Boca Raton: CRC Press, 2015. ISBN: 9780429152931. DOI: 10.1201/b18852. URL: https://www.taylorfrancis.com/books/9781482282375.

[10] Qirui Wang, Hanping Mao, and Qinglin Li. “Modelling and simulation of the grain threshing process based on the discrete element method”. In: *Computers and Electronics in Agriculture* 178 (2020), p. 105790. ISSN: 01681699. DOI: 10.1016/j.compag.2020.105790.

[11] Todd Wisdom, Mike Jacobs, and James Chaponnel. “GeoWasteTM – continuous comingled tailings for large-scale mines”. In: *Proceedings of the 21st International Seminar on Paste and Thickened Tailings*. Proceedings of the International Seminar on Paste and Thickened Tailings. Australian Centre for Geomechanics, Perth, 2018, pp. 465–472. DOI: 10.36487/ACG{\textunderscore}rep/1805{\textunderscore}38{\textunderscore}Wisdom.

[12] Fimbinger E. Platzer F. “Simulation of complex-shaped particle breakage using the Discrete Element Method”. In: *World Academy of Science, Engineering and Technology, International Journal of Chemical and Molecular Engineering* 2022.Vol:16, No:9 (), pp. 79–85.

[13] P. A. Cundall and O. D. L. Strack. “Discussion: A discrete numerical model for granular assemblies”. In: *Géotechnique* 30.3 (1980), pp. 331–336. ISSN: 0016-8505. DOI: 10.1680/geot.1980.30.3.331.

[14] John A. Hudson, ed. *Comprehensive rock engineering: Principles, practice & projects*. 1. ed. Oxford [u.a.]: Pergamon Press, 1993. ISBN: 9780080406152.

[15] Nicholas J. Brown, Jian-Fei Chen, and Jin Y. Ooi. “A bond model for DEM simulation of cementitious materials and deformable structures”. In: *Granular Matter* 16.3 (2014), pp. 299–311. ISSN: 1434-5021. DOI: 10.1007/s10035-014-0494-4.

[16] M. Cervera and M. Chiumenti. “Smeared crack approach: back to the original track”. In: *International Journal for Numerical and Analytical Methods in Geomechanics* 30.12 (2006), pp. 1173–1199. ISSN: 0363-9061. DOI: 10.1002/nag.518.

[17] Simon-Nicolas Roth, Pierre Léger, and Azzeddine Soulaïmani. “A combined XFEM–damage mechanics approach for concrete crack propagation”. In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 923–955. ISSN: 00457825. DOI: 10.1016/j.cma.2014.10.043.

[18] Lanhao Zhao et al. “A fracture model for the deformable spheropolygon-based discrete element method”. In: *Granular Matter* 24.2 (2022). ISSN: 1434-5021. DOI: 10.1007/s10035-022-01206-w.

[19] Ante Munjiza. *The combined finite-discrete element method*. Reprint. Hoboken, NJ: Wiley, 2010. ISBN: 0470841990. URL: http://www.loc.gov/catdir/description/wiley041/2003025485.html.

[20] Peter Wriggers and B. Avci. “Discrete Element Methods: Basics and Applications in Engineering”. In: *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*. Ed. by Riva, Laura de Lorenzis, and Alexander Düster. Vol. 599. CISM International Centre for Mechanical Sciences. Cham: Springer International Publishing, 2020, pp. 1–30. ISBN: 978-3-030-37517-1. DOI: 10.1007/978-3-030-37518-8_1.

[21] Heinrich Hertz. “Ueber die Berührung fester elastischer Körper”. In: *crll* 1882.92 (1882), pp. 156–171. ISSN: 0075-4102. DOI: 10.1515/crll.1882.92.156.

[22] R. D. Mindlin. “Compliance of Elastic Bodies in Contact”. In: *Journal of Applied Mechanics* 16.3 (1949), pp. 259–268. ISSN: 0021-8936. DOI: 10.1115/1.4009973.

[23] R. D. Mindlin and H. Deresiewicz. “Elastic Spheres in Contact Under Varying Oblique Forces”. In: *Journal of Applied Mechanics* 20.3 (1953), pp. 327–344. ISSN: 0021-8936. DOI: 10.1115/1.4010702.

[24] BECKER 3D GMBH. *ThreeParticle/CAE [software]. Version R6.0.5*. 2022. URL: http://becker3d.com/.

[25] Narcés Jiménez-Herrera, Gabriel K.P. Barrios, and Luís Marcelo Tavares. “Comparison of breakage models in DEM in simulating impact on particle beds”. In: *Advanced Powder Technology* 29.3 (2018), pp. 692–706. ISSN: 09218831. DOI: 10.1016/j.apt.2017.12.006.

[26] Wei Zhou et al. “Discrete element modeling of particle breakage considering different fragment replacement modes”. In: *Powder Technology* 360 (2020), pp. 312–323. ISSN: 00325910. DOI: 10.1016/j.powtec.2019.10.002.

[27] D. Cantor et al. “Three-dimensional bonded-cell model for grain fragmentation”. In: *Computational Particle Mechanics* 4.4 (2017), pp. 441–450. ISSN: 2196-4378. DOI: 10.1007/s40571-016-0129-0.

[28] Damien André et al. “Discrete element method to simulate continuous material by using the cohesive beam model”. In: *Computer Methods in Applied Mechanics and Engineering* 213-216 (2012), pp. 113–125. ISSN: 00457825. DOI: 10.1016/j.cma.2011.12.002.

[29] D. O. Potyondy and P. A. Cundall. “A bonded-particle model for rock”. In: *International Journal of Rock Mechanics and Mining Sciences* 41.8 (2004), pp. 1329–1364. ISSN: 13651609. DOI: 10.1016/j.ijrmms.2004.09.011.

[30] S. P. Timoshenko. “LXVI. On the correction for shear of the differential equation for transverse vibrations of prismatic bars”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 41.245 (1921), pp. 744–746. ISSN: 1941-5982. DOI: 10.1080/14786442108636264.

[31] S. P. Timoshenko. “X. On the transverse vibrations of bars of uniform cross-section”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 43.253 (1922), pp. 125–131. ISSN: 1941-5982. DOI: 10.1080/14786442208633855.

[32] R. v. Mises. “Mechanik der festen Körper im plastisch- deformablen Zustand”. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1913 (1913), pp. 582–592. URL: http://eudml.org/doc/58894.

[33] Qing Lin et al. “Imaging opening-mode fracture in sandstone under three-point bending: A direct identification of the fracture process zone and traction-free crack based on cohesive zone model”. In: *International Journal of Rock Mechanics and Mining Sciences* 136 (2020), p. 104516. ISSN: 13651609. DOI: 10.1016/j.ijrmms.2020.104516.

[34] Hu Chen et al. “A Particle-Based Cohesive Crack Model for Brittle Fracture Problems”. In: *Materials (Basel, Switzerland)* 13.16 (2020). ISSN: 1996-1944. DOI: 10.3390/ma13163573.

[35] Chunyang Cui et al. “Discrete-Element Modeling of Cemented Granular Material Using Mixed-Mode Cohesive Zone Model”. In: *Journal of Materials in Civil Engineering* 32.4 (2020). ISSN: 0899-1561. DOI: 10.1061/(ASCE)MT.1943-5533.0003069.

[36] D. O. Potyondy. “A Flat-Jointed Bonded-Particle Material For Hard Rock”. In: OnePetro, 2012.

[37] ITASCA CONSULTING GROUP, INC. *PFC (Particle Flow Code) Version 6.0, Documentation*. URL: http://docs.itascacg.com/pfc600/common/contactmodel/flatjoint/doc/manual/cmflatjoint.html#potyondy2013.

[38] Vinh D. X. Nguyen et al. “Discrete element method using cohesive plastic beam for modeling elasto-plastic deformation of ductile materials”. In: *Computational Particle Mechanics* 8.3 (2021), pp. 437–457. ISSN: 2196-4378. DOI: 10.1007/s40571-020-00343-4.

[39] C. J. Coetzee. “Review: Calibration of the discrete element method”. In: *Powder Technology* 310 (2017), pp. 104–142. ISSN: 00325910. DOI: 10.1016/j.powtec.2017.01.015.

[40] Wei Zhou et al. “Macro–micro responses of crushable granular materials in simulated true triaxial tests”. In: *Granular Matter* 17.4 (2015), pp. 497–509. ISSN: 1434-5021. DOI: 10.1007/s10035-015-0571-3.

[41] Catherine O’Sullivan. *Particulate Discrete Element Modelling: A Geomechanics Perspective*. First edition. Vol. volume 4. Applied geotechnics. Boca Raton, FL: CRC Press, 2014. ISBN: 0203880986. URL: http://site.ebrary.com/lib/alltitles/Doc?id=10477452.

[42] Johannes Grotendorst, Dominik Marx, and Alejandro Muramatsu, eds. *Quantum simulations of complex many-body systems: from theory to algorithms: Winterschool, 25 February - 1 March 2002, Rolduc Conference Centre, Kerkrade, The Netherlands : poster presentations*. Vol. vol. 11. NIC series. Jülich: NIC Directors, 2002. ISBN: 3000090584. URL: http://hdl.handle.net/2128/8597.

[43] H. A. Carmona et al. “Fragmentation processes in impact of spheres”. In: *Physical Review E* 77.5 Pt 1 (2008), p. 051302. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.77.051302.

[44] H. A. Janssen. “Getreidedruck in Silozellen”. In: *Z. Ver. Dt. Ing. 39* 1895 (), pp. 1045–1049.

[45] Y. Ma et al. “Packing Irregular Objects in 3D Space via Hybrid Optimization”. In: *Computer Graphics Forum* 37.5 (2018), pp. 49–59. ISSN: 01677055. DOI: 10.1111/cgf.13490.

[46] Elias Lozano et al. “An efficient algorithm to generate random sphere packs in arbitrary domains”. In: *Computers & Mathematics with Applications* 71.8 (2016), pp. 1586–1601. ISSN: 08981221. DOI: 10.1016/j.camwa.2016.02.032.

[47] Katalin Bagi. “An algorithm to generate random dense arrangements for discrete element simulations of granular assemblies”. In: *Granular Matter* 7.1 (2005), pp. 31–43. ISSN: 1434-5021. DOI: 10.1007/s10035-004-0187-5.

[48] Jinjin Ge and Ying Xu. “A Method for Making Transparent Hard Rock-Like Material and Its Application”. In: *Advances in Materials Science and Engineering* 2019 (2019), pp. 1–14. ISSN: 1687-8434. DOI: 10.1155/2019/1274171.

[49] Q. B. Zhang and J. Zhao. “A Review of Dynamic Experimental Techniques and Mechanical Behaviour of Rock Materials”. In: *Rock Mechanics and Rock Engineering* 47.4 (2014), pp. 1411–1478. ISSN: 0723-2632. DOI: 10.1007/s00603-013-0463-y.

[50] DIN Deutsches Institut für Normung e.V. *Prüfverfahren für Naturstein - Bestimmung der Biegefestigkeit unter Mittellinienlast*. URL: https://www.beuth.de/de/norm/din-en-12372/345323652.

[51] A. Coviello, R. Lagioia, and R. Nova. “On the Measurement of the Tensile Strength of Soft Rocks”. In: *Rock Mechanics and Rock Engineering* 38.4 (2005), pp. 251–273. ISSN: 0723-2632. DOI: 10.1007/s00603-005-0054-7.

[52] DIN Deutsches Institut für Normung e.V. *Prüfverfahren für Naturstein - Bestimmung der Biegefestigkeit unter Drittellinienlast*. URL: https://www.beuth.de/de/norm/din-en-13161/108348551.

[53] American Society for Testing and Materials. *Standard Test Method for Flexural Strength of Dimension Stone*. URL: https://www.astm.org/c0880_c0880m-18e01.html.

[54] American Society for Testing and Materials. *Standard Test Method for Splitting Tensile Strength of Cylindrical Concrete Specimens*. URL: https://www.astm.org/c0496_c0496m-17.html.

[55] American Society for Testing and Materials. *Standard Test Method for Splitting Tensile Strength of Intact Rock Core Specimens*. URL: https://www.astm.org/d3967-16.html.

[56] ISRM. “Suggested methods for determining tensile strength of rock materials”. In: *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* 15.3 (1978), pp. 99–103. ISSN: 01489062. DOI: 10.1016/0148-9062(78)90003-7.

[57] C. Fairhurst. “On the validity of the ‘Brazilian’ test for brittle materials”. In: *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* 1.4 (1964), pp. 535–546. ISSN: 01489062. DOI: 10.1016/0148-9062(64)90060-9.

[58] American Society for Testing and Materials. *Standard Test Method for Direct Tensile Strength of Intact Rock Core Specimens*. URL: https://www.astm.org/d2936-08.html.

[59] American Society for Testing and Materials. *Standard Test Method for Direct Tensile Strength of Intact Rock Core Specimens*. URL: https://www.astm.org/d2936-20.html.

[60] E. Hoeck. “Fracture of anisotropic rock”. In: *Journal of the South African Institute of Mining and Metallurgy* 40 (1964), pp. 501–518. URL: https://journals.co.za/doi/pdf/10.10520/aja0038223x_3802.

[61] Matthew A. Perras and Mark S. Diederichs. “A Review of the Tensile Strength of Rock: Concepts and Testing”. In: *Geotechnical and Geological Engineering* 32.2 (2014), pp. 525–546. ISSN: 0960-3182. DOI: 10.1007/s10706-014-9732-0.

[62] American Society for Testing and Materials. *Standard Test Methods for Compressive Strength and Elastic Moduli of Intact Rock Core Specimens under Varying States of Stress and Temperatures*.

[63] American Society for Testing and Materials. *Standard Test Method for Performing Laboratory Direct Shear Strength Tests of Rock Specimens Under Constant Normal Force*. URL: https://www.astm.org/d5607-16.html.

[64] B. Hopkinson. "X. A method of measuring the pressure produced in the detonation of high, explosives or by the impact of bullets". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 213.497-508 (1914), pp. 437–456. ISSN: 0264-3952. DOI: 10.1098/rsta.1914.0010.

[65] H. Kolsky. "An Investigation of the Mechanical Properties of Materials at very High Rates of Loading". In: *Proceedings of the Physical Society. Section B* 62.11 (1949), pp. 676–700. ISSN: 0370-1301. DOI: 10.1088/0370-1301/62/11/302.

[66] Q. B. Zhang and J. Zhao. "Determination of mechanical properties and full-field strain measurements of rock material under dynamic loads". In: *International Journal of Rock Mechanics and Mining Sciences* 60 (2013), pp. 423–439. ISSN: 13651609. DOI: 10.1016/j.ijrmms.2013.01.005.

[67] Tomas Möller and Ben Trumbore. "Fast, minimum storage ray/triangle intersection". In: *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*. Ed. by John Fujii. New York, New York, USA: ACM Press, 2005, p. 7. DOI: 10.1145/1198555.1198746.

[68] A. Fuhrmann, G. Sobotka, C. Groß. "Distance fields for rapid collision detection in physically based modeling". In: *Proceedings of GraphiCon* 2003 (), pp. 58–65.

[69] MathWorks. *MATLAB [software]. Version R2020a*. 2020. URL: https://de.mathworks.com/products/matlab.html.

[70] Li Ge Wang, Ruihuan Ge, and Xizhong Chen. "Establishing an oblique impact breakage master curve using a DEM bonded contact model". In: *Computers and Geotechnics* 145 (2022), p. 104668. ISSN: 0266352X. DOI: 10.1016/j.compgeo.2022.104668.

[71] MOORE E. F. "The shortest path through a maze". In: *Proc. Int. Symp. Switching Theory, 1959* 0 (1959), pp. 285–292. URL: https://ci.nii.ac.jp/naid/10015375086/.

[72] Tamir Brosh, Haim Kalman, and Avi Levy. "Fragments spawning and interaction models for DEM breakage simulation". In: *Granular Matter* 13.6 (2011), pp. 765–776. ISSN: 1434-5021. DOI: 10.1007/s10035-011-0286-z.

[73] Rayleigh. "On Waves Propagated along the Plane Surface of an Elastic Solid". In: *Proceedings of the London Mathematical Society* s1-17.1 (1885), pp. 4–11. ISSN: 00246115. DOI: 10.1112/plms/s1-17.1.4.

[74] Catherine O'Sullivan and Jonathan D. Bray. "Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme". In: *Engineering Computations* 21.2/3/4 (2004), pp. 278–303. ISSN: 0264-4401. DOI: 10.1108/02644400410519794.

[75] James M. Gere and Barry J. Goodno. *Mechanics of materials*. 8th ed. Stamford, CT: Cengage Learning, 2013.

[76] Xiaobin Ding et al. "Effect of Model Scale and Particle Size Distribution on PFC3D Simulation Results". In: *Rock Mechanics and Rock Engineering* 47.6 (2014), pp. 2139–2156. ISSN: 0723-2632. DOI: 10.1007/s00603-013-0533-1.

[77] Y. B. Lu and Q. M. Li. “About the dynamic uniaxial tensile strength of concrete-like materials”. In: *International Journal of Impact Engineering* 38.4 (2011), pp. 171–180. ISSN: 0734743X. DOI: 10.1016/j.ijimpeng.2010.10.028.

[78] Sang Ho Cho, Yuji Ogata, and Katsuhiko Kaneko. “Strain-rate dependency of the dynamic tensile strength of rock”. In: *International Journal of Rock Mechanics and Mining Sciences* 40.5 (2003), pp. 763–777. ISSN: 13651609. DOI: 10.1016/S1365-1609(03)00072-8.

[79] X. Q. Zhou and H. Hao. “Modelling of compressive behaviour of concrete-like materials at high strain rate”. In: *International Journal of Solids and Structures* 45.17 (2008), pp. 4648–4661. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2008.04.002.

[80] Stef Lommen, Dingena Schott, and Gabriel Lodewijks. “DEM speedup: Stiffness effects on behavior of bulk material”. In: *Particuology* 12 (2014), pp. 107–112. ISSN: 16742001. DOI: 10.1016/j.partic.2013.03.006.

[81] Alexander BECKER. “API Guide”. In: *(ThreeParticle Simulation)* ThreeParticle 2019 R6.0 : 2022 ().