THE UNIVERSITY *of* EDINBURGH

Efficient simulation of nonspherical particles using the discrete element method

---

Di Peng

8th August 2020

The University of Edinburgh

THE UNIVERSITY
of EDINBURGH

School of Engineering
Institute for Infrastructure and Environment
Granular Mechanics and Industrial Infrastructure Group

A thesis submitted for the degree of Doctor of Philosophy

Efficient simulation of nonspherical particles using the discrete element method

Di Peng

External reviewer — Prof. Yuntian Feng
College of Engineering
Swansea University

Internal reviewer — Dr. Stefanos Aldo Papanicolopulos
Institute for Infrastructure and Environment
The University of Edinburgh

Supervisors — Dr. Kevin J. Hanley
Prof. Jin Y. Ooi

8th August 2020

Di Peng

Efficient simulation of nonspherical particles using the discrete element method

A thesis submitted for the degree of Doctor of Philosophy, 8th August 2020

Reviewers: Prof. Yuntian Feng and Dr. Stefanos Aldo Papanicolopulos

Supervisors: Dr. Kevin J. Hanley and Prof. Jin Y. Ooi

The University of Edinburgh

Granular Mechanics and Industrial Infrastructure Group

Institute for Infrastructure and Environment

School of Engineering

Thomas Bayes Road, Alexander Graham Bell Building, The King's Buildings

EH9 3FG Edinburgh

Abstract

The discrete element method (DEM) is a numerical simulation approach for particulate systems proposed during the 1970s. The computational expense of DEM traditionally limited the simulations to small numbers of highly-idealised particles, typically disks or spheres. However, continual increases of computational power means that it is now feasible to incorporate nonspherical particles in DEM simulations. However, there are still significant gaps in the theory that need to be addressed before nonspherical particles find widespread adoption, particularly in industry. This thesis sought to develop some important theoretical aspects of nonspherical particle simulations, and increase the efficiency of these simulations.

Contact detection is a major issue in simulating nonspherical particles. The first original scientific chapter of this thesis describes a novel contact-detection algorithm between convex polyhedra and superquadrics, which generally refer to blocky and round particles, respectively. The contact detection is based on a highly efficient 'search and return' method. The algorithm has been successfully validated for all types of contact between polyhedra and superquadrics. This algorithm makes it possible to simulate a system containing particles of both blocky and round shapes.

Selecting a stable, efficient time step is essential for any DEM simulation; choosing a larger time step will increase a simulation's efficiency. The second scientific chapter of this thesis presents a method for calculating the critical time step for systems of nonspherical particles in DEM analyses. The critical time step was analytically derived from the amplification matrix of the simulation and is explicit with damping considered. For underdamped cases, this approach gives a similar critical time step for spheres compared with previous studies. Moreover, this approach is applicable to underdamped, critically-damped and overdamped cases while previous studies were restricted to underdamped cases.

The final scientific chapter of this thesis is an application: simulating ellipsoidal beans in a rotating drum. Laboratory experiments were performed in which the

system was recorded by a high-speed camera and the images were analysed with particle image velocimetry (PIV) for validation of DEM results. The interaction of particles with the drum's surface was the main focus of this study. Both soybeans and red beans slide along the drum in the simulation and experiment. This was observed from velocity analysis of the PIV data and mobilised friction using the DEM data. This sliding was relative to the dynamic angle of repose, but the particle translational velocities predicted from DEM and PIV differed while the simulated dynamic angle of repose was close to that in experiments, indicating that the DEM model did not exactly match the physical experiment. Other micro-scale behaviours in the system were also investigated.

This thesis consists of both theoretical development and a practical application. The former provides some indication of how to simulate nonspherical particle systems in DEM more efficiently and more possibilities to simulate systems of diverse particle shapes, while the latter provides insights into a common engineering system of industrial relevance.

Publications related to this thesis:

1. D Peng and KJ Hanley, Contact detection between convex polyhedra and superquadrics in discrete element codes, Powder Technology, 2019, 356: 11-20, `https://doi.org/10.1016/j.powtec.2019.07.082`.

2. D Peng, SJ Burns and KJ Hanley, Critical time step for DEM simulations using a Hertzian contact model and Euler integrator, Proceedings of the 8th International Conference on Discrete Element Methods (DEM8), 2019, `https://mercurylab.co.uk/dem8/wp-content/uploads/sites/4/2019/07/208.pdf`.

3. D Peng, SJ Burns and KJ Hanley, Critical time step for DEM simulations of convex particles with central symmetry, International Journal for Numerical Methods in Engineering, 2020 (under review).

Acknowledgements

Time flies fast. Three years, around one thousand days, have witnessed the progress of my PhD. Getting the job done, I hereby sincerely acknowledge all those who helped me a lot throughout my years spent in Edinburgh.

First I would like to thank my principal supervisor, Dr. Kevin Hanley. Kevin supports me during my three years' PhD extremely much, both academically and in living. I would also like to acknowledge my secondary supervisor, Prof. Jin Ooi, for his great inspirations contributing this thesis. Ms. Nannan Huang is acknowledged for her indelible contribution to this research. Other faculty in the granular group, including Dr. Jin Sun, Dr. Stefanos Papanicolopulos, Dr. Chris Beckett and Mr. Jim Hutcheson are acknowledged for their thoughtful help throughout these years.

Second I would thank all my colleagues in granular group in Edinburgh: Dr. John Paul Morrissey, Dr. Lige Wang, Dr. Zeynep Karatza, Dr. Xizhong Chen, Dr. Tim Najuch, Dr. Shane Burns, Dr. Rangarajan Radhakrishnan, Dr. Rosario Capozza, Dr. Chongqiang Zhu, Dr. Joel Keishing, Mr. Stefan Pantaleev, Mr. Yang Cui, Ms. Milena Velikova, Mr. Stephanos Constandinou and Mr. Masahiko Watanabe. Other guys from cooperative groups in China: Mr. Jiawei Xu from Zhejiang University, Mr. Lin Bu from Shandong University are also acknowledged. Colleagues from DEM Solutions Ltd. where I used to carry on internship are acknowledged: Dr. Marina Sousani, Mr. Carles Bosch Padros, Ms. Sophie Broad, Mr. Rich Rowan and all other colleagues in the engineering department. Dr. Alexander Podlozhnyuk from DCS Computing GmbH is acknowledged for his great aid on my PhD research. In addition, Dr. Andrew Firth and Mr. Songbo Wang from School of Engineering, University of Edinburgh are acknowledged. In addition, Prof. Yun Bai, Prof. Yu Huang, Prof. Zhenming Shi, Prof. Zixin Zhang and Dr. Xin Huang from Tongji University are also acknowledged.

Third I would thank all my friends I have made in Edinburgh during my three years of life in Edinburgh: Ms. Min Zou from Panda Kitchen Ltd., Lili Zhang, Jianlong Xu, Ke Sui and Jialin Hao from China Red Catering Ltd. Also my friends in China,

Mr. Yuejun Li, Mr. Shuang Qiu, Mr. Xian Wang and Mr. Hengxi Ren, who supported me throughout my PhD, are acknowledged.

Finally, my acknowledgements are given to my parents, Mr. Guobin Peng and Ms. Yi Wang, who always support me, not only financially but also mentally. Funding from the Univerisity of Edinburgh Global Research Scholarship and the China Scholarship Council (No. 201606260230) is also acknowledged. I would prefer to end with a poem:

长夜过尽显澄空，

The clear blue sky appears after the long night.

层峦叠嶂群芳红。

Endless mountains are in the eyes, flowers are all red.

寒窗廿载终不悔，

I have been studying hard for more than twenty years, and it is truly worthy.

他日向东看巨龙。

Just see the huge dragon at the east someday.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Discrete element method (DEM)

1.1.1 Overview

The discrete element method (DEM), also known as distinct element method, was first introduced by Cundall and Strack (1979). DEM treats the system as a collection of particles (elements) and discretises the time into a large number of small time intervals, each of which is called a time step. During each time step, particles interact with each other, which is done by determining which particles are in contact (contact detection) and evaluating contact forces between them using a specific contact model specifying a proper force-displacement law (a mechanical constitutive model). Then the acceleration (both translational and rotational components) of each particle is derived from the particle's net force and moment using Newton's second law. Then two linear integrals are computed to work out the particle's displacement and rotation during a time step, with a variety of integrators which can be either explicit (more common) or implicit. By deducing the movement of all particles at the micro-scale for each time step, the change of the system at the macro-scale is ascertained. By repeatedly performing the same calculation process, the evolution of the particle system is simulated.

DEM has some similarity to molecular dynamics (MD, Aktulga et al. (2012)), which is a numerical method widely used in physics. For example, both methods deal with multi-particle systems and perform time evolution by steps. The differences between MD and DEM are: (1) particles are shapeless and dimensionless in MD; (2) particle kinematics are generally controlled by a potential function and force field in MD rather than Newton's kinematics in DEM; (3) unlike DEM, particles interact each other without a physical contact in MD. However, the similarity between the methods means that some codes for MD that have been further developed to do DEM simulations, e.g., LAMMPS (Plimpton (1995)).

DEM has become a widely adopted numerical method in many areas of engineering, including geotechnical engineering which is its origin (Cundall and Strack (1979), Thornton (2000), McDowell et al. (2006) and Chen et al. (2011)), structural (Lemos

(2007)) and environmental engineering (Cleary and Prakash (2004)), chemical engineering (Rhodes et al. (2001), Anand et al. (2008) and Sakai and Koshizuka (2009)) as well as material science (Buttlar and You (2001), You and Buttlar (2004) and Sun et al. (2017)), agriculture (Tijskens et al. (2003)), biology and medicine (Genda et al. (2001) and Abraham et al. (2013)). O'Sullivan (2014) reviewed DEM as a numerical modelling method from the perspective of geotechnical engineering.

DEM has some advantages as a simulation tool. For example, compared with the finite element method (FEM), DEM with an explicit integrator eliminates any possibility of convergence problems. In addition, DEM regards particles as individual elements which can move around freely and interact with other particles, which allows DEM to model systems with large deformation / fracture. By contrast, in FEM simulations, the neighbouring elements always interact, as numerical data transfer relies on joints of these linked elements. This may cause problems when large deformations occur.

1.1.2 DEM codes

There are numerous DEM codes available, both commercial and open-source, providing a variety of choices. Commercial codes include PFC (*PFC - Particle Flow Code, Version 6.0* (2018), newest version 6.0), EDEM (*EDEM 2019 User Guide* (2019), newest version 2020.0) and MatDEM (Liu et al. (2013), Liu et al. (2015) and Liu et al. (2017)), and open-source codes include YADE-DEM (*Yade Documentation 2nd ed.* (2015)), LAMMPS (Plimpton (1995)) along with its modification LIGGGHTS (Kloss et al. (2012)) and the recently released code MercuryDPM (Thornton et al. (2012) and Weinhart et al. (2012)).

EDEM and YADE-DEM have graphical user interfaces (GUIs) and all the operations are done within the GUI; there is no need for scripting. PFC also has a GUI, but the simulations can only be carried out via input scripts with specific grammar (FISH). MatDEM is a MATLAB (*MATLAB version 9.5.0 (R2018b)* (2018)) based code and is run in the MATLAB interface where a GUI is available. LAMMPS and LIGGGHTS do not have a GUI, and the simulations are carried out via input scripts with specific grammar which is different from that in PFC. MercuryDPM is a set of kernels for DEM simulation and users need to compile their own C/C++ codes as input files describing the problem and call the kernels for simulation.

PFC has only a Windows release while YADE-DEM and MercuryDPM have only Linux releases. EDEM, LAMMPS and LIGGGHTS can run on both Windows and Linux systems. MatDEM can also run on Windows and Linux, as MATLAB is on both systems.

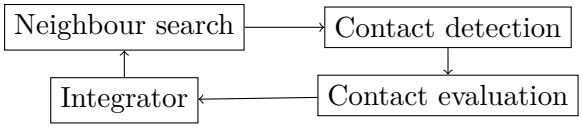


Fig. 1.1.: Process of DEM during each time step.

1.1.3 Basic algorithm of DEM

DEM simulation involves repeating the same process during each time step of the simulation. The process can be divided into four parts which are shown in Figure 1.1.

1.1.3.1. Neighbour search

The first step is neighbour search, which finds potential contacts, i.e., particles in close proximity within the particle domain. This reduces the computational cost of contact detection (Zhao et al. (2006)). There are many particles within a system, and there is no need to check each pair of them. Instead, the system can be divided geometrically into mesh grids that contain a number of particles (Rapaport (2004) and Xu et al. (2011)), so that particles in non-neighbouring mesh cells cannot be in contact. This could save a lot of effort as only particles within the same/neighbouring mesh cells should be checked for contact. Recently there have been developments on neighbour search for systems with nonspherical particles (Yan and Regueiro (2018)).

The size of the mesh grid is important. Too course or too fine meshes will both slow down the DEM simulation. In addition, the number of cells sometimes corresponds to the number of processors for DEM simulation (Xu et al. (2011)). Usually, DEM codes provide automatic mesh size estimation (e.g., EDEM, PFC, LIGGGHTS, etc.); some codes even optimise the mesh during a simulation.

1.1.3.2. Contact detection

For two neighbouring particles, it is crucial to tell whether they are in contact. That is contact detection. The sphere is the most common shape of particle in 3D

DEM. Taking spherical particles as an example, the contact detection criterion can be expressed as

$$\boldsymbol{d} = \boldsymbol{x_2} - \boldsymbol{x_1} \tag{1.1}$$

$$\delta = r_1 + r_2 - |\boldsymbol{d}| \tag{1.2}$$

$$\textbf{IsContact} = \begin{cases} 0 & \text{if } \delta \leq 0 \\ 1 & \text{if } \delta > 0 \end{cases} \tag{1.3}$$

where r_1, r_2 are the radii of two spherical particles, $\boldsymbol{x_1}, \boldsymbol{x_2}$ are their respective centre locations, δ is contact overlap and $\boldsymbol{d}$ is a vector linking their centres. For nonspherical particles, the contact detection is much more complex, e.g. Cundall (1988) and Podlozhnyuk et al. (2017). This thesis also describes a contact detection algorithm for nonspherical particles in detail in Chapter 2. For a simple introduction in this section, all the following equations are sphere based.

If the two spheres are in contact, then we can define the contact normal direction as:

$$\boldsymbol{n} = \boldsymbol{d}/|\boldsymbol{d}| \tag{1.4}$$

Along with the overlap δ obtained above, the contact detection process is complete.

1.1.3.3. Contact evaluation

Based on contact normal direction and contact overlap obtained above, the contact force can be worked out. This is the function of the contact model. For spherical contact, the contact plane passes through the contact point with normal direction of $\boldsymbol{n}$. The contact force can be separated into two parts: a normal part which is perpendicular to the contact plane and a shear (tangential) part which is within the contact plane. The simplest and most widely used contact model, the linear (Hooke) model (Cundall and Strack (1979)), regards contact behaviour between two particles as a spring where the force is linearly related to deformation, which is overlap in DEM. The linear model can be applied for both the normal and shear directions. However, the shear direction is slightly different from normal, as the shear force must be computed incrementally while the normal force does not need to be. In addition, sliding will occur when the shear force exceeds the limit of

frictional force at the contact. At this time, the shear force will be restricted to the frictional limit.

$$\boldsymbol{F} = \boldsymbol{F}_n + \boldsymbol{F}_s \tag{1.5}$$

$$\boldsymbol{F}_n = k_n \delta \boldsymbol{n} \tag{1.6}$$

$$\Delta \boldsymbol{F}_s = k_s \Delta \boldsymbol{s}_s \tag{1.7}$$

$$|\boldsymbol{F}_s| \leq \mu |\boldsymbol{F}_n| \tag{1.8}$$

where $\boldsymbol{F}$ is total contact force, $\boldsymbol{F}_n$, $\boldsymbol{F}_s$ are normal and shear forces, respectively, $\boldsymbol{s}_s$ is shear displacement at the contact, k_n, k_s are normal and shear contact stiffnesses, respectively, and μ is the interparticle frictional coefficient.

For a linear model, k_n, k_s are constants during a simulation. A more advanced nonlinear model is Hertz-Mindlin model (Mindlin (1953)). The normal and shear contact stiffnesses k_n, k_s contain $\sqrt{r\delta}$ so that they are not constants, in which $r = r_1 r_2/(r_1 + r_2)$ is the equivalent radius of two particles.

$$k_n = \frac{4}{3} E' \sqrt{r\delta} \tag{1.9}$$

$$k_s = 8 G' \sqrt{r\delta} \tag{1.10}$$

where E' and G' are equivalent Young's and shear moduli, respectively:

$$\frac{1}{E'} = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} \tag{1.11}$$

$$\frac{1}{G'} = \frac{2(2-\nu_1)(1+\nu_1)}{E_1} + \frac{2(2-\nu_2)(1+\nu_2)}{E_2} \tag{1.12}$$

where E_1, E_2, ν_1, ν_2 are the Young's moduli and Poisson's ratios of the two particles, respectively. Both linear and Hertz models yield a contact force, a combination of normal and shear components, at the contact point. The contact force contributes to the translational and rotational acceleration of both particles.

Some complex models, for example, for rolling resistance (Iwashita and Oda (1998) in 2D, Ai et al. (2011) in 3D) will give another moment (torque) at the contact along with a contact force:

$$\Delta M = k_M \Delta \theta \text{ (2D)} \tag{1.13}$$

$$\Delta \boldsymbol{M} = \boldsymbol{k}_M \Delta \boldsymbol{\theta} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_M & 0 \\ 0 & 0 & k_M \end{bmatrix} \Delta \boldsymbol{\theta} \text{ (3D)} \tag{1.14}$$

where θ and $\boldsymbol{\theta}$ are the rotational displacements of the contact in 2D and 3D, respectively, and k_M is the rotational stiffness. Some bond models (e.g. Potyondy

and Cundall (2004) and Brown et al. (2014)) also give a moment at the contact between bonded particles. Compared to the unbonded model, in the bond model the bond dominates the mechanical behaviour of the contact, until it breaks, after which the contact behaviour generally returns to that of an unbonded model.

In real systems, energy loss always takes place; in DEM, in addition to the frictional sliding mentioned above, one main way of energy dissipation is through damping. The most common form of damping is viscous damping, which is usually applied via a dashpot force that is proportional to contact relative velocity (the relative velocity between two particles in contact) along with the contact force (e.g. PFC and LAMMPS/LIGGGHTS). The coefficient of viscous damping may be related to a physical parameter such as the coefficient of restitution, e.g., LAMMPS/LIGGGHTS:

$$c = \sqrt{\frac{4m'k_n}{1 + (\frac{\pi}{\ln e})^2}} \tag{1.15}$$

$$\frac{1}{m'} = \frac{1}{m_1} + \frac{1}{m_2} \tag{1.16}$$

where c is the viscous damping coefficient for both normal and shear directions, e is the coefficient of restitution, m' is equivalent particle mass, and m_1 and m_2 are particle masses respectively.

Apart from viscous damping, other forms of damping exist, for example, "local damping" in PFC, which applies a body force to each particle in the opposite direction of particle velocity (Cundall (1987)).

1.1.3.4. Integrator

Once the net forces and moments are known during a time step, the acceleration, velocity and displacement of each particle, both translational and rotational, must be computed. The acceleration is computed via Newton's second law given particle mass m and inertia I, while the velocity and displacement are computed via integration. The position (and orientation, if applicable) of particles can be then updated for the next time step.

$$\ddot{\boldsymbol{x}} = \sum_{i=1}^{n} \boldsymbol{F}_i / m \tag{1.17}$$

$$\ddot{\boldsymbol{\theta}} = \sum_{i=1}^{n} \boldsymbol{M}_i / I \tag{1.18}$$

where $\boldsymbol{F}_i$ and $\boldsymbol{M}_i$ are forces and moments on the particle and $I = \frac{2mr^2}{5}$ is the moment of inertia of a sphere.

There are several integrators in common use to obtain particle velocity and displacement from acceleration. The Euler integrator (Euler (1768)) is a simple integrator, for which the integration between the nth and $(n+1)$th step can be expressed as follows:

$$\dot{\boldsymbol{x}}_{n+1} = \dot{\boldsymbol{x}}_n + \ddot{\boldsymbol{x}}_n \Delta t \tag{1.19}$$

$$\dot{\boldsymbol{\theta}}_{n+1} = \dot{\boldsymbol{\theta}}_n + \ddot{\boldsymbol{\theta}}_n \Delta t \tag{1.20}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \dot{\boldsymbol{x}}_n \Delta t \tag{1.21}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \dot{\boldsymbol{\theta}}_n \Delta t \tag{1.22}$$

where Δt is the time step. The Euler integrator is not widely used in DEM simulations due to the fact that it uses the integrated velocity at the end of a time step for displacement integration, which will amplify the error.

Another integrator, the velocity Verlet integrator (Verlet (1967)), does not have this problem and is more widely used in DEM codes, e.g., PFC and LAMMPS/LIGGGHTS. The main characteristic of the velocity Verlet integrator is that it integrates the velocity at the middle point of each step and displacement at the start/end of the step:

$$\dot{\boldsymbol{x}}_{n+\frac{1}{2}} = \dot{\boldsymbol{x}}_{n-\frac{1}{2}} + \ddot{\boldsymbol{x}}_n \Delta t \tag{1.23}$$

$$\dot{\boldsymbol{\theta}}_{n+\frac{1}{2}} = \dot{\boldsymbol{\theta}}_{n-\frac{1}{2}} + \ddot{\boldsymbol{\theta}}_n \Delta t \tag{1.24}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \dot{\boldsymbol{x}}_{n+\frac{1}{2}} \Delta t \tag{1.25}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \dot{\boldsymbol{\theta}}_{n+\frac{1}{2}} \Delta t \tag{1.26}$$

Compared with the Euler integrator, the velocity Verlet integrator has more accurate integration of displacement. Some DEM codes like EDEM allow the user to determine which integrator to use, providing more possibilities and comparisons.

1.1.4 Postprocessing of DEM datasets

Once a DEM simulation is finished, raw data are exported, consisting of particle and contact information. Generally, information other than that from raw data needs to be investigated for the following analysis, for example, in soil mechanics, the stress and strain of the soil is required rather than the velocities of each soil particle. Therefore, it is crucial to postprocess the raw data to obtain this information.

1.1.4.1. Visualisation

One main postprocessing operation is visualisation, as EDEM, PFC and YADE-DEM have their own GUIs. Thus various variables, including particle translational/rotational acceleration/velocity/displacement, contact force (normal/shear/total), etc. can be visualised in these codes without external software. However, LAMMPS/LIGGGHTS and MercuryDPM users have to visualise their dataset on their own. A widely used open-source package is the visualisation toolkit (VTK) (Schroeder et al. (2006)) by Kitware. LAMMPS/LIGGGHTS can write output as a formatted VTK dataset. Kitware also publishes a user interface called ParaView (Ayachit (2015)) which is powered by VTK. ParaView can run on both Windows and Linux systems and is user-friendly with a GUI interface. ParaView can also run with Python scripts. This thesis uses ParaView for visualising superquadric particles.

1.1.4.2. Coarse graining

One may be interested in continuum quantities such as stress or strain of a specific region within the system. A single particle cannot reflect such information. PFC has a capability called "measurement circle/region" (circle for version 3.0-4.0, region for version 5.0-6.0) for interpreting stress and strain within *PFC - Particle Flow Code, Version 6.0* (2018). The measurement circle/region is a circle in 2D and a spherical surface in 3D within which stress and strain can be directly computed. Similarly, LAMMPS/LIGGGHTS can set up a "region" and group all the particles in it for further computation.

In recent years, a technique of "coarse graining" was introduced (Noid et al. (2008a), Noid et al. (2008b) and Goldhirsch (2010)). The coarse graining method first establishes a coarse graining function, $\phi(\boldsymbol{r}-\boldsymbol{r}_i)$, usually a Gaussian function, to ensure that the integral of mass within a specific volume equals to the total mass of the particles within the volume:

$$\rho(\boldsymbol{r}) = \sum_i m_i \phi(\boldsymbol{r}-\boldsymbol{r}_i) \tag{1.27}$$

where $\boldsymbol{r}$ is centre of mass of the volume, m_i is mass of each particle and $\boldsymbol{r}_i$ is the centre of mass of each particle within the volume. With this function, a lot of variables can be exported coarse-grained, such as velocity:

$$\boldsymbol{V}(\boldsymbol{r}) = [\sum_i m_i \boldsymbol{v}_i \phi(\boldsymbol{r}-\boldsymbol{r}_i)]/\rho(\boldsymbol{r}) = \frac{\sum_i m_i \boldsymbol{v}_i \phi(\boldsymbol{r}-\boldsymbol{r}_i)}{\sum_i m_i \phi(\boldsymbol{r}-\boldsymbol{r}_i)} \tag{1.28}$$

where $\boldsymbol{v}_i$ is the velocity of each particle within the volume.

Course graining is now widely used (Weinhart et al. (2016), Chu et al. (2016), Daraio et al. (2019), Xie et al. (2019b) and Tausendschön et al. (2020)) and has been implemented into Iota (*Iota User Manual* (2019)), which is the implementation used in this thesis. Iota has a user-friendly GUI which is easy to operate, and supports various of input data formats (EDEM, LIGGGHTS-DUMP and LIGGGHTS-VTK).

1.1.5 High-performance computation for DEM

DEM requires a large number of computations within each time step, and usually the time step is less than 1×10^{-4} s. This leads to very large computational effort to simulate even one second in the real world. It is impossible to carry out real-time DEM simulation with present computational capacity. However, some improvements upon traditional single-processor CPU implementations can process.

Most supercomputers have central processing units (CPUs) in grids. Some are instead equipped with graphics processing units (GPUs). Each CPU forms a node within the grid. In addition, one CPU may contain multiple processors. Since a large particle system can be divided into cells for a DEM simulation, it is natural to consider one CPU/processor calculating one cell. Indeed that is the approach that is adopted. However, as particles may pass from one cell to another, efficiently transferring data between nodes is crucial to achieve good parallel scaling performance. There are mainly two programming approaches used for parallelisation: one is a protocol called message passing interface (MPI, Walker (1992)); the other is an application programming interface (API) called open multi-processing (OpenMP, Dagum and Menon (1998)). MPI is typically used for parallelising among different nodes (CPUs) while OpenMP is used for parallelising within a node (CPU). In this thesis, the DEM simulations have been performed on a supercomputer at the University of Edinburgh with MPI.

Parallelisation on GPUs can also be possible. Recently general-purpose computing on graphics processing units (GPGPU) (Lee et al. (2009)) is popular, with open computing language (OpenCL, Stone et al. (2010)) as an implementation. Compared with a CPU, a GPU has a slower clock frequency but many more processors. Therefore, a GPU may be faster than a CPU once the system is large enough and divided into large number of grid cells via GPU threads. There are recently more and more DEM studies using GPU parallelisation: Qi et al. (2015), Lisitsa et al. (2018) and Sousani et al. (2019). However, a DEM code must be carefully written to exploit the available performance afforded by GPU parallelisation. LAMMPS can include a C++ library called Kokkos (Edwards et al. (2014)) which allows simulations to run efficiently on both GPUs and CPUs with multiple threads, without rewriting the code.

1.1.6 DEM coupling with other numerical methods

DEM is not always used alone. There are several other numerical methods that DEM can couple with. Computational fluid dynamics (CFD) has historically been the most popular, e.g., Zhong et al. (2016), Chu et al. (2016), Jing et al. (2016), Sun and Xiao (2016), Xu et al. (2017), Varas et al. (2017), Miao et al. (2017), Wu et al. (2017), Sousani et al. (2019) and Schrader et al. (2019). In addition, DEM can also be coupled with FEM, e.g. Liu et al. (2020). PFC version 6.0 provides a coupling interface with the FEM code FLAC (*FLAC, Version 8.1* (2020) and *FLAC3D, Version 7.0* (2020)). DEM can also be coupled with lattice Boltzmann method (LBM), etc. (Sun et al. (2013), Leonardi et al. (2014) and Cui et al. (2014)).

1.2 Particle shape in DEM

Particle shape is recognised to have a strong impact on the behaviour of granular systems (Lu et al. (2015)). Initially Cundall and Strack (1979) used 2D discs, due to limited computer power available at the time. Only 2D DEM was was considered during the early 1980s (Bathurst and Rothenburg (1988) and Ting et al. (1989)) until Cundall (1988) first proposed the three-dimensional discrete element model, after which 3D DEM started to appear and the particles switched from 2D discs to 3D spheres (Taylor and Preece (1992)). A sphere is a perfectly homogeneous shape in 3D. Its homogeneity makes contacts between spheres easy to detect and contact overlap determination algorithms are simple and efficient for spheres (Equations (1.1-1.3)). More advanced particle modelling (i.e., multispheres) can still rely on this spherical contact resolution algorithm.

Apart from some codes using polyhedral particles (e.g., Rocky DEM (*Rocky DEM Software Overveiw*)), almost all DEM codes use spheres as their basic element for simulation. However, spheres have a lot of disadvantages in modelling sharp and blocky particles such as rocks. Although Cundall and Strack (1979) initially performed DEM to model rocks, rock and its fragments are highly nonspherical, which motivated the development of nonspherical particle modelling. In addition, particles in reality, e.g., soils, are never perfectly smooth. A review paper of Lu et al. (2015) systematically summarised the methods for modelling nonspherical particles. Generally, these methods can be divided into two categories in 3D: (1) sphere-based methods which adopt spheres as the fundamental element; (2) nonsphere-based methods which use a different fundamental element in the DEM.

1.2.1 Sphere-based methods

Sphere-based methods still adopt spheres as the fundamental shape of element but improve on the usual spherical particle model. The rolling resistance model was first introduced by Iwashita and Oda (1998) in 2D to reflect the coarse surface of soil particles. Sands modelled with perfect spheres have lower friction angles compared to reality, even if the friction coefficient of the particle is set to an unrealistically large value. The problem is that sand particles usually have asperities which somewhat restrict particle rotation when two particles come into contact; however, perfectly smooth spheres cannot capture this characteristic. Therefore, the rolling resistance model introduces one more DEM parameter, the rolling friction, to particle contacts. The rolling resistance model can capture the correct strength behaviour of sands. Rolling resistance has been extended to 3D (Wensrich and Katterfeld (2012)) and even supplemented with more complex twisting resistance in 3D (Jiang et al. (2015)), with another parameter, the twisting resistance coefficient, involved. Ai et al. (2011) assessed multiple rolling resistance models, and more development work continues in this area (Xie et al. (2019a) and Huang et al. (2017)).

However, rolling resistance has some limitations and is lacking in realism. A better sphere-based method involves aggregating more than one sphere into a whole particle, e.g., "multisphere" in LIGGGHTS (Kloss et al. (2012)) or "clump" in PFC (*PFC - Particle Flow Code, Version 6.0* (2018)). The main characteristic of multisphere particles is that during translation and rotation of the whole particle, its inner structure remains rigid and no relative displacement of the forming spheres is allowed. Indeed, no contact assessment takes place for these forming spheres. Figure 1.2 shows a huge sphere with several small spheres embedded in its surface to form a multisphere particle. This particle is believed to physically reflect spheres with rolling resistance. Moreover, as the forming spheres do not interact with each other, they can have any overlaps and be placed with any alignment, which enables infinite shapes of the multisphere particle. Therefore, multisphere is a flexible way to model nonspherical particles, which has been adopted in many DEM studies (Song et al. (2006), Kruggel-Emden et al. (2008) and Markauskas et al. (2010)). Theoretically, multispheres can model any particle shape, as long as there are enough forming spheres of small size; however, the increasing number of forming spheres will increase the calculation effort as the contact resolution for multispheres is still based on these spheres. In addition, the accepted time step in DEM decreases with particle size (which will be discussed in Chapter 3), leading to a huge increase of computational effort as discussed in Section 1.1. Another way of joining particles together is not to rigidly fix them but bond them, e.g., with a bond contact model (e.g., Cheng et al. (2003) and Wang and Yan (2012)). The bond

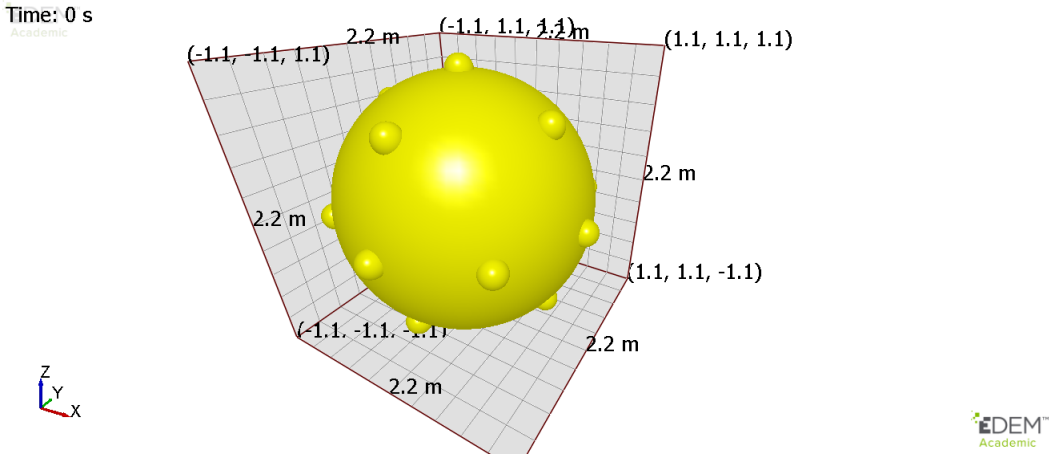


Fig. 1.2.: Illustration of a multisphere particle using EDEM *EDEM 2019 User Guide* (2019).

contact model acts like glue between particles to bind them, although relative translational and/or rotational displacement is allowed. Compared to rigid multispheres, the bonded multisphere can not only reflect particle surface roughness and particle shape but also particle deformation. This is especially useful for soft particles.

As basic elements are used for evaluating interparticle contact, one other method is to form several virtual spheres/circles and use the space intersected by these elements (e.g., overlaps between two virtual spheres) to represent the real particle. This method is surprisingly useful for simulating tablet-shaped particles as tablets can generally be represented by sphere/circle intersections (Song et al. (2006) and Kodam et al. (2012)). However, the particle shapes supported are limited with this method.

1.2.2 Nonsphere-based methods

Forming a nonspherical particle without spherical fundamental elements will make the simple spherical contact detection algorithm invalid. Therefore, specific contact detection algorithms for nonspherical particles become necessary (Lu et al. (2015)). These contact detection algorithms can generally be divided into two types. One describes the particle surface as a mathematical function, e.g., ellipsoids in Equation (1.29) where a, b and c are half the lengths of the principal axes. Ellipsoids are one of the nonspherical shapes that were first studied (Lin and Ng (1997)). Note that spheres can also be described like this as spheres are a special case of ellipsoids.

$$F(x,y,z) = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 = 0 \tag{1.29}$$

A polyhedron can be expressed as Equation (1.30), from Boon et al. (2013), where A_i, B_i, C_i are components of normal vectors of the ith surface plane, and D_i is the distance from the origin to the ith surface plane:

$$F(x, y, z) = \sum_{i=1}^{n} \langle A_i x + B_i y + C_i z - D_i \rangle = 0 \tag{1.30}$$

In Equation (1.30), the Macaulay bracket refers to:

$$\langle x \rangle = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{1.31}$$

By functional intersection and interpolation etc., the contact can be resolved based on the surface functions of two particles. This is called continuous function representation (CFR). Generally, the objective of all CFR contact detection algorithms can be described as follows (Houlsby (2009)):

$$\text{Minimise } F_A + F_B \text{ while } F_A - F_B = 0 \tag{1.32}$$

as long as the particle surface functions are wholly differentiable. CFR is a relatively straightforward method to deal with nonspherical particles, as many abstract shapes can be described using equations of the form $F(x, y, z) = 0$. CFR can even be divided into two subtypes: analytical CFR and numerical/iterative CFR, depending on whether the contact detection algorithm works analytically or numerically. Analytical CFR is usually used for particles with a linear surface function, e.g., polyhedral particles; one representative example of analytical CFR is Cundall's common plane (CP) method for detecting polyhedral particle contacts (Cundall (1988) and Nezami et al. (2006)). In addition, advanced methods of analytical CFR exist (Feng and Owen (2004)). Numerical CFR is usually used for particles with curved surfaces to get numerical results for contact point, etc. Contact detection with CFR usually shares the aim described in Equation (1.32). Numerical CFR has been developed since 1992 (Williams and Pentland (1992), Lin and Ng (1995), Mustoe and Miyata (2001) and Lu et al. (2012)). One example is the modelling of superquadrics (Podlozhnyuk et al. (2017)) which has been adopted in this thesis. For a particle surface described by spline functions (Andrade et al. (2012)), iterative methods such as Newton's method can be adopted. Iterative CFR can be applicable to round-cornered polyhedra, as long as the flat polyhedral surface is processed properly to smoothly join the rounded corner (Boon et al. (2013)).

The alternative to CFR is to describe the particle surface as a set of discrete points (Williams and O'Connor (1995), Williams and O'Connor (1999) and Lu et al. (2012)). Contact detection relies on these representative points. This is called discrete function representation (DFR). Compared with CFR, DFR is usually less

efficient in memory usage, as an accurate particle surface representation requires many points. More discussions on frontiers of CFR and DFR will be in Sec. 2.1.

There are also methods which, in principle, can be applied to any shape of particle. Feng et al. (2012) proposed a method of determining contact orientation based on the potential field which is related to particle shape. Dong et al. (2015) proposed a contact resolution method based on orientation discretisation: the contact region is divided into cells and each cell is categorised as inside or outside of the particle intersection. Their specific approach requires a pre-computed database. The Gilbert-Johnson-Keerthi (GJK) algorithm (Gilbert et al. (1988)) is used in computer graphics for detecting penetration or finding the minimum distance between two objects. The original GJK algorithm has been optimised both for speed (Ong and Gilbert (1997), Bergen (1999) and Cameron (1997a)) and for specific shapes such as polyhedra (Cameron (1997b)). However, the GJK algorithm is accurate only for shapes with flat surfaces; for shapes with curved surfaces such as superquadrics, discretisation is necessary. Another option for simulating nonspherical particles and their contacts is digitisation (Jia and Williams (2001) and Dong et al. (2015)). This method splits the simulation domain into fine meshes and recognise particles as a collection of neighbouring mesh cells. Then the particle contact, including its position and orientation, can be obtained with the mean of overlapping meshes of two particles. This method requires a fine mesh size, which can hamper efficiency.

There are still many areas requiring further research on nonspherical particles in DEM. All the discussion in this thesis will focus on nonspherical particle systems, although spherical particles will be introduced for comparison.

1.3 Motivation and outline of this thesis

According to the literature survey on particle shape above, advanced contact detection is always needed for simulating nonspherical particle systems which are not based on spherical fundamental particles. This constitutes the first motivation of this thesis. In this thesis, Chapter 2 provides a novel contact detection algorithm between polyhedra and superquadrics. Along with polyhedron–polyhedron and superquadric–superquadric contact detection algorithms which already exist (see Chapter 2), this algorithm will enable the simulation of systems with mixed polyhedra (blocky particles) and superquadrics (rounded particles).

The greatest disadvantage of DEM is that it requires extremely large computational effort. For a specific time period to be simulated, the total computational effort

depends on the number of steps and the number of particles. The number of steps is the time period divided by time step. This time step cannot be too large; otherwise the system simulated will be unstable. Chapter 3 of this thesis will discuss this thoroughly.

Finally, a practical experimental and DEM study is carried out, studying natural beans which are regarded as ellipsoids in a rotating drum (Chapter 4), for which the theory of previous chapters has been implemented. The particle image velocimetry (PIV) technique has been used for analysis of images taken during the experiment, and the coarse graining (CG) technique has been used for analysis of DEM simulation data. The results of experiments and DEM simulations are compared, and particle–drum interaction is investigated.

This thesis provides a combined study on both theory and application of nonspherical particles, giving more options for nonspherical particle simulation and guidance on optimising the efficiency of nonspherical particle simulations that can be practically used in large-scale DEM simulations of real systems.

2

Contact detection between convex polyhedra and superquadrics in discrete element codes[1]

2.1 Introduction

Since its formulation by Cundall and Strack (1979) in the 1970s, the discrete element method (DEM) has become extremely popular as it allows exploration of the micro-scale behaviour that determines the bulk-scale response of any particulate system. The most commonly used particle shape in DEM simulations is the sphere, which is both conceptually and computationally simple. The large majority of commercial codes such as PFC (*PFC - Particle Flow Code, Version 6.0* (2018)) and EDEM (*EDEM 2019 User Guide* (2019)), along with open-source codes such as YADE (*Yade Documentation 2nd ed.* (2015)) and LIGGGHTS (Kloss et al. (2012)), adopt spheres as the basic shape of element. However, in reality many particles are poorly described by spheres and some problems cannot be adequately addressed without considering particle shape more accurately.

In recognition of the importance of particle shape, there has been a major increase in scientific interest in recent years in DEM modelling of non-spherical particle systems (Lu et al. (2015)). Many approaches have been proposed which has been discussed already in Chapter 1. Compared to spheres, accurately resolving the contacts between complex fundamental particles such as superquadrics or polyhedra is difficult and computationally expensive. Two approaches are commonly used: discrete function representation (DFR) and continuous function representation (CFR) (Hogue (1998)). DFR (Williams and O'Connor (1995) and Williams and O'Connor (1999)) is based on discretising the surface of a particle into a set of nodes. CFR detects and evaluates interparticle contacts through direct manipulation of the equations which mathematically describe a particle's shape. CFR has two subtypes: analytical CFR is used to resolve contacts between polyhedra (Feng and Owen (2004) and Nezami et al. (2007)) while numerical CFR is used for resolving contacts between particles with curved surfaces (Williams and Pentland

[1]This chapter originates from D Peng and KJ Hanley, Contact detection between convex polyhedra and superquadrics in discrete element codes, Powder Technology, 2019, 356: 11-20, https://doi.org/10.1016/j.powtec.2019.07.082.

(1992), Lin and Ng (1995), Mustoe and Miyata (2001), Andrade et al. (2012) and Lu et al. (2012)). These CFR approaches tend to be applicable to particles only of specific shapes, e.g., cylinders (Feng et al. (2017)) or convex particles represented using the potential particle approach (Houlsby (2009) and Boon et al. (2013)). There are established analytical CFR methods for detecting and evaluating contacts between polyhedra, e.g., the 3DEC (*3DEC - Three-Dimensional Distinct Element Code, Ver. 5.2* (2016)) code adopts Cundall's contact plane (CP) method (Cundall (1988)). PFC 6.0 allows the simulation of systems of mixed spheres and polyhedra using a variant of the contact detection algorithm proposed by Nezami et al. (2006). Overlaps between ellipsoids may be detected efficiently using CFR-based algorithms, e.g. Xu and Chen (2012) and Zhu et al. (2019). A CFR approach for contact resolution of convex superquadrics (Podlozhnyuk et al. (2017)) has been implemented in LIGGGHTS (Kloss et al. (2012)).

While there are established approaches for polyhedron–polyhedron (PH–PH) and superquadric–superquadric (SQ–SQ) contact detection, an open problem is contact detection in mixed polyhedron–superquadric (PH–SQ) systems. Such simulations are desirable as many real systems contain a broad range of particle shapes. Superquadrics are appropriate for simulating rounded particles but cannot capture the high degree of angularity provided by polyhedra. Hence, having both shapes available in a single simulation gives a broader range of possible particle shapes than solely polyhedra or superquadrics.

In this chapter, a novel contact detection algorithm for a convex polyhedron and superquadric is presented which is efficient and reasonably simple. An associated contact evaluation method has also been proposed to determine the contact normal and interparticle overlap. The contact detection algorithm has been implemented in a C++ code, enabling verification of the algorithm and quantification of its run time.

2.2 Development of convex PH–SQ contact detection algorithm

In Sections 2.2.1 and 2.2.2, the superquadric (SQ) and polyhedron (PH) particles are respectively introduced and an existing algorithm for SQ–SQ contact detection is presented. The newly developed algorithm for PH–SQ contact detection is then described in Section 2.2.3 based on this prior knowledge.

2.2.1 Definition of and contact detection between convex superquadrics

The surface of a SQ particle may be written as a function

$$F(\boldsymbol{X}) = F(x,y,z) = \left(\left|\frac{x}{a}\right|^{n_2} + \left|\frac{y}{b}\right|^{n_2}\right)^{\frac{n_1}{n_2}} + \left|\frac{z}{c}\right|^{n_1} - 1 = 0 \tag{2.1}$$

where $\boldsymbol{X}$ is a point defined by three coordinates (x,y,z) in 3D space (Podlozhnyuk et al. (2017)). a,b,c are shape parameters (half-lengths of the particle along the x,y,z axes) and n_1,n_2 are blockiness parameters which define a SQ's shape. $n_1 = n_2 = 2$ corresponds to an ellipsoid; in addition, if $a = b = c$, the ellipsoid becomes a sphere. The LIGGGHTS implementation of SQs requires $n_1, n_2 \geq 2$ (*LIGGGHTS(R)-PUBLIC Documentation* (2016)). This prohibition of non-convex SQs is to avoid issues with multiple contacts between two SQs.

If $F(\boldsymbol{X}) < 0$, $\boldsymbol{X}$ lies within the particle; if $F(\boldsymbol{X}) > 0$, $\boldsymbol{X}$ is outside of the particle. The special case $F(\boldsymbol{X}) = 0$ means that $\boldsymbol{X}$ lies on the particle's surface. Consider two SQs A and B that are potentially in contact. The surfaces of both particles can be written as $F_A(\boldsymbol{X})$ and $F_B(\boldsymbol{X})$, or simply F_A and F_B. Contact detection between A and B is equivalent to the following constrained optimisation (Houlsby (2009)):

$$\text{Minimise } F_A + F_B \text{ while } F_A - F_B = 0 \tag{2.2}$$

Podlozhnyuk et al. (2017) substituted Equation (2.1) into Equation (2.2), and used the Lagrange multiplier method to obtain a solution. Suppose the Lagrange function

$$L(x,y,z,\lambda) = F_A + F_B + \lambda(F_A - F_B). \tag{2.3}$$

Thus,

$$\nabla L = \mathbf{0} \iff \nabla(F_A+F_B) = -\lambda\nabla(F_A-F_B) \iff (1+\lambda)\nabla F_A+(1-\lambda)\nabla F_B = \mathbf{0} \tag{2.4}$$

By combining Equation (2.4), which provides three equations, with $\frac{\partial L}{\partial \lambda} = 0$, or equivalently $F_A - F_B = 0$, a set of four equations is obtained:

$$\boldsymbol{\Phi} = \begin{bmatrix} (1+\lambda)\frac{\partial F_A}{\partial x} + (1-\lambda)\frac{\partial F_B}{\partial x} \\ (1+\lambda)\frac{\partial F_A}{\partial y} + (1-\lambda)\frac{\partial F_B}{\partial y} \\ (1+\lambda)\frac{\partial F_A}{\partial z} + (1-\lambda)\frac{\partial F_B}{\partial z} \\ F_A - F_B \end{bmatrix} = \mathbf{0} \tag{2.5}$$

There must be a solution $(x_0, y_0, z_0, \lambda_0)$ to Equation (2.5). The multi-unknown Newton's method can be used to obtain this solution:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ \lambda_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \\ z_n \\ \lambda_n \end{bmatrix} - \boldsymbol{J}^{-1}\boldsymbol{\Phi} \tag{2.6}$$

where $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{\Phi}$. The detailed solution procedure can be found in Podlozhnyuk et al. (2017), resulting in the point (x_0, y_0, z_0), which is the midpoint of two points $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$: the closest point to the centre of SQ B on SQ A's surface and the closest point to the centre of SQ A on SQ B's surface, respectively. We can evaluate whether or not (x_0, y_0, z_0) is the contact point with the following criterion:

$$F_A(x_0, y_0, z_0) < 0 \quad \text{and} \quad F_B(x_0, y_0, z_0) < 0 \tag{2.7}$$

If Equation (2.7) is true, A and B are in contact; otherwise the two SQs are not in contact. If F_B were substituted by the function of a planar wall, SQ–wall contact detection can be accomplished in a similar manner to find the point on the surface of the SQ which is closest to the wall, or the deepest point beneath the wall surface if the SQ contacts the wall (Podlozhnyuk et al. (2017)).

2.2.2 Definition of and points within convex polyhedra

The surface function describing a polyhedron is given as Equation (2.8) (Boon et al. (2013)):

$$F(\boldsymbol{X}) = F(x, y, z) = \sum_{i=1}^{n} \langle f_i(x, y, z) \rangle = \sum_{i=1}^{n} \langle A_i x + B_i y + C_i z - D_i \rangle = 0 \tag{2.8}$$

where n is the total number of faces comprising the PH, $f_i(x, y, z)$ is the function of the i^{th} plane of the PH with outer normal vector (A_i, B_i, C_i), and D_i is the perpendicular distance from the origin to the i^{th} plane of the PH. The Macaulay brackets in Equation (2.8) indicate that positive terms retain their values while negative terms are set to zero. Equation (2.8) is equivalent to

$$F(\boldsymbol{X}) \leq 0 \iff \forall f_i(\boldsymbol{X}) \leq 0 \tag{2.9a}$$

$$F(\boldsymbol{X}) > 0 \iff \exists f_i(\boldsymbol{X}) > 0. \tag{2.9b}$$

Equation (2.9a) means that if a point is within a PH, all PH surface functions must be negative at that point. Conversely, if a point makes one or more PH surface functions positive, that point must be outside the PH (Equation (2.9b)). Thus Equation (2.9a) can be used to confirm a possible contact point, while Equation

(2.9b) can be used to eliminate invalid candidate contact points. Equation (2.9) is used extensively in the algorithm presented in Section 2.2.3.

2.2.3 Resolving contact or non-contact between a PH and a SQ

In total, there are six distinct cases that must be considered in this algorithm: three cases in which the PH and SQ are in contact, and three in which they are not. We have chosen a triangular prism with five faces as the PH and an ellipsoid as the SQ for most of the figures in Section 2.2.3. After the six cases have been described, a flowchart is shown as Figure 2.7 which summarises the steps in the algorithm.

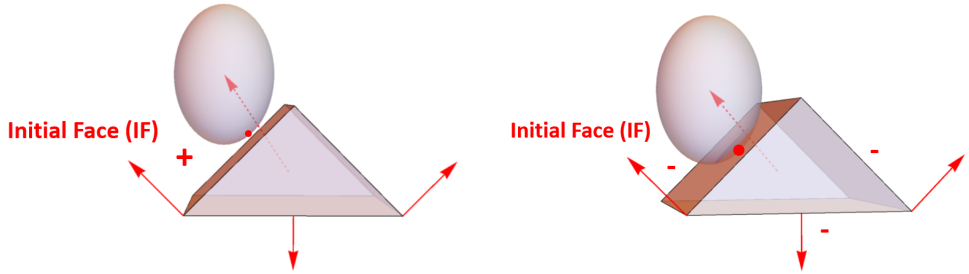
2.2.3.1. Defining the 'Initial Face' (IF) and positive/negative faces

We define the initial face (IF) as the face of the PH whose outer normal vector has the maximum dot product, i.e., is most closely aligned, with the vector joining the centroids of the PH and SQ. An example IF is shown in Figure 2.1(a). The IF is used in the first step to judge whether or not the particles are in contact. Positive/ negative faces are assigned that status by their positions relative to a fixed point $\boldsymbol{X}$. Each face lies on an infinite plane; if the point $\boldsymbol{X}$ is above that plane, where the outer normal vector indicates the positive direction, that face is positive for point $\boldsymbol{X}$. Conversely, if $\boldsymbol{X}$ is beneath the plane, the face is negative for $\boldsymbol{X}$. In order to clearly present a 3D shape on , only three of the faces of the triangular prism PH are labelled as positive or negative in the figures in this section; the other faces are always negative.

In this algorithm, the SQ–planar wall contact detection approach summarised in Section 2.2.1 (Podlozhnyuk et al. (2017)) is applied to one or more specific planes enclosing faces of the PH. This yields one or more points on the surface of the SQ which are potentially contact points.

2.2.3.2. Case 1: Positive IF

Consider Figure 2.1(a) in which the PH and SQ are not in contact. The point highlighted on the surface of the SQ has been identified as a potential contact point based on Section 2.2.1. However, this point makes the IF positive. Based on Equation (2.9b), there must be no contact and the contact detection can be



(a) Case 1: non-contact due to a positive IF (b) Case 2: the simplest contact case with all PH faces negative

Fig. 2.1.: The simplest (non-)contact cases for face contact of a PH and SQ.

terminated. Case 1 is highly efficient as there is no need to check any PH faces apart from the IF.

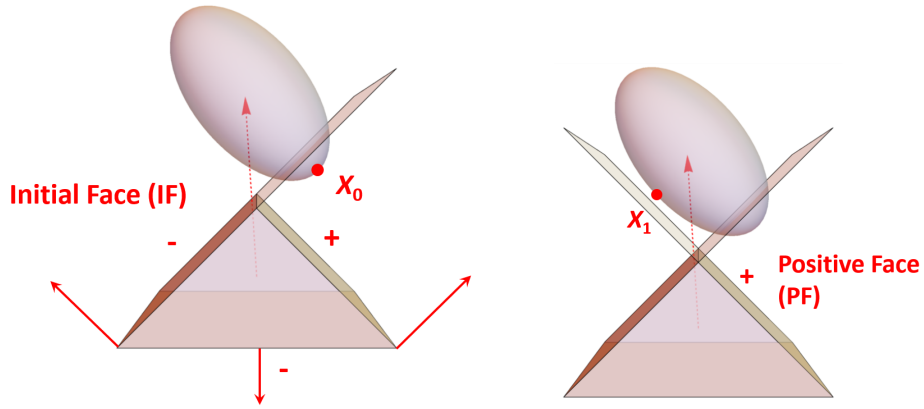
2.2.3.3. Case 2: All faces negative

Case 2, in which all faces are negative, is shown in Figure 2.1(b). The highlighted point must be the contact point according to Equation (2.9a). Case 2 is somewhat less efficient than Case 1 as the positive/negative assessment must be made for all faces of the PH, not only the IF.

2.2.3.4. Cases 3 & 4: Negative IF, resolve through switch to 'Positive Face' (PF)

Cases 3 (non-contact) and 4 (contact) arise when the IF is negative for the potential contact point denoted as $\boldsymbol{X}_0$ but one other face is positive. The scenario in which more than one face is positive is addressed in Section 2.2.3.5. Figure 2.2(a) illustrates Case 3: $\boldsymbol{X}_0$ makes one face, which is not the IF, positive. In this situation, we switch from the IF to this newly identified positive face (PF) and repeat the same procedure as before. A new potential contact point $\boldsymbol{X}_1$ is identified by applying the procedure outlined in Section 2.2.2 to the plane which includes the PF (Figure 2.2(b)). Since the PF remains positive, the same conclusion of no contact is drawn, based on Equation (2.9b), as for Case 1.

Case 4 is the analogue of Case 3 for which there is a PH–SQ contact. This is shown in Figure 2.3. As usual, the IF is found initially, resulting in a potential contact point $\boldsymbol{X}_0$ (Figure 2.3(a)). Since $\boldsymbol{X}_0$ makes a face other than the IF positive, it cannot be the contact point but another point on the SQ surface may still be in contact with the PH. Attention is switched from the IF to the PF (Figure 2.3(b)).



(a) Positive and negative faces and potential contact point $\boldsymbol{X}_0$ based on the IF

(b) After switching attention to the PF

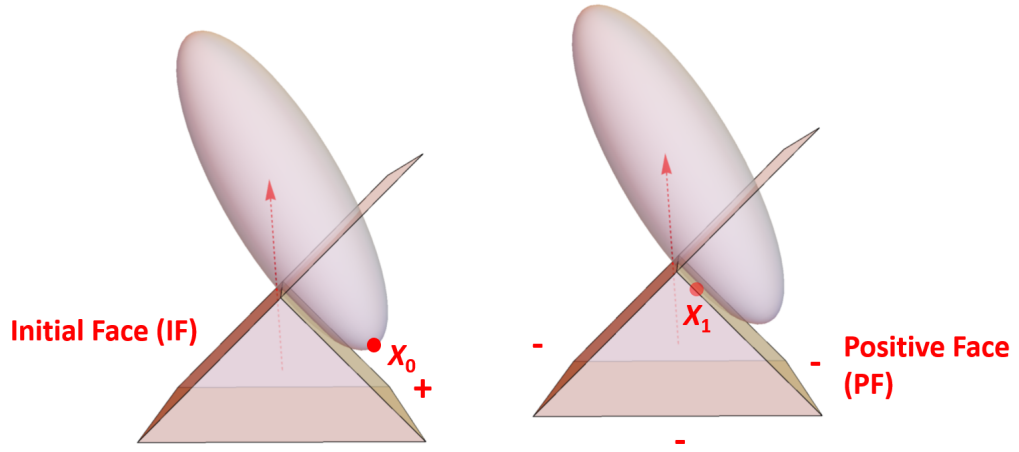
Fig. 2.2.: Non-contact of a PH and SQ determined by switching from the IF to the PF (Case 3).

The potential contact point $\boldsymbol{X}_1$ makes all PH faces negative. As concluded for Case 2, $\boldsymbol{X}_1$ is the contact point.

2.2.3.5. Cases 5 & 6: Negative IF, search and return for edge/vertex contact

In Cases 3 and 4, an unambiguous assessment of contact/non-contact could be made after switching from the initial face to the one positive face. However, one could imagine a scenario in which the contact status remains uncertain after switching to the one PF. For example, resolving a contact between a SQ and the edge or vertex of a PH, rather than its face, cannot be adequately resolved using Cases 1–4. Furthermore, Cases 3 and 4 are inapplicable if there is more than one positive face. In these situations, Cases 5 (non-contact) and 6 (contact) apply.

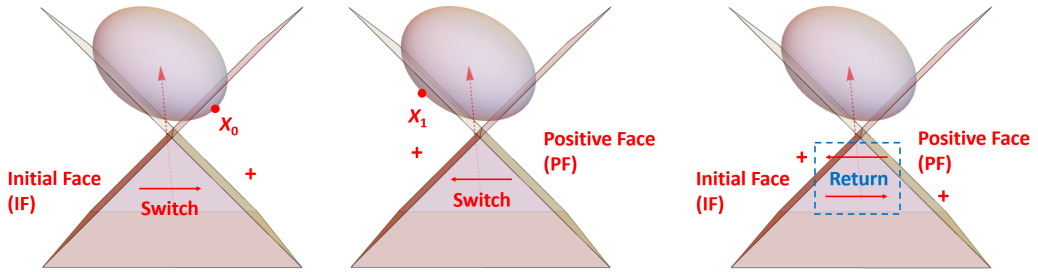
Consider Figure 2.4(a). As for Cases 3 and 4, a potential contact point $\boldsymbol{X}_0$ is found initially using the IF obtained by joining the centroids of the two shapes. Since this makes the rightmost face positive, another potential contact point $\boldsymbol{X}_1$ is found based on the PF (Figure 2.4(b)). However, $\boldsymbol{X}_1$ makes the PF negative but the IF positive, thus creating an endless alternating cycle between these two faces. This is resolved by checking whether the edge common to the two faces contacts the SQ (Figure 2.4(c)). Figure 2.5 is a similar scenario in which there is indeed a contact with the common edge. This is termed the 'search and return' (SR) method for contact determination. This also allows the detection of contacts between a SQ and a vertex of a PH: since a vertex belongs to multiple edges of a PH, contact can be assessed using any of the edges linking to it.



(a) Positive and negative faces and potential contact point $\boldsymbol{X}_0$ based on the IF

(b) Contact point $\boldsymbol{X}_1$ found after switching to the PF

Fig. 2.3.: Contact point $\boldsymbol{X}_1$ found after switching from the IF to the PF (Case 4).

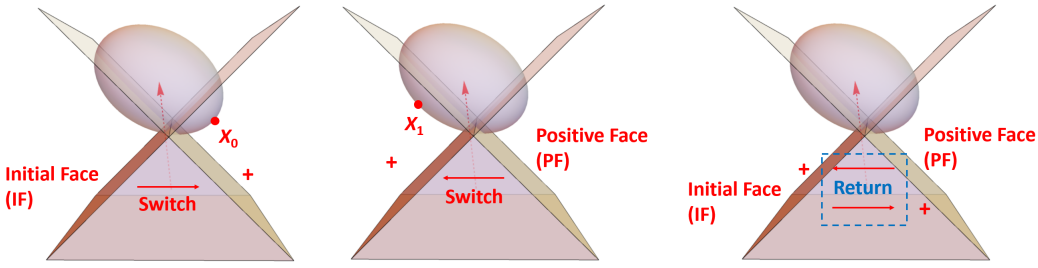


(a) First switch from IF to PF

(b) Second switch from PF back to IF

(c) Check the common edge

Fig. 2.4.: SR method for determining SQ contact with a PH edge/vertex (Case 5: non-contact).

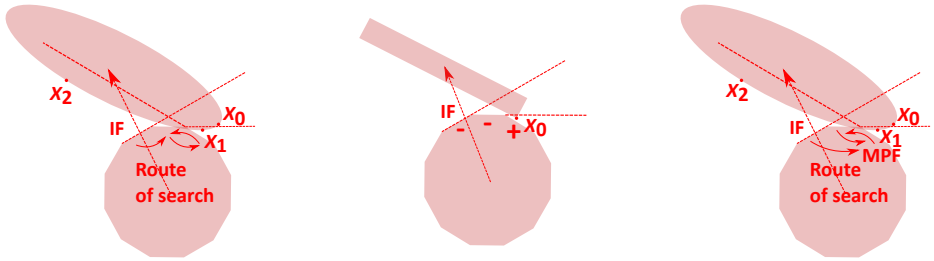


(a) First switch from IF to PF

(b) Second switch from PF back to IF

(c) Check the common edge

Fig. 2.5.: SR method for determining SQ contact with a PH edge/vertex (Case 6: contact found).



(a) Multiple searches before a return

(b) Positive face not adjacent to IF

(c) New search route for the amended SR method

Fig. 2.6.: Extreme PH–SQ contact situations which cannot be addressed using the simple SR method.

In Figures. 2.4 and 2.5, the return appears after only one switch from IF to another face. However, more iterations may be needed before a return appears, e.g., Figure 2.6(a). This is a limitation of the simple SR method. Furthermore, consider Figure 2.6(b) which shows a 2D projection of a cuboidal SQ contacting a many-faceted PH. Starting from the IF in the usual manner, both the IF and all of the faces surrounding it are negative. However, since one face of the PH remains positive, $\boldsymbol{X}_0$ cannot be definitively identified as the contact point. The SR method described above needs to be amended to account for this general situation.

This is simply done. As before, start from the IF and find $\boldsymbol{X}_0$. The check of plane function values with $\boldsymbol{X}_0$ is expanded to all PH faces, rather than those faces adjacent to the IF. The face with maximum positive value (termed the 'maximal positive face' or MPF) becomes the starting point for the next search. This amended initial search procedure is suitable for Figure 2.6(a), as demonstrated in Figure 2.6(c), and is also applicable to all other scenarios, including Figure 2.6(b). As the SQ surface is considered in finding the MPF, this amended SR method also improves efficiency by reducing the scope of the search to the area in which contact is most likely. Only neighbouring faces need to be checked in the following searches to find a return.

There is a subtle difference to note between contacts with the faces of PHs, e.g., Figure 2.1(b) or Figure 2.3, and contacts with the edges or vertices of PHs, e.g., Figure 2.5 or Figure 2.6. Face contacts yield a contact point on the surface of the SQ while edge/vertex contacts yield a contact point which is on an edge of the PH.

This algorithm is only applicable to a convex PH and SQ: the outer normal vector is used in the approach for PHs, while the SQs must be convex to avoid the possibility of multiple contacts as the algorithm presented in this section gives at most one contact point. Furthermore, the overlap between a PH and a SQ should be small

to ensure accuracy. This is because an identified contact point is always located on the surface of the PH or SQ; as overlaps become large, the validity of choosing a surface point as the single contact point becomes questionable.

The complete algorithm is summarised in Figure 2.7. The six cases leading to termination of the algorithm are shown in this flowchart. For odd cases, a contact is not found; for even cases, a contact is found.

2.2.4 Contact evaluation based on the contact point

If a contact exists, a single contact point $\boldsymbol{X}_c$ is found by following the algorithm in Figure 2.7. The next step is to determine the contact normal and the interparticle overlap, both of which are necessary to calculate a contact force. It is noted that these quantities are not well defined in general for irregularly shaped particles (Feng et al. (2017)); however, the approach proposed here gives physically reasonable values. Three cases must be considered separately: face contact, edge contact and vertex contact.

For face contact, the contact point lies on a single face of the PH which is therefore the contact plane. The normal vector of the contact plane is

$$\boldsymbol{n} = \boldsymbol{n}_c \tag{2.10}$$

where $\boldsymbol{n}_c$ is the normal vector of the PH face which contains the contact point $\boldsymbol{X}_c$. The interparticle overlap is the distance between $\boldsymbol{X}_c$ and the contact plane (Figure 2.8(a)). This method originates from Feng et al. (2012).

For edge contact, it would not be physically meaningful to adopt either of the PH faces forming the edge as the contact plane. Hence, a virtual contact plane is sought. Its normal vector is chosen to be the normalised sum of the normal vectors of the two faces forming the edge:

$$\boldsymbol{n} = \frac{\boldsymbol{n}_1 + \boldsymbol{n}_2}{|\boldsymbol{n}_1 + \boldsymbol{n}_2|} \tag{2.11}$$

where $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are the normal vectors of the two faces forming the edge. The contact plane passes through $\boldsymbol{X}_c$ with normal vector $\boldsymbol{n}$. The overlap is computed between the SQ and the contact plane (Figure 2.8(b), Podlozhnyuk et al. (2017)).

For vertex contact, a virtual contact plane is again required. Its normal vector is chosen to be the normalised sum of the direction vectors $\boldsymbol{l}$ of all n edges meeting at the vertex:

$$\boldsymbol{n} = \frac{\sum_{i=1}^{n} \boldsymbol{l}_i}{|\sum_{i=1}^{n} \boldsymbol{l}_i|} \tag{2.12}$$

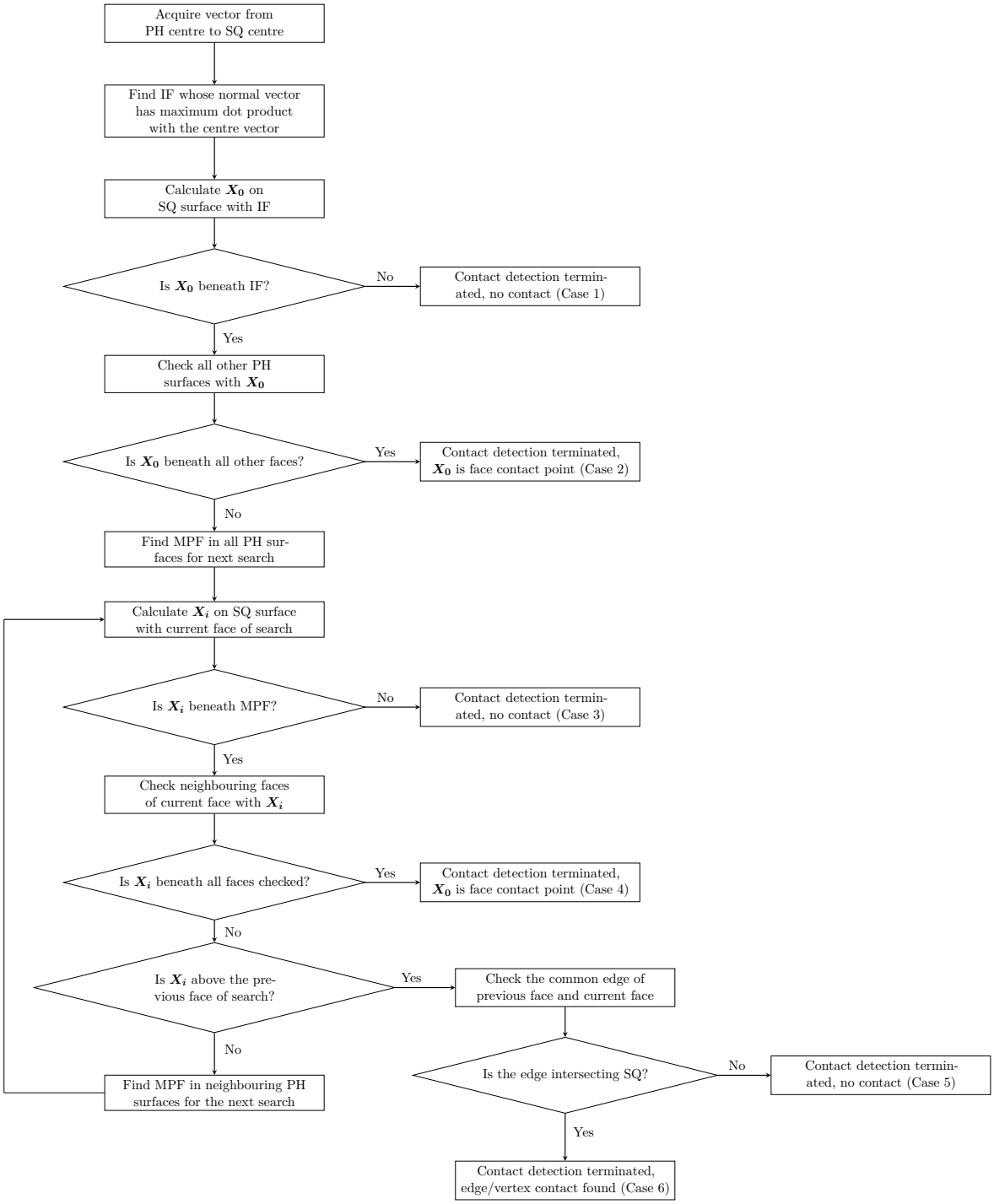
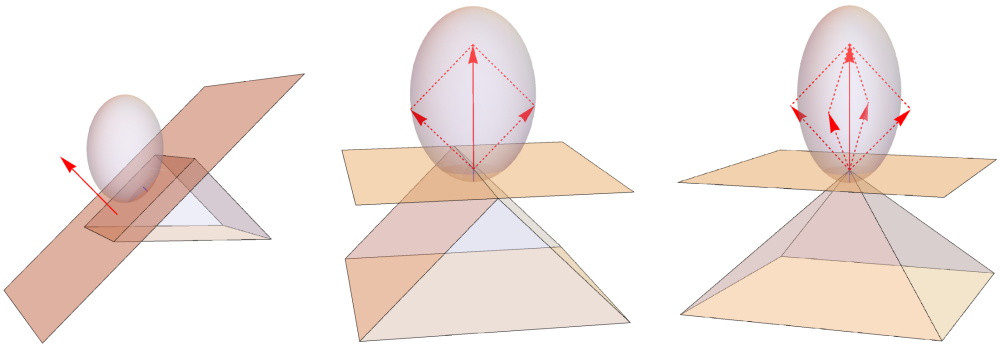


Fig. 2.7.: Flowchart showing the complete PH–SQ contact detection algorithm.



(a) For face contact (b) For edge contact (c) For vertex contact

Fig. 2.8.: Contact evaluation for three different contact cases.

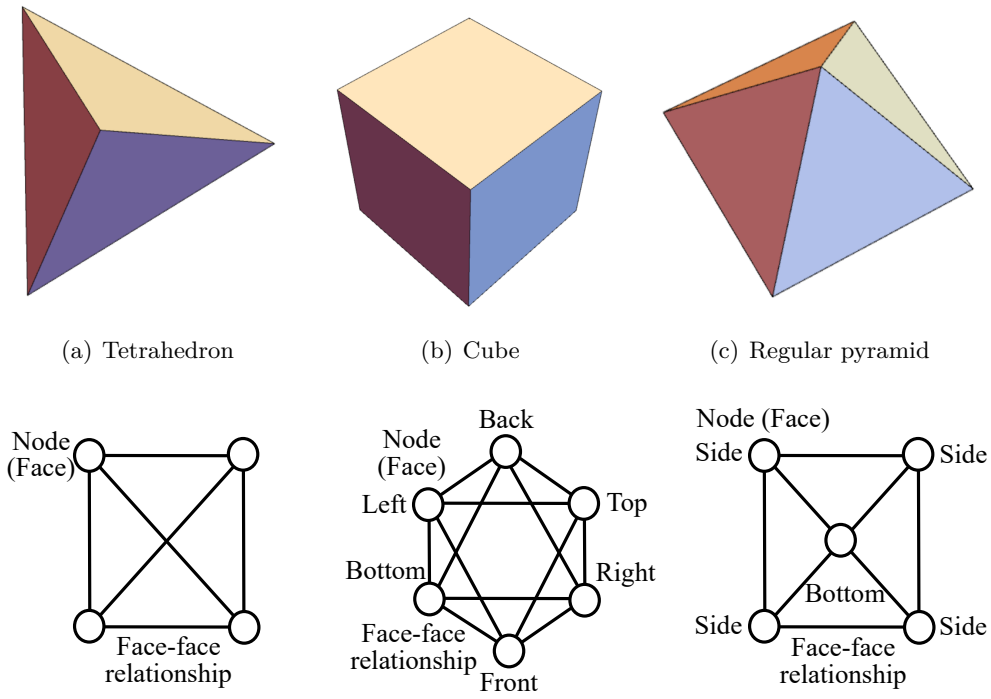
$\boldsymbol{l}$ always points towards the vertex contact. This notation emphasises the adoption of the direction vectors of edges rather than normal vectors of faces for the edge contact case. As for the edge contact case, the overlap is computed between the SQ and the contact plane (Figure 2.8(c), Podlozhnyuk et al. (2017)).

The contact evaluation approach stated here is a simple one which only takes the geometry of the PH into consideration. There can be a leap of contact normal from face contact to vertex contact when a SQ translates along a PH face towards the vertex of a PH. A more comprehensive approach that ensures energy conservation is needed for these cases (Feng et al. (2017)).

2.3 Implementation and verification of the contact detection algorithm

2.3.1 Implementation of algorithm

LIGGGHTS-PUBLIC v3.8.0 (*LIGGGHTS(R)-PUBLIC Documentation* (2016)) implements the contact detection approach described in Section 2.2.1 for SQ–SQ, SQ–planar wall and SQ–line segment contacts. The LIGGGHTS implementation was used as a basis for the code implementation described in this chapter. Since the algorithm requires knowledge of neighbouring faces, it is beneficial to build a relationship map of PH surface faces into the implementation. A graph model in which nodes are PH surface faces and links are relationships between these faces was adopted to satisfy this requirement. Two faces link only when they have a common edge, not if they only share a vertex. Figure 2.9 shows three simple examples of this graph model. All faces are linked to each other for a tetrahedron; for a cube, each face is linked to four other faces; for a regular pyramid, any side face links to



(a) Tetrahedron (b) Cube (c) Regular pyramid

(d) Tetrahedron model (e) Cube model (f) Regular pyramid model

Fig. 2.9.: Describing PHs using a graph model.

all other faces except the face directly opposite, while the square bottom face links to all faces.

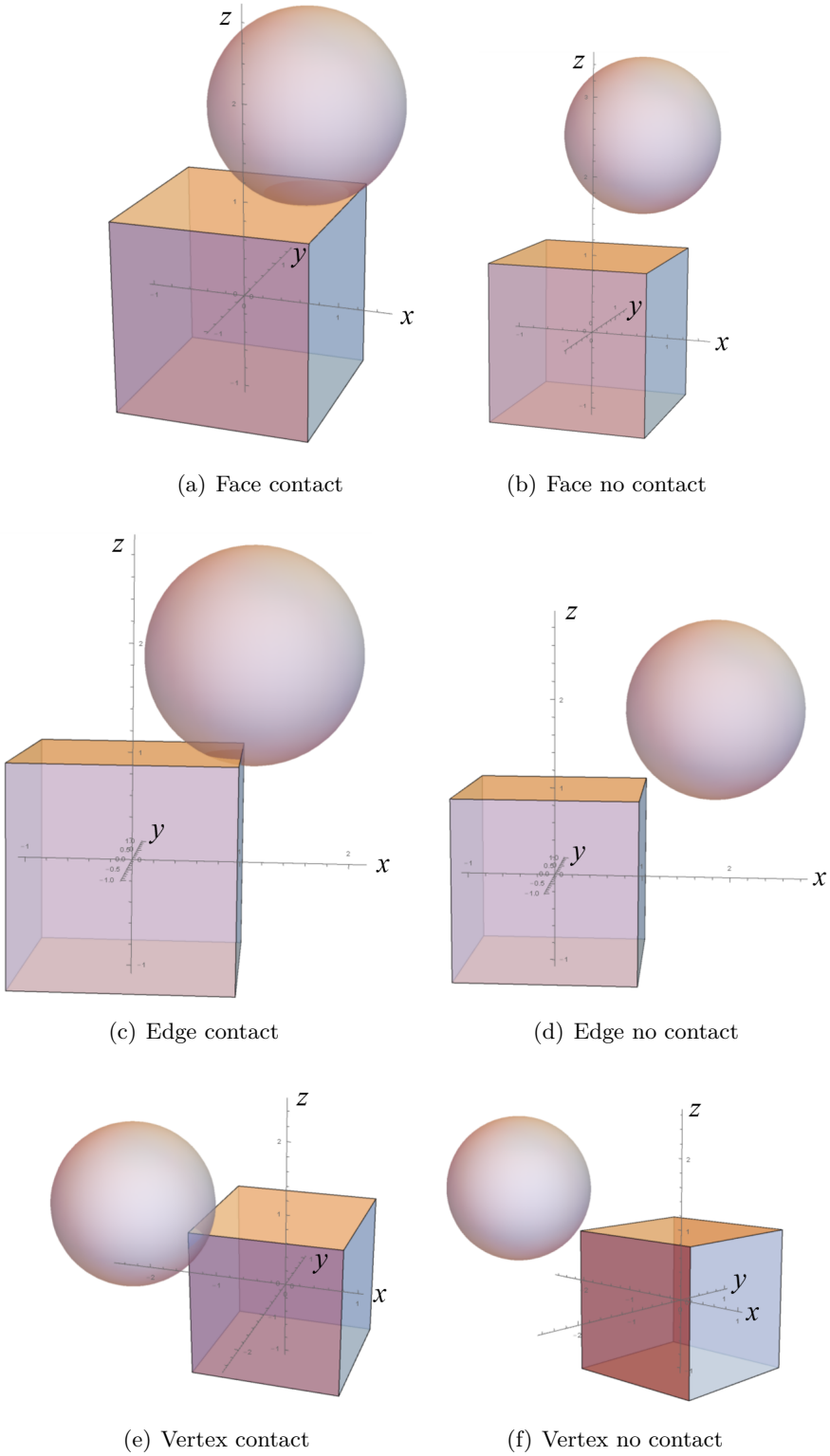
This PH model was implemented in C++ by creating a class which includes a set of planes enclosing the PH faces (stored using the outer normal vector and one specific point on each plane) and a 2D adjacency array to store the relationships between faces/planes. This is an easy way to model a static graph (Mehlhorn (2008)). Within this class, the vertices are stored in a disordered array list. The vertices are traversed only when edge/vertex contact judgement is needed. This traversal was not optimised to improve upon the time complexity of $O(n)$ (Sedgewick (1984)) for the naive implementation. However, as finding the first MPF has the time complexity of $O(n)$ and the number of faces involved is apparently lower than n after the first MPF is found, with proper topology implemented the time complexity of $O(n)$ can be reached for the algorithm. This class was combined with the SQ implementation from LIGGGHTS to create a standalone program for verifying one PH contacting one SQ. Note that the algorithm in this chapter has not been implemented into LIGGGHTS or any other discrete element code due to a lack of access to a code which contains both SQ and PH modelling capabilities. Implementing the missing PH capabilities into the version of LIGGGHTS used for this research would have taken a prohibitively long time and was not in line with the primary objective of this PhD study.

Tab. 2.1.: Program results for sphere-cube contact detection.

Simulations in Figure 2.10	Program Outputs	Exit code
(1a) Face contact	Contact point found! At: 0.5, 0.5, 0.9	2
(1b) Face no contact	Initial face does not intersect with SQ. No contact.	1
(2a) Edge contact	Contact point found! At: 1, -4.37013E-9, 1	6
(2b) Edge no contact	PH surface intersecting edge does not intersect with SQ. No contact.	5
(3a) Vertex contact	Contact point found! At: -1, -1, 1	6
(3b) Vertex no contact	PH surface intersecting edge does not intersect with SQ. No contact.	5

2.3.2 Verification of the algorithm

Initially, spheres, the simplest SQs, were placed in a wide range of contacting and non-contacting positions relative to a cube. Consider a sphere of radius 1 m (a special SQ) contacting a cube of 2 m side length in each Cartesian direction. The centre of cube is fixed at the origin, while the centre of sphere changes. Figure 2.10 shows three different contact types with both contact and non-contact, and Table 2.1 shows the outputs of the program for these three situations. For Figure 2.10(a), IF is the top face of cube, and the contact point found is the point that the sphere dips deepest into IF. Then the contact detection is terminated with a face contact point found. For Figure 2.10(b), IF remains the top face but the sphere is far from IF so no contact is found. For Figure 2.10(c), IF is still the top face and is X_0 =(1.1, 0.0, 0.9), which makes the right face of cube positive while all other faces are negative. Then MPF is the right face, based on which the second point X_1 =(0.1, 0.0, 1.9) is found. That makes the top face (IF) positive, from which a return is performed. Then the intersecting edge of two faces is checked and the deepest point into sphere (1.0, 0.0, 1.0) on the edge is found (in Table 2.1 $y = -4.37013 \times 10^{-9}$ which is effectively zero). For Figure 2.10(c), a similar procedure is followed but there is no contact between the edge and the sphere. For Figure 2.10(c), it is hard to determine IF as three faces (front, left and top) all have their normal vectors the maximum dot product with the vector joining the centres of the cube and sphere. However, no matter which face is chosen as IF, similar process of edge contact judgement will happen to any one within three intersecting edges of any two of these three faces. No matter which edge is chosen for judgement, the result is the same: contact point found at vertex (-1.0, -1.0, 1.0). That is why edge contact judgment can also solve vertex contact at the same time, which is mentioned in Section 2.2.3.5. For Figure 2.10(d), no matter which route is chosen, no contact is the outcome.



(a) Face contact (b) Face no contact

(c) Edge contact (d) Edge no contact

(e) Vertex contact (f) Vertex no contact

Fig. 2.10.: Three sphere-cube contact types.

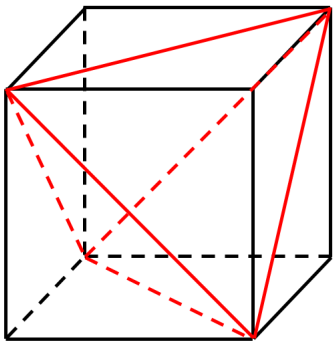
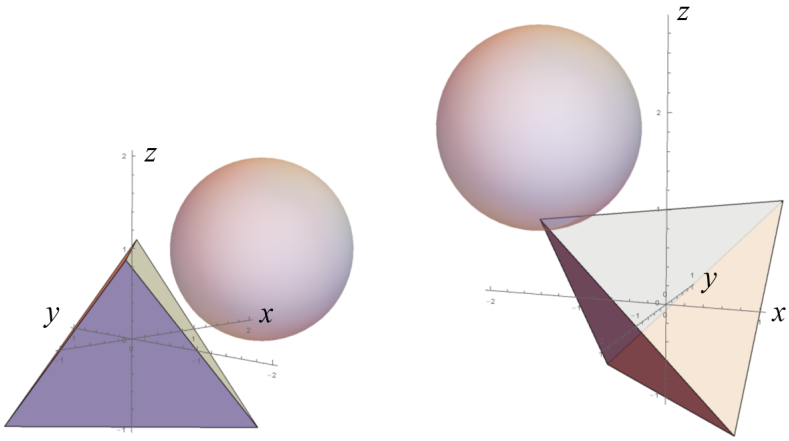


Fig. 2.11.: Tetrahedron in a cube.

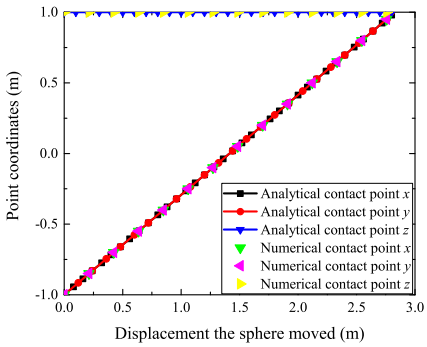
Another validation example is the same sphere contacting a tetrahedron. The tetrahedron can be generated within a cube, just as Figure 2.11. We first try a face non-contact situation, Figure 2.12(a). The output of the program is "Initial face does not intersect with SQ. No contact", as is shown in Figure 2.12(a). Then the sphere centre's z coordinate is fixed at $z = 1.9$, which is above the top edge of tetrahedron, while the sphere's centre is moved linearly from (-1, -1, 1.9) to (1, 1, 1.9), as shown in Figure 2.12(b). Both analytical results and validation program results are shown in Figure 2.12(c). In this situation, only edge contact happens and the deepest point of the top edge of tetrahedron in the sphere is always the one right under the sphere centre as the sphere moves its position. The validation program matches the analytical results.

Several other cases were considered in which the SQ is not a sphere. Figure 2.13(a) shows an ellipsoid contacting a vertex of cube, just slightly different from Figure 2.10(e). One of the shape parameters of SQ $c = 2$, while other two shape parameters $a = b = 1$. The IF can be set to any of the three potential faces without impacting the result of (-1, -1, 1), as shown in 2.13(a). The other example is a cylinder contacting an edge of the cube in Figure 2.13(b). Choosing $n_1 = 1000$ and $n_2 = 2$ produces a cylindrical SQ (*LIGGGHTS(R)-PUBLIC Documentation* (2016)). The IF can be selected as either the top or right face of the cube, and the final judgement will lie on their intersecting edge. Results of these two situations are shown in Table 2.2. The results show that the SR method can detect these different types of PH–SQ contact properly.

It is noted that all the verification cases are special cases of SQs (spheres, ellipsoids, cylinders). For arbitrary SQs, it can be very difficult to derive the analytical result for verification. As all the verifications in this chapter are quantitative to ensure accuracy, verifications for irregular cases of SQs are therefore not considered.

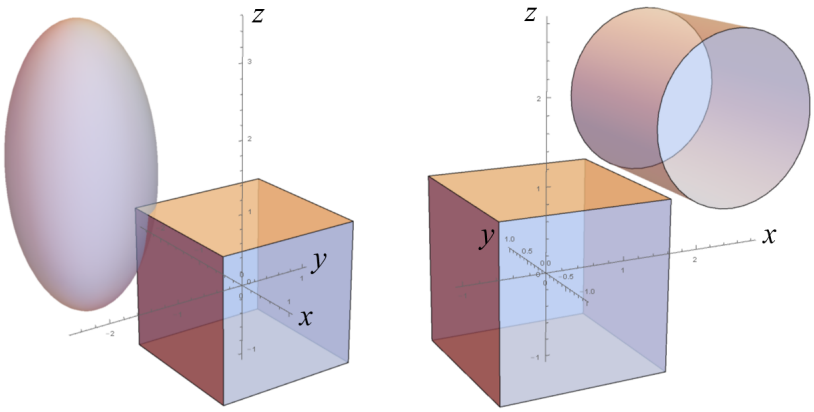


(a) Face no contact (b) Edge contact



(c) Edge contact results

Fig. 2.12.: Validation of sphere and tetrahedron contact.



(a) Ellipsoid contact a vertex of cube (b) Cylinder contact an edge of cube

Fig. 2.13.: Other validation examples with nonspherical SQ.

Tab. 2.2.: Program results for nonspherical SQ-cube contact detection.

Situations in Figure 10	Program Outputs	Exit code
(a) Ellipsoid contact a vertex of cube	Contact point found! At: -1, -1, 1	6
(b) Cylinder contact an edge of cube	PH surface intersecting edge does not intersect with SQ. No contact.	5

Tab. 2.3.: Comparison of the Monte Carlo verification with the analytical result based on volumes.

	Cases 1/3/5 (no contact)	Cases 2/4/6 (contact)	Case 8	Total
Monte Carlo count	745619	217499	36882	1000000
Monte Carlo (%)	74.56	21.75	3.69	100.00
Analytical volume (m^3)	160.962	47.038	8	216
Analytical %	74.52	21.78	3.70	100.00

After successfully carrying out this initial stage of verification, the Monte Carlo method was adopted to randomly position a spherical SQ of radius 1 m in a large cubical space (edge length 6 m) containing a smaller cubical PH of edge length 2 m. The centre of the SQ could be any point within the 6 m × 6 m × 6 m cubical space shown in pink in Figure 2.14. The cubical PH is shown in green in this figure. The centres of any spheres which contact the cube must lie within the round-cornered cubic space shown in yellow in Figure 2.14. 1000000 points were randomly generated within the pink cube as SQ centre points. This led to three possible outcomes for each point, with reference to Figure 2.14:

1. The centre was inside the green PH. These points were neither identified as contacts nor as non-contacts but were counted separately as 'Case 8'.

2. The centre was inside the yellow round-cornered cubic space but not inside the green PH. These are contact points, i.e., Cases 2, 4 or 6.

3. The centre was outside the yellow region, i.e., no contact (Cases 1, 3 or 5).

The code described in Section 2.3.1 was applied to each SQ centre point generated, leading to one of these three outcomes for each point. Analytically, the frequency with which these outcomes were observed must be proportional to the volumes of the three regions on Figure 2.14. $V_{green} = 2^3 = 8\,m^3$, $V_{yellow} \approx 47.038\,m^3$, and $V_{pink} = 6^3 - V_{green} - V_{yellow} \approx 160.962\,m^3$. This is exactly what was observed, as shown in Table 2.3. The difference between the analytical and Monte Carlo results is less than 0.05%, further demonstrating the reliability of this algorithm for resolving PH–SQ contact.

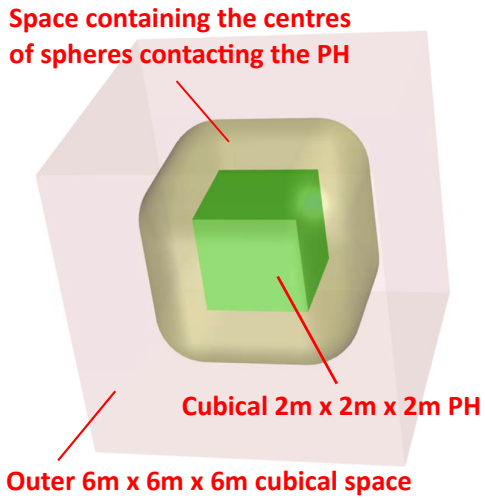


Fig. 2.14.: Setup for the Monte Carlo verification.

2.4 Computational efficiency of algorithm

After the code had been verified, it could be used to assess the computational efficiency of the contact detection algorithm. A similar Monte Carlo approach was used for this purpose with the centre points of 10000 spherical SQs being randomly located within the space shown on Figure 2.14. Instead of only a cubical PH, all five convex regular polyhedra were used for assessing computational efficiency: a tetrahedron, cube, octahedron, dodecahedron and icosahedron which have 4, 6, 8, 12 and 20 faces, respectively. Sphere centres within the PHs (Case 8) were disregarded for this benchmarking exercise.

Each determination of contact/non-contact was repeated 5000 times to ensure robust data were gathered for total elapsed time. Times were measured using the *ctime* library (*Programming languages – C (ISO/IEC 9899:1990)* (1990)); this was confirmed to give almost identical results to class *std::chrono::system_clock* (*Information technology – Programming languages – C++ (ISO/IEC 14882:2011)* (2011)). All benchmarks were run on a laptop computer with an Intel Core i7-6600U 2.60 GHz CPU.

The results are shown in Figure 2.15. The elapsed calculation times are categorised by case and accumulated curves are adopted for representation. It is noted that, for certain PHs, some cases cannot occur, e.g., Case 4 for a cube. For any PH, Cases 1 and 2 are clearly the fastest, then Cases 3 and 4 where present, and finally Cases 5 and 6 are the slowest by a significant margin, which is expected from Figure 2.7. The odd, non-contact cases are generally slightly faster than the corresponding

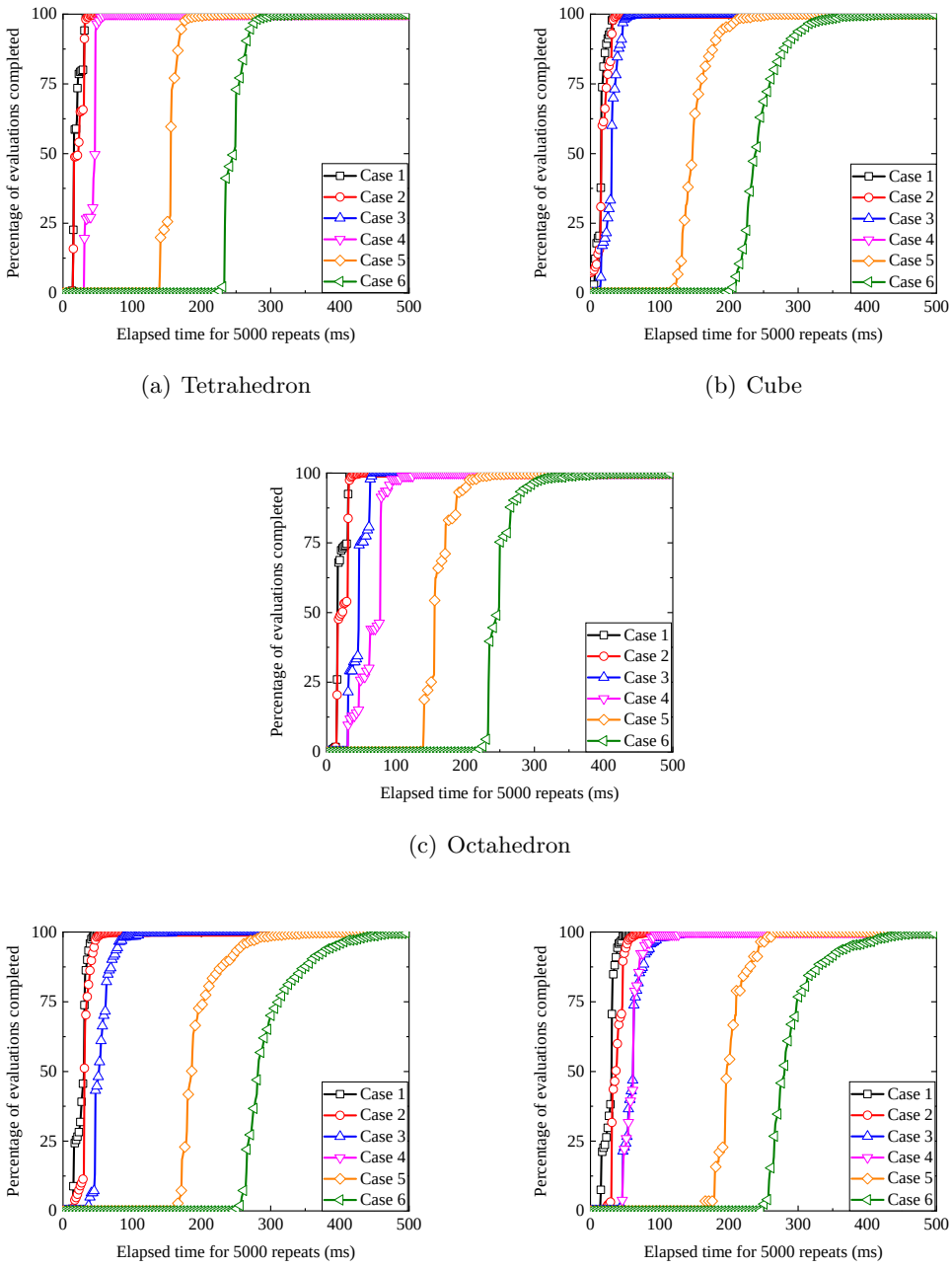
contact cases, as termination of the algorithm occurs sooner for the non-contact cases than for the contact cases.

The effect of increasing the number of PH faces (and also edges and vertices) is perhaps not as pronounced as might be expected. Comparing the graphs for the five different polyhedra on Figure 2.15, there is a small time penalty associated with increasing the number of faces but this is modest. This algorithm is therefore particularly efficient for polyhedra with a large number of faces.

2.5 Conclusions

This chapter presents a novel contact detection algorithm for a convex polyhedron (PH) and superquadric (SQ) in DEM. This algorithm is based on contact detection between a SQ and a planar wall, and terminates in one of six cases for each contact event: three cases in which the PH and the SQ are in contact, and three in which they are not. This contact detection algorithm has been implemented as a standalone program, based on the existing SQ implementation in LIGGGHTS (*LIGGGHTS(R)-PUBLIC Documentation* (2016)).

This program was used to verify the correctness of the algorithm and evaluate its efficiency using the Monte Carlo method. The evaluation time largely depends on the point at which the algorithm terminates. Non-contact cases are generally slightly faster than the equivalent contact cases. The effect of increasing the number of PH faces on the evaluation time is small, indicating that the proposed algorithm is particularly efficient for many-faceted PHs.



(a) Tetrahedron

(b) Cube

(c) Octahedron

(d) Dodecahedron

(e) Icosahedron

Fig. 2.15.: Evaluation times, subdivided by case, for different types of PH.

3

Critical time step for discrete element method simulations of nonspherical particles[1]

3.1 Introduction

The discrete element method (DEM) was proposed by Cundall and Strack (1979) for modelling systems of particles and their interactions. The particles in DEM are modelled as rigid bodies which are allowed to overlap when they come into contact, from which interparticle forces are calculated using a contact model. Particle motions can be described as a system of second-order differential equations which require solution using a suitable numerical integration scheme. Typically an explicit, conditionally stable integration scheme of second-order accuracy is chosen (Rougier et al. (2004)). Stability is contingent on choosing a sufficiently small time step. However, as reducing the time step increases the computational cost of a simulation, it is desirable that the chosen time step is not unnecessarily small. This is an important consideration since DEM simulations often require considerable computational effort: a fact which has motivated the adoption of high-performance computing and the development of parallelised codes.

DEM often uses disks (2D) or spheres (3D) as the shape of element, again to reduce the computational cost. When a linear contact model is adopted, the critical time step has a $\sqrt{\frac{m}{k}}$ dependency where m and k respectively represent some measure of particle mass and contact stiffness. Cundall and Strack (1979) estimated a time step with this dependency for a single particle of mass m connected to the ground by a spring of stiffness k. Belytschko (1983) proposed modal decomposition of the equations of motion to obtain a simple system with a single degree of freedom for explicit finite element analyses. The eigenvalues of the amplification matrix for this system cannot exceed 1 in magnitude (Wood (1990)) which gives the maximum stable time step. O'Sullivan and Bray (2004) drew an analogy between particles in a DEM simulation and nodes of a finite element mesh. Based on this, they found critical time steps for regular packings of uniform disks and spheres. Even though

[1]This chapter originates from D Peng, SJ Burns and KJ Hanley, Critical time step for DEM simulations of convex particles with central symmetry, International Journal for Numerical Methods in Engineering, 2020 (under review).

these approaches are not strictly applicable to nonlinear systems, they have nonetheless been extended to systems of disks/spheres with nonlinear contact models (Tsuji et al. (1993), Tu and Andrade (2008) and Rojek et al. (2016)) and rotational resistance (Huang et al. (2017)). Otsubo et al. (2017) estimated the critical time step using a nonlinear contact model based on eigenvalue decomposition of the global mass and stiffness matrices for idealised and randomly packed assemblies of spheres. Tavarez and Plesha (2007) constructed mass and stiffness matrices for a DEM unit cell of monosized spheres to obtain an upper-bound estimate of the critical time step. Burns et al. (2019) developed a framework for selecting a stable time step for both linear and nonlinear interactions of spheres by analysing the equations of motion as a nonlinear map. The most common approach to estimate a critical time step for nonlinear contact between spheres is fundamentally based on the propagation of Rayleigh waves through a particle assembly, e.g., Thornton and Randall (1988) and Li et al. (2005).

These prior studies on the critical time step considered disks or spheres. However, advances in computational power have made it increasingly feasible to run DEM simulations using particle shapes other than disks or spheres. Lu et al. (2015) discuss the increased scientific interest in modelling systems of nonspherical particles and the various methods being used to achieve this, e.g., ellipsoids, superquadrics (a generalisation of ellipsoids), polyhedra or rigid clusters composed of multiple spheres. At present, time steps for nonspherical particle systems are often found by trial and error though some limited progress has been made to rigorously establish critical time steps. Hart et al. (1988) conservatively recommended a critical time step proportional to $\sqrt{\frac{m}{k}}$ for the block-based 3DEC code, taking m as the smallest particle mass and k as the largest contact stiffness in the system. The commercial DEM code PFC (*PFC - Particle Flow Code, Version 6.0* (2018)) automatically estimates the critical time step based on two constraints: one using contact stiffness matrices under the assumption that the degrees of freedom decouple from each other; the other a kinematic constraint which depends on the maximum components of translational velocity and acceleration among all particles in the system. The critical time step is taken as the minimum value (*PFC - Particle Flow Code, Version 6.0* (2018)). This methodology is applied for all particle shapes including spheres, multi-sphere clusters and polyhedra. However, as the maximum components of translational velocity and acceleration cannot be known before analysis, the critical time step is recomputed during the simulation based on the continually changing particle velocities and accelerations which "may be relatively expensive" (*PFC - Particle Flow Code, Version 6.0* (2018)). Burns and Hanley (2017) obtained critical time steps proportional to $\sqrt{\frac{m}{k}}$ for any planar rigid body subject to linear damping and forcing. A limitation of this analysis is that particle rotations have one degree of freedom, within the contact plane, which does not represent all possible particle kinematics.

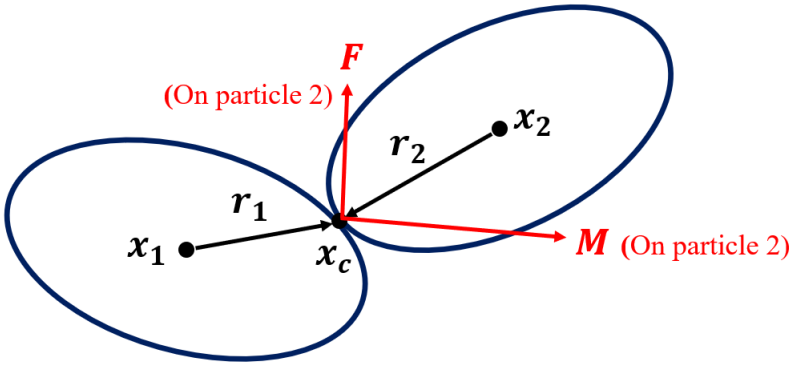


Fig. 3.1.: Illustration of two nonspherical particles in contact

In this chapter, upon this previous work a similar criterion is performed to establish critical time steps for a subset of nonspherical particles including ellipsoids, convex superquadrics, and polyhedra which are both convex and central symmetric. Two analyses are compared: one based on the application of Belytschko's criterion to the characteristic equation at the interparticle contact and the other based on the eigenvalues of the amplification matrix. Both assume a linear contact model and damping.

3.2 Relative acceleration between two contacting particles

Consider two arbitrarily shaped particles with centres of mass $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, contacting at $\boldsymbol{x}_c$ as shown in Figure 3.1. Force $\boldsymbol{F}$ and moment $\boldsymbol{M}$ act on particle 2. The global moment on each particle can be expressed for any arbitrary global coordinate system as

$$\boldsymbol{r}_1 = \boldsymbol{x}_c - \boldsymbol{x}_1 \tag{3.1}$$

$$\boldsymbol{r}_2 = \boldsymbol{x}_c - \boldsymbol{x}_2 \tag{3.2}$$

$$\boldsymbol{M}_{G1} = \boldsymbol{r}_1 \times (-\boldsymbol{F}) + (-\boldsymbol{M}) \tag{3.3}$$

$$\boldsymbol{M}_{G2} = \boldsymbol{r}_2 \times \boldsymbol{F} + \boldsymbol{M} \tag{3.4}$$

Moments on each particle may be converted from a global to a local coordinate system using rotation matrices:

$$\boldsymbol{M}_{L1} = \boldsymbol{R}_1 \boldsymbol{M}_{G1} \tag{3.5}$$

$$\boldsymbol{M}_{L2} = \boldsymbol{R}_2 \boldsymbol{M}_{G2} \tag{3.6}$$

where $\boldsymbol{M}_{Gj} = [M_{Gjx}, M_{Gjy}, M_{Gjz}]^T$ and $\boldsymbol{M}_{Lj} = [M_{Ljx}, M_{Ljy}, M_{Ljz}]^T$, $j = 1, 2$. The rotation matrices can be written in terms of unit quaternions:

$$\boldsymbol{R}_1 = \begin{bmatrix} q_{w1}^2 + q_{x1}^2 - q_{y1}^2 - q_{z1}^2 & 2q_{x1}q_{y1} - 2q_{w1}q_{z1} & 2q_{x1}q_{z1} + 2q_{w1}q_{y1} \\ 2q_{x1}q_{y1} + 2q_{w1}q_{z1} & q_{w1}^2 - q_{x1}^2 + q_{y1}^2 - q_{z1}^2 & 2q_{y1}q_{z1} - 2q_{w1}q_{x1} \\ 2q_{x1}q_{z1} - 2q_{w1}q_{y1} & 2q_{y1}q_{z1} + 2q_{w1}q_{x1} & q_{w1}^2 - q_{x1}^2 - q_{y1}^2 + q_{z1}^2 \end{bmatrix} \tag{3.7}$$

$$\boldsymbol{R}_2 = \begin{bmatrix} q_{w2}^2 + q_{x2}^2 - q_{y2}^2 - q_{z2}^2 & 2q_{x2}q_{y2} - 2q_{w2}q_{z2} & 2q_{x2}q_{z2} + 2q_{w2}q_{y2} \\ 2q_{x2}q_{y2} + 2q_{w2}q_{z2} & q_{w2}^2 - q_{x2}^2 + q_{y2}^2 - q_{z2}^2 & 2q_{y2}q_{z2} - 2q_{w2}q_{x2} \\ 2q_{x2}q_{z2} - 2q_{w2}q_{y2} & 2q_{y2}q_{z2} + 2q_{w2}q_{x2} & q_{w2}^2 - q_{x2}^2 - q_{y2}^2 + q_{z2}^2 \end{bmatrix} \tag{3.8}$$

Suppose the moments of inertia of these two particles are:

$$\boldsymbol{I}_1 = \begin{bmatrix} I_{xx1} & I_{xy1} & I_{xz1} \\ I_{yx1} & I_{yy1} & I_{yz1} \\ I_{zx1} & I_{zy1} & I_{zz1} \end{bmatrix} \tag{3.9}$$

$$\boldsymbol{I}_2 = \begin{bmatrix} I_{xx2} & I_{xy2} & I_{xz2} \\ I_{yx2} & I_{yy2} & I_{yz2} \\ I_{zx2} & I_{zy2} & I_{zz2} \end{bmatrix} \tag{3.10}$$

These can be computed knowing each particle's geometry. We define a, b, c to be the half-lengths in the x, y, z directions of an axis-aligned bounding box around any particle in its local coordinate system. The local rotational accelerations of the particles are

$$\ddot{\boldsymbol{\theta}}_{L1} = \boldsymbol{I}_1^{-1}(\boldsymbol{M}_{L1} - \boldsymbol{\omega}_{L1} \times \boldsymbol{I}_1\boldsymbol{\omega}_{L1}) \tag{3.11}$$

$$\ddot{\boldsymbol{\theta}}_{L2} = \boldsymbol{I}_2^{-1}(\boldsymbol{M}_{L2} - \boldsymbol{\omega}_{L2} \times \boldsymbol{I}_2\boldsymbol{\omega}_{L2}) \tag{3.12}$$

where $\boldsymbol{\omega}_{L1} = \dot{\boldsymbol{\theta}}_{L1}$ and $\boldsymbol{\omega}_{L2} = \dot{\boldsymbol{\theta}}_{L2}$ are the rotational velocities of the two particles in their local coordinate systems. These can be returned to the global coordinate system using the rotation matrices:

$$\ddot{\boldsymbol{\theta}}_{G1} = \boldsymbol{R}_1^{-1}\ddot{\boldsymbol{\theta}}_{L1} \tag{3.13}$$

$$\ddot{\boldsymbol{\theta}}_{G2} = \boldsymbol{R}_2^{-1}\ddot{\boldsymbol{\theta}}_{L2} \tag{3.14}$$

$$\boldsymbol{\omega}_{G1} = \boldsymbol{R}_1^{-1}\boldsymbol{\omega}_{L1} = [\omega_{x1}, \omega_{y1}, \omega_{z1}]^T \tag{3.15}$$

$$\boldsymbol{\omega}_{G2} = \boldsymbol{R}_2^{-1}\boldsymbol{\omega}_{L2} = [\omega_{x2}, \omega_{y2}, \omega_{z2}]^T \tag{3.16}$$

The translational accelerations of the two particles at the contact point are

$$\ddot{\boldsymbol{x}}_{c1} = \ddot{\boldsymbol{\theta}}_{G1} \times \boldsymbol{r}_1 + \boldsymbol{\omega}_{G1} \times (\boldsymbol{\omega}_{G1} \times \boldsymbol{r}_1) - \frac{\boldsymbol{F}}{m_1} \tag{3.17}$$

$$\ddot{\boldsymbol{x}}_{c2} = \ddot{\boldsymbol{\theta}}_{G2} \times \boldsymbol{r}_2 + \boldsymbol{\omega}_{G2} \times (\boldsymbol{\omega}_{G2} \times \boldsymbol{r}_2) + \frac{\boldsymbol{F}}{m_2} \tag{3.18}$$

Finally, the relative translational and rotational accelerations at the contact may be computed:

$$\ddot{\boldsymbol{x}}_c = \ddot{\boldsymbol{x}}_{c2} - \ddot{\boldsymbol{x}}_{c1} \tag{3.19}$$

$$\ddot{\boldsymbol{\theta}}_c = \ddot{\boldsymbol{\theta}}_{G2} - \ddot{\boldsymbol{\theta}}_{G1} \tag{3.20}$$

3.3 Worst case scenario for deriving maximum relative acceleration at the contact

3.3.1 Basic assumptions

In the commercial PFC code (*PFC - Particle Flow Code, Version 6.0* (2018)), the kinematic time step constraint, which in addition to a stiffness constraint determines the simulation time step, is based on the maximum components of translational velocity and acceleration among all particles in the system. For this two-particle case, a similar assumption is made that the maximum relative translational acceleration at the contact, i.e., the maximum magnitude of $\ddot{\boldsymbol{x}}_c$, determines the critical time step. $\ddot{\boldsymbol{x}}_c$ appears in the differential equation describing the system's dynamics at the contact:

$$\boldsymbol{M}\ddot{\boldsymbol{x}}_c + \boldsymbol{C}\dot{\boldsymbol{x}}_c + \boldsymbol{K}\boldsymbol{x}_c = \boldsymbol{F} \tag{3.21}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ refer to the particle mass, damping and contact stiffness matrices, respectively (Burns et al. (2019)).

The magnitude of $\ddot{\boldsymbol{x}}_c$ depends on up to 18 independent parameters for moments of inertia, as shown by Equations (3.9) and (3.10). In order to find a practical solution, several additional assumptions are made to constrain the analysis:

1. Only a single contact is possible between two particles;

2. Each particle is central symmetric with a local coordinate system coinciding with the principal axes of moment of inertia, i.e. $I_{ij} = 0$ when $i \neq j$;

3. $a \geq b \geq c$ and $I_{zz} \geq I_{yy} \geq I_{xx}$ (the former ensuring the latter for most shapes).

Therefore, each particle has three non-zero parameters defining its moment of inertia: I_{xx}, I_{yy}, I_{zz}. These assumptions restrict the applicability of this analysis to a subset of shapes including ellipsoids, convex superquadrics, and some polyhedra

(e.g., regular octahedra) which are both convex and central symmetric. It is also noted that this analysis is limited to a single contact per particle; multiple simultaneous contacts require the critical time step to be reduced from the two-particle estimate derived in this chapter (Otsubo et al. (2017)).

3.3.2 Numerical study to find out the worst case scenario

The location of the contact point $\boldsymbol{x}_c$ on the surface of particle 1 and the orientation of particle 2 relative to particle 1 both affect $\ddot{\boldsymbol{x}}_c$. It is worth investigating in which situations the $\ddot{\boldsymbol{x}}_c$ gets its largest magnitude. Consider the same two ellipsoids as in Section 3.7, both with unit density. The centre of ellipsoid 1 is fixed at the origin $(0, 0, 0)$ with an axis-aligned orientation $(q_x = q_y = q_z = 0, q_w = 1)$. The contact force and rotational velocities of the two particles give a total of 9 unique parameters. Treating each of these parameters independently would make the analysis infeasible. Thus the number of parameters is reduced, i.e., the analysis is constrained, by fixing the contact force at $(1, 1, 1)$ and setting the particle rotational velocities to zero for both particles. Overlaps are restricted to negligible values. The configuration of the contacting ellipsoid 2 is altered in two different ways:

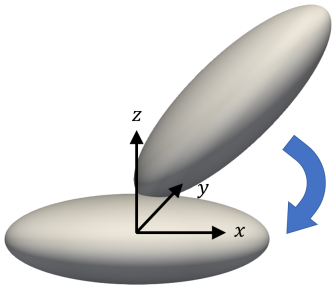
Change centre of ellipsoid 2 Move the centre of ellipsoid 2 without changing its fixed orientation, giving a range of contact points on the surface of ellipsoid 1

Change orientation of ellipsoid 2 Change the orientation of ellipsoid 2 without changing the contact point between the ellipsoids

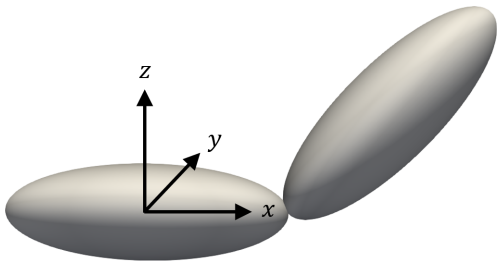
3.3.2.1. Change centre of ellipsoid 2

The orientation of ellipsoid 2 is fixed, with a rotation of 45° around the minus y axis ($q_x = q_z = 0$, $q_y = -\sin(\frac{\pi}{8})$, $q_w = \cos(\frac{\pi}{8})$, Gaschler (2016)). Its centre is moved within Octant I $(x, y, z > 0)$ as shown in Figure 3.2(a) to generate contact points covering one-eighth of the surface of ellipsoid 1. Two scenarios are considered as depicted in Figures 3.2(a) and 3.2(c). The only difference between these scenarios is a 90° rotation of ellipsoid 2 around its local x axis.

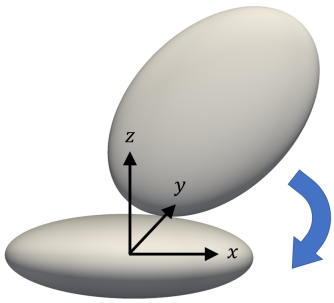
Figure 3.3 shows the partial surface of ellipsoid 1, coloured by the magnitude of relative translational acceleration at the contact. For both scenarios, the maximum acceleration occurs at the longitudinal tip of ellipsoid 1. The relative positions of the ellipsoids when $\ddot{\boldsymbol{x}}_c$ is a maximum is shown in Figures 3.2(b) and 3.2(d) for scenarios 1 and 2, respectively.



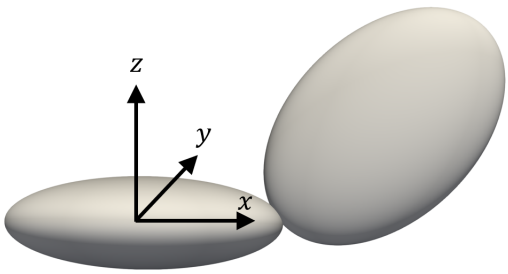
(a) Sketch of ellipsoid 2 changing position (Scenario 1)



(b) Configuration for maximum $\ddot{\boldsymbol{x}}_c$ (Scenario 1)

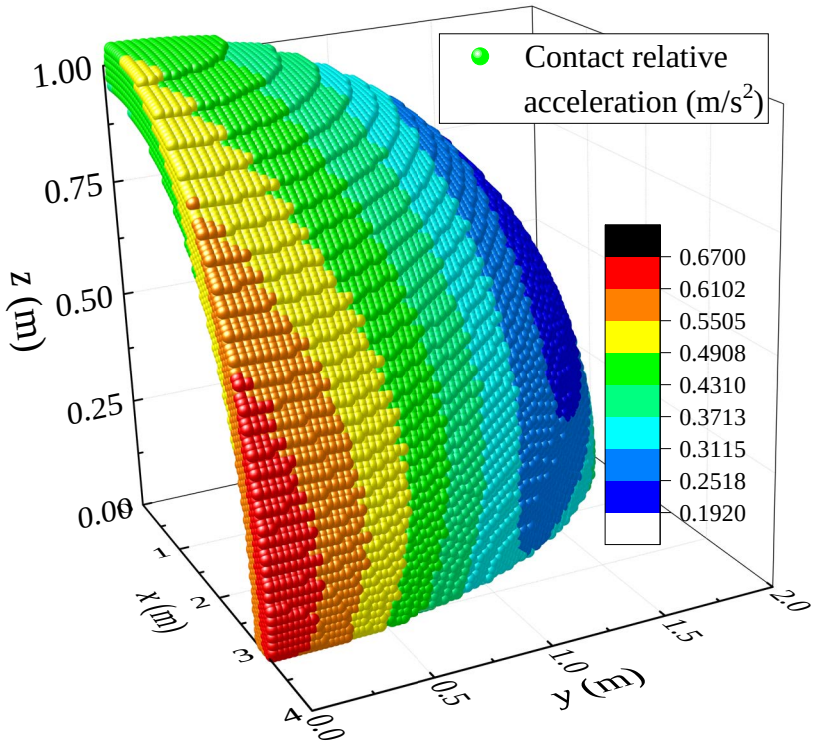


(c) Sketch of ellipsoid 2 changing position (Scenario 2)

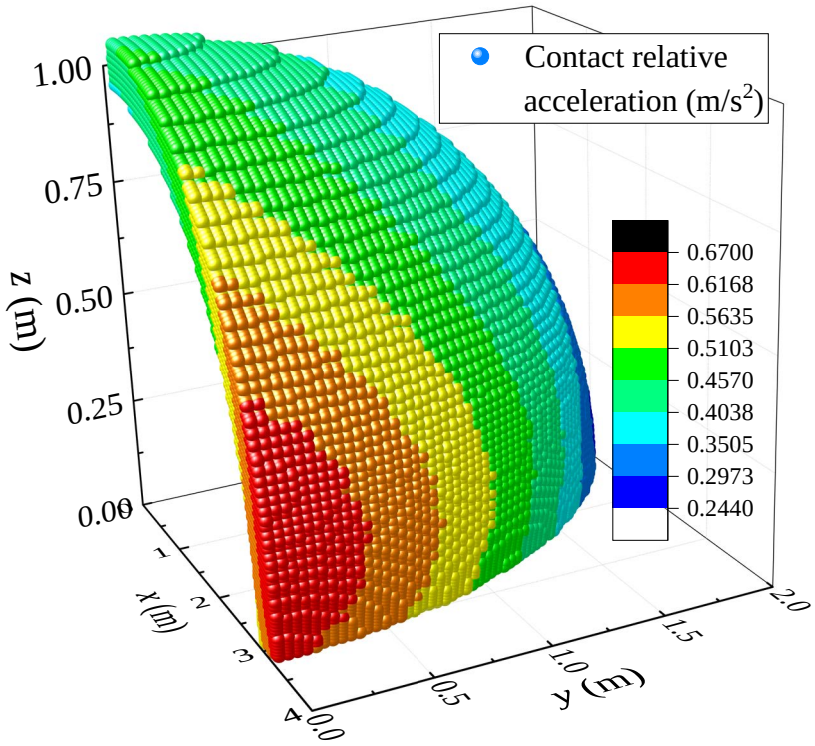


(d) Configuration for maximum $\ddot{\boldsymbol{x}}_c$ (Scenario 2)

Fig. 3.2.: Illustration of the two scenarios for the centre of ellipsoid 2 moving relative to ellipsoid 1 (a & c), and the configuration of the two ellipsoids when the magnitude of relative translational acceleration at the contact is a maximum (b & d)



(a) Scenario in Figure 3.2(a)



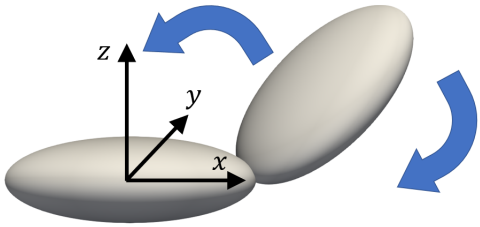
(b) Scenario in Figure 3.2(c)

Fig. 3.3.: Partial surface of ellipsoid 1, coloured by the magnitude of relative translational acceleration at the contact as the centre of ellipsoid 2 is moved

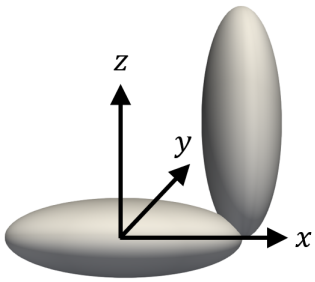
3.3.2.2. Change orientation of ellipsoid 2

The contact point is fixed, within a small tolerance, at $(3, 0, 0)$: the tip of ellipsoid 1 at which $\ddot{\boldsymbol{x}}_c$ is a maximum. The orientation of ellipsoid 2 is changed by moving its centre point within Octant I as indicated in Figures 3.4(a) and 3.4(c). The same two scenarios are considered as before, differing only by a 90° rotation of ellipsoid 2 around its local x axis.

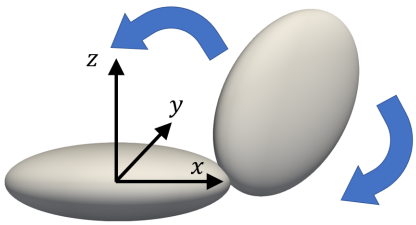
Figure 3.5 shows the surface formed from the centre points of ellipsoid 2, coloured as before by the magnitude of $\ddot{\boldsymbol{x}}_c$. For both scenarios, the magnitude of $\ddot{\boldsymbol{x}}_c$ approaches a similar maximum value at three distinct orientations. Two of these orientations, shown in red at the left and bottom of Figure 3.5, correspond to non-orthogonal contact configurations. The third, at $z = 3\,\mathrm{m}$, is an orthogonal configuration which is more convenient for analysis than the non-orthogonal configurations. Therefore, the orthogonal configurations shown in Figures 3.4(b) and 3.4(d), for scenarios 1 and 2, respectively, are identified as those giving the maximum magnitude of relative translational acceleration at the contact.



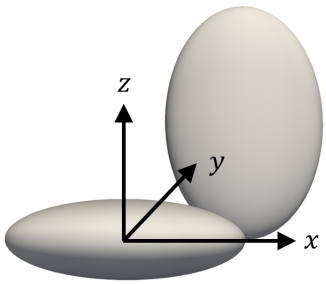
(a) Sketch of ellipsoid 2 changing orientation (Scenario 1)



(b) Configuration for maximum $\ddot{x}_c$ (Scenario 1)

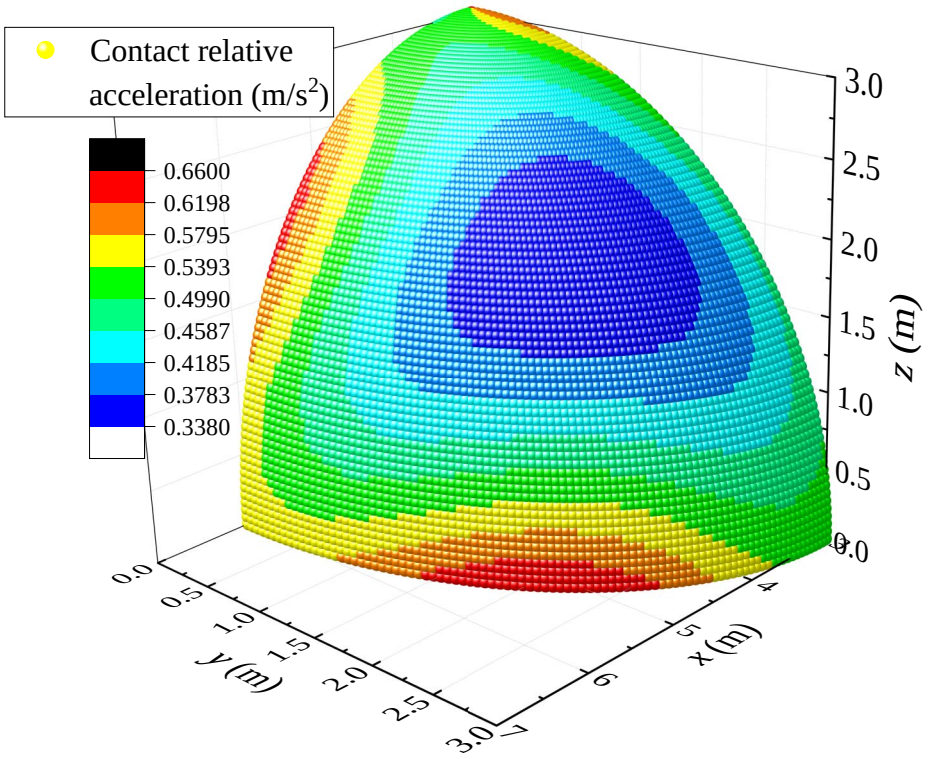


(c) Sketch of ellipsoid 2 changing orientation (Scenario 2)

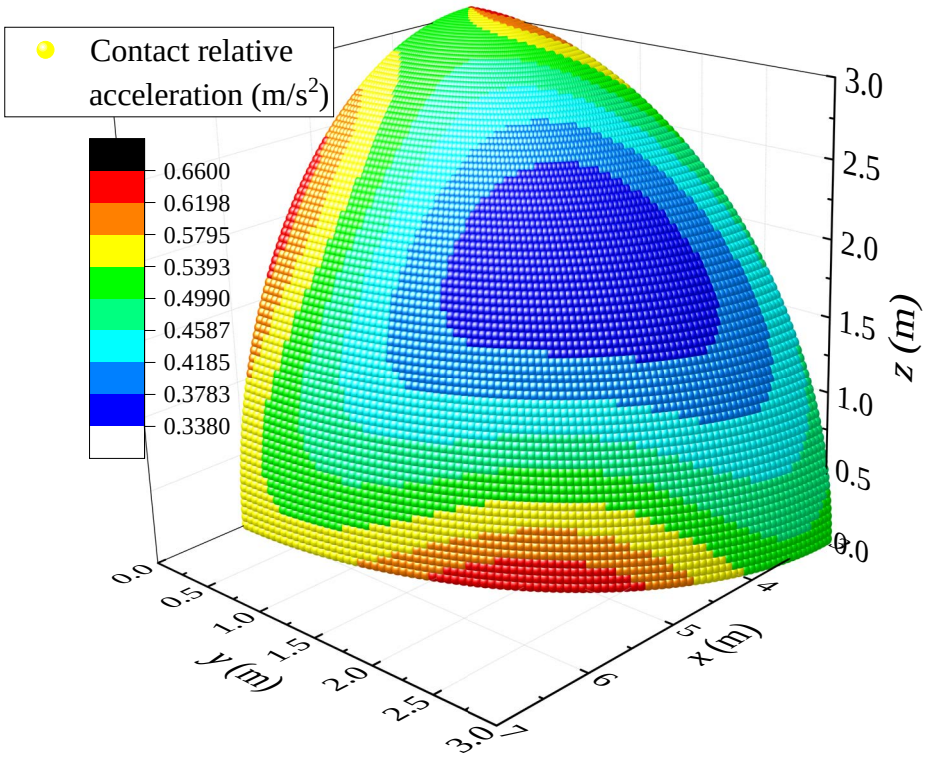


(d) Configuration for maximum $\ddot{x}_c$ (Scenario 2)

Fig. 3.4.: Illustration of the two scenarios in which the orientation of ellipsoid 2, and hence the position of its centre, are changed relative to ellipsoid 1 (a & c), and the configuration of the two ellipsoids when the magnitude of relative translational acceleration at the contact is a maximum (b & d)



(a) Scenario in Figure 3.4(a)



(b) Scenario in Figure 3.4(c)

Fig. 3.5.: Surface formed by the centre points of ellipsoid 2 as its orientation is changed, coloured by the magnitude of relative translational acceleration at the contact

3.4 Maximum relative translational acceleration at the contact

According to the analysis in Section 3.3, the worst case scenario for the maximum $\ddot{\boldsymbol{x}}_c$ is when the two particles are in orthogonal tip-to-tip contact. Figure 3.6 shows three of the infinitely many configurations of particles 1 and 2 in orthogonal tip-to-tip contact, in which the two ellipsoids both have semi axis lengths of a =3 m, b =2 m and c =1 m, respectively from major to minor. The expressions for the relative translational acceleration at the contact are shown as Equations (3.22)-(3.24) for Figures 3.6(a)-(c), respectively, obtained by combining Equations (3.1)-(3.19):

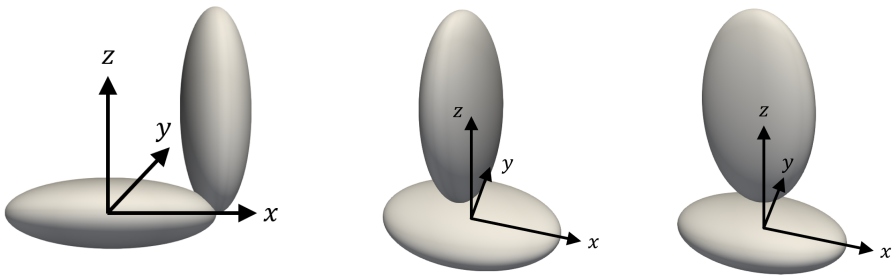
$$\ddot{\boldsymbol{x}}_c = \begin{bmatrix} \frac{a_2^2}{I_{yy2}} + \frac{1}{m_1} + \frac{1}{m_2} & & \\ 0 & \frac{a_1^2}{I_{zz1}} + \frac{a_2^2}{I_{zz2}} + \frac{1}{m_1} + \frac{1}{m_2} & 0 \\ 0 & 0 & \frac{a_1^2}{I_{yy1}} + \frac{1}{m_1} + \frac{1}{m_2} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} a_1(\omega_{y1}^2 + \omega_{z1}^2) + \frac{a_2\omega_{x2}\omega_{z2}(I_{zz2}-I_{xx2}-I_{yy2})}{I_{yy2}} \\ \frac{a_1\omega_{x1}\omega_{y1}(I_{yy1}-I_{zz1}-I_{xx1})}{I_{zz1}} + \frac{a_2\omega_{y2}\omega_{z2}(I_{yy2}-I_{zz2}-I_{xx2})}{I_{zz2}} \\ a_2(\omega_{x2}^2 + \omega_{y2}^2) + \frac{a_1\omega_{x1}\omega_{z1}(I_{zz1}-I_{xx1}-I_{yy1})}{I_{yy1}} \end{bmatrix} \tag{3.22}$$

$$\ddot{\boldsymbol{x}}_c = \begin{bmatrix} \frac{c_1^2}{I_{yy1}} + \frac{a_2^2}{I_{yy2}} + \frac{1}{m_1} + \frac{1}{m_2} & 0 & 0 \\ 0 & \frac{c_1^2}{I_{xx1}} + \frac{a_2^2}{I_{zz2}} + \frac{1}{m_1} + \frac{1}{m_2} & 0 \\ 0 & 0 & \frac{1}{m_1} + \frac{1}{m_2} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} \frac{c_1\omega_{x1}\omega_{z1}(I_{xx1}-I_{yy1}-I_{zz1})}{I_{yy1}} + \frac{a_2\omega_{x2}\omega_{z2}(I_{zz2}-I_{xx2}-I_{yy2})}{I_{yy2}} \\ \frac{c_1\omega_{y1}\omega_{z1}(I_{yy1}-I_{zz1}-I_{xx1})}{I_{xx1}} + \frac{a_2\omega_{y2}\omega_{z2}(I_{yy2}-I_{zz2}-I_{xx2})}{I_{zz2}} \\ c_1(\omega_{x1}^2 + \omega_{y1}^2) + a_2(\omega_{x2}^2 + \omega_{y2}^2) \end{bmatrix} \tag{3.23}$$

$$\ddot{\boldsymbol{x}}_c = \begin{bmatrix} \frac{a_2^2}{I_{xx2}} + \frac{c_1^2}{I_{yy1}} + \frac{1}{m_1} + \frac{1}{m_2} & 0 & 0 \\ 0 & \frac{c_1^2}{I_{xx1}} + \frac{a_2^2}{I_{zz2}} + \frac{1}{m_1} + \frac{1}{m_2} & 0 \\ 0 & 0 & \frac{1}{m_1} + \frac{1}{m_2} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} \frac{c_1\omega_{x1}\omega_{z1}(I_{xx1}-I_{yy1}-I_{zz1})}{I_{yy1}} + \frac{a_2\omega_{x2}\omega_{z2}(I_{zz2}-I_{xx2}-I_{yy2})}{I_{xx2}} \\ \frac{c_1\omega_{y1}\omega_{z1}(I_{yy1}-I_{zz1}-I_{xx1})}{I_{xx1}} + \frac{a_2\omega_{y2}\omega_{z2}(I_{xx2}-I_{yy2}-I_{zz2})}{I_{zz2}} \\ c_1(\omega_{x1}^2 + \omega_{y1}^2) + a_2(\omega_{x2}^2 + \omega_{y2}^2) \end{bmatrix} \tag{3.24}$$

We can write a generic expression for $\ddot{\boldsymbol{x}}_c$ as

$$\ddot{\boldsymbol{x}}_c = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \tag{3.25}$$



(a) (b) (c)

Fig. 3.6.: Three examples showing particle 2 changing its position and orientation relative to particle 1

where A_x, A_y, A_z are comprehensive inverse inertia coefficients and $\Omega_x, \Omega_y, \Omega_z$ are terms related to the particles' rotational velocities. With the assumption that $a \geq b \geq c$ and $I_{zz} \geq I_{yy} \geq I_{xx}$, it can be stated that

$$A_{x/y/z} \leq A_{max} = \frac{a_1^2}{I_{xx1}} + \frac{a_2^2}{I_{xx2}} + \frac{1}{m_1} + \frac{1}{m_2} \tag{3.26}$$

Equation (3.25) shows that, for nonspherical particles, the relative translational acceleration at the contact is influenced by two factors: the contact force, which is related to the interparticle overlap, and particle rotation. We denote these two terms as the 'translation-dominant' and 'rotation-dominant' terms which redwill be analysed separately, i.e., critical time steps are obtained by neglecting either the 'rotation-dominant' or 'translation-dominant' terms in Sections 3.5 and 3.6, respectively. In practice, the translation-dominant time step is several orders of magnitude smaller than the rotation-dominant time step in most scenarios and hence determines the critical time step.

3.5 Critical time step for translation-dominant scenarios

If the rotation-dominant term is neglected, i.e., $\Omega_x, \Omega_y, \Omega_z = 0$,

$$\ddot{\boldsymbol{x}}_c = A_{max}\boldsymbol{F} \tag{3.27}$$

3.5.1 Belytschko's criterion

For an explicit finite element method analysis, Belytschko (1983) expressed the critical time step as

$$\Delta t_{cri} = \frac{2}{\phi_{max}} \tag{3.28}$$

for a linear system with Rayleigh damping where ϕ_{max} is the largest natural angular frequency of the system. The linear contact model is given by Equation (3.29) in which k_j and c_j are the contact stiffness and viscous damping coefficient, respectively, in the j^{th} direction:

$$F_j = -k_j x_{cj} - c_j \dot{x}_{cj} \quad (j = 1, 2, 3) \tag{3.29}$$

Combining Equations (3.27) and (3.29), the ordinary differential equation of the contact point is

$$\ddot{x}_{cj} + c_j A_{max} \dot{x}_{cj} + k_j A_{max} x_{cj} = 0 \quad (j = 1, 2, 3) \tag{3.30}$$

which has the characteristic equation

$$s^2 + c_j A_{max} s + k_j A_{max} = 0 \tag{3.31}$$

The roots of the characteristic equation are

$$s = \frac{1}{2}\left(-c_j A_{max} \pm \sqrt{c_j^2 A_{max}^2 - 4k_j A_{max}}\right) \tag{3.32}$$

The overdamped case, for which $c_j^2 A_{max}^2 - 4k_j A_{max} > 0$, does not have a natural angular frequency. Hence Δt_{cri} cannot be derived from Belytschko's criterion. The underdamped case, for which $c_j^2 A_{max}^2 - 4k_j A_{max} < 0$, has a maximum natural angular frequency of (Chopra (2012))

$$\phi_{max} = \frac{\sqrt{4k_j A_{max} - c_j^2 A_{max}^2}}{2} \tag{3.33}$$

According to Equation (3.28), the critical time step

$$\Delta t_{cri} = \frac{2}{\phi_{max}} = \frac{4}{\sqrt{4k_j A_{max} - c_j^2 A_{max}^2}} \tag{3.34}$$

In the undamped case, $c_j = 0$ so

$$\Delta t_{cri} = \frac{2}{\sqrt{k_j A_{max}}} \tag{3.35}$$

Equation (3.34) has some similarities to the collision time between particles reported by Silbert et al. (2001), although that has no dependence on the particle shape:

$$\Delta t_{cri} = \frac{\pi}{\sqrt{\frac{2k_n}{m} - \frac{c_n^2}{4m^2}}} \tag{3.36}$$

A fraction of this collision time has been adopted in other studies as a suitable simulation time step.

The critical damping is obtained by setting $c_j^2 A_{max}^2 - 4k_j A_{max} = 0$:

$$c_{j,cri} = 2\sqrt{\frac{k_j}{A_{max}}} \tag{3.37}$$

Since Belytschko's criterion (Equation (3.28)) requires the system to have a natural angular frequency, it is applicable only to the underdamped case. The normal and shear damping coefficients need to be maintained below the critical value (Equation (3.37)) to make this approach valid: a significant limitation. In Section 3.5.2, an alternative is proposed which does not have this limitation, i.e., is applicable irrespective of the choice of damping coefficients.

3.5.2 Amplification matrix method

Linearity is a requirement of the amplification matrix method (Burns and Hanley (2017)) by which the critical time step can be derived from the eigenvalues of the amplification matrix. The linear contact model adopted is shown as Equation (3.29). An Euler integrator is adopted which gives two equations:

$$\frac{\boldsymbol{x}_{c,n+1} - \boldsymbol{x}_{c,n}}{\Delta t} = \dot{\boldsymbol{x}}_{c,n} \tag{3.38}$$

$$\frac{\dot{\boldsymbol{x}}_{c,n+1} - \dot{\boldsymbol{x}}_{c,n}}{\Delta t} = \ddot{\boldsymbol{x}}_{c,n} \tag{3.39}$$

n and $n+1$ denote successive time steps. Combining this integrator with the linear contact model results in Equation (3.40) where $K_j = -k_j A_{max}$ and $C_j = -c_j A_{max}$, $j = 1, 2, 3$:

$$\begin{bmatrix} x_{cj,n+1} \\ \dot{x}_{cj,n+1} \\ \ddot{x}_{cj,n+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ K_j & C_j & 0 \end{bmatrix} \begin{bmatrix} x_{cj,n} \\ \dot{x}_{cj,n} \\ \ddot{x}_{cj,n} \end{bmatrix} \tag{3.40}$$

Thus the amplification matrix is 3×3. Its eigenvalues can be obtained analytically using the characteristic polynomial given by Equation (3.41). It is noted that the commonly used velocity Verlet integrator (Verlet (1967)) and its variants would result in a 5×5 amplification matrix. In addition, the Euler and velocity Verlet integrators give identical critical time steps for undamped systems, which is discussed in Appendix A.

$$\begin{bmatrix} 1-\lambda & \Delta t & 0 \\ 0 & 1-\lambda & \Delta t \\ K_j & C_j & -\lambda \end{bmatrix} = 0 \tag{3.41}$$

The three eigenvalues are

$$\lambda = \begin{cases} \dfrac{D}{3\sqrt[3]{2}} + \dfrac{\sqrt[3]{2}(3C_j\Delta t + 1)}{3D} + \dfrac{2}{3} \\ \dfrac{\left(\sqrt{3}i - 1\right) D}{6\sqrt[3]{2}} - \dfrac{\left(1 + \sqrt{3}i\right)(3C_j\Delta t + 1)}{3 \times 2^{2/3} D} + \dfrac{2}{3} \\ \dfrac{\left(-\sqrt{3}i - 1\right) D}{6\sqrt[3]{2}} - \dfrac{\left(1 - \sqrt{3}i\right)(3C_j\Delta t + 1)}{3 \times 2^{2/3} D} + \dfrac{2}{3} \end{cases} \tag{3.42}$$

where

$$D = \sqrt[3]{\sqrt{\left(-9C_j\Delta t + 27K_j\Delta t^2 - 2\right)^2 - 4(3C_j\Delta t + 1)^3} - 9C_j\Delta t + 27K_j\Delta t^2 - 2} \tag{3.43}$$

$|\lambda| < 1$ is required for stability, with the critical time step being found from $|\lambda| = 1$. Considering only the real λ,

$$\frac{D}{3\sqrt[3]{2}} + \frac{\sqrt[3]{2}(3C_j\Delta t + 1)}{3D} + \frac{2}{3} = \pm 1 \tag{3.44}$$

which has two solutions for D:
$D = \frac{\sqrt[3]{2}}{2}\left(-5 \pm \sqrt{21 - 12C_j\Delta t}\right)$ or $D = \frac{\sqrt[3]{2}}{2}\left(-1 \pm \sqrt{-3 - 12C_j\Delta t}\right)$. Equating these solutions to Equation (3.43) gives two meaningful values for the time step:

$$\Delta t = \frac{C_j \pm \sqrt{C_j^2 - 4K_j}}{K_j} \tag{3.45}$$

A third solution, $\Delta t = -\frac{1}{3C_j}$, is disregarded since this gives $D = 0$ on substitution into Equation (3.43).

Since $C_j, K_j < 0$, $\frac{C_j+\sqrt{C_j^2-4K_j}}{K_j} < 0$. Therefore, the critical time step may be expressed as

$$\Delta t_{cri} = \frac{C_j - \sqrt{C_j^2 - 4K_j}}{K_j} = \frac{\sqrt{4k_j A_{max} + c_j^2 A_{max}^2} + c_j A_{max}}{k_j A_{max}} \tag{3.46}$$

In the undamped case, $C_j = 0$ so

$$\Delta t_{cri} = \sqrt{-\frac{4}{K_j}} = \frac{2}{\sqrt{k_j A_{max}}} \tag{3.47}$$

which matches the critical time step from Belytschko's criterion in Equation (3.35).

For physical realism (Thornton et al. (2011)), the normal contact stiffness $k_n \geq k_s$, the shear contact stiffness. Conservatively taking the larger,

$$\Delta t_{cri} = \frac{2}{\sqrt{k_n A_{max}}} \tag{3.48}$$

for the undamped case. Even though Equation (3.48) contains k_n, the shear direction is usually the one which limits the critical time step as implied by Equations (3.22)-(3.24). This is consistent with the findings of Tu and Andrade (2008) and Burns et al. (2019), among others.

3.5.3 Special case of undamped spheres

For two identical spheres of radius r and mass m, $a_1 = a_2 = r$, $I_{xx1} = I_{xx2} = \frac{2}{5}mr^2$ so $A_{max} = \frac{7}{m}$. Substituting into Equation (3.48), the critical time step for spheres can be obtained as

$$\Delta t_{cri,sphere} = \frac{2}{\sqrt{7}}\sqrt{\frac{m}{k_n}} \tag{3.49}$$

Thus the expected result of $\alpha\sqrt{\frac{m}{k}}$ is recovered for a two-sphere interaction with a linear contact model. Our α of $\frac{2}{\sqrt{7}}$ is considerably more conservative than $\alpha = \sqrt{2}$ found by Burns et al. (2019) for the same two-sphere configuration. A large factor of safety must be applied for multi-contact situations comprising more than two spheres (Otsubo et al. (2017)); for multiple contacts, a wide range of α values have been recommended, e.g., 0.17 (O'Sullivan and Bray (2004)), 0.5 (Tavarez and Plesha (2007)) or 1.27 (Tu and Andrade (2008)).

3.6 Critical time step for rotation-dominant scenarios

One could envisage a scenario in which particles have large initial rotational velocities, or particles acquire high rotational velocities via moments induced by fluid motion, e.g., through coupling with computational fluid dynamics (CFD). If the translation-dominant term in Equation (3.25) is neglected, using Equations (3.1)-(3.18) and Equation (3.20) the following expressions are obtained for Figures 3.6(a)-(c), respectively:

$$\ddot{\boldsymbol{\theta}}_c = \begin{bmatrix} -\frac{\omega_{y1}\omega_{z1}(I_{yy1}-I_{zz1})}{I_{xx1}} - \frac{\omega_{x2}\omega_{y2}(I_{xx2}-I_{yy2})}{I_{zz2}} \\ \frac{\omega_{x1}\omega_{z1}(I_{xx1}-I_{zz1})}{I_{yy1}} + \frac{\omega_{x2}\omega_{z2}(I_{xx2}-I_{zz2})}{I_{yy2}} \\ -\frac{\omega_{x1}\omega_{y1}(I_{xx1}-I_{yy1})}{I_{zz1}} - \frac{\omega_{y2}\omega_{z2}(I_{yy2}-I_{zz2})}{I_{xx2}} \end{bmatrix} \tag{3.50}$$

$$\ddot{\boldsymbol{\theta}}_c = \begin{bmatrix} -\frac{\omega_{y1}\omega_{z1}(I_{yy1}-I_{zz1})}{I_{xx1}} - \frac{\omega_{x2}\omega_{y2}(I_{xx2}-I_{yy2})}{I_{zz2}} \\ \frac{\omega_{x1}\omega_{z1}(I_{xx1}-I_{zz1})}{I_{yy1}} + \frac{\omega_{x2}\omega_{z2}(I_{xx2}-I_{zz2})}{I_{yy2}} \\ -\frac{\omega_{x1}\omega_{y1}(I_{xx1}-I_{yy1})}{I_{zz1}} - \frac{\omega_{y2}\omega_{z2}(I_{yy2}-I_{zz2})}{I_{xx2}} \end{bmatrix} \tag{3.51}$$

$$\ddot{\boldsymbol{\theta}}_c = \begin{bmatrix} -\frac{\omega_{y1}\omega_{z1}(I_{yy1}-I_{zz1})}{I_{xx1}} + \frac{\omega_{x2}\omega_{y2}(I_{xx2}-I_{yy2})}{I_{zz2}} \\ \frac{\omega_{x1}\omega_{z1}(I_{xx1}-I_{zz1})}{I_{yy1}} + \frac{\omega_{y2}\omega_{z2}(I_{yy2}-I_{zz2})}{I_{xx2}} \\ -\frac{\omega_{x1}\omega_{y1}(I_{xx1}-I_{yy1})}{I_{zz1}} - \frac{\omega_{x2}\omega_{z2}(I_{xx2}-I_{zz2})}{I_{yy2}} \end{bmatrix} \tag{3.52}$$

Defining the maximum magnitude of particle rotational velocity ω_{max} as $\max(|\boldsymbol{\omega}_G|)$,

$$|\ddot{\theta}_{cx/y/z}| \leq \omega_{max}^2(B_{max1} + B_{max2}) \tag{3.53}$$

where

$$B_{max} = \max\left(\frac{|I_{xx} - I_{yy}|}{I_{zz}}, \frac{|I_{xx} - I_{zz}|}{I_{yy}}, \frac{|I_{yy} - I_{zz}|}{I_{xx}}\right) \tag{3.54}$$

In addition, note that

$$\max(|\ddot{\theta}_{cx/y/z}|) = \max\left(\frac{|\omega_{2x/y/z} - \omega_{1x/y/z}|}{\Delta t_{cri}}\right) = \frac{2\omega_{max}}{\Delta t_{cri}} \tag{3.55}$$

Combining Equations (3.53) and (3.55),

$$\Delta t_{cri} = \frac{2}{\omega_{max}(B_{max1} + B_{max2})} \tag{3.56}$$

For system containing particles of different types, B_{max} can be the maximum obtainable, leading to the following rotation-dominant critical time step:

$$\Delta t_{cri} = \frac{1}{\omega_{max} B_{max}} \tag{3.57}$$

This expression requires estimation or evaluation of the maximum particle angular velocity during a simulation.

3.7 Numerics

As the translation-dominant time step is generally orders of magnitude smaller than the rotation-dominant time step, that is the focus of these numerics. However, as an illustration, Section 3.7.4 compares the translation-dominant and rotation-dominant time steps for a study of particle aspect ratio.

3.7.1 Damping coefficient

Three equations are presented in Sections 3.5.1 and 3.5.2 to determine a critical time step when viscous damping is valid:

1. Equation (3.34) derived based on Belytschko's criterion. Applicable only to underdamped systems; considers particle shape in its formulation

2. Equation (3.36) from Silbert et al. (2001). Applicable only to underdamped systems; no consideration of particle shape

3. Equation (3.46) derived based on the amplification matrix method. Applicable to both underdamped and overdamped systems; considers particle shape in its formulation

To enable a fair comparison across a range of damping coefficients, consider the spherical particle case in Section 3.5.3 where $A_{max} = \frac{7}{m}$. Suppose $k_n = k_s = k$ and $c_n = c_s = c$. We define a normalised dimensionless number $\beta = \frac{c}{\sqrt{mk}}$ to quantify the amount of damping present. The three equations listed above can be rewritten for identical spherical particles in terms of β as follows:

$$\Delta t_{cri}\sqrt{\frac{k}{m}} = \frac{4}{\sqrt{28 - 49\beta^2}} \tag{3.58}$$

$$\Delta t_{cri}\sqrt{\frac{k}{m}} = \frac{\pi}{\sqrt{2 - \frac{\beta^2}{4}}} \tag{3.59}$$

$$\Delta t_{cri}\sqrt{\frac{k}{m}} = \beta + \sqrt{\beta^2 + \frac{4}{7}} \tag{3.60}$$

for Equations (3.34), (3.36) and (3.46), respectively. $\Delta t_{cri}\sqrt{\frac{k}{m}}$ is dimensionless. Figure 3.7 shows the evolution of $\Delta t_{cri}\sqrt{\frac{k}{m}}$ with β. The curves for Equation (3.58), derived based on Belytschko's criterion (Belytschko (1983)), and Equation (3.59) from Silbert et al. (2001), both contain vertical asymptotes (at $\beta = \frac{2}{\sqrt{7}}$ and $2\sqrt{2}$, respectively) which correspond to critical damping. Beyond these critical damping

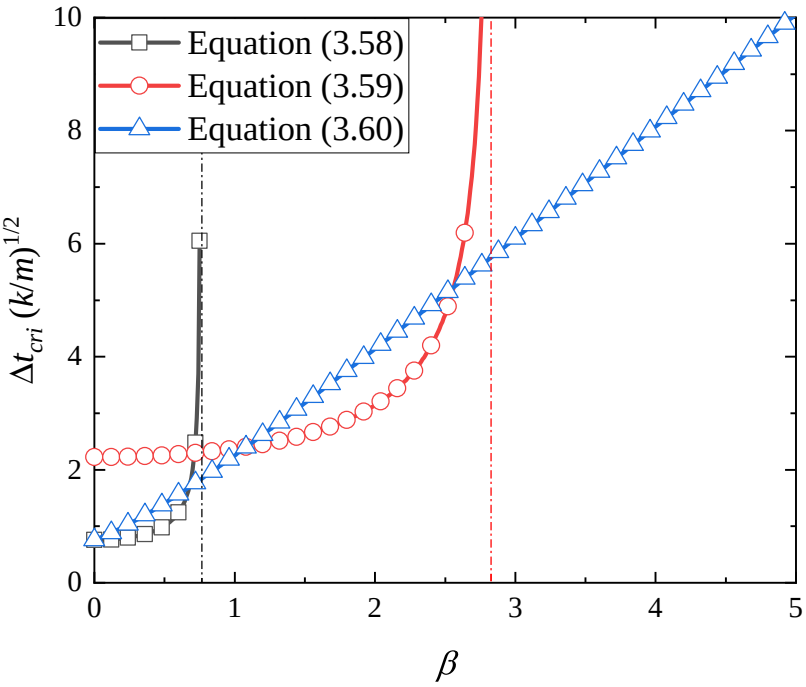


Fig. 3.7.: Nondimensionalised critical time step for two contacting spheres against the damping coefficient β for Equations (3.58)-(3.60)

values, complex values of Δt_{cri} appear which lack a physical meaning. Hence, Equations (3.58) and (3.59) are applicable only to underdamped cases. Equation (3.60), based on the amplification matrix method, is applicable at all levels of damping. For low-damping cases, Equations (3.58) and (3.60) are similar – and are identical in the undamped case as already seen from Equations (3.35) and (3.48). Since the critical time step based on the amplification matrix method is the most generally applicable, that is adopted for the remaining numerics.

3.7.2 Contact stiffness

Consider the two ellipsoids shown in Figure 3.6 as representative particles which satisfy the assumptions in Section 3.4. Each ellipsoid is identical, with density of $1000\,\mathrm{kg\,m^{-3}}$ and contact stiffness $k_n = k_s$ ranging from $1 \times 10^{10}\,\mathrm{N\,m^{-1}}$ to $10 \times 10^{10}\,\mathrm{N\,m^{-1}}$. We introduce a new dimensionless number, N, to quantify the damping for nonspherical particles in terms of A_{max}: $c_n = c_s = N\sqrt{k_n/A_{max}}$. The evolution of Δt_{cri} with k_n is shown in Figure 3.8 for $N = 0$, 1, 2, 3, 4 or 5. Decreasing the particle stiffness increases the critical time step irrespective of damping.

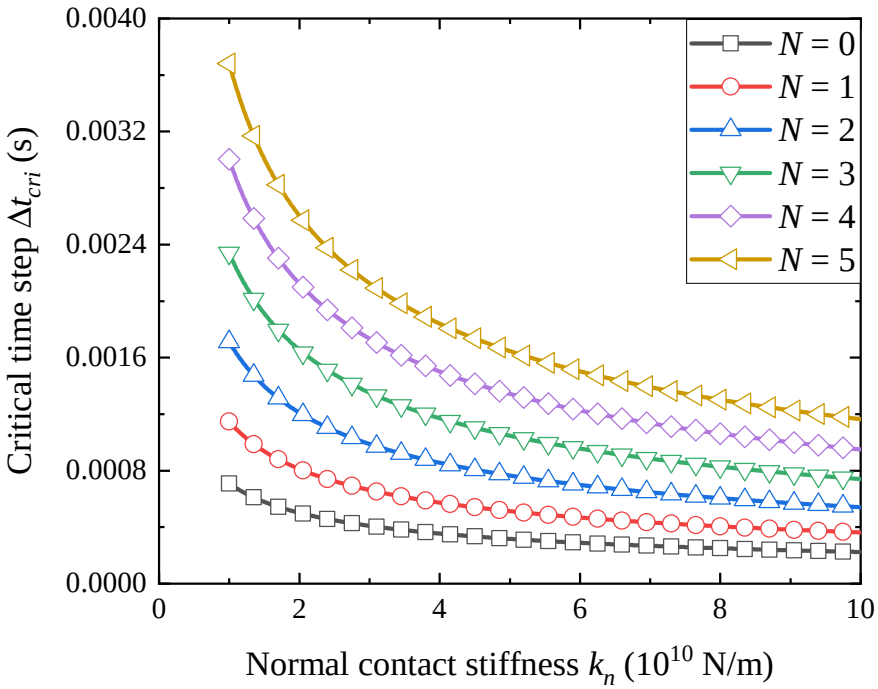


Fig. 3.8.: Critical time step for two contacting ellipsoids with semi-axis lengths $a = 3\,\mathrm{m}$, $b = 2\,\mathrm{m}$, $c = 1\,\mathrm{m}$ as the normal contact stiffness k_n varies from $1 \times 10^{10}\,\mathrm{N\,m^{-1}}$ to $10 \times 10^{10}\,\mathrm{N\,m^{-1}}$ at a fixed particle density ρ of $1000\,\mathrm{kg\,m^{-3}}$

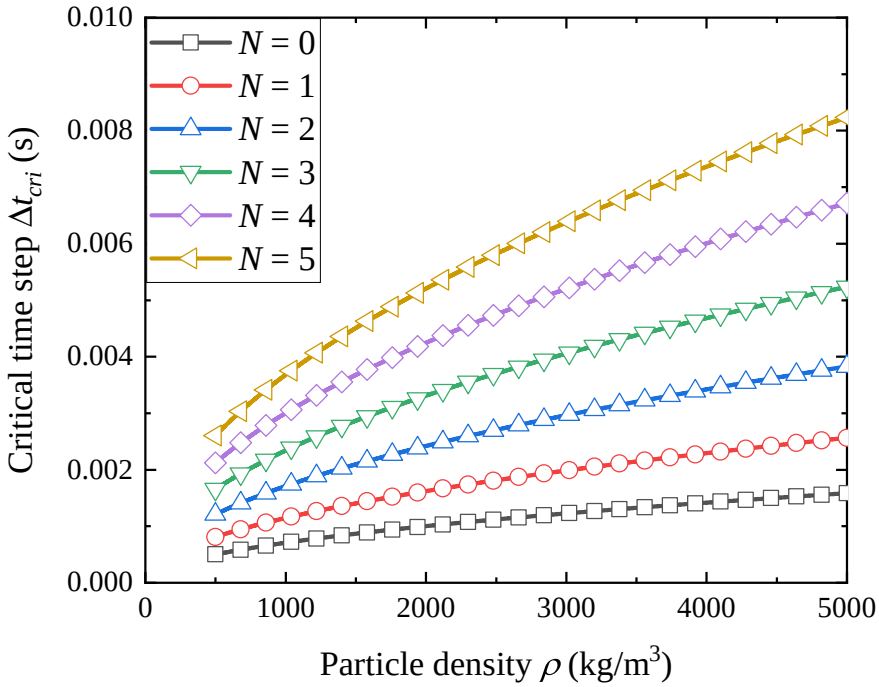


Fig. 3.9.: Critical time step for two contacting ellipsoids with semi-axis lengths $a = 3\,\mathrm{m}$, $b = 2\,\mathrm{m}$, $c = 1\,\mathrm{m}$ as the particle density ρ varies from $500\,\mathrm{kg\,m^{-3}}$ to $5000\,\mathrm{kg\,m^{-3}}$ at a fixed normal contact stiffness k_n of $1 \times 10^{10}\,\mathrm{N\,m^{-1}}$

3.7.3 Particle density

The same two ellipsoids are adopted as in Section 3.7.2. Now the contact stiffness is fixed at $k_n = k_s = 1 \times 10^{10}\,\mathrm{N\,m^{-1}}$ while the density is varied from $500\,\mathrm{kg\,m^{-3}}$ to $5000\,\mathrm{kg\,m^{-3}}$. Figure 3.9 shows that increasing the particle density, and hence mass, increases the critical time step. This is the motivation for 'density scaling' in which the particle densities are artificially increased by orders of magnitude in quasi-static simulations, enabling an increase of time step and a reduction of computation time.

3.7.4 Particle aspect ratio

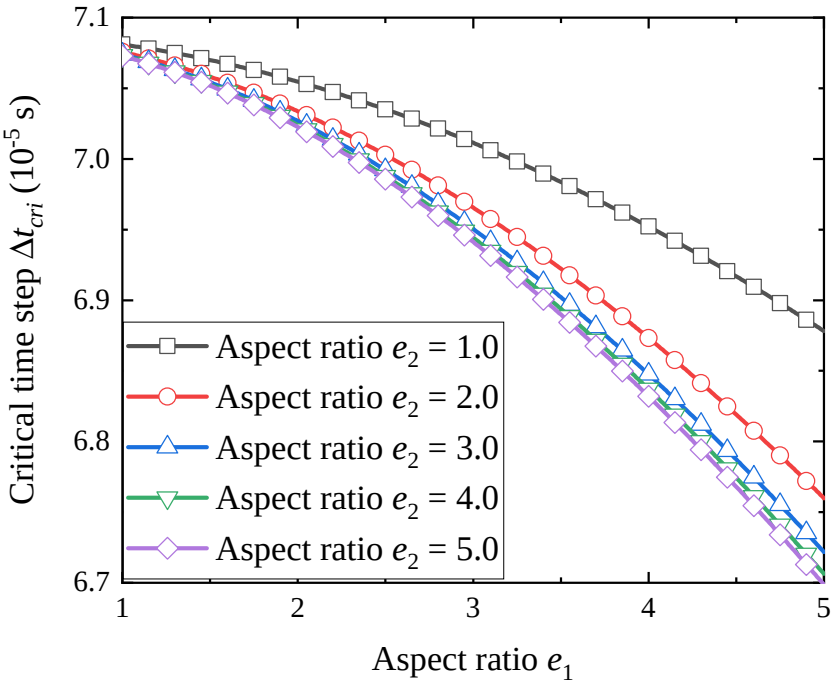
For a nonspherical particle, one would expect a change of aspect ratio to change a particle's moment of inertia, and thereby change the critical time step in accordance with Section 3.5. Choosing the two standard ellipsoids with density of $1000\,\mathrm{kg\,m^{-3}}$, contact stiffness of $k_n = k_s = 1 \times 10^{10}\,\mathrm{N\,m^{-1}}$ and no damping, fix the volume $V = \frac{4}{3}\pi abc = 8\pi\,\mathrm{m}^3$ of each ellipsoid and systematically vary two aspect ratios, defined as $e_1 = \frac{a}{b}$, $e_2 = \frac{b}{c}$. Since $a \geq b \geq c$, $e_1, e_2 \geq 1$. The maximum rotational velocity has

been set at $\omega_{max} = 1\,\mathrm{rad\,s^{-1}}$. Figure 3.10 shows the evolution of the critical time step with aspect ratios e_1 or e_2, considering both the translation- and rotation-dominant expressions for the critical time step. Using the translation-dominant expression, the critical time step decreases as the aspect ratios e_1 or e_2 increase, i.e., a nonspherical particle necessitates a smaller critical time step to ensure stability than a spherical particle of equivalent volume. The same is true using the rotation-dominant expression. It is therefore not advisable to assume equivalent diameters in order to calculate a simulation time step if particle aspect ratios differ significantly from one. The conservative assumption that $A \leq A_{max}$ in Equation (3.26) provides an easy and unified approach for determining critical time step; however, A_{max} makes the critical time step derived considerably more conservative than the one from exact mechanical analysis and may reduce the variation of critical time steps as particle aspect ratio changes. The translation-dominant time step is more than four orders of magnitude smaller than the rotation-dominant time step for this representative situation; for the DEM simulation in Chapter 4, the translation-dominant time step is two orders of magnitude smaller than the rotation-dominant time step.

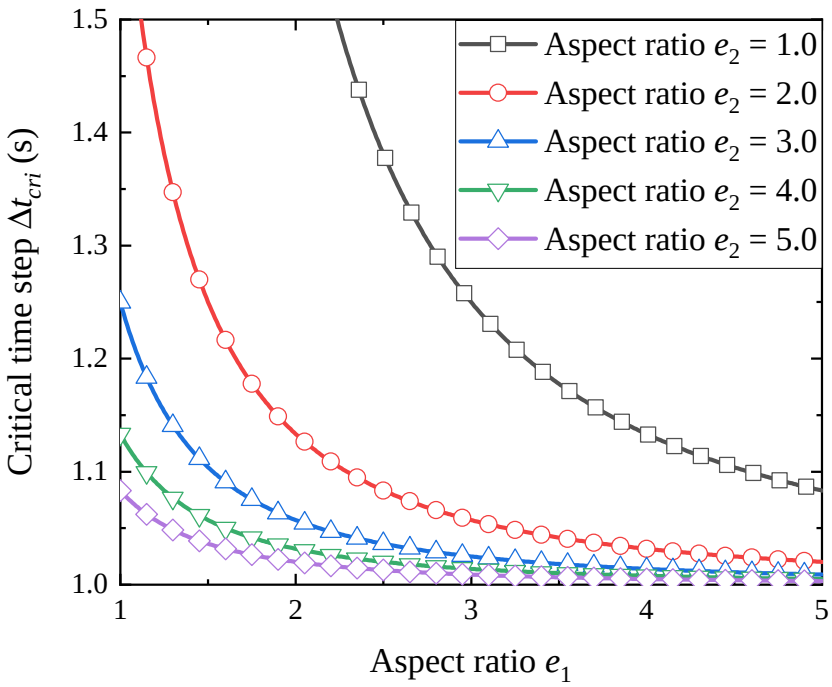
3.8 Validation of critical time step

In this section we will validate the critical time step derived using amplification matrix method. For validation cases using ellipsoids, LIGGGHTS-PUBLIC 3.8.0 (*LIGGGHTS(R)-PUBLIC Documentation* (2016)), which has superquadric support, is adopted. Suppose two ellipsoids of the same size in Section 3.3 put into a $20\,\mathrm{m}\times20\,\mathrm{m}\times20\,\mathrm{m}$ box centred at the origin. The centres of two ellipsoids are $(-5,0,0)$ and $(5,0,0)$ respectively (Figure 3.11). Both ellipsoids have same density of $1000\,\mathrm{kg\,m^{-3}}$ and random orientations (the random seed keeps the same so that in the following simulations the orientations keep the same). The ellipsoid 2 is assigned with the velocity of $5\,\mathrm{m\,s^{-1}}$ towards minus x direction to impact ellipsoid 1. Then a period of time 1000 s is simulated, during which two ellipsoids impact with each other and the walls. At the end two ellipsoids do not have any contact towards particles or walls. The box boundary has the same material property as the particle, and the linear Hooke stiffness model is adopted with $\mu = 0$ and contact stiffness as variable. The coefficient of restitution is set to 1.0 to make up a system without energy loss. In this case, the critical time step is calculated as

$$\Delta t_{cri} = \frac{2}{\sqrt{k_n\left(\frac{a^2}{\frac{4\pi}{15}\rho abc(b^2+c^2)} + \frac{1}{\frac{4\pi}{3}\rho abc}\right)}} \tag{3.61}$$



(a) Translation-dominant expression for the critical time step



(b) Rotation-dominant expression for the critical time step

Fig. 3.10.: Critical time step for two contacting ellipsoids as the particle aspect ratios e_1 and e_2 are systematically varied from 1 to 5 at a fixed normal contact stiffness $k_n = 1 \times 10^{10}\,\mathrm{N\,m^{-1}}$, particle density $\rho = 1000\,\mathrm{kg\,m^{-3}}$ and particle volume $V = 8\pi\,\mathrm{m^3}$

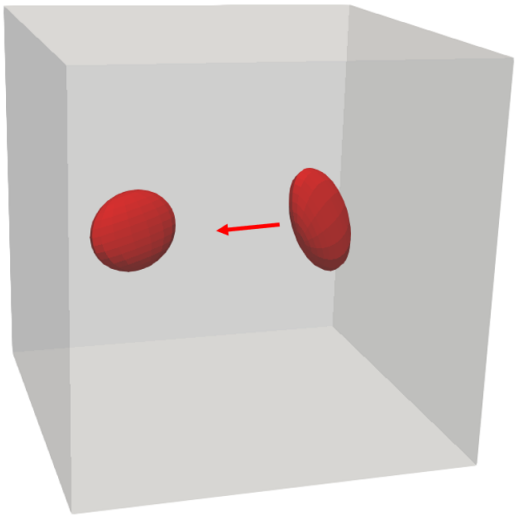
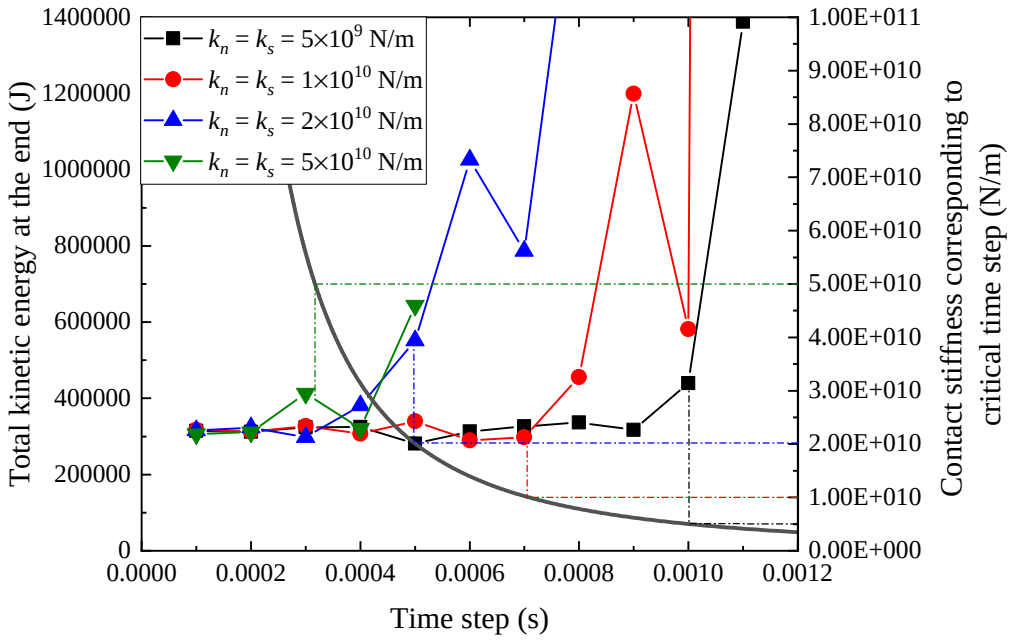
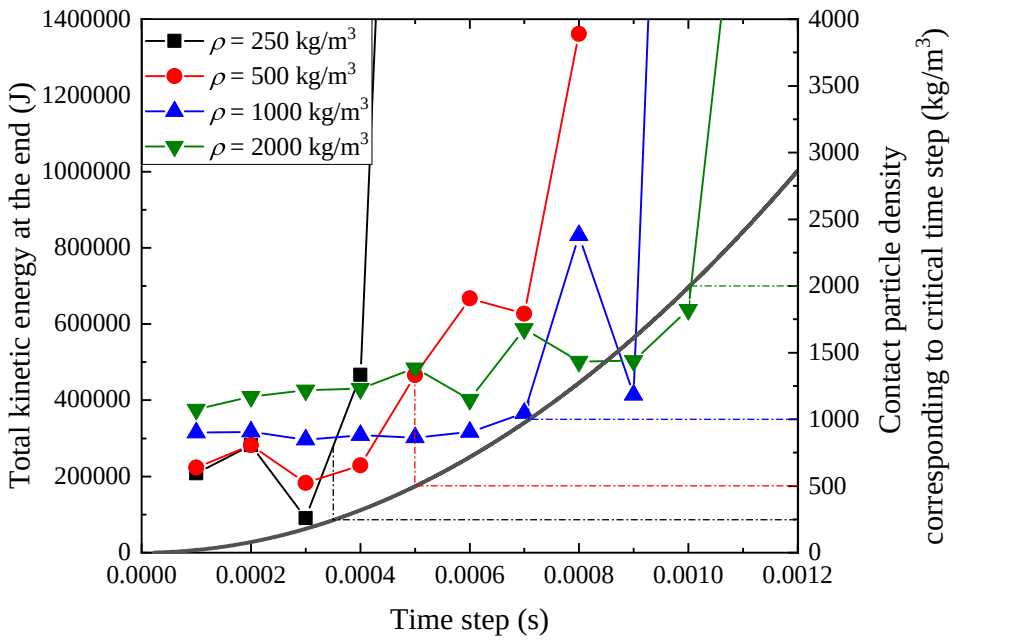


Fig. 3.11.: Illustration of validation scenario.

where $a = 3\,\mathrm{m}$, $b = 2\,\mathrm{m}$, $c = 1\,\mathrm{m}$. The formula of mass and moment of inertia of ellipsoid can be found in Jaklič et al. (2000). Figure 3.12(a) shows the total kinetic energy of the system at the end simulated in different time step. In the figure, the critical time step Δt_{cri} is related to contact stiffness, which is marked by the solid curve, as critical time step will decrease as contact stiffness increases. The corresponding Δt_{cri} is marked in vertical dashed dot line for each simulation case with specific contact stiffness. If the time step is not stable, the total energy in the system will rise as the simulation proceeds. It is shown in the figure that the total kinematic energy at the end increases rapidly when the time step exceeds Δt_{cri}, indicating that the amplification matrix method makes good prediction of critical time step for this case. If the contact stiffness is fixed as $k_n = k_s = 1 \times 10^{10}\,\mathrm{N\,m^{-1}}$ while particle density changes and all other variables keep the same, another figure can be drawn to show the impact of particle density (Figure 3.12(b)). Note that the initial kinetic energy of the system changes with particle density at this time. The critical time step increases with particle density, and the critical time step has been predicted properly for all the cases. Furthermore, apart from LIGGGHTS, EDEM Academic 2019.0 (*EDEM 2019 User Guide* (2019)) and PFC 5.00.40 (*PFC - Particle Flow Code, Version 6.0* (2018)) are also used for spherical particles for validation. By simply replacing ellipsoids in the validation case above with spheres of radius $r = 2\,\mathrm{m}$ similar results as Figures 3.12(a) and 3.12(b) can be generated for three codes, which are shown in Figure 3.13. Despite that the sphere 2 has the initial velocity of (-5, -0.5, 0.5) $\mathrm{m\,s^{-1}}$, all other variables are kept the same as above. The evolution of energy within the system for different time step is consistent among different DEM codes. However, when the time step gets too large and the system gets unstable,



(a) For different contact stiffnesses



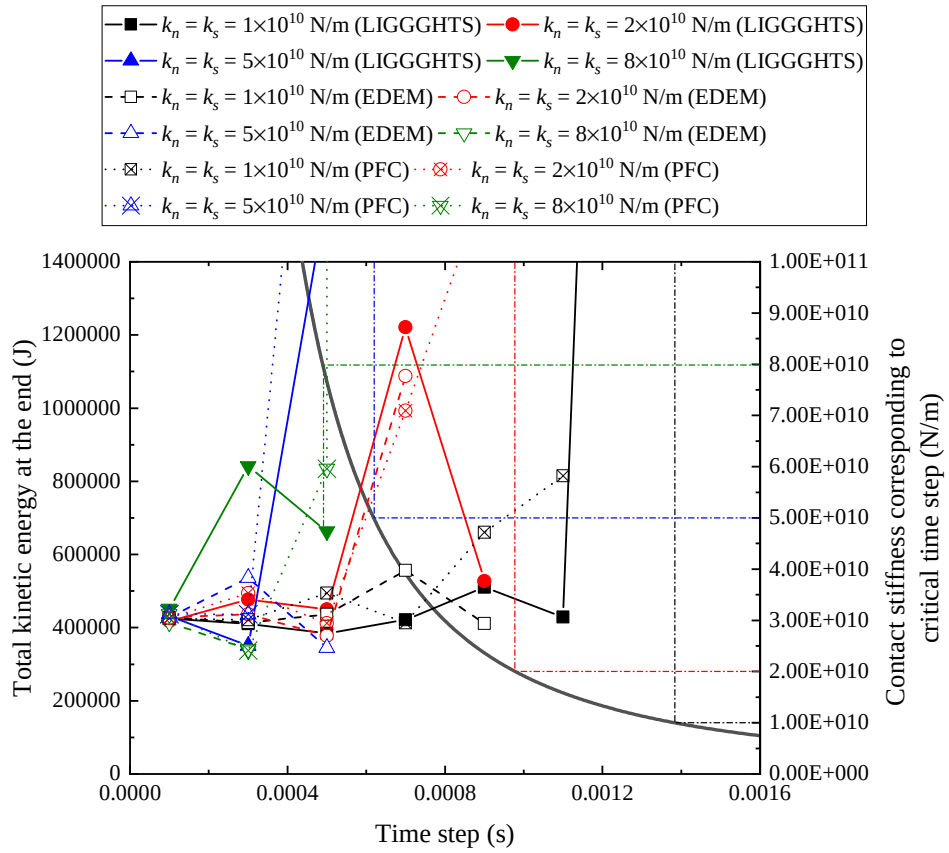
(b) For different particle densities

Fig. 3.12.: Kinematic energy evolution as time step changes for ellipsoids.

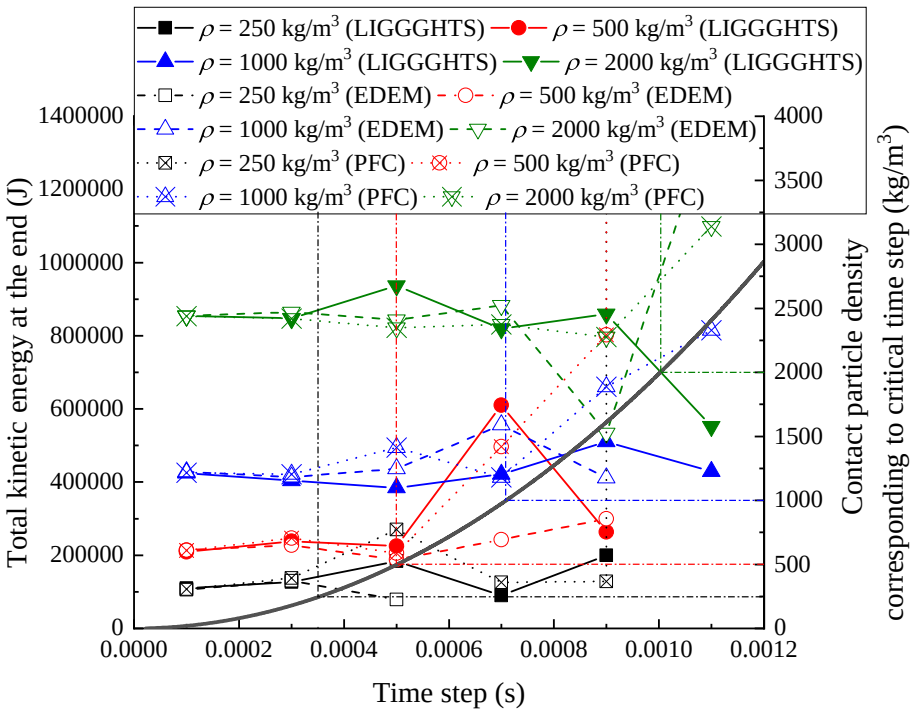
EDEM and LIGGGHTS will remove particles that moves through the boundary of domain while PFC keeps particles at the corner of domain. For cases that particles are removed, the data will not be shown on the figure, which is why there are data points missing for EDEM and LIGGGHTS results in Figure 3.13. Compared with ellipsoids above, the amplification matrix method in this study predicts less conservative critical time step for spheres. The reason is that this method considers the aspect ratio of nonspherical particle and makes a shrink in Equation (3.26), which controls the critical time step derived to be more conservative. The larger aspect ratio a nonspherical particle has, the more conservative the critical time step will be derived. Validation cases above have shown that for a nonspherical particle with apparent aspect ratio (approximately 3:2:1) the shrink will give a critical time step which is very close to the trial and error result. However, for spheres with aspect ratio of 1, the shrink has malfunctioned, giving a larger and unconservative time step. Generally a safety factor ranged from 0.1 to 0.9 will be given to the calculated critical time step in modelling spheres during simulation (*EDEM 2019 User Guide* (2019) and *PFC - Particle Flow Code, Version 6.0* (2018)), but for nonspherical particles, this safety factor could be larger than that for spheres.

3.9 Conclusions

This chapter has proposed a practical criterion for determining the critical time step for nonspherical particles which are convex and possess central symmetry, e.g., ellipsoids, convex superquadrics and certain polyhedra. The derivation using the amplification matrix method is based on linearity so is limited to a linear contact model with optional viscous damping. The derived expression, which applies across the whole range of damping from undamped to overdamped, has the expected $\sqrt{\frac{m}{k}}$ dependence on particle mass and contact stiffness. Hence, as the mass increases or the stiffness decreases, the critical time step increases. A second methodology based on Belytschko's criterion yielded the same critical time step as the amplification matrix method in the absence of damping. As particle shapes become increasingly nonspherical, quantified in this chapter by the deviation from unity of the aspect ratios of ellipsoids, the critical time step reduces nonlinearly. This indicates that the assumption of an 'equivalent sphere' for estimating a critical time step is potentially nonconservative.



(a) For different contact stiffnesses



(b) For different particle densities

Fig. 3.13.: Kinematic energy evolution as time step changes for spheres.

4 Experimental and numerical study of nonspherical particles in a rotating drum

4.1 Introduction

Rotating drums are commonplace in agricultural and chemical engineering, e.g., drying process of grains and fertilisers. Experimental studies on rotating drums date back to at least (Briscoe et al. (1984)) in which silica sand in a rotating drum was studied. That study was based on micro scale where particle-wall friction and normal interaction were discussed in a preliminary manner. The sand was considered aspherical. The particle-wall normal and shear stresses were found to be proportional to particle mass from the experimental data in that study. In recent years, the investigations have become more and more detailed. Yamamoto et al. (2016) and Chou et al. (2019) investigated the particle density and size effect, respectively, on the behaviour of the material in the rotating drum. With more techniques, more insights can be revealed in the rotating drum in the laboratory. Dubé et al. (2013) used X-rays to scan nonspherical particles during rotation and post-processed the information to get particle-scale velocity data. Li et al. (2018) used speckle visibility spectroscopy (SVS) to investigate a quasi-2D rotating drum where the drum was very thin in the longitudinal direction and the section was observed and investigated. The avalanche dynamics of granular beds were carefully investigated. Zhang et al. (2020) used the 3D particle tracking velocimetry (PTV) technique to generate micro-scale information in the rotating drum, along with DEM simulations using LIGGGHTS Kloss et al. (2012). However, all these experimental studies have focused on the materials in the rotating drum rather than the particle-wall interaction.

In addition to experiments, numerical simulations also play a role. Pöschel and Buchholtz (1995) simulated particles in a rotating drum using molecular dynamics (MD), which is algorithmically similar to DEM. Both spherical and nonspherical particles were simulated. The results matched previous experimental data (e.g. Briscoe et al. (1984) and Rajchenbach (1990)). The dynamic angle of repose, the flow regime and their relationship with particle rotational velocity were revealed. Recently, DEM studies on rotating drums have become more focused. Höhner et

al. (2014) investigated the effect of polyhedral particle shape while Freireich et al. (2009) performed DEM parameter sensitivity analysis, the former on dynamic angle of repose and the latter particle collision frequency and other variables at the particle scale. A series of studies have been carried out using ellipsoidal particles in DEM simulations in which the flow regime (He et al. (2019a)) and segregation of binary sized particles (He et al. (2019c)) were investigated. Moreover, the impact of particle shape on segregation was also revealed in He et al. (2019b), providing guidance on the behaviour of nonspherical particles in the rotating drum. An alternative approach to numerically study rotating drums is the finite element method (FEM), in which the particles are regarded as a continuum that can be meshed (Zheng et al. (2019)). However, all the numerical studies on the rotating drum above did not investigate particle-wall interaction; even Zhang et al. (2020) who used a flighted rotating drum in both experiments and simulations. Both experiments and simulations have lacked a detailed study on particle-wall interaction, which contributes the main motivation of this chapter.

The flow of solid particles can also be treated as a fluid, since fluid units (cells) can be separated and each has its own mass and stiffness. The flowing particles may or may not contact, which are correspondingly continuous and discontinuous fluids. In this way, the rheology, in other words, the characteristics of the "fluid" in the rotating drum can be studied. Rajchenbach (1990) discovered the flow of particles in the rotating drum is always switching between continuity and discontinuity, with the criterion given. A recent DEM study has revealed six different flow regimes in the rotating drum: slipping, slumping, rolling, cascading, cataracting and centrifuging (He et al. (2019a)). Some parameters for fluids can be generated, e.g., Froude number, for this kind of flow (Mellmann (2001)). However, the flow regime still does not take particle-wall interaction into consideration.

Boundary conditions are important in modelled tests in engineering. For example, in direct shear tests in soil mechanics, the shear box should be lubricated on the surface to avoid extra shear resistance of the specimen which induces increase of the shear strength of the soil (Nova (2010)). In addition, boundary effects have large impacts on the results of centrifuge tests in geotechnical engineering (Whitman and Lambe (1986), Lee (1990), Bolton et al. (1999), Lee et al. (2012), Ullah et al. (2014) and Ullah et al. (2017)). In-situ cone penetration tests also contain boundary effects (Been et al. (1986), Bolton et al. (1993) and Puppala et al. (1995)). In agricultural engineering, the discharge of a silo is also impacted by the boundary condition, i.e., particle-wall interaction between the material and the silo (Rotter et al. (1998), Zhong et al. (2001), Sanad et al. (2001), González-Montellano et al. (2011) and Wang et al. (2013)). Therefore, it is worth investigating the boundary effect in the rotating drum.

In this study, experiments using a rotating drum will be carried out with two types of beans. Images will be taken by hi-speed camera and PIV is adopted for analysis of experimental results. The particle sliding along the drum surface is investigated using these experimental results. DEM simulations of the experiment are also carried out, with the results calibrated at macro-scale using the experimental results. Coarse graining (CG) is adopted for generating the particle velocity near the surface of the drum, and the result is compared with that from PIV. In addition, particle-wall frictional coefficient is investigated statistically, showing the degree of sliding of the beans along the inner surface of the drum.

4.2 Experimental study

Several experiments are carried out by Huang (2019) for both red beans and soybeans. The images (raw data) taken by Huang (2019) during these experiments are adopted for analysis in this chapter. The images are analysed by the author using particle image velocimetry (PIV) to extract particle velocities.

4.2.1 Particle image velocimetry (PIV)

Particle image velocimetry (PIV) aims to track particle motion in the particulate system and extract micro-scale kinematic information. PIV first calculates the correlation c of a specific area of dimensions $M \times N$ between two consecutive images for the given displacement $(\Delta P_y, \Delta P_x)$ and given rotation $\Delta\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ of the area using the following criterion (Chung et al. (2010)):

$$c(\Delta P_y, \Delta P_x, \Delta\theta) = \sum_{i=1}^{M}\sum_{j=1}^{N} P_1(i, j, \Delta\theta) P_2(i + \Delta P_y, j + \Delta P_x) \tag{4.1}$$

where i, j refer to coordinates of pixels within the area and P_1 and P_2 refer to the pixel value (grey scale) of the coordinate (i, j) within the area in the former and latter images, respectively. P_1 is related to $\Delta\theta$ to reflect the scenarios in which the area rotates, while P_2 is related to translational changes of the area between two images. By maximising c with different sets of ΔP_y, ΔP_x and $\Delta\theta$) (ΔP_y and ΔP_x should not exceed the maximum displacement of particles between two images), the most likely motion of the area can be derived, and the velocity of this area can be then derived. It is important to maintain c at a relatively high value to ensure the quality of correlation, i.e. if c is very low after trying all the possible combinations of ΔP_y, ΔP_x and $\Delta\theta$), the computation of area motion may not be very reliable. Particle tracking velocimetry (PTV) is similar to PIV, though PTV can capture the motion of a single particle while PIV focuses on the motion of a specific area

(Zhang et al. (2020)). PIV/PTV are used in various areas. Niedostatkiewicz and Tejchman (2007) used PIV to analyse silo discharge; Chung et al. (2010) used PTV to investigate the behaviour of nonspherical polyformaldehyde particles; DeJong et al. (2006) used PIV to investigate soil-structure interaction behaviour. White et al. (2003) used PIV for analysing geotechnical problems, especially the deformation of soil, i.e., strain, which can be combined with the stress measured to produce a constitutive model. The PIV analyser adopted in this study, GeoPIV-RG (Stanier et al. (2016)), originates from that. Compared with GeoPIV (White et al. (2003)), GeoPIV-RG provides an automatic updating scheme so that the position of a specific area in the latter image will be adopted as the position of the reference area for the following analysis, if the correlation of the previous two images is successful (Stanier et al. (2016)). This could be very convenient for analysing the strain/displacement path of particles along a long image series in soil mechanics. However, in this study this feature is useless, as the reference cell position in this study is always fixed (see Section 4.2.4) and the tracking of the cell position is not needed. As this feature cannot be switched off in GeoPIV-RG, for each experimental scenario, ten pairs of the images from ten time points are selected for PIV analysis, while each pair of the images is consecutive.

4.2.2 Particle description

The particles used in this study are two types of beans: red bean and black beans, both coming from China. These two types of beans are selected for their robustness during multiple experiments, in which very few beans break. Figure 4.1 shows the two beans (Huang (2019)). The red bean is very smooth on the surface, while the soybean is relatively rougher. This phenomenon is linked to the dynamic angle of repose of the two types of beans discussed in Section 4.4. The red bean is considered as an ellipsoid, while the soybean is considered as a sphere for comparison. The sizes of the beans are taken from the average of 5 measurements of different beans. The half axial lengths of the red bean are $a = 4.5 \times 10^{-3}\,\mathrm{m}$, $b = c = 3 \times 10^{-3}\,\mathrm{m}$ while the radius of the soybean is $r = 3.5 \times 10^{-3}\,\mathrm{m}$. The volumes of red bean and soybean particles can be calculated as follows:

$$V_r = \frac{4}{3}\pi abc \tag{4.2}$$

$$V_s = \frac{4}{3}\pi r^3 \tag{4.3}$$

Fig. 4.1.: Red bean (left) and soybean (right) (Huang (2019)).

The mass of a bean is calculated as follows:

$$m = \frac{M}{N} \tag{4.4}$$

while $M = 400\,\mathrm{g}$ is the total mass of the beans in a package and N is the number of the beans counted within that package. The red bean and soybean particle masses are $m_r = 0.21\,\mathrm{g}$ and $m_s = 0.24\,\mathrm{g}$, respectively. Based on the mass and volume, the densities of red bean and soybean can be calculated as approximately $\rho_r = 1250\,\mathrm{kg\,m^{-3}}$ and $\rho_s = 1300\,\mathrm{kg\,m^{-3}}$, respectively.

4.2.3 Geometry description

Figure 4.2 is an image of the drum. Its inner diameter is $d_D = 28\,\mathrm{cm}$, its outer diameter is $D_D = 31\,\mathrm{cm}$, and its depth is $l_D = 10\,\mathrm{cm}$. The drum is made of ABS plastic with a Young's modulus of 2 GPa and Poisson's ratio of 0.35, which are adopted in the DEM simulations in Section 4.3. The inner surface of the drum is relatively smooth. Three rotational velocities were chosen for the drum: $1.745\,\mathrm{rad\,s^{-1}}$, $3.222\,\mathrm{rad\,s^{-1}}$ and $5.856\,\mathrm{rad\,s^{-1}}$.

4.2.4 Measurement

The hi-speed camera used in this study is a Photron FASTCAM SA2 Huang (2019). The image size taken is 1024×1024 px. The frame rate is $f = 500\,\mathrm{s^{-1}}$. An external lamp provides extra illumination (Figure 4.3). The camera will capture images for a

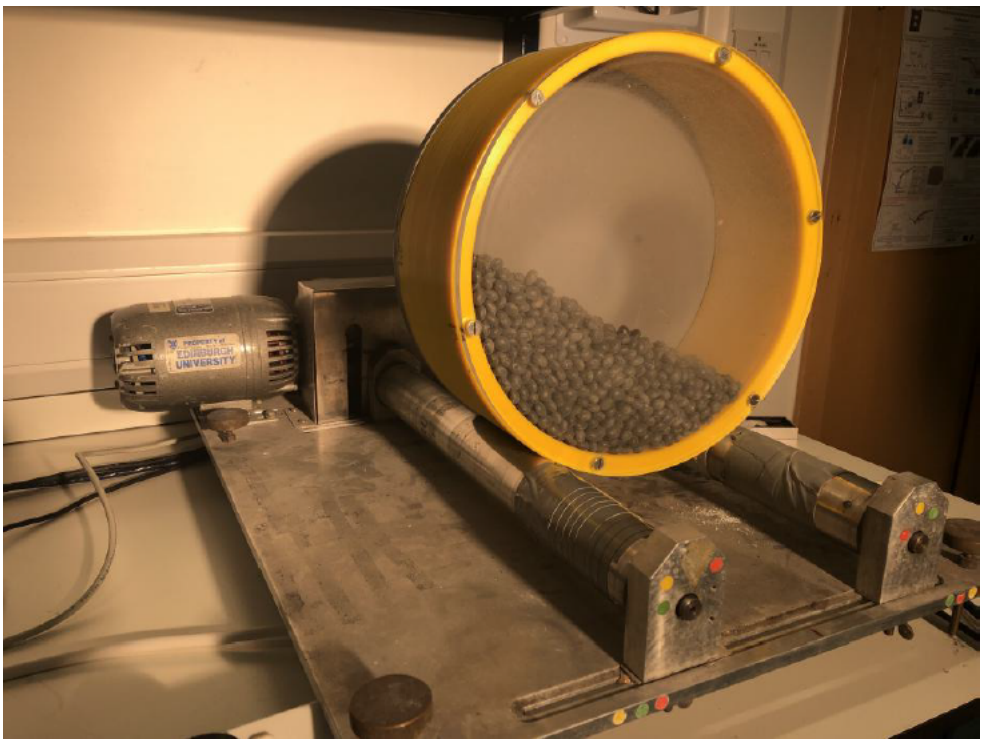


Fig. 4.2.: The rotating drum.

period of no less than two consecutive seconds for analysis, once the system within the drum has reached a steady state.

Figure 4.4 shows sample images for introducing the mesh and cells for analysis. For PIV analysis, the mesh size (diameter) is 45 px and the spacing of the mesh is 45 px, and only meshes that are within the material will be analysed with PIV (Figure 4.4(a)). Three specific positions in the drum are taken for analysis, which are called "Front", "Bottom" and "Back" and are marked in Figure 4.4(b). These three specific positions are unchanged for all analyses. Since the sample image is the scenario with highest dynamic angle of repose, the "Front" position in all the scenarios will capture the foot area of the material. However, this position yields fluctuating data as particles travel rapidly, which will be discussed in Section 4.4. The outer dimension of the drum in the image is $P_D = 1013$ px, therefore, the velocity of the particle extracted via PIV can be expressed as

$$|\boldsymbol{v}| = \frac{D_D}{P_D} f \Delta P \tag{4.5}$$

where $|\boldsymbol{v}|$ is the magnitude of velocity, $\Delta P = \sqrt{\Delta P_x^2 + \Delta P_y^2}$ is the relative pixel displacement between two images where P_x and P_y are its components in the x and y directions, respectively.

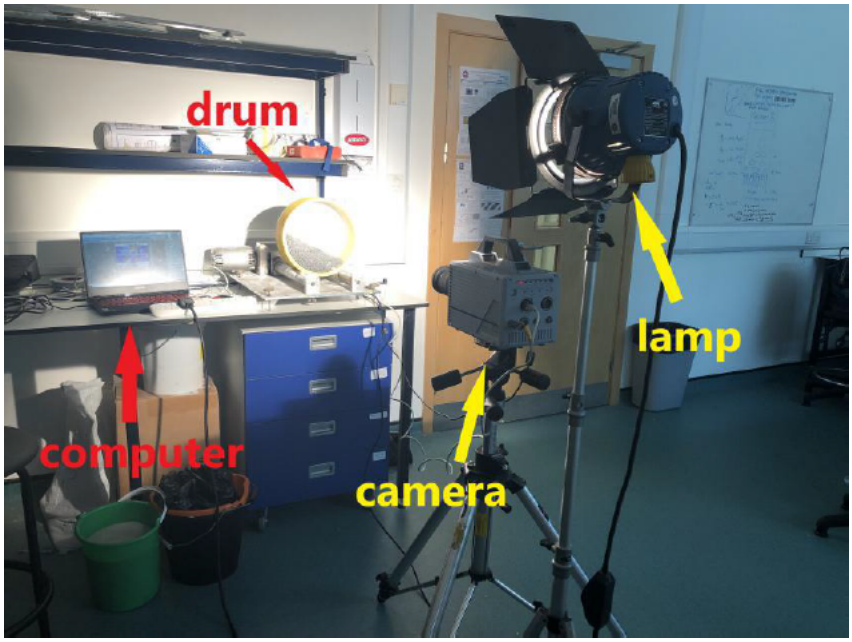
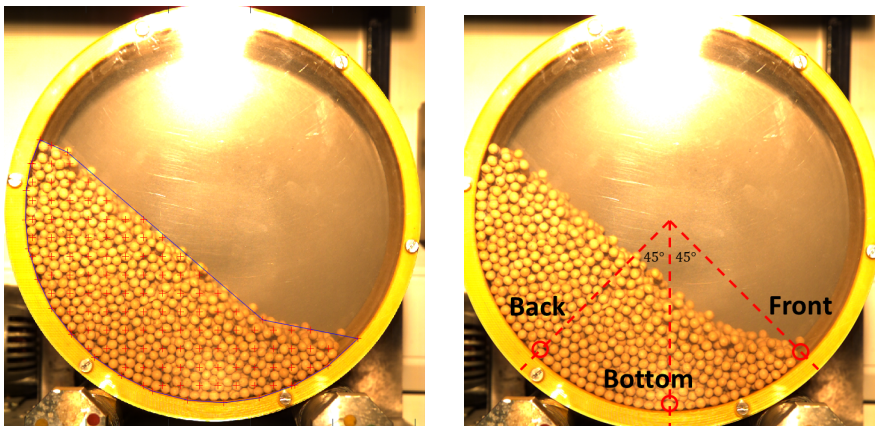


Fig. 4.3.: Experimental setup.



(a) Illustration of meshes.

(b) Illustration of Back, Bottom and Front cells.

Fig. 4.4.: Illustration of meshes and cells selected for analysis.

4.2.5 Experiment process

For each experimental scenario, 1200 g of the beans are inserted into the rotating drum, which correspond to approximately 5700 red beans or 5100 soybeans. These numbers of particles are adopted in the DEM simulations in Section 4.3. If filled flat, 1200 g of beans will fill the drum with the volume of $9.67 \times 10^{-4}\,\mathrm{m}^3$.

4.3 Numerical study

The simulation scenario is kept the same as the experiments in Section 4.2.2 to enable comparison between DEM modelling and reality. The open-source code LIGGGHTS (Kloss et al. (2012)) is adopted. Both red beans and soybeans are both modelled using superquadric particle shapes in LIGGGHTS (Podlozhnyuk et al. (2017)). Although soybeans are spheres so could be modelled with spherical particles, in this chapter the soybeans are also modelled using superquadrics for consistency with the ellipsoidal red beans.

4.3.1 Coarse graining in DEM

Coarse graining is an increasingly commonly used method for post-processing DEM data (Noid et al. (2008a), Noid et al. (2008b), Goldhirsch (2010), Weinhart et al. (2016), Chu et al. (2016), Daraio et al. (2019), Xie et al. (2019b) and Tausendschön et al. (2020)). In this chapter, a coarse graining software, Iota (*Iota User Manual* (2019)), is used for interpreting particle velocity at the surface of the drum. A coarse graining function $\phi(\boldsymbol{x}-\boldsymbol{x}_i)$ is initially established which is usually a Gaussian function. This function ensures that the integral of mass within a specific volume equals to the total mass of particles within the volume:

$$m(\boldsymbol{x}) = \sum_i m_i \phi(\boldsymbol{x} - \boldsymbol{x}_i) \tag{4.6}$$

where $\boldsymbol{x}$ is the centre of mass of the volume, m_i is the mass of each particle and $\boldsymbol{x}_i$ is the centre of mass of each particle within the volume. The velocity of this specific volume can then be interpreted as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{\sum_i m_i \boldsymbol{v}_i \phi(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum_i m_i \phi(\boldsymbol{x} - \boldsymbol{x}_i)} \tag{4.7}$$

where $\boldsymbol{v}_i$ is the velocity of each particle within the volume.

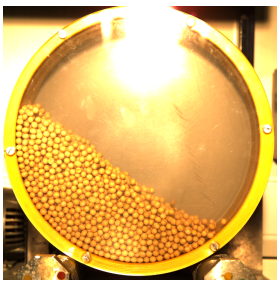
Tab. 4.1.: Frictional coefficients.

Bean type	Particle-particle	Particle-wall
Red bean	0.27	0.18
Soybean	0.48	0.48

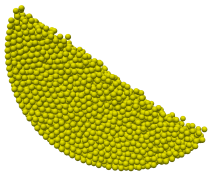
4.3.2 DEM parameters and simulation process

In addition to the particle parameters in Section 4.2.2, DEM requires more parameters for simulation. The Hooke contact model is adopted for all the DEM simulations to simplify the modelling. The elastic modulus and Poisson's ratio adopted, $E = 5 \times 10^8$ Pa and $\nu = 0.35$, are values for a similar material, Chinese black beans, from Johnstone (2010). The same parameters are set for both red beans and soybeans for simplicity and consistency in DEM analysis, as the particle-wall friction coefficient is assumed to be the key factor influencing on particle-wall interaction. Both beans have the same coefficient of restitution of 0.65, as a drop test of a bean from a height of 1 m will give a rebound height of approximately 0.42 m. The characteristic impact velocity which is related to normal and shear contact stiffness is set to be $1\,\mathrm{m\,s^{-1}}$ (*LIGGGHTS(R)-PUBLIC Documentation* (2016)). The frictional coefficients, both particle-particle and particle-wall, are calibrated via trial-and-error during the simulation, with the criterion of obtaining the same dynamic angle of repose as in the experiment. Table 4.1 shows the frictional coefficients for DEM simulations. The friction coefficient of soybeans calibrated is much higher than the realistic value (i.e. 0.24 in *Eurocode 1 – Actions on structures, Part 4: Actions on silos and tanks* (2003)). One probable reason is that the soybeans in DEM simulations are represented by spheres, although they are indeed not perfectly spherical. The other possible reason is that calibration of DEM parameters relying on macro behaviour may lead to unrealistic results, with more detailed discussions provided in Section 4.4. Viscous damping is applied in normal direction of the contacts; in the shear direction, the viscous damping is disabled. This is due to a bug in the public version of LIGGGHTS (LIGGGHTS-PUBLIC 3.8.0) which is discussed in Appendix B. The time step used for simulation is determined via Chapter 3. Taking red beans as the example, if the system is translation-dominant, the critical time step is 1.57×10^{-5} s; if the system is rotation-dominant, as the maximum particle rotational velocity is $1.14 \times 10^2\,\mathrm{rad\,s^{-1}}$ obtained from the demonstration simulation, the critical time step is 2.27×10^{-2} s. Then the translation-dominant scenario is adopted as the critical time step. A safety factor of 0.2 is set, multiplying the critical timestep calculated. The time step is therefore set as 3.13×10^{-6} s for red bean systems and 4.74×10^{-6} s for soybean systems.

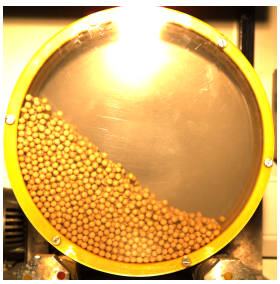
Each simulation starts by randomly generating superquadric particles within the drum. Then 10 s are allocated for particles to pack under gravity, after which the



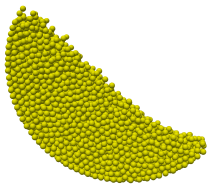
(a) Experiment, soybean 1.745 rad s^{-1}



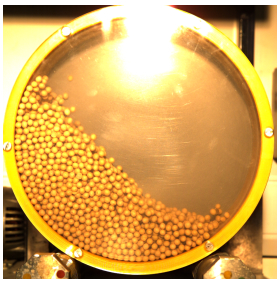
(b) Simulation, soybean 1.745 rad s^{-1}



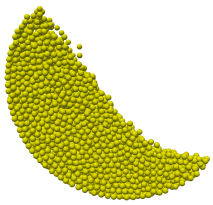
(c) Experiment, soybean 3.222 rad s^{-1}



(d) Simulation, soybean 3.222 rad s^{-1}



(e) Experiment, soybean 5.856 rad s^{-1}



(f) Simulation, soybean 5.856 rad s^{-1}

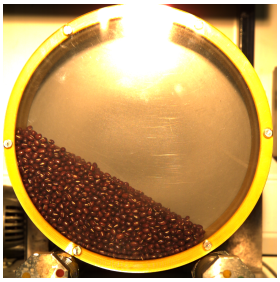
Fig. 4.5.: Experimental and simulation scenarios of soybeans in the rotating drum.

drum starts to rotate and another 10 s are given for achieving the steady state. Finally, 2 s are given for calculation and exporting micro-scale data for comparison with PIV results.

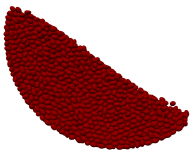
4.4 Results and analysis

4.4.1 Dynamic angle of repose

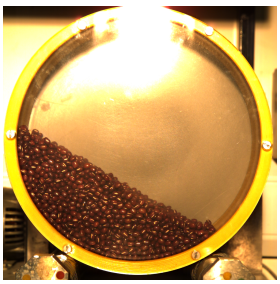
The views of soybeans and red beans in the rotating drum at steady state for both experiment and simulation are shown in Figures 4.5 and 4.6, respectively. The dynamic angle of repose is measured based on these images. The experimental images come from Huang (2019) and are analysed by the author.



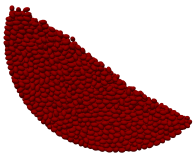
(a) Experiment, red bean $1.745\,\mathrm{rad\,s^{-1}}$



(b) Simulation, red bean $1.745\,\mathrm{rad\,s^{-1}}$

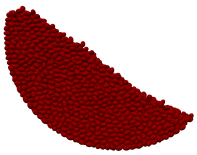


(c) Experiment, red bean $3.222\,\mathrm{rad\,s^{-1}}$



(d) Simulation, red bean $3.222\,\mathrm{rad\,s^{-1}}$

(e) Experiment, red bean $5.856\,\mathrm{rad\,s^{-1}}$



(f) Simulation, red bean $5.856\,\mathrm{rad\,s^{-1}}$

Fig. 4.6.: Experimental and simulation scenarios of red beans in the rotating drum.

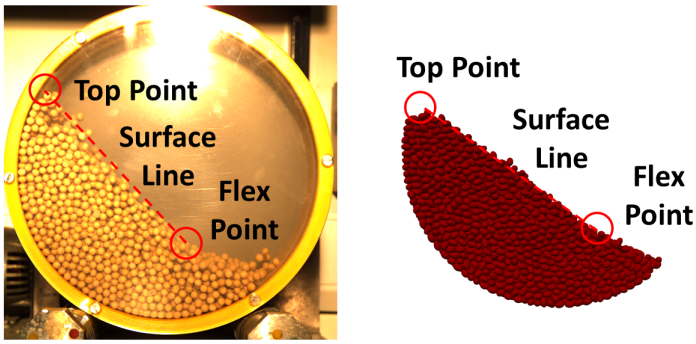


Fig. 4.7.: Illustration of flex point and surface line.

Tab. 4.2.: Dynamic angles of repose of all the scenarios of soybean.

Rotating speed	Angle of repose (experiment)	Angle of repose (simulation)
$1.745\,\mathrm{rad\,s^{-1}}$	38°	36°
$3.222\,\mathrm{rad\,s^{-1}}$	42°	43°
$5.856\,\mathrm{rad\,s^{-1}}$	48°	50°

The angles of repose measured for soybeans and redbeans are shown in Table 4.2 and Table 4.3, respectively. The surface of the material is not always flat in the rotating drum (Figures 4.5 and 4.6), and the dynamic angle of repose is selected to be the angle between the surface line (Figure 4.7) and the horizontal line. On the surface of the material (both for experiments and simulations) there is usually a flex point where the angle of repose will change (Figure 4.7). The surface line joins the flex point and the top of the material in Figure 4.7. Sometimes there is no flex point (e.g. experiment, red bean $1.745\,\mathrm{rad\,s^{-1}}$), and thus the flex point is the foot point of the material. The dynamic angles of repose are similar for both experiment and simulation, due to the fact that the friction coefficients are calibrated using the dynamic angles of repose. The dynamic angle of repose increases with rotating speed for soybeans, but remains almost the same for red beans whatever the rotating speed is. This is perhaps surprising as generally the nonspherical red beans should have more interlocking and thus give a higher angle of repose. However, as mentioned in Section 4.2.2, the red bean is very smooth on the surface. This will be discussed further in Section 4.4.3.

Tab. 4.3.: Dynamic angles of repose of all the scenarios of red beans.

Rotating speed	Angle of repose (experiment)	Angle of repose (simulation)
$1.745\,\mathrm{rad\,s^{-1}}$	31°	31°
$3.222\,\mathrm{rad\,s^{-1}}$	31°	31°
$5.856\,\mathrm{rad\,s^{-1}}$	31°	31°

4.4.2 Bean velocity at drum surface

Figure 4.8 and Figure 4.11 show the evolution of cell velocity within the drum during a 2 s period of steady state. Both PIV and DEM results are shown in the same figure for a specific scenario, in which the dashed horizontal line in the figure marks the translational velocity the of drum's surface. For both PIV and DEM, very large fluctuations occur from the results interpreted from the Front cell. This corresponds to the description in Section 4.2.2 that particles travel rapidly at the Front cell, which makes the interpretation of particle velocity unstable. For soybeans, both PIV and DEM results show that the particle velocity at the drum's surface is lower than the translational velocity of the drum, meaning that sliding always occurs at the particle-wall surface. The discrepancy of the two velocities increases with rotating speed, meaning that the sliding increases with the rotating speed.

In addition, the particle velocity derived from DEM is less than that from PIV, and the difference increases with the rotation speed of the drum. One possible reason is that soybeans are not perfectly smooth spheres; there is a small degree of non-sphericity and unevenness in the surface. To further investigate this, a rolling resistance model (an elastic-plastic spring-dashpot (EPSD) model with an additional twisting torque contribution based on Ai et al. (2011) in LIGGGHTS (*LIGGGHTS(R)-PUBLIC Documentation* (2016))) is added to the present soybean simulation scenarios. The dimensionless rolling stiffness coefficient is 1.0 and rolling friction coefficient is 0.5 for both particle-particle and particle-wall contacts. Note that these parameters are only typical values which have not been calibrated, as these simulations with rolling resistance are only to show that the dynamic angle of repose can be increased by implementing rolling resistance model. No viscous damping is set in the rotational direction. The snapshots of steady states and coarse-grained particle velocity results of scenarios with rolling resistance are shown in Figures 4.9 and 4.10. Compared to the scenarios without rolling resistance, the dynamic angle of repose increases. The particle velocity also increases and becomes closer to the translational velocity of the drum, meaning that the sliding will be reduced and the dynamic angle of repose will be further increased if particle-particle and/or particle-wall interlocking increases. As mentioned in Section 4.3, the Hooke contact model has been selected to make the contact modelling simple, and the friction coefficient was calibrated to be an unrealistically high value for soybeans. The results of scenarios with rolling resistance show that the friction coefficient of soybeans may not be sufficient for modelling, as some surface roughness of soybeans also plays a role in the particle-wall interaction.

For red beans, the particle velocity from DEM is no less than that from PIV. Though both PIV and DEM results show that massive sliding occurs between red beans and the drum's surface, a much larger difference exists between particle velocity and translational velocity of the drum for red beans than soybeans. Massive sliding may be a reason that the dynamic angle of repose of red beans remains the same as the rotating speed increases.

The difference between PIV and DEM on particle velocity at the micro-scale is apparent and different for soybeans and red beans, although the dynamic angle of repose at the macro-scale has been calibrated in DEM to be very close to the experimental result. This shows that calibrating DEM parameters (even key parameters) with macro-scale behaviour generally does not result in the same match in the micro-scale at the same time. Freireich et al. (2009) reached the same conclusion during the DEM parameter sensitivity study.

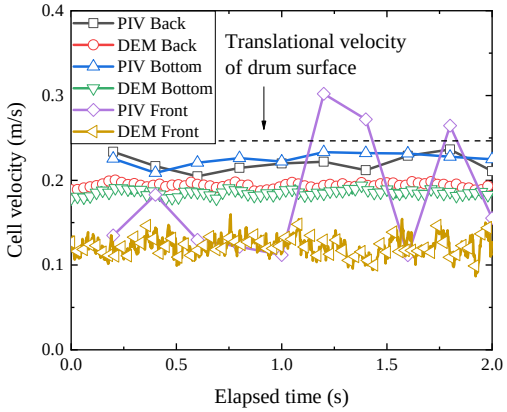
4.4.3 Particle-wall friction

Micro-scale frictional information can be extracted from DEM datasets. Selecting the time point of t =20 s at which steady state has been achieved, the distribution of friction coefficients of all particle-wall contacts at this time are shown in Figure 4.12. The friction coefficient of each contact is calculated as follows:

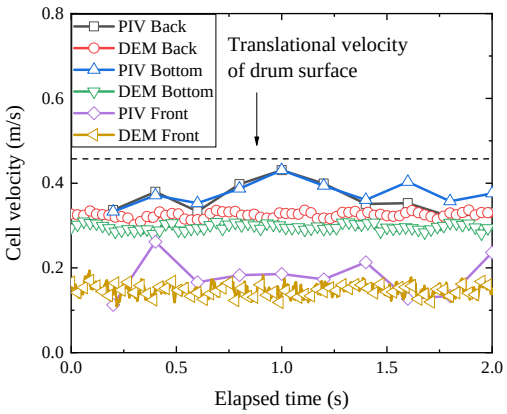
$$\mu = \frac{|\boldsymbol{F}_s|}{|\boldsymbol{F}_n|} \tag{4.8}$$

where $\boldsymbol{F}_n$ and $\boldsymbol{F}_s$ are normal and shear contact forces, respectively. The vertical dash-dotted line marks the limit value of μ. Note that the contacts between particles and side walls are not taken into account. From the figure, the difference between the two types of beans can be seen. Approximately 40-50% of the particle-wall contacts do not slide for soybeans, while almost all the particle-wall contacts slide for red beans. This can be the reason that the dynamic angle of repose of red beans remains the same as rotating speed increases, as the red bean-drum surface is almost perfectly smooth.

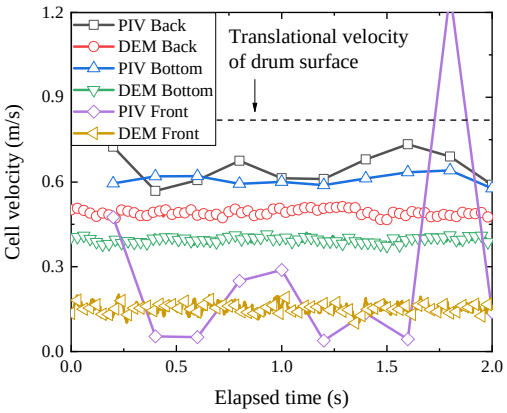
The geometrical distribution of particle-wall friction coefficient is shown in Figure 4.13 for soybeans and Figure 4.14 for red beans. The black plots in both figures refer to sliding. For red beans, the sliding occurs all along the drum surface and is larger than for soybeans, which corresponds to the results shown in Figure 4.12.



(a) $1.745\,\mathrm{rad\,s^{-1}}$

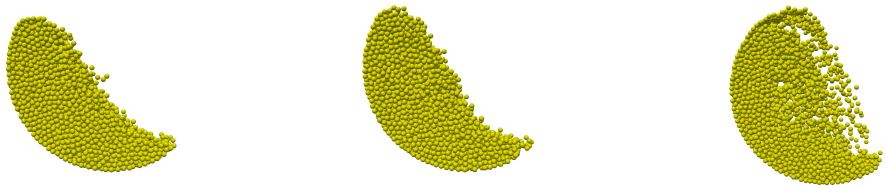


(b) $3.222\,\mathrm{rad\,s^{-1}}$



(c) $5.856\,\mathrm{rad\,s^{-1}}$

Fig. 4.8.: Cell velocity at the surface of the drum (soybeans).

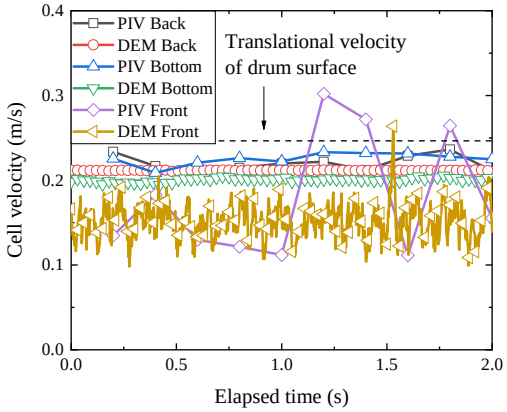


(a) $1.745\,\mathrm{rad\,s^{-1}}$ (b) $3.222\,\mathrm{rad\,s^{-1}}$ (c) $5.856\,\mathrm{rad\,s^{-1}}$

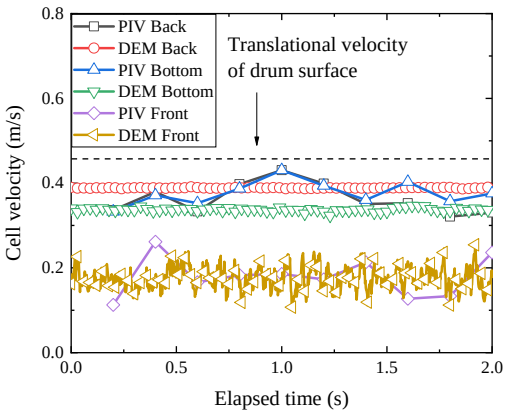
Fig. 4.9.: Simulation scenarios of soybeans with rolling resistance in the rotating drum.

4.5 Conclusions

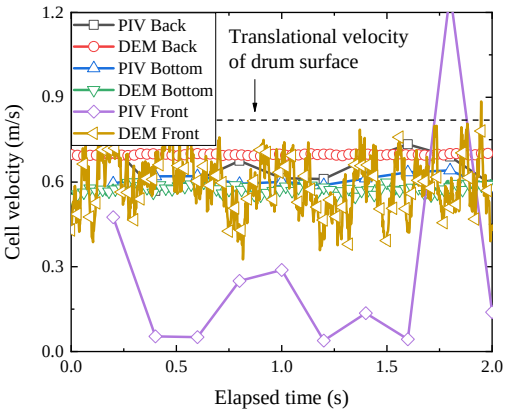
This chapter performs a combination of experimental and numerical study of beans in the rotating drum. Red beans and soybeans are selected as the granular materials and the particle-wall interaction is the focus of the investigation, which is lacking in previous studies. The dynamic angles of repose from the experiments are adopted for calibrating the particle frictional coefficients in DEM, and thus the dynamic angles of repose in DEM are close to those in the experiments. The dynamic angles of repose increase with rotating speed for soybean but do not increase for red beans. The particle velocity is interpreted using PIV from hi-speed camera images in the experiment and using coarse-graining from datasets in the DEM simulation. However, the particle velocities in the micro-scale from PIV and DEM do not match while the macro-scale dynamic angles of repose are the same, which shows the discrepancy of DEM modelling that the micro- and macro-scale behaviour of the system cannot be captured at the same time. Both soybeans and red beans slide along the drum. Almost all the red beans are sliding while half of the soybeans are, which may be the reason why the dynamic angle of repose of red beans does not change as the rotating speed increases.



(a) $1.745\,\mathrm{rad\,s^{-1}}$

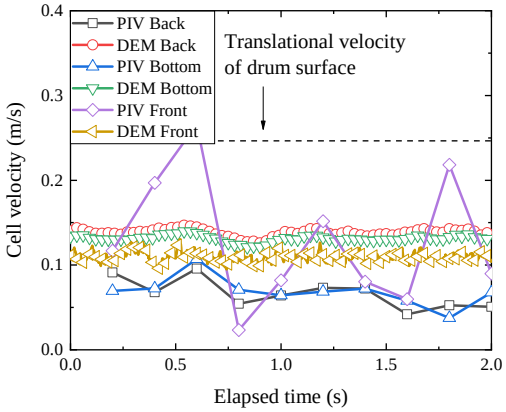


(b) $3.222\,\mathrm{rad\,s^{-1}}$

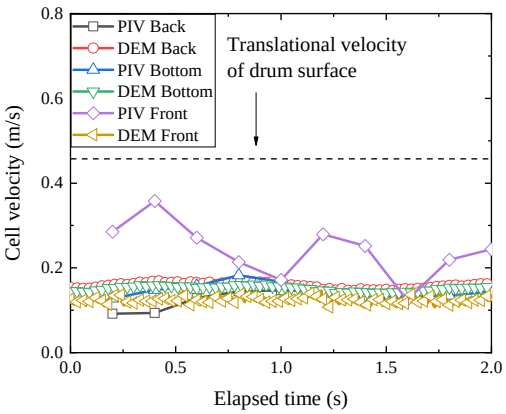


(c) $5.856\,\mathrm{rad\,s^{-1}}$

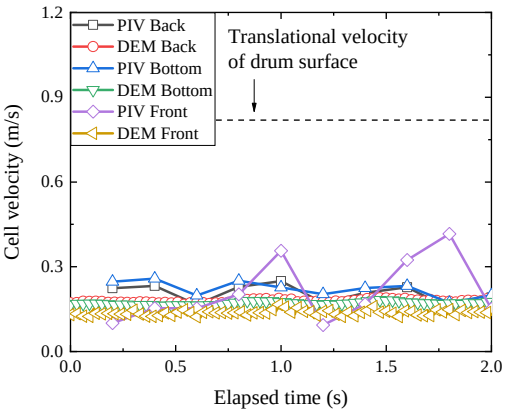
Fig. 4.10.: Cell velocity at the surface of the drum (soybeans with rolling resistance).



(a) $1.745\,\mathrm{rad\,s^{-1}}$

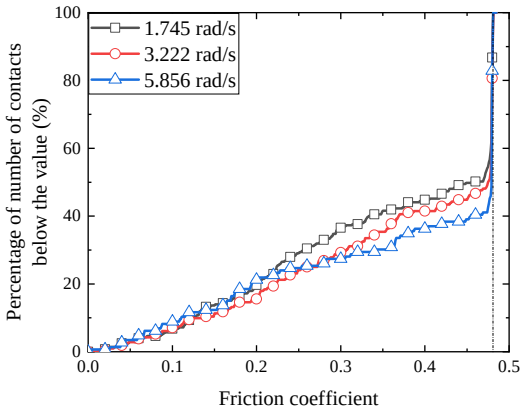


(b) $3.222\,\mathrm{rad\,s^{-1}}$

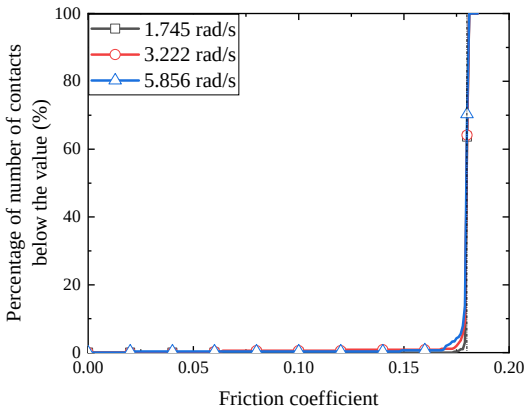


(c) $5.856\,\mathrm{rad\,s^{-1}}$

Fig. 4.11.: Cell velocity at the surface of the drum (red beans).

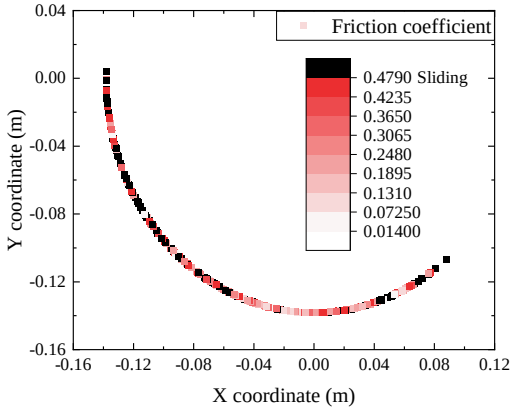


(a) Soybeans

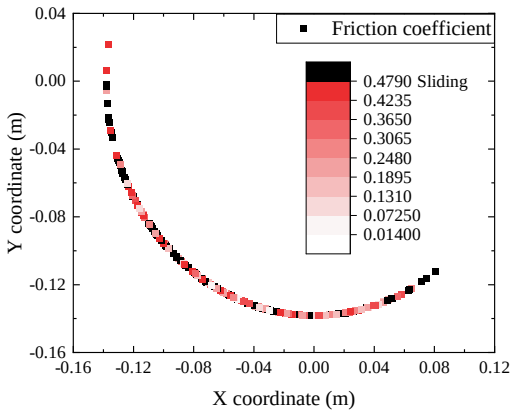


(b) Red beans

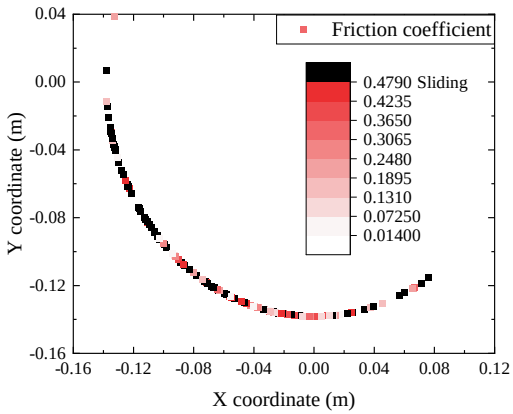
Fig. 4.12.: Distribution of particle-wall friction coefficient for both types of beans.



(a) $1.745\,\mathrm{rad\,s^{-1}}$

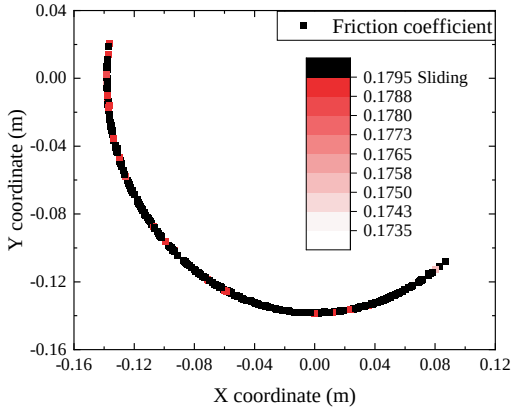


(b) $3.222\,\mathrm{rad\,s^{-1}}$

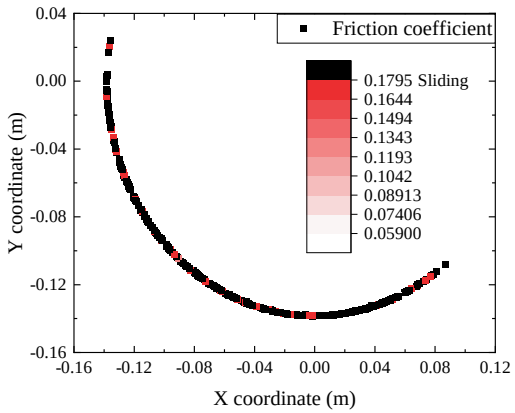


(c) $5.856\,\mathrm{rad\,s^{-1}}$

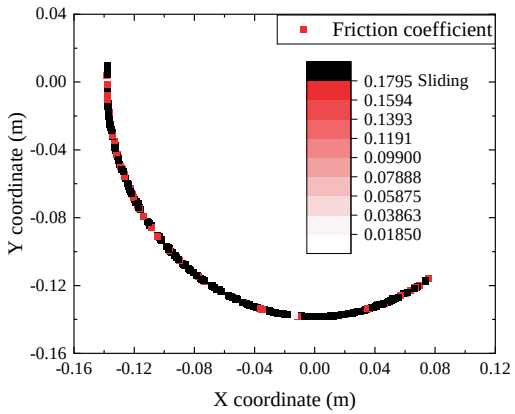
Fig. 4.13.: Particle-wall friction coefficient along drum surface for soybeans.



(a) $1.745\,\mathrm{rad\,s^{-1}}$



(b) $3.222\,\mathrm{rad\,s^{-1}}$



(c) $5.856\,\mathrm{rad\,s^{-1}}$

Fig. 4.14.: Particle-wall friction coefficient along drum surface for red beans.

5 Concluding remarks

5.1 Conclusions

This thesis has focused on nonspherical particles in discrete element method (DEM) simulations from several different perspectives. The practical simulation of nonspherical particles which are not based on spheres needs a robust contact detection algorithm as a basis. This research has firstly developed a novel contact detection algorithm between polyhedra (PH) and superquadrics (SQ). Secondly, in order to improve the efficiency of a simulation, it is essential to select a time step which is not overly conservative to ensure numerical stability. There is a lack of time step determination criteria in the literature for nonspherical particle systems; this thesis has developed such a criterion for central symmetric nonspherical particle systems, e.g., SQs. This thesis finally investigates an engineering application – rotating drums – using this newly-developed time step approach, focusing on particle–wall interaction which is lacking in previous studies. The main conclusions arising from this body of research are:

1. The novel contact detection algorithm mentioned above has been developed which is based on a previous contact detection algorithm between a SQ particle and a planar wall (Podlozhnyuk et al. (2017)). As the PH faces are finite polygons rather than infinite planes, the potential intersecting point found between an SQ and a PH face may not always be the true point of intersection. Therefore, a novel "search and return" (SR) method is used to determine contact rigorously. In addition, this algorithm has been optimised to remove any redundant steps. The time complexity of this algorithm is $O(n)$, where n is the number of faces in the PH. This algorithm has been implemented into a C++ standalone program, validated to be reliable, and is efficient.

2. A criterion for determining the critical time step for a system comprising central symmetric nonspherical particles has been established. Based on the two-particle contact framework in a previous study (Burns et al. (2019)), the contact relative translational and rotational accelerations have been analysed and the critical time step is assumed to occur when the relative translational acceleration is a maximum. The maximum relative translational acceleration occurs where particles are in tip-to-tip orthogonal contact. Based on this,

the critical time step is obtained analytically for translation-dominant scenarios with maximum relative translational acceleration (usually limiting) and rotation-dominant scenarios with maximum relative rotational acceleration. In working out critical time step for translation-dominant scenarios, an amplification matrix method is adopted apart from the conventional Belytschko criterion (Belytschko (1983)). This method assumes a linear contact model and Euler integrator, and is applicable to all underdamped, critically damped and overdamped cases, while previous approaches (e.g., Belytschko (1983) and Silbert et al. (2001)) can only cover underdamped cases. The derived critical time step increases with particle density and damping coefficient but decreased contact stiffness and particle aspect ratio for translation-dominant scenarios. For rotation-dominant scenarios, evaluation of the maximum rotational velocity of particles during the simulation is needed for deriving the critical time step.

3. Both experimental and numerical investigations of red beans and soybeans in a rotating drum have been carried out. Rather than the materials within the drum, the particle–wall interaction is the focus of the investigation. The friction coefficient for the DEM simulations was calibrated by matching the dynamic angles of repose in DEM to those in the experiments. Red beans and soybeans have different surface roughness which may be the reason why the dynamic angle of repose increases with rotating speed for soybeans but not for red beans. The particle velocity is extracted from experimental images with particle image velocimetry (PIV) and from DEM datasets using coarse-graining. However, the particle velocities extracted from PIV and DEM do not match, despite the calibration. The micro- and macro-scale behaviour of the system cannot be captured at the same time when relying solely on a macro-scale calibration. Both types of the beans slide along the drum; however, almost all the red beans slide while half of the soybeans do. This may be the reason for different dynamic angles of repose.

5.2 Suggested future work

1. While a contact detection algorithm for polyhedra and superquadrics has been developed (Chapter 2), no implementation has been carried out in DEM. In order to simulate a system with a mixture of polyhedral (blocky) and superquadric (rounded) particles, and take advantage of this algorithm, implementations of PH–PH, SQ–SQ and PH–SQ contact detection algorithms are needed in the same DEM code.

2. The time step determination criterion in Chapter 3 has many restrictions. For example, it requires the nonspherical particles to be central symmetric. Although superquadrics can fulfil this requirement, polyhedra and multisphere clumps generally do not. However, the contact framework for two nonspherical particles described before Section 3.2 is generally applicable, even to asymmetric nonspherical particles. If the moment of inertia matrix of the particle is allowed to be non-diagonal, then the study could be expanded to a system with asymmetric nonspherical particles – a more generalised case.

3. The materials used in Chapter 4 are natural beans which contain varying particle sizes and shapes. These aspects can be controlled via manufacturing particles to our specifications, e.g., by 3D printing and mould injection etc.

A

Critical time step for DEM simulations using a Hertzian contact model and Euler integrator[1]

A.1 Introduction

Burns et al. (2019) adopted a commonly used velocity Verlet integration scheme (Verlet (1967)). However, it is not known what effect the choice of integration scheme has on the critical time step. Therefore, in this section, the methodology developed by Burns et al. (2019) has been extended to an Euler integrator (Euler (1768)).

A.2 Mathematical derivation

Consider two identical spheres, A and B, of mass m, radius r, shear modulus G, Young's modulus E, Poisson's ratio ν, and moment of inertia $I = \frac{2}{5}mr^2$. These two spheres have a common contact point c. The spheres are oriented such that the branch vector joining the sphere centres, along which the normal force acts, is aligned in the Cartesian z direction. The tangential force acts in the $x - y$ plane. There is no damping in the system.

We define three additional quantities for convenience:

$$E^* = \frac{E}{2(1-\nu^2)} \tag{A.1}$$

$$G^* = \frac{G}{4(2-\nu)(1+\nu)} \tag{A.2}$$

$$r^* = \frac{r}{2} \tag{A.3}$$

[1]This appendix originates from D Peng, SJ Burns and KJ Hanley, Critical time step for DEM simulations using a Hertzian contact model and Euler integrator, Proceedings of the 8th International Conference on Discrete Element Methods (DEM8), 2019, `https://mercurylab.co.uk/dem8/wp-content/uploads/sites/4/2019/07/208.pdf`.

Using this notation and frame of reference, the relative acceleration at the contact point of the spheres during the impact phase at some time step $n+1$ can be written as

$$\begin{bmatrix} \ddot{x}_{cx,n+1} \\ \ddot{x}_{cy,n+1} \\ \ddot{x}_{cz,n+1} \end{bmatrix} = \begin{bmatrix} \frac{2r^2}{I} + \frac{2}{m} & 0 & 0 \\ 0 & \frac{2r^2}{I} + \frac{2}{m} & 0 \\ 0 & 0 & \frac{2}{m} \end{bmatrix} \begin{bmatrix} -8G^* \sqrt{r^* |x_{cz,n+1}|} x_{cx,n+1} \\ -8G^* \sqrt{r^* |x_{cz,n+1}|} x_{cy,n+1} \\ -\frac{4}{3} E^* \sqrt{r^* |x_{cz,n+1}|} x_{cz,n+1} \end{bmatrix} \tag{A.4}$$

$$= \begin{bmatrix} -\frac{14G\sqrt{r/2}}{m(2-\nu)(1+\nu)} & 0 & 0 \\ 0 & -\frac{14G\sqrt{r/2}}{m(2-\nu)(1+\nu)} & 0 \\ 0 & 0 & -\frac{4E\sqrt{r/2}}{3m(1-\nu^2)} \end{bmatrix} \begin{bmatrix} \sqrt{|x_{cz,n+1}|} x_{cx,n+1} \\ \sqrt{|x_{cz,n+1}|} x_{cy,n+1} \\ \sqrt{|x_{cz,n+1}|} x_{cz,n+1} \end{bmatrix}$$

for a Hertzian contact model. $\dot{x}_{cx/y/z}$ represents the relative velocity at the contact point while $x_{cx/y/z}$ represents the relative displacement at the contact point, i.e., x_{cz} represents the normal overlap while the particles are in contact. For the Euler integrator, the relative contact velocity and displacement can be expressed vectorially as

$$\dot{\boldsymbol{x}}_{c,n+1} = \dot{\boldsymbol{x}}_{c,n} + \Delta t \ddot{\boldsymbol{x}}_{c,n+1} \tag{A.5}$$

$$\boldsymbol{x}_{c,n+1} = \boldsymbol{x}_{c,n} + \Delta t \dot{\boldsymbol{x}}_{c,n+1} \tag{A.6}$$

where Δt is the time step. Equation (A.4) may be substituted into Equation (A.5) for the relative acceleration. It is noted that the relative displacement/velocity/acceleration at the interparticle contact is easily related to the displacement/velocity/acceleration of A and B:

$$\ddot{\boldsymbol{x}}_{c,n} = \ddot{\boldsymbol{x}}_{B,n} - \ddot{\boldsymbol{x}}_{A,n} \tag{A.7}$$

$$\dot{\boldsymbol{x}}_{c,n} = \dot{\boldsymbol{x}}_{B,n} - \dot{\boldsymbol{x}}_{A,n} \tag{A.8}$$

$$\boldsymbol{x}_{c,n} = \boldsymbol{x}_{B,n} - \boldsymbol{x}_{A,n} \tag{A.9}$$

Here the same assumptions as Burns et al. (2019) are made in order to determine the critical time step for this system. This requires considering the normal and tangential directions separately.

A.2.1 Normal direction

The impact is divided into two phases:

Phase 1 The system goes from initial contact to the point of maximum compression. We propose that at least one time step is needed to capture the dynamics of this compression phase.

Phase 2 The system returns from maximum compression to a touching contact. As for Phase 1, at least one time step is needed to capture the dynamics of Phase 2.

The boundary conditions for Phase 1 are $\dot{x}_{cz,n} = v_{z,i}$ (an initial incident velocity), $\dot{x}_{cz,n+1} = 0$ and, from Equation (A.6), $x_{cz,n} \approx x_{cz,n+1} \approx \Delta t v_{z,i}$. Using these boundary conditions in Equation (A.5), it can be obtained that

$$0 = v_{z,i} - \Delta t \frac{4E\sqrt{r/2}}{3m(1-\nu^2)} \left(\Delta t v_{z,i}\right)^{\frac{3}{2}} \tag{A.10}$$

Solving for Δt,

$$\Delta t = \left(\frac{3m(1-\nu^2)}{4E\sqrt{r v_{z,i}/2}}\right)^{\frac{2}{5}} \tag{A.11}$$

In the absence of damping, Phase 2 is symmetric to Phase 1 with the boundary conditions $\dot{x}_{cz,n} = 0$, $\dot{x}_{cz,n+1} = v_{z,i}$ as there is no energy dissipation and, from Equation (A.6), $x_{cz,n} \approx x_{cz,n+1} \approx \Delta t v_{z,i}$. Hence, Equation (A.11) is also recovered as the critical time step for Phase 2.

A.2.2 Shear direction

The proposed shear bound follows a similar logic to the normal bounds (Burns et al. (2019)): the tangential velocity cannot be brought to zero in a single time step, i.e., a velocity reversal cannot take place in the first time step after collision. In the limit of zero tangential velocity at time step $n+1$, the boundary conditions are $\dot{x}_{cx,n+1} = \dot{x}_{cy,n+1} = 0$, $x_{cj,n} \approx x_{cj,n+1} \approx \Delta t v_{j,i}$ for $j = x/y/z$ and $\dot{x}_{cj,n} = v_{j,i}$ by definition. Equation (A.5) gives

$$0 = v_{x,i} - \Delta t \frac{14G\sqrt{r/2}}{m(2-\nu)(1+\nu)} \sqrt{\Delta t v_{z,i}} \Delta t v_{x,i} \tag{A.12}$$

$$0 = v_{y,i} - \Delta t \frac{14G\sqrt{r/2}}{m(2-\nu)(1+\nu)} \sqrt{\Delta t v_{z,i}} \Delta t v_{y,i} \tag{A.13}$$

Solving for Δt,

$$\Delta t = \left(\frac{m(2-\nu)(1+\nu)}{14G\sqrt{r v_{z,i}/2}}\right)^{\frac{2}{5}} \tag{A.14}$$

A.2.3 Critical time step

We now have two equations defining the critical time step for the normal and shear directions: Equation (A.11) and Equation (A.14), respectively. Both equations have

the same dependence on particle mass, radius and impact velocity. Taking the ratio between the two and substituting $E = 2G\,(1+\nu)$:

$$\left(\frac{\Delta t_{\text{normal}}}{\Delta t_{\text{shear}}}\right)^{\frac{5}{2}} = \frac{\frac{3m(1-\nu^2)}{4E\sqrt{rv_{z,i}/2}}}{\frac{m(2-\nu)(1+\nu)}{14G\sqrt{rv_{z,i}/2}}} = \frac{21G(1-\nu)}{2E(2-\nu)} = \frac{21(1-\nu)}{4(1+\nu)(2-\nu)} \tag{A.15}$$

For physically realistic values of $-1 < \nu < 0.5$, Equation (A.15) decreases as ν increases, attaining a minimum value of $\frac{7}{6}$ when $\nu = 0.5$. Therefore, Δt_{normal} is always larger than Δt_{shear} for $-1 < \nu < 0.5$, meaning that Equation (A.14) is the critical time step. This result is the same as Burns et al. (2019). Using a different approach, Tu and Andrade (2008) also found that the critical time step for an undamped system is dictated by the shear rather than the normal bounds.

A.3 Comparison with velocity Verlet integrator

Assume values of $G = 300\,\text{MPa}$, $\nu = 0.25$, $\rho = 1000\,\text{kg}\,\text{m}^{-3}$ and $r = 1.5\,\text{mm}$ which represent an industrial powder (Gao et al. (2013)). These parameters were also used by Burns et al. (2019) in their verification. Fig. A.1 compares the critical time step given by Equation (A.14) for an Euler integrator with the undamped critical time step obtained by Burns et al. (2019) for a velocity Verlet integration scheme. Both show an identical decrease of critical time step as the impact velocity increases.

A.4 Discussion

For the undamped case considered in this section, the same critical time step is obtained (pertaining to the shear rather than the normal direction) for the Euler integrator as for the velocity Verlet integrator. The critical time step is a function of particle shear moduli, radii, densities, Poisson's ratio and the impact velocity of the particles. Increasing the impact velocity, reducing the particle density ($\propto$ mass) or increasing its stiffness reduce the critical time step. A limitation of this analysis is its restriction to two contacting spherical particles. Otsubo et al. (2017) related the critical time step to the maximum particle coordination number for a specific polydisperse system of spherical particles; it is proposed that the same relationship could be applied to the time step obtained from this two-particle analysis. The authors are currently conducting research to quantify critical time steps for two nonspherical particles using an Euler integrator.

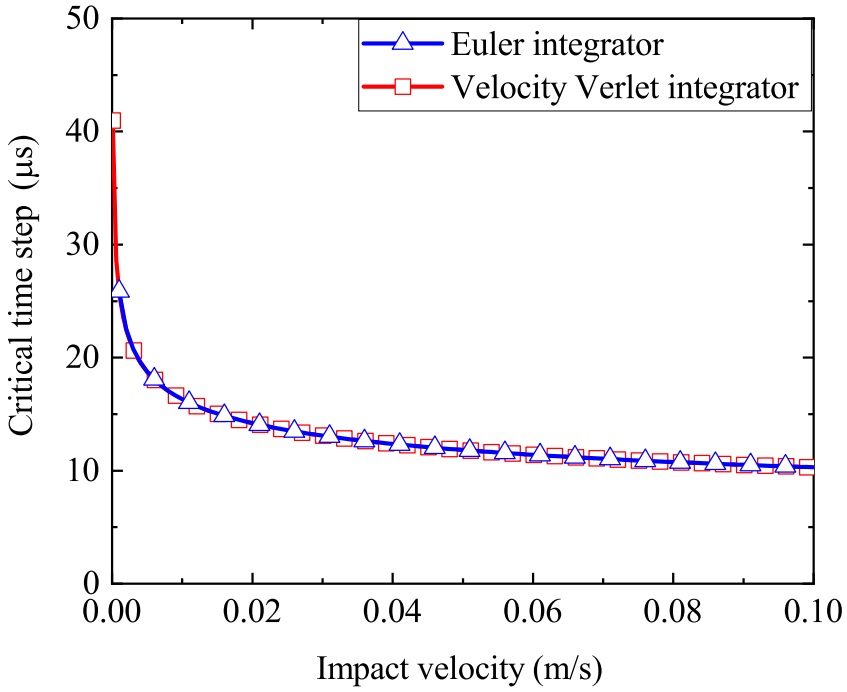


Fig. A.1.: Comparison of critical time steps as the impact velocity varies from zero to $0.1\,\mathrm{m\,s^{-1}}$ for two integration schemes: Euler as investigated in this chapter and velocity Verlet from Burns et al. (2019).

B

Possible bug in LIGGGHTS-PUBLIC 3.8.0

During the analysis in Chapter 4, one bug in LIGGGHTS-PUBLIC 3.8.0 has been found. In LIGGGHTS-PUBLIC 3.8.0, outputting contact information usually needs a "compute" command. If a compute exists, the contact model will be run twice during each time step, once for the DEM simulation and once for the compute command output, which can be regarded as a "shadow run". If sliding occurs, in the first run, the contact relative shear displacement is updated using the particle relative motion from contact resolution, and at this time the shear force will be normally resolved, as will the corresponding shear displacement. However, truncation errors will lead the resolved shear force and displacement being slightly more or less than the frictional limit. In the "shadow run", the contact resolution does not run for a second time and the shear displacement is not updated. Instead, the shear displacement keeps the resolved value after the last run. The contact model is called and if the shear displacement is slightly below the frictional limit, no sliding will be concluded and the shear force will be updated with a viscous dashpot force (the shear viscous damping only occurs when there is no sliding, see tangential history model in *LIGGGHTS(R)-PUBLIC Documentation* (2016)). Therefore, the shear force output is the damping force rather than the frictional limit, which is not correct. According to the analysis above, this bug will only occur in LIGGGHTS when the "compute" command is deployed to obtain the contact information, and it does not affect the DEM simulation. One of the easiest ways to resolve this problem without modifying the source code is to disable the shear viscous damping, and that is why the shear viscous damping is off in all the DEM simulations in this chapter. Note that the premium version of LIGGGHTS, LIGGGHTS-PREMIUM, does not have this problem.

Bibliography

3DEC - Three-Dimensional Distinct Element Code, Ver. 5.2 (2016). Itasca Consulting Group, Inc. Minneapolis: Itasca (cit. on p. 18).

Abraham, Christine L, Steve A Maas, Jeffrey A Weiss et al. (2013). "A new discrete element analysis method for predicting hip joint contact stresses". In: *Journal of Biomechanics* 46.6, pp. 1121–1127 (cit. on p. 2).

Ai, Jun, Jian-Fei Chen, J Michael Rotter and Jin Y Ooi (2011). "Assessment of rolling resistance models in discrete element simulations". In: *Powder Technology* 206.3, pp. 269–282 (cit. on pp. 5, 11, 79).

Aktulga, Hasan Metin, Joseph C Fogarty, Sagar A Pandit and Ananth Y Grama (2012). "Parallel reactive molecular dynamics: Numerical methods and algorithmic techniques". In: *Parallel Computing* 38.4-5, pp. 245–259 (cit. on p. 1).

Anand, Anshu, Jennifer S Curtis, Carl R Wassgren, Bruno C Hancock and William R Ketterhagen (2008). "Predicting discharge dynamics from a rectangular hopper using the discrete element method (DEM)". In: *Chemical Engineering Science* 63.24, pp. 5821–5830 (cit. on p. 2).

Andrade, José E, Keng-Wit Lim, Carlos F Avila and Ivan Vlahinić (2012). "Granular element method for computational particle mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 241, pp. 262–274 (cit. on pp. 13, 18).

Ayachit, Utkarsh (2015). *The ParaView Guide: A Parallel Visualization Application*. Kitware (cit. on p. 8).

Bathurst, RJ and L Rothenburg (1988). "Micromechanical Aspects of Isotropic Granular Assemblies With Linear Contact Interactions". In: *Journal of Applied Mechanics* 55.1, pp. 17–23 (cit. on p. 10).

Been, K, JHA Crooks, DE Becker and MG Jefferies (1986). "The cone penetration test in sands: part I, state parameter interpretation". In: *Géotechnique* 36.2, pp. 239–249 (cit. on p. 68).

Belytschko, Ted (1983). "An overview of semidiscretization and time integration procedures". In: *Computational Methods for Transient Analysis(A 84-29160 12-64). Amsterdam, North-Holland, 1983*, pp. 1–65 (cit. on pp. 39, 52, 57, 90).

Bergen, Gino van den (1999). "A fast and robust GJK implementation for collision detection of convex objects". In: *Journal of Graphics Tools* 4.2, pp. 7–25 (cit. on p. 14).

Bolton, Malcolm D, Meen-Wah Gui, J Garnier et al. (1999). "Centrifuge cone penetration tests in sand". In: *Géotechnique* 49.4, pp. 543–552 (cit. on p. 68).

Bolton, MD, MW Gui and R Phillips (1993). "Review of miniature soil probes for model tests". In: *Proceedings of the 11th Southeast Asian Geotechnical Conference*. South East Asian Geotechnical Society, pp. 85–90 (cit. on p. 68).

Boon, CW, GT Houlsby and S Utili (2013). "A new contact detection algorithm for three-dimensional non-spherical particles". In: *Powder Technology* 248, pp. 94–102 (cit. on pp. 13, 18, 20).

Briscoe, BJ, L Pope and MJ Adams (1984). "Interfacial friction of powders on concave counterfaces". In: *Powder Technology* 37.1, pp. 169–181 (cit. on p. 67).

Brown, Nicholas J, Jian-Fei Chen and Jin Y Ooi (2014). "A bond model for DEM simulation of cementitious materials and deformable structures". In: *Granular Matter* 16.3, pp. 299–311 (cit. on p. 6).

Burns, Shane J and Kevin J Hanley (2017). "Establishing stable time-steps for DEM simulations of non-collinear planar collisions with linear contact laws". In: *International Journal for Numerical Methods in Engineering* 110.2, pp. 186–200 (cit. on pp. 40, 53).

Burns, Shane J, Petri T Piiroinen and Kevin J Hanley (2019). "Critical time step for DEM simulations of dynamic systems using a Hertzian contact model". In: *International Journal for Numerical Methods in Engineering* 119.5, pp. 432–451 (cit. on pp. 40, 43, 55, 89, 93–97).

Buttlar, William G and Zhanping You (2001). "Discrete element modeling of asphalt concrete: microfabric approach". In: *Transportation Research Record* 1757.1, pp. 111–118 (cit. on p. 2).

Cameron, Stephen (1997a). "A comparison of two fast algorithms for computing the distance between convex polyhedra". In: *IEEE Transactions on Robotics and Automation* 13.6, pp. 915–920 (cit. on p. 14).

– (1997b). "Enhancing GJK: Computing minimum and penetration distances between convex polyhedra". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 4. IEEE, pp. 3112–3117 (cit. on p. 14).

Chen, RP, LJ Tang, DS Ling and YM Chen (2011). "Face stability analysis of shallow shield tunnels in dry sandy ground using the discrete element method". In: *Computers and Geotechnics* 38.2, pp. 187–195 (cit. on p. 1).

Cheng, YP, Y Nakata and MD Bolton (2003). "Discrete element simulation of crushable soil". In: *Géotechnique* 53.7, pp. 633–641 (cit. on p. 11).

Chopra, Anil K. (2012). *Dynamics of Structures*. Fourth Edition, International Edition. Harlow: Pearson Education (cit. on p. 52).

Chou, SH, FC Yang and SS Hsiau (2019). "Influence of interstitial fluid viscosity and particle size on creeping granular flow in a rotating drum". In: *International Journal of Multiphase Flow* 113, pp. 179–190 (cit. on p. 67).

Chu, Kaiwei, Jiang Chen and Aibing Yu (2016). "Applicability of a coarse-grained CFD-DEM model on dense medium cyclone". In: *Minerals Engineering* 90, pp. 43–54 (cit. on pp. 9, 10, 74).

Chung, YC, SS Hsiau, HH Liao and JY Ooi (2010). "An improved PTV technique to evaluate the velocity field of non-spherical particles". In: *Powder Technology* 202.1-3, pp. 151–161 (cit. on pp. 69, 70).

Cleary, Paul W and Mahesh Prakash (2004). "Discrete-element modelling and smoothed particle hydrodynamics: potential in the environmental sciences". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 362.1822, pp. 2003–2030 (cit. on p. 2).

Cui, Xilin, Jun Li, Andrew Chan and David Chapman (2014). "Coupled DEM-LBM simulation of internal fluidisation induced by a leaking pipe". In: *Powder Technology* 254, pp. 299–306 (cit. on p. 10).

Cundall, PA (1987). "Distinct element models of rock and soil structure". In: *Analytical and Computational Methods in Engineering Rock Mechanics*, pp. 129–163 (cit. on p. 6).

Cundall, Peter A (1988). "Formulation of a three-dimensional distinct element model. Part I. A scheme to detect and represent contacts in a system composed of many polyhedral blocks". In: *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* 25.3, pp. 107–116 (cit. on pp. 4, 10, 13, 18).

Cundall, Peter A and Otto D L Strack (1979). "A discrete numerical model for granular assemblies". In: *Géotechnique* 29.1, pp. 47–65 (cit. on pp. 1, 4, 10, 17, 39).

Dagum, Leonardo and Ramesh Menon (1998). "OpenMP: an industry standard API for shared-memory programming". In: *IEEE Computational Science and Engineering* 5.1, pp. 46–55 (cit. on p. 9).

Daraio, D, J Villoria, A Ingram et al. (2019). "Investigating grinding media dynamics inside a vertical stirred mill using the discrete element method: Effect of impeller arm length". In: *Powder Technology* (cit. on pp. 9, 74).

DeJong, Jason T, David J White and Mark F Randolph (2006). "Microscale observation and modeling of soil-structure interface behavior using particle image velocimetry". In: *Soils and Foundations* 46.1, pp. 15–28 (cit. on p. 70).

Dong, Kejun, Chuncheng Wang and Aibing Yu (2015). "A novel method based on orientation discretization for discrete element modeling of non-spherical particles". In: *Chemical Engineering Science* 126, pp. 500–516 (cit. on p. 14).

Dubé, Olivier, Ebrahim Alizadeh, Jamal Chaouki and François Bertrand (2013). "Dynamics of non-spherical particles in a rotating drum". In: *Chemical Engineering Science* 101, pp. 486–502 (cit. on p. 67).

EDEM 2019 User Guide (2019). DEM Solutions Ltd. https://www.edemsimulation.com/ (cit. on pp. 2, 12, 17, 63, 65).

Edwards, H. Carter, Christian R. Trott and Daniel Sunderland (2014). "Kokkos: Enabling manycore performance portability through polymorphic memory access patterns". In: *Journal of Parallel and Distributed Computing* 74.12, pp. 3202 –3216 (cit. on p. 9).

Euler, Leonhard (1768). *Institutionum Calculi Integralis*. Petropoli: Impensis Academiae Imperialis Scientiarum (cit. on pp. 7, 93).

Eurocode 1 – Actions on structures, Part 4: Actions on silos and tanks (2003). European Committee for Standardization (cit. on p. 75).

Feng, YT and DRJ Owen (2004). "A 2D polygon/polygon contact model: algorithmic aspects". In: *Engineering Computations* 21.2/3/4, pp. 265–277 (cit. on pp. 13, 17).

Feng, YT, K Han and DRJ Owen (2012). "Energy–conserving contact interaction models for arbitrarily shaped discrete elements". In: *Computer Methods in Applied Mechanics and Engineering* 205, pp. 169–177 (cit. on pp. 14, 26).

– (2017). "A generic contact detection framework for cylindrical particles in discrete element modelling". In: *Computer Methods in Applied Mechanics and Engineering* 315, pp. 632–651 (cit. on pp. 18, 26, 28).

FLAC, Version 8.1 (2020). Itasca Consulting Group, Inc. Minneapolis: Itasca (cit. on p. 10).

FLAC3D, Version 7.0 (2020). Itasca Consulting Group, Inc. Minneapolis: Itasca (cit. on p. 10).

Freireich, Ben, Jim Litster and Carl Wassgren (2009). "Using the discrete element method to predict collision-scale behavior: a sensitivity analysis". In: *Chemical Engineering Science* 64.15, pp. 3407–3416 (cit. on pp. 68, 80).

Gao, Yijie, Fani Boukouvala, William Engisch et al. (2013). "Improving continuous powder blending performance using projection to latent structures regression". In: *Journal of Pharmaceutical Innovation* 8.2, pp. 99–110 (cit. on p. 96).

Gaschler, Andre (2016). *3D Rotation Converter*. `https://github.com/gaschler/rotationconverter` (cit. on p. 44).

Genda, Eiichi, Norimasa Iwasaki, Guoan Li et al. (2001). "Normal hip joint contact pressure distribution in single-leg standing–effect of gender and anatomic parameters". In: *Journal of Biomechanics* 34.7, pp. 895–905 (cit. on p. 2).

Gilbert, Elmer G, Daniel W Johnson and S Sathiya Keerthi (1988). "A fast procedure for computing the distance between complex objects in three-dimensional space". In: *IEEE Journal on Robotics and Automation* 4.2, pp. 193–203 (cit. on p. 14).

Goldhirsch, Isaac (2010). "Stress, stress asymmetry and couple stress: from discrete particles to continuous fields". In: *Granular Matter* 12.3, pp. 239–252 (cit. on pp. 8, 74).

González-Montellano, C, F Ayuga and JY Ooi (2011). "Discrete element modelling of grain flow in a planar silo: influence of simulation parameters". In: *Granular Matter* 13.2, pp. 149–158 (cit. on p. 68).

Hart, R, P A Cundall and J Lemos (1988). "Formulation of a three-dimensional distinct element model – Part II. Mechanical calculations for motion and interaction of a system composed of many polyhedral blocks". In: *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* 25.3, pp. 117–125 (cit. on p. 40).

He, SY, JQ Gan, David Pinson, AB Yu and ZY Zhou (2019a). "Flow regimes of cohesionless ellipsoidal particles in a rotating drum". In: *Powder Technology* 354, pp. 174–187 (cit. on p. 68).

He, SY, JQ Gan, David Pinson and ZY Zhou (2019b). "Particle shape-induced radial segregation of binary mixtures in a rotating drum". In: *Powder Technology* 341, pp. 157–166 (cit. on p. 68).

He, SY, JQ Gan, D Pinson, AB Yu and ZY Zhou (2019c). "Radial segregation of binary-sized ellipsoids in a rotating drum". In: *Powder Technology* 357, pp. 322–330 (cit. on p. 68).

Hogue, C (1998). "Shape representation and contact detection for discrete element simulations of arbitrary geometries". In: *Engineering Computations* 15.3, pp. 374–390 (cit. on p. 17).

Höhner, D, S Wirtz and V Scherer (2014). "A study on the influence of particle shape and shape approximation on particle mechanics in a rotating drum using the discrete element method". In: *Powder Technology* 253, pp. 256–265 (cit. on p. 67).

Houlsby, G T (2009). "Potential particles: a method for modelling non-circular particles in DEM". In: *Computers and Geotechnics* 36.6, pp. 953–959 (cit. on pp. 13, 18, 19).

Huang, Nannan (2019). "Granular flow dynamics of non-spherical particles in rotating drums using PIV". MSc Dissertation. The University of Edinburgh (cit. on pp. 69–71, 76).

Huang, Xin, Kevin J Hanley, Catherine O'Sullivan and Chung-Yee Kwok (2017). "Implementation of rotational resistance models: a critical appraisal". In: *Particuology* 34, pp. 14–23 (cit. on pp. 11, 40).

Information technology – Programming languages – C++ (ISO/IEC 14882:2011) (2011). International Organization for Standardization (cit. on p. 35).

Iota User Manual (2019). Particle Analytics Limited (cit. on pp. 9, 74).

Iwashita, Kazuyoshi and Masanobu Oda (1998). "Rolling resistance at contacts in simulation of shear band development by DEM". In: *Journal of engineering mechanics* 124.3, pp. 285–292 (cit. on pp. 5, 11).

Jaklič, Aleš, Aleš Leonardis and Franc Solina (2000). *Segmentation and Recovery of Superquadrics*. Springer Science & Business Media (cit. on p. 63).

Jia, Xiaodong and Richard A Williams (2001). "A packing algorithm for particles of arbitrary shapes". In: *Powder Technology* 120.3, pp. 175–186 (cit. on p. 14).

Jiang, Mingjing, Zhifu Shen and Jianfeng Wang (2015). "A novel three-dimensional contact model for granulates incorporating rolling and twisting resistances". In: *Computers and Geotechnics* 65, pp. 147–163 (cit. on p. 11).

Jing, L, CY Kwok, YF Leung and YD Sobral (2016). "Extended CFD-DEM for free-surface flow with multi-size granules". In: *International journal for numerical and analytical methods in geomechanics* 40.1, pp. 62–79 (cit. on p. 10).

Johnstone, Mical William (2010). "Calibration of DEM models for granular materials using bulk physical tests". PhD thesis. The University of Edinburgh (cit. on p. 75).

Kloss, Christoph, Christoph Goniva, Alice Hager, Stefan Amberger and Stefan Pirker (2012). "Models, algorithms and validation for opensource DEM and CFD-DEM". In: *Progress in Computational Fluid Dynamics, an International Journal* 12.2–3, pp. 140–152 (cit. on pp. 2, 11, 17, 18, 67, 74).

Kodam, Madhusudhan, Jennifer Curtis, Bruno Hancock and Carl Wassgren (2012). "Discrete element method modeling of bi-convex pharmaceutical tablets: contact detection algorithms and validation". In: *Chemical Engineering Science* 69.1, pp. 587–601 (cit. on p. 12).

Kruggel-Emden, H, S Rickelt, S Wirtz and V Scherer (2008). "A study on the validity of the multi-sphere Discrete Element Method". In: *Powder Technology* 188.2, pp. 153–165 (cit. on p. 11).

Lee, Chung-Jung, Yu-Chen Wei and Yu-Chieh Kuo (2012). "Boundary effects of a laminar container in centrifuge shaking table tests". In: *Soil Dynamics and Earthquake Engineering* 34.1, pp. 37–51 (cit. on p. 68).

Lee, Say Yong (1990). "Centrifuge modelling of cone penetration testing in cohesionless soils". PhD thesis. University of Cambridge (cit. on p. 68).

Lee, Seyong, Seung-Jai Min and Rudolf Eigenmann (2009). "OpenMP to GPGPU: a compiler framework for automatic translation and optimization". In: *ACM Sigplan Notices* 44.4, pp. 101–110 (cit. on p. 9).

Lemos, José V (2007). "Discrete element modeling of masonry structures". In: *International Journal of Architectural Heritage* 1.2, pp. 190–213 (cit. on p. 1).

Leonardi, Alessandro, Falk K Wittel, Miller Mendoza and Hans J Herrmann (2014). "Coupled DEM-LBM method for the free-surface simulation of heterogeneous suspensions". In: *Computational Particle Mechanics* 1.1, pp. 3–13 (cit. on p. 10).

Li, R, H Yang, G Zheng and QC Sun (2018). "Granular avalanches in slumping regime in a 2D rotating drum". In: *Powder Technology* 326, pp. 322–326 (cit. on p. 67).

Li, Yanjie, Yong Xu and Colin Thornton (2005). "A comparison of discrete element simulations and experiments for 'sandpiles' composed of spherical particles". In: *Powder Technology* 160.3, pp. 219–228 (cit. on p. 40).

LIGGGHTS(R)-PUBLIC Documentation (2016). Version 3.X. DCS Computing GmbH. https://www.cfdem.com/media/DEM/docu/Manual.html (cit. on pp. 19, 28, 32, 36, 61, 75, 79, 99).

Lin, Xiaoshan and T-T Ng (1997). "A three-dimensional discrete element model using arrays of ellipsoids". In: *Géotechnique* 47.2, pp. 319–329 (cit. on p. 12).

Lin, Xiaoshan and Tang-Tat Ng (1995). "Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 19.9, pp. 653–659 (cit. on pp. 13, 18).

Lisitsa, Vadim V, Vladimir A Tcheverda and Victoria V Volianskaia (2018). "GPU-based implementation of discrete element method for simulation of the geological fault geometry and position". In: *Supercomputing Frontiers and Innovations* 5.3, pp. 46–50 (cit. on p. 9).

Liu, Chun, David D Pollard and Bin Shi (2013). "Analytical solutions and numerical tests of elastic and failure behaviors of close-packed lattice for brittle rocks and crystals". In: *Journal of Geophysical Research: Solid Earth* 118.1, pp. 71–82 (cit. on p. 2).

Liu, Chun, David D Pollard, Kai Gu and Bin Shi (2015). "Mechanism of formation of wiggly compaction bands in porous sandstone: 2. Numerical simulation using discrete element method". In: *Journal of Geophysical Research: Solid Earth* 120.12, pp. 8153–8168 (cit. on p. 2).

Liu, Chun, Qiang Xu, Bin Shi, Shang Deng and Honghu Zhu (2017). "Mechanical properties and energy conversion of 3D close-packed lattice model for brittle rocks". In: *Computers & Geosciences* 103, pp. 12–20 (cit. on p. 2).

Liu, Quansheng, Weiqin Wang and Hao Ma (2020). "Parallelized combined finite-discrete element (FDEM) procedure using multi-GPU with CUDA". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 44.2, pp. 208–238 (cit. on p. 10).

Lu, G, JR Third and CR Müller (2012). "Critical assessment of two approaches for evaluating contacts between super-quadric shaped particles in DEM simulations". In: *Chemical Engineering Science* 78, pp. 226–235 (cit. on pp. 13, 18).

Lu, G, J R Third and C R Müller (2015). "Discrete element models for non-spherical particle systems: from theoretical developments to applications". In: *Chemical Engineering Science* 127, pp. 425–465 (cit. on pp. 10, 12, 17, 40).

Markauskas, Darius, Rimantas Kačianauskas, Algis Džiugys and Robertas Navakas (2010). "Investigation of adequacy of multi-sphere approximation of elliptical particles for DEM simulations". In: *Granular Matter* 12.1, pp. 107–123 (cit. on p. 11).

MATLAB version 9.5.0 (R2018b) (2018). The MathWorks Inc. (cit. on p. 2).

McDowell, GR, O Harireche, H Konietzky, SF Brown and NH Thom (2006). "Discrete element modelling of geogrid-reinforced aggregates". In: *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering* 159.1, pp. 35–48 (cit. on p. 1).

Mehlhorn, Kurt (2008). *Algorithms and Data Structures: The Basic Toolbox*. Berlin: Springer (cit. on p. 29).

Mellmann, Jochen (2001). "The transverse motion of solids in rotating cylinders–forms of motion and transition behavior". In: *Powder Technology* 118.3, pp. 251–270 (cit. on p. 68).

Miao, Zhen, Zongyan Zhou, AB Yu and Yansong Shen (2017). "CFD-DEM simulation of raceway formation in an ironmaking blast furnace". In: *Powder Technology* 314, pp. 542–549 (cit. on p. 10).

Mindlin, Raymond D (1953). "Elastic spheres in contact under varying oblique forces". In: *Journal of Applied Mechanics* 20, pp. 327–344 (cit. on p. 5).

Mustoe, GGW and M Miyata (2001). "Material flow analyses of noncircular-shaped granular media using discrete element methods". In: *Journal of Engineering Mechanics* 127.10, pp. 1017–1026 (cit. on pp. 13, 18).

Nezami, Erfan G, Youssef MA Hashash, Dawei Zhao and Jamshid Ghaboussi (2006). "Shortest link method for contact detection in discrete element method". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 30.8, pp. 783–801 (cit. on pp. 13, 18).

Nezami, Erfan G, Youssef M A Hashash, Dawei Zhao and Jamshid Ghaboussi (2007). "Simulation of front end loader bucket–soil interaction using discrete element method". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 31.9, pp. 1147–1162 (cit. on p. 17).

Niedostatkiewicz, Maciej and Jacek Tejchman (2007). "Investigations of porosity changes during granular silo flow using Electrical Capacitance Tomography (ECT) and Particle Image Velocimetry (PIV)". In: *Particle & Particle Systems Characterization* 24.4-5, pp. 304–312 (cit. on p. 70).

Noid, William George, Jhih-Wei Chu, Gary S Ayton et al. (2008a). "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models". In: *The Journal of Chemical Physics* 128.24, p. 244114 (cit. on pp. 8, 74).

Noid, William George, Pu Liu, Yanting Wang et al. (2008b). "The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models". In: *The Journal of Chemical Physics* 128.24, p. 244115 (cit. on pp. 8, 74).

Nova, Roberto (2010). *Soil mechanics*. London: ISTE – Wiley (cit. on p. 68).

Ong, Chong Jin and Elmer G Gilbert (1997). "The Gilbert-Johnson-Keerthi distance algorithm: A fast version for incremental motions". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 2. IEEE, pp. 1183–1189 (cit. on p. 14).

O'Sullivan, Catherine (2014). *Particulate Discrete Element Modelling: A Geomechanics Perspective*. CRC Press (cit. on p. 2).

O'Sullivan, Catherine and Jonathan D Bray (2004). "Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme". In: *Engineering Computations* 21.2/3/4, pp. 278–303 (cit. on pp. 39, 55).

Otsubo, Masahide, Catherine O'Sullivan and Tom Shire (2017). "Empirical assessment of the critical time increment in explicit particulate discrete element method simulations". In: *Computers and Geotechnics* 86, pp. 67–79 (cit. on pp. 40, 44, 55, 96).

PFC - Particle Flow Code, Version 6.0 (2018). Itasca Consulting Group, Inc. Minneapolis: Itasca (cit. on pp. 2, 8, 11, 17, 40, 43, 63, 65).

Plimpton, Steve (1995). "Fast parallel algorithms for short-range molecular dynamics". In: *Journal of Computational Physics* 117.1, pp. 1–19 (cit. on pp. 1, 2).

Podlozhnyuk, Alexander, Stefan Pirker and Christoph Kloss (2017). "Efficient implementation of superquadric particles in Discrete Element Method within an open-source framework". In: *Computational Particle Mechanics* 4.1, pp. 101–118 (cit. on pp. 4, 13, 18–21, 26, 28, 74, 89).

Pöschel, Thorsten and Volkhard Buchholtz (1995). "Complex flow of granular material in a rotating cylinder". In: *Chaos, Solitons & Fractals* 5.10, pp. 1901–1912 (cit. on p. 67).

Potyondy, David O and PA Cundall (2004). "A bonded-particle model for rock". In: *International Journal of Rock Mechanics and Mining Sciences* 41.8, pp. 1329–1364 (cit. on p. 5).

Programming languages – C (ISO/IEC 9899:1990) (1990). International Organization for Standardization (cit. on p. 35).

Puppala, Anand J, Yalcin B Acar and Mehmet T Tumay (1995). "Cone penetration in very weakly cemented sand". In: *Journal of Geotechnical Engineering* 121.8, pp. 589–600 (cit. on p. 68).

Qi, Ji, Kuan-Ching Li, Hai Jiang, Qingguo Zhou and Lei Yang (2015). "GPU-accelerated DEM implementation with CUDA". In: *International Journal of Computational Science and Engineering* 11.3, pp. 330–337 (cit. on p. 9).

Rajchenbach, Jean (1990). "Flow in powders: From discrete avalanches to continuous regime". In: *Physical Review Letters* 65.18, p. 2221 (cit. on pp. 67, 68).

Rapaport, Dennis C (2004). *The art of molecular dynamics simulation*. Cambridge University Press (cit. on p. 3).

Rhodes, Martin J, XS Wang, M Nguyen, P Stewart and Kurt Liffman (2001). "Use of discrete element method simulation in studying fluidization characteristics: influence of interparticle force". In: *Chemical Engineering Science* 56.1, pp. 69–76 (cit. on p. 2).

Rocky DEM Software Overveiw (cit. on p. 10).

Rojek, Jerzy, Szymon Nosewicz, Kamila Jurczak et al. (2016). “Discrete element simulation of powder compaction in cold uniaxial pressing with low pressure”. In: *Computational Particle Mechanics* 3.4, pp. 513–524 (cit. on p. 40).

Rotter, JM, JMFG Holst, JY Ooi and AM Sanad (1998). “Silo pressure predictions using discrete–element and finite–element analyses”. In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356.1747, pp. 2685–2712 (cit. on p. 68).

Rougier, E, A Munjiza and N W M John (2004). “Numerical comparison of some explicit time integration schemes used in DEM, FEM/DEM and molecular dynamics”. In: *International Journal for Numerical Methods in Engineering* 61.6, pp. 856–879 (cit. on p. 39).

Sakai, Mikio and Seiichi Koshizuka (2009). “Large-scale discrete element modeling in pneumatic conveying”. In: *Chemical Engineering Science* 64.3, pp. 533–539 (cit. on p. 2).

Sanad, AM, JY Ooi, JMFG Holst and JM Rotter (2001). “Computations of granular flow and pressures in a flat-bottomed silo”. In: *Journal of Engineering Mechanics* 127.10, pp. 1033–1043 (cit. on p. 68).

Schrader, Marcel, Kathrin Pommerehne, Silas Wolf et al. (2019). “Design of a CFD-DEM-based method for mechanical stress calculation and its application to glass bead-enhanced cultivations of filamentous Lentzea aerocolonigenes”. In: *Biochemical Engineering Journal* 148, pp. 116–130 (cit. on p. 10).

Schroeder, Will, Ken Martin and Bill Lorensen (2006). *The Visualization Toolkit (4th ed.)* Kitware (cit. on p. 8).

Sedgewick, Robert (1984). *Algorithms*. Reading, Massachusetts: Addison-Wesley (cit. on p. 29).

Silbert, Leonardo E, Deniz Ertaş, Gary S Grest et al. (2001). “Granular flow down an inclined plane: Bagnold scaling and rheology”. In: *Physical Review E* 64.5, p. 051302 (cit. on pp. 53, 57, 90).

Song, Yongxin, Richard Turton and Ferhan Kayihan (2006). “Contact detection algorithms for DEM simulations of tablet-shaped particles”. In: *Powder Technology* 161.1, pp. 32–40 (cit. on pp. 11, 12).

Sousani, Marina, Andrew M Hobbs, Adam Anderson and Richard Wood (2019). “Accelerated heat transfer simulations using coupled DEM and CFD”. In: *Powder Technology* 357, pp. 367–376 (cit. on pp. 9, 10).

Stanier, Sam A, Justin Blaber, W Andy Take and DJ White (2016). “Improved image-based deformation measurement for geotechnical applications”. In: *Canadian Geotechnical Journal* 53.5, pp. 727–739 (cit. on p. 70).

Stone, JE, D Gohara and G Shi (2010). “OpenCL: A parallel programming standard for heterogeneous computing systems”. In: *Computing in Science Engineering* 12.3, pp. 66–73 (cit. on p. 9).

Sun, Rui and Heng Xiao (2016). “SediFoam: A general-purpose, open-source CFD-DEM solver for particle-laden flow with emphasis on sediment transport”. In: *Computers & Geosciences* 89, pp. 207–219 (cit. on p. 10).

Sun, Wai Ching, Matthew R Kuhn and John W Rudnicki (2013). "A multiscale DEM-LBM analysis on permeability evolutions inside a dilatant shear band". In: *Acta Geotechnica* 8.5, pp. 465–480 (cit. on p. 10).

Sun, Wei, Kepeng Hou, Zhiquan Yang and Yiming Wen (2017). "X-ray CT three-dimensional reconstruction and discrete element analysis of the cement paste backfill pore structure under uniaxial compression". In: *Construction and Building Materials* 138, pp. 69–78 (cit. on p. 2).

Tausendschön, Josef, Jari Kolehmainen, Sankaran Sundaresan and Stefan Radl (2020). "Coarse graining Euler-Lagrange simulations of cohesive particle fluidization". In: *Powder Technology* (cit. on pp. 9, 74).

Tavarez, Federico A and Michael E Plesha (2007). "Discrete element method for modelling solid and particulate materials". In: *International Journal for Numerical Methods in Engineering* 70.4, pp. 379–404 (cit. on pp. 40, 55).

Taylor, Lee M and Dale S Preece (1992). "Simulation of blasting induced rock motion using spherical element models". In: *Engineering Computations* 9.2, pp. 243–252 (cit. on p. 10).

Thornton, Anthony, Thomas Weinhart, Stefan Luding and Onno Bokhove (2012). "Modeling of particle size segregation: calibration using the discrete particle method". In: *International Journal of Modern Physics C* 23.08, p. 1240014 (cit. on p. 2).

Thornton, C and CW Randall (1988). *Applications of theoretical contact mechanics to solid particle system simulation, in Satake, M and Jenkins, JT (Eds.), Micromechanics of Granular Materials*. Elsevier (cit. on p. 40).

Thornton, C, SJ Cummins and PW Cleary (2011). "An investigation of the comparative behaviour of alternative contact force models during elastic collisions". In: *Powder Technology* 210, pp. 189–197 (cit. on p. 55).

Thornton, Colin (2000). "Numerical simulations of deviatoric shear deformation of granular media". In: *Géotechnique* 50.1, pp. 43–53 (cit. on p. 1).

Tijskens, Engelbert, Herman Ramon and Josse De Baerdemaeker (2003). "Discrete element modelling for process simulation in agriculture". In: *Journal of Sound and Vibration* 266.3, pp. 493–514 (cit. on p. 2).

Ting, John M, Brent T Corkum, Claudia R Kauffman and Carlo Greco (1989). "Discrete numerical model for soil mechanics". In: *Journal of Geotechnical Engineering* 115.3, pp. 379–398 (cit. on p. 10).

Tsuji, Y, T Kawaguchi and T Tanaka (1993). "Discrete particle simulation of two-dimensional fluidized bed". In: *Powder Technology* 77.1, pp. 79–87 (cit. on p. 40).

Tu, Xuxin and José E Andrade (2008). "Criteria for static equilibrium in particulate mechanics computations". In: *International Journal for Numerical Methods in Engineering* 75.13, pp. 1581–1606 (cit. on pp. 40, 55, 96).

Ullah, Shah Neyamat, Yu Xia Hu, David White and Samuel Stanier (2014). "Lateral boundary effect in centrifuge tests for spudcan penetration in uniform clay". In: *Applied Mechanics and Materials*. Vol. 553. Trans Tech Publ, pp. 458–463 (cit. on p. 68).

Ullah, Shah Neyamat, Yuxia Hu, Samuel Stanier and David White (2017). "Lateral boundary effects in centrifuge foundation tests". In: *International Journal of Physical Modelling in Geotechnics* 17.3, pp. 144–160 (cit. on p. 68).

Varas, AE Carlos, EAJF Peters and JAM Kuipers (2017). "CFD-DEM simulations and experimental validation of clustering phenomena and riser hydrodynamics". In: *Chemical Engineering Science* 169, pp. 246–258 (cit. on p. 10).

Verlet, Loup (1967). "Computer" experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules". In: *Physical Review* 159.1, p. 98 (cit. on pp. 7, 54, 93).

Walker, David W (1992). *Standards for message-passing in a distributed memory environment*. Tech. rep. Oak Ridge National Lab., TN (United States) (cit. on p. 9).

Wang, Jianfeng and Haibin Yan (2012). "DEM analysis of energy dissipation in crushable soils". In: *Soils and Foundations* 52.4, pp. 644–657 (cit. on p. 11).

Wang, Yin, Yong Lu and Jin Y Ooi (2013). "Numerical modelling of dynamic pressure and flow in hopper discharge using the Arbitrary Lagrangian–Eulerian formulation". In: *Engineering structures* 56, pp. 1308–1320 (cit. on p. 68).

Weinhart, Thomas, Anthony R Thornton, Stefan Luding and Onno Bokhove (2012). "From discrete particles to continuum fields near a boundary". In: *Granular Matter* 14.2, pp. 289–294 (cit. on p. 2).

Weinhart, Thomas, Carlos Labra, Stefan Luding and Jin Y Ooi (2016). "Influence of coarse-graining parameters on the analysis of DEM simulations of silo flow". In: *Powder Technology* 293, pp. 138–148 (cit. on pp. 9, 74).

Wensrich, CM and A Katterfeld (2012). "Rolling friction as a technique for modelling particle shape in DEM". In: *Powder Technology* 217, pp. 409–417 (cit. on p. 11).

White, DJ, WA Take and MD Bolton (2003). "Soil deformation measurement using particle image velocimetry (PIV) and photogrammetry". In: *Géotechnique* 53.7, pp. 619–631 (cit. on p. 70).

Whitman, Robert V and Philip C Lambe (1986). "Effect of boundary conditions upon centrifuge experiments using ground motion simulation". In: *Geotechnical Testing Journal* 9.2, pp. 61–71 (cit. on p. 68).

Williams, John R and Ruaidhrí O'Connor (1999). "Discrete element simulation and the contact problem". In: *Archives of Computational Methods in Engineering* 6.4, pp. 279–304 (cit. on pp. 13, 17).

Williams, John R and Ruaidhri O'Connor (1995). "A linear complexity intersection algorithm for discrete element simulation of arbitrary geometries". In: *Engineering Computations* 12.2, pp. 185–201 (cit. on pp. 13, 17).

Williams, John R and Alex P Pentland (1992). "Superquadrics and modal dynamics for discrete elements in interactive design". In: *Engineering Computations* 9.2, pp. 115–127 (cit. on pp. 13, 17).

Wood, W. L. (1990). *Practical time-stepping schemes*. Oxford: Clarendon Press (cit. on p. 39).

Wu, Hao, Nan Gui, Xingtuan Yang, Jiyuan Tu and Shengyao Jiang (2017). "Numerical simulation of heat transfer in packed pebble beds: CFD-DEM coupled with particle thermal radiation". In: *International Journal of Heat and Mass Transfer* 110, pp. 393–405 (cit. on p. 10).

Xie, Changhua, Huaqing Ma and Yongzhi Zhao (2019a). “Investigation of modeling non-spherical particles by using spherical discrete element model with rolling friction”. In: *Engineering Analysis with Boundary Elements* 105, pp. 207–220 (cit. on p. 11).

Xie, Zhouzun, Yansong Shen, Kazuya Takabatake, Akira Yamaguchi and Mikio Sakai (2019b). “Coarse-grained DEM study of solids sedimentation in water”. In: *Powder Technology* (cit. on pp. 9, 74).

Xu, Huibin, Wenqi Zhong, Zhulin Yuan and AB Yu (2017). “CFD-DEM study on cohesive particles in a spouted bed”. In: *Powder Technology* 314, pp. 377–386 (cit. on p. 10).

Xu, Ji, Huabiao Qi, Xiaojian Fang et al. (2011). “Quasi-real-time simulation of rotating drum using discrete element method with parallel GPU computing”. In: *Particuology* 9.4, pp. 446–450 (cit. on p. 3).

Xu, W and H Chen (2012). “Mesostructural characterization of particulate composites via a contact detection algorithm of ellipsoidal particles”. In: *Powder Technology* 221, pp. 296–305 (cit. on p. 18).

Yade Documentation 2nd ed. (2015). The Yade Project. http://yade-dem.org/doc/ (cit. on pp. 2, 17).

Yamamoto, Michinori, Shingo Ishihara and Junya Kano (2016). “Evaluation of particle density effect for mixing behavior in a rotating drum mixer by DEM simulation”. In: *Advanced Powder Technology* 27.3, pp. 864–870 (cit. on p. 67).

Yan, Beichuan and Richard Regueiro (2018). “Comparison between O (n2) and O (n) neighbor search algorithm and its influence on superlinear speedup in parallel discrete element method (DEM) for complex-shaped particles”. In: *Engineering Computations* (cit. on p. 3).

You, Zhanping and WG Buttlar (2004). “Discrete element modeling to predict the modulus of asphalt concrete mixtures”. In: *Journal of Materials in Civil Engineering* 16.2, pp. 140–146 (cit. on p. 2).

Zhang, Lanyue, Zhaochen Jiang, Fabian Weigler et al. (2020). “PTV measurement and DEM simulation of the particle motion in a flighted rotating drum”. In: *Powder Technology* 363, pp. 23–37 (cit. on pp. 67, 68, 70).

Zhao, Dawei, Erfan G Nezami, Youssef MA Hashash and Jamshid Ghaboussi (2006). “Three-dimensional discrete element simulation for granular materials”. In: *Engineering Computations* (cit. on p. 3).

Zheng, Qijun, Liang Bai, Liuyimei Yang and Aibing Yu (2019). “110th anniversary: continuum modeling of granular mixing in a rotating drum”. In: *Industrial & Engineering Chemistry Research* 58.41, pp. 19251–19262 (cit. on p. 68).

Zhong, Wenqi, Aibing Yu, Xuejiao Liu, Zhenbo Tong and Hao Zhang (2016). “DEM/CFD-DEM modelling of non-spherical particulate systems: theoretical developments and applications”. In: *Powder Technology* 302, pp. 108–152 (cit. on p. 10).

Zhong, Z, JY Ooi and JM Rotter (2001). “The sensitivity of silo flow and wall stresses to filling method”. In: *Engineering structures* 23.7, pp. 756–767 (cit. on p. 68).

Zhu, Z, W Xu and H Chen (2019). “The fraction of overlapping interphase around 2D and 3D polydisperse non-spherical particles: Theoretical and numerical models”. In: *Computer Methods in Applied Mechanics and Engineering* 345, pp. 728–747 (cit. on p. 18).