# *Simulating hot asphalt compaction in Bullet Physics*

Discrete modelling of the superpave gyratory compaction process
by implementation of Burgers' contact model in Bullet Physics

*Mark Jol*

TU Delft

# *Simulating hot asphalt compaction in Bullet Physics*

Discrete modelling of the superpave gyratory compaction process
by implementation of Burgers' contact model in Bullet Physics

By

*Mark Jol*

in partial fulfilment of the requirements for the degree of

*Master of Science*
in Civil Engineering

at the Delft University of Technology,
to be defended publicly on Wednesday 10 August 2022, at 13:00.

| **Student number** | **Project duration** |
|---|---|
| 4536819 | September 2021 – August 2022 |

**Thesis committee**

| | |
|---|---|
| ir. Cor Kasbergen | Delft University of Technology, supervisor |
| dr. Kumar Anupam | Delft University of Technology, supervisor |
| dr. Valeri Markine | Delft University of Technology |
| dr. Loay Al Khateeb | Norwegian University of Science and Technology |

Image front page: unsplash.com/@sidrus

## *Preface*

First of all, I would like to thank Cor Kasbergen and Kumar Anupam, for I owe it to them that I could graduate on such an amazing subject. I am very grateful for their support and guidance throughout the academic year. I hope that they will further build on the knowledge acquired during this research, and that my contribution to the academic world will be appreciated. I also would like to thank Loay Al Khateeb and Valeri Markine for being committee members, as well as their valuable feedback towards the end of the graduation process.

The image on the front cover is not an arbitrary one. Although the picture contains a road and thus represents asphalt, its meaning is much deeper. The road represents the journey ahead. The end of the road cannot be seen, meaning that the destination is as yet unknown. The road is surrounded by grass, plants and trees, representing the essential connection with nature. The destination shapes the path. I shape the destination.

A regularly asked question is, what will I do after I graduate? And the most common choice would be an office job. But this is not for me. I did not come here to work. So, what will I do? My main purpose here is do develop myself, and how better to do that than by travelling. I will take the road to the unknown. I will take the road to nowhere. That is, the path to Now Here.

I've had a wonderful adventure here at Delft University of Technology, and I am ready to take on a new one. There is one being I've yet to thank, and that is the Universal Intelligence. I am very grateful for the moment to moment guidance I've received and still will receive. For the path to the unknown is ultimately not unknown.

*Mark Jol*
July 2022

*Can you remain unmoving*
*till the right action arises by itself?*

*~ Lao Tzu*

## Contents

## Abstract

Asphalt concrete is the most used material in road construction. Analysing the behaviour and properties of this material is vital for modern day society. Currently, this is mostly done in labs using actual asphalt mixtures. However, this is a very expensive and time consuming process. An upcoming alternative approach is the use of computational models. The finite (FEM) and discrete element methods (DEM) have been used in the past, but both of these have shown significant disadvantages regarding the modelling of discrete particle movement and shape, respectively. An upcoming alternative is the use of physics engines, such as Bullet Physics, to model porous media. This method can have substantial benefits in terms of costs and research time. Also, phenomena can be visualised that cannot easily be observed in experiments.

This research focusses on the utility of Bullet Physics for modelling hot asphalt compaction. Therefore, a complex contact model is implemented which can describe the contact forces of the bituminous mixture. The superpave gyratory compaction process is digitally modelled. The model has been programmed in PyBullet, an open source physics engine, programmable in Python. A parametric study has been performed, which reveals the significance of certain properties, which cannot easily be investigated during laboratory compaction.

Bullet Physics was not initially designed for scientific research in the field of structural engineering. Therefore, alterations are needed to make the software usable. The implementation of a complex contact model is challenging. Although the Burgers' contact forces can be correctly described, it proves difficult to implement custom contact forces directly in PyBullet. Two attempts have been made. In the case of a custom integration scheme, the computation time proved too long to be applicable for large scale simulations. In case of a direct implementation in Bullet Physics with the application of external forces, instability occurs. The only correct way to implement a custom contact model is by altering the source code.

During the simulations with a simpler contact model, substantial improvements of digital simulations over actual experiments are presented. The consistency proves far better than the prescribed minimum. The influence of inertia and friction can be assessed. It turns out that inertia, as well as the friction of the mould, can be disregarded. Another phenomenon that can be clearly illustrated is the revolving of aggregates inside the mould. Further analysis has shown that the average contact area depends on degradation, but in a typical asphalt mixture does not depend on the size of the individual elements, and could rather be treated as a constant. Further analysis shows that Bullet Physics is incredibly efficient, thus yielding the possibility of performing large scale simulations within reasonable time.

# 1 Introduction

## 1.1 Relevance

Compaction is an essential part of the pavement quality [1]. Mixture characteristics are performance based on the air voids content [2]. If the compaction is insufficient, this leads to lower strength and a higher air void percentage, in which moisture can penetrate, leading to damage during winter [1]. To study asphalt compaction, various laboratory experiments exist [3]. One of these is the superpave gyratory compaction method [2]. During this process, the degradation and temperature are well maintained, whereas in the field this can vary greatly over time as well as over the volume [4].

In order to better understand the compaction process and save valuable resources, many researchers have attempted to digitally model this process [5, 6, 7, 8]. Some researchers used the finite element method, which is an efficient method for analysing continuous materials [9, 10, 11, 12]. However, asphalt mixtures consist of discrete construction aggregate elements, coated in bitumen, and are therefore not continuous. The discrete element method on the other hand generally simulates simple spheres, which can accurately model aggregate movement, but not its shape. Both the finite and discrete element method are known to be useful, but also computationally expensive methods. Very powerful computers are employed, which can still require considerable computation time, thus defeating its purpose. A new and upcoming alternative is the use of physics engines. These can best be compared with the discrete element method, but with custom shapes for the particles. They are also known to be very computationally efficient, and are already used to investigate the behaviour of granular material [13, 14, 15]. Therefore, it might prove useful to investigate the utility of physics tools for analysing hot asphalt compaction.

Previous research shows promising results regarding digital simulations [5–12]. Al Khateeb et al. [8] simulated the micromechanical behaviour of asphalt mixtures using the discrete element method, and incorporated a new contact model. This was applied to simulate and validate the overall process of the superpave gyratory compaction laboratory tests, as well as field compaction. The model was employed to assess the air voids distribution showing mixed agreement. Komaragiri et al. [16] simulated laboratory compaction using Bullet Physics [17]. Viscous damping and cohesive forces were added manually to the model. PyBullet [18], the Python implementation of Bullet Physics [17], does not assess the contact area of two colliding elements [18]. Therefore Komaragiri et al. [16] estimated the contact area based on the surface area of the smallest particle. Although the results were not validated, this shows that also physics engines have great potential.

Based on literature studies, it could that the ideal simulation tool has not yet been developed. The finite element method is unable to accurately model particle movement. On the other hand, most discrete element tools suffer from simplicity in particle shape [7, 8]. Both DEM and FEM are often carried out with commercial packages, which could be financially expensive, requiring the user to limit the size and time of the simulation.

Physics Engines seem to be a viable alternative (some of them are freeware), however, these are mostly developed for simple animations in video games.

## 1.2 Research goal

The main research goal of this report is to assess the utility of Bullet Physics for analysing hot asphalt compaction, and more specifically, for analysing the superpave gyratory compaction process. In order to describe the contact behaviour of the bituminous mixture, Burgers' contact model is implemented. Furthermore, a geometrical model of the superpave gyratory compaction process is created, and the advantages of a simulation are mapped out. This research focusses on the laboratory compaction, and may serve as a basis for field compaction for future research.

## 1.3 Research questions

Keeping the research goal as described above in view, the following research questions were created.

1. Can a complex contact model, such as the Burgers' contact model, be implemented directly in PyBullet?
2. Is Bullet Physics able to simulate the gyratory compaction process, and hot asphalt compaction in general?
3. Does Bullet Physics hold potential for future research when analysing hot asphalt compaction?

## 1.4 Structure

These research questions will be investigated, and the outcomes will be presented in the different chapters of the report. The structure of the report is as follows.

Chapter 2 performs a literature review. This chapter introduces the subject, and the field, in depth. Also, a table is presented with the definitions of physical quantities used throughout the report. This chapter further studies the gyratory compaction process, reviews previous research, explores the utilities of Bullet Physics, and presents some technical details supporting the research.

Chapter 3 explores Burgers' contact model. Also, an analytical and numerical solution are presented. A novelty is also presented: a semi-analytical solution, where the contact forces are analytically solved over a small time step. A detailed guide on how to implement these models is also presented.

Chapter 4 describes the digital model. Not only does it describe the geometry of the aggregates, mortar layer, and digital mould, but it also presents a workable implementation, as well as detailed descriptions of the aggregate placement and post-processing.

Chapter 5 verifies the contact model by comparison of the numerical solution with the analytical one in a simple experiment, and shows that the model is correct. This chapter also suggests an improvement for Burgers' contact model.

Chapter 6 presents the results and possible applications of the digital gyratory compaction model. Also, substantial benefits of simulations are presented. Phenomena that cannot easily be seen or investigated in actual experiments are presented in this chapter.

Chapter 7 briefly presents the conclusions by answering the research questions, and elaborates on opportunities for further research.

Appendix A presents an exercise for the reader to program the discrete element method in Python.

Appendix B presents a part of the programming code by which the reader can perform the numerical simulations.

Appendices C, D and E present technical details for further implementation.

Appendix F presents the derivation for the equation of motion for an improved contact model.

It is highlighted that for easier understanding of the reader, some material related to different phases of the research are presented as videos [19].

# 2 Literature review

This chapter gives a comprehensive overview of all relevant information that can be found on computationally modelling the superpave gyratory compaction process in Bullet Physics.

## 2.1 Definitions

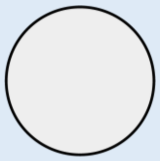
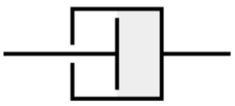
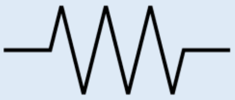
This section gives an overview of all definitions used in this report. Numerous physical quantities are consistently used throughout the report (see Table 2.1). Also, a mechanical scheme is employed in this report (see Table 2.2).

Table 2.1: Applied quantities

| Quantity | Symbol | Unit |
|---|---|---|
| acceleration tensor | $\boldsymbol{a}$ | $mm/s^2$ |
| air void content | $v$ | - |
| angle of gyration | $\phi$ | $deg$ |
| area (contact area) of two colliding elements | $A_c$ | $mm^2$ |
| area (surface area) of the aggregate | $A_{agg}$ | $mm^2$ |
| area (surface area) of the mould | $A_{mld}$ | $mm^2$ |
| damper – Kelvin-Voigt component | $c_K$ | $Ns/mm$ |
| damper – Maxwell component | $c_M$ | $Ns/mm$ |
| damping – critical | $c_{cr}$ | $Ns/mm$ |
| damping ratio Burgers' contact force | $\zeta$ | - |
| deformation – elastic | $u_e$ | $mm$ |
| deformation – plastic | $u_p$ | $mm$ |
| deformation – total | $u$ | $mm$ |
| deformation – viscoelastic | $u_{ve}$ | $mm$ |
| delta factors semi-analytical solution | $\delta_{F_0}, \delta_{F_1},$<br>$\delta_{u_1}, \delta_{u_2}$ | - |
| diameter of the mould | $d_{mld}$ | $mm$ |
| distance between two rays | $\Delta r$ | $mm$ |
| eigen frequency Burgers' contact force | $\omega$ | $(rad/s)^2$ |
| factor – computation time factor | $f_t$ | - |
| factor – element factor | $f_e$ | - |
| force – Kelvin-Voigt component | $F_K$ | $N$ |
| force – Maxwell component | $F_M$ | $N$ |
| friction coefficient | $\mu$ | - |
| gamma factors semi-analytical solution | $\gamma_{F_0}, \gamma_{F_1},$<br>$\gamma_{u_1}, \gamma_{u_2}$ | - |
| height of the fully compacted specimen | $h_{\min}$ | $mm$ |
| height of the specimen | $h$ | $mm$ |
| identity tensor | $\boldsymbol{I}$ | - |
| inertia tensor | $\boldsymbol{J}$ | |
| Lambert W function | $W$ | - |
| length of a ray | $r_i$ | $mm$ |
| moment tensor | $\boldsymbol{M}$ | $N\ mm$ |

| | | |
|---|---|---|
| normal tensor | $\boldsymbol{n}$ | - |
| number of elements | $n_{el}$ | - |
| number of gyrations | $n_g$ | - |
| number of mould lamellae | $n_{mld}$ | - |
| number of rays | $n_r$ | - |
| order – time order | $O$ | - |
| Period of rotation of the mould | $T_{mld}$ | $s$ |
| position tensor | $\boldsymbol{p}$ | $mm$ |
| quaternion | $\boldsymbol{q}$ | - |
| relative damping – largest term | $Z^+$ | $rad/s$ |
| relative damping – omega term | $Z_\omega$ | $rad/s$ |
| relative damping – smallest term | $Z^-$ | $rad/s$ |
| rotation around the global $x, y, z$-axis | $\alpha, \beta, \gamma$ | $deg$ |
| rotation tensor | $\boldsymbol{R}$ | - |
| rotational acceleration tensor | $\boldsymbol{\alpha}$ | $rad/s^2$ |
| rotational velocity tensor | $\boldsymbol{\omega}$ | $rad/s$ |
| spring stiffness – Kelvin-Voigt component | $k_K$ | $N/mm$ |
| spring stiffness – Maxwell component | $k_M$ | $N/mm$ |
| temperature of compaction | $T$ | $^\circ C$ |
| thickness of the mortar layer | $\tau_{mor}$ | $mm$ |
| time – simulation time | $t$ | $s$ |
| time – computation time | $t_c$ | $h$ |
| time step | $\Delta \mathrm{t}$ | $s$ |
| velocity tensor | $\boldsymbol{v}$ | $mm/s$ |
| volume of the aggregates | $V_{agg}$ | $mm^3$ |
| volume of the mortar | $V_{mor}$ | $mm^3$ |

Table 2.2: Applied mechanical symbols

| Name | Symbol | Mechanical symbol |
|---|---|---|
| **mass** | $m$ |  |
| **damper** | $c$ |  |
| **spring** | $k$ |  |

## 2.2 Tensor notation

Throughout the report, first and second order tensors are used. These are represented in bold. First order tensor are presented in lower case, whereas the second order tensors are presented with capital letters. With the exception of the first order force and moment tensor. In the text it is always clarified whether a tensor is of the first or second order. Furthermore, it is assumed common knowledge that a rotation matrix is orthogonal, and therefore its transpose equals its inverse. Equation 2.1 shows the second order rotation tensor from global to local coordinates.

$$\boldsymbol{R}^T = \boldsymbol{R}^{-1} \tag{2.1}$$

## 2.3 Superpave gyratory compaction

This paragraph specifies the superpave gyratory compaction process, which is a standardized tool for determining the void curve of an asphalt mixture, according to NEN-EN 12697-31, 35 [2, 20] and NEN-EN 13108-1 [21]. This information is essential, as this forms the basis for all simulations. Also, some differences between lab and field compaction are mentioned.

Superpave gyratory compaction (SGC) is achieved by combining a rotary shearing action and a vertical resultant force applied by a mechanical head. This method can be used for the determination of the density, or air void content of a mixture versus the number of gyrations. Figure 2.1 shows the 3D motion diagram of the specimen. The bituminous mixture is contained within a cylindrical mould and kept at a constant temperature throughout the whole duration of the test. Compaction is achieved by the simultaneous action of a low static compression $F$, and of the shearing action resulting from the motion of the axis of the mould. The ends of the specimen remain perpendicular to the axis of the conical surface. The mould is lubricated with a coating of glycerised soda oleate or silicon grease to reduce friction between the mould and the specimen [2].

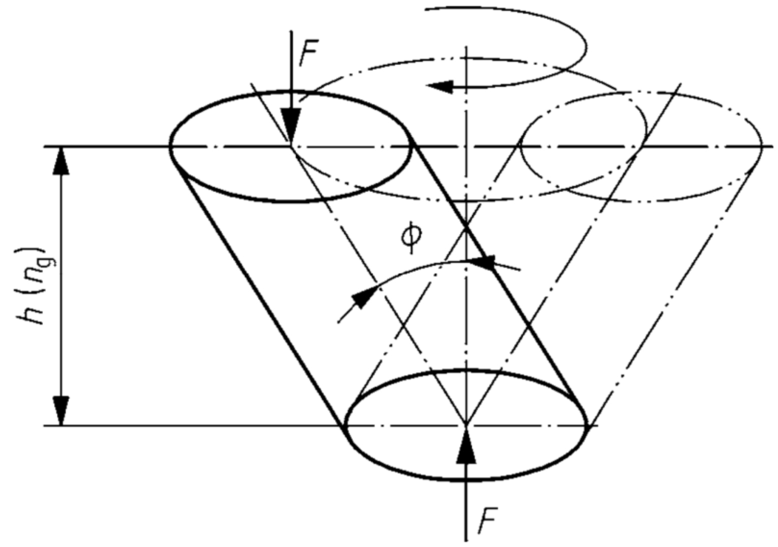


Figure 2.1: Specimen 3D motion diagram [2]

A preload is applied between 150 and 3000 $N$. During testing, a pressure of 600 $kPa$ is applied. The angle of gyration, $\phi$, is usually 0.82°, but may vary. The rotational frequency equals $1/2\ Hz$. The mould diameter could be taken as 100, 150, or 160 $mm$. The first of which may only contain aggregates with a diameter less than 16 $mm$, whereas the latter two may contain aggregates with a diameter up to 31.5 $mm$. The height of the mould is not specified, but must be compatible with the volume of the material under test [2].

Equation 2.2 shows how to determine the mean air void fraction $v$, as a function of the number of gyrations $n_g$, and height of the fully compressed specimen $h_{min}$. The number of gyrations usually varies between 0 and 100, but may go up to 500, for determining the density curve. The height of the fully compressed specimen represents the situation where the specimen does not contain any air void content [2].

$$v(n_g) = 1 - \frac{h_{min}}{h(n_g)} \tag{2.2}$$

$$with$$
$$h_{min} = \frac{V_{agg} + V_{mor}}{A_{mld}} \quad ; \quad 0.66\, d_{mld} \leq h_{min} \leq 1.05\, d_{mld}$$

The mean of at least three test results are used. For the test to be successful, the coefficient of variation of the heights of the specimens must be lower than 1.5% after 20 gyrations [2].

The result of the SGC contains the following:

- Identification of the mixture, and method of manufacture of the mixture.
- Type and model of gyratory compactor.
- Internal diameter, minimal height of specimen, speed of rotation, temperature of the test, the angle of inclination, force, and method of calibration.
- Void content for the prescribed numbers of gyrations.
- The operational details, and anomalies, if any, which might have affected the results.

This process is used to simulate the field compaction process. However, there are a few differences between the two, which impacts performance and test results. These factors are to be considered in case field compaction is digitally simulated.

1. Lab compaction happens in a very controlled environment, where the degradation, temperature and moisture content are carefully chosen. Whereas in the field, these parameters might differ.
2. The mixing process takes longer in the field compaction, leading to aged binder, which significantly increases its stiffness.
3. The compaction process in the field is done by rollers, and takes much longer. This has a significant impact on the compaction temperature, which varies much more than in laboratory compaction [4].

### *2.4 Literature review*

Much research has been done on the simulation of hot asphalt compaction [5–12]. Two studies that are particularly relevant are the ones of Al Khateeb et al. [8] and Komaragiri et al. [16]. This paragraph explorers and discusses the results of these papers.

Al Khateeb et al. [8] developed a computational model of asphalt using the commercial EDEM software. Coated aggregates were simulated using clusters of spheres of different sizes. Burgers' model was used for the contact between the particles. The parameters were determined based on dynamic shear rheometer tests. Subsequently, the results were compared with an SGC test to validate the model. Also, an X-ray CT scan was conducted to assess the air void fraction at different locations in the sample. Aggregates were modelled slightly bigger than their actual size to represent the mortar, a continuous glue layer around the aggregates, as well. Different contact conditions were applied when there was a mortar–mortar , mortar–aggregate, or aggregate–aggregate contact. Al Khateeb et al. [8] determined the macroscale parameters of Burgers' model for different compaction temperatures. Also, the influence of temperature drop during pavement construction was analysed. The result of the simulation was in agreement with a laboratory test. It was clearly shown that the DEM model gives accurate results and can be used additionally to laboratory testing. Also, this paper showed that the DEM has great benefits in terms of deriving properties of the specimen that in some cases cannot be seen in a laboratory test, such as the tracking of elements.

Al Khateeb et al. [8] also showed some limitations of their model. Elements smaller than $2\,mm$ were neglected due to limitations in computational power. Instead, these were included in the mortar layer. Furthermore, they did not adhere to the minimum amount of specimen required for SGC testing, as described in NEN-EN 12697-31 [2], which might have influenced the results [16]. Also, the system specifications and computation time were not mentioned, making it unclear whether this method is applicable on a regular pc. What is more, the X-ray CT scan showed air void content over the height of the specimen, which was compared to the air void content in the DEM model. It showed that the air void content on the top was overestimated. This was likely to be caused by the fact that the smaller particles were mostly to be found in bottom half of the specimen, resulting in larger gaps in the top. A recommendation for further research is to incorporate the complex morphological characteristics, shapes and angularities of aggregates more realistically. In this, a physics engine, such as Bullet Physics can be useful.

Also Komaragiri et al. [16] created a computation model for analysing gyratory compaction using a physics engine, and validate the correctness of this model by means of parametric study. They used the open source Bullet Physics engine [17], which allows for the implementation of realistic particle shapes using triangular face tessellations. Therefore, a number of aggregates were 3D laser-scanned, and included in their model. In order to accurately model the contact forces between the elements, Komaragiri et al. [16] made two additions to the software: cohesive forces normal to the contact, and viscous damping tangential to the contact. These were implemented using a Python script, and were applied as external forces on all interacting objects at the beginning of each time step. In order to validate the results of their model, three parametric studies have been performed, where the angle of gyration, compaction pressure and height of the specimen were varied, showing mixed agreements with laboratory experiments. Nonetheless, they concluded that that the developed model can reasonably be used for simulation hot asphalt compaction. Another useful finding is that the simulation resulted in a very resonable computation time, allowing for the digital analysis on a regular deskop pc.

Komaragiri et al. [16] identified some limitations to their model. In order to achieve a reasonable computation time, any element smaller than 4.75 $mm$ was not explicitly modelled. Instead, this was considered as part of the mortar layer. Furthermore, all elements were assumed to be rigid bodies, not allowing for the breakage of particles in the digital model. What is more, to simulate the gyration of the specimen, instead of rotating the mould, Komaragiri et al. [16] gyrated the top end plate, while keeping the mould steady. This did not sufficiently match the laboratory procedure, as the aggregates did not experience a revolving motion inside the mould. Also, PyBullet does not assess the contact area of two colliding elements. Therefore Komaragiri et al. [16] estimated the contact area as half the surface area of the smallest particle for determining viscous and cohesive forces. The latter assumption is further examined in Paragraph 6.5. Further recommendations are the creation of a digial library of scanned aggregates, with different shapes and sizes, and the addition of friction.

## *2.5 Bullet Physics and its utilities*

This paragraph elaborates on Bullet Physics, its advantages and disadvantages, and possible utility for modelling hot asphalt compaction.

As opposed to most commercial software, Bullet Physics is an open source, free-to-use, and cross-platform tool [18]. Bullet utilizes the CPU, as well as the GPU to shorten computation time [17]. Also, the Python wrapper PyBullet makes it easily accessible for most researchers. Running a simulation requires 3D objects, which can be created using any 3D drawing tool, such as SketchUp [22] or Blender [23]. A library with simple objects is also available for the user. This includes a simple sphere, cube, robot, and ground surface, among others. An elaborate and easy to understand quickstart guide can be found on the official PyBullet website [18].

Bullet Physics is widely used in simulations, video games, and movies [18]. In 2015, the main authors won the Academy Awards for their significant contributions to the development of motion pictures [24]. Since then, a few researchers have investigated the use of this software for the simulation of granular materials [13, 14, 15]. The physics engine is also very popular in the field of robotics [25, 26, 27]. As opposed to video games and animations, scientific research in the field of pavement engineering requires high levels of accuracy. The main focus of Bullet is currently on applications that do not require such accuracy.

Another interesting feature of Bullet Physics is that it doesn’t have the classical ‘truncation error’, where the precision of the simulation increases with a smaller step size. Instead, the projected Gauss Seidel solver [28, 29] is implemented, where multiple iterations are used per time step to ensure the precision of the simulation, regardless of the step size. A standard step size of 1/240 seconds in combination with 50 iterations were carefully set by the developers to ensure reasonable stability and accuracy of the simulation, while significantly reducing computation time [18].

A shortcoming of Bullet Physics for the field of engineering is its focus on simplicity. To give a few examples: Bullet determines dynamical properties, such as the friction coefficient of two colliding elements, by multiplying the user defined friction coefficients of the two individual objects. Whereas in reality, this dynamical property is determined based on both materials and cannot be determined based on solely the individual friction coefficients. Also, complex contact models to describe the behaviour of viscoelastic material are not (yet) supported. Since Bullet is open source, improvements in this regard are possible [17].

## 2.6 An introduction to quaternions

### Euler rotation and gimbal locking

Since a custom integration method is being implemented, a method of describing an object's orientation is required. The most common description of a rotation in 3D space is Euler rotation. Extrinsic Euler rotation could be described as the consequent rotation about the $x$, $y$, and $z$-axes. This way, the orientation of an object in 3D space can be fully described [30]. Figure 2.2 gives a visual representation [31].

There are however a few downsides to this method. First of all, due to the order of rotation, an additional rotation cannot easily be added. Secondly, it is not always possible to continuously describe an object's orientation under a certain angular velocity. The latter phenomenon is called 'gimbal locking' [30]. Gimbal locking is the loss of one angular degree of freedom, and occurs when the second successive rotation nears $\pm 90°$, or $\pi/2$ $rad$. The two other axes are then aligned, and the third rotation yields the same result as the first one. Thus, effectively losing a degree of freedom [30].

There are several ways to avoid gimbal locking. Resetting the gimbals under certain conditions, as well as adding a fourth gimbal are viable solutions [30]. In this report, the use of gimbals to describe an object's orientation is disregarded completely. Instead the use of quaternions was chosen, which is depicted in Figure 2.3 [32].

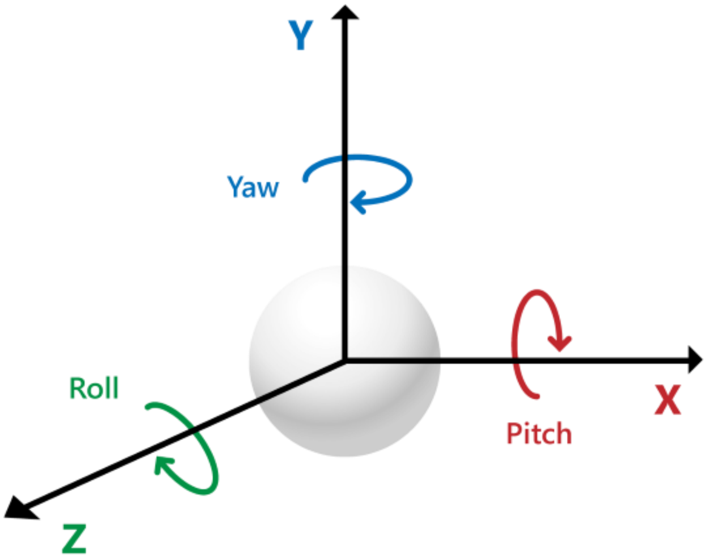


Figure 2.2: Euler rotation [31]

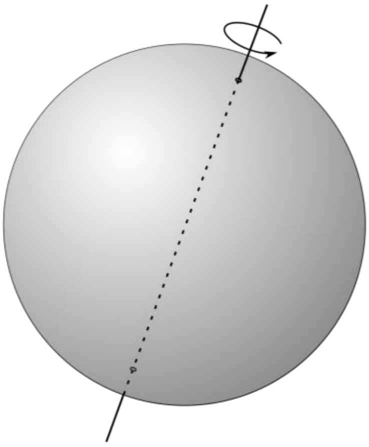


Figure 2.3: Quaternion rotation [32]

### Definition of a quaternion

The use of quaternions is a more modern approach in describing the orientation of an object in 3D space. A quaternion describes a single rotation around a vector. Not only can the orientation of an object be fully described, but this method can also describe a continuous rotational velocity under any orientation [30].

A quaternion is a first order tensor, consisting of four scalars. Equation 2.3 shows a quaternion, where $\omega$ is a scalar value that represents an angle of rotation, and $x$, $y$, $z$ correspond to an axis of rotation about which the angle of rotation is performed. In literature, the position of $\omega$ can be in the first or last entry [30].

$$\boldsymbol{q} = \begin{pmatrix} x \\ y \\ z \\ \omega \end{pmatrix} \tag{2.3}$$

In this report, the rotational scalar is always presented in the last entry of the quaternion. Also, quaternions are always assumed to be unit quaternions, meaning that the magnitude of this vector equals 1. Not only is the quaternion more convenient to work with than Euler rotation, but this method is also computationally more efficient [30].

### Euler to quaternion conversion

Since angular velocities and accelerations are not dependent on the orientation of an object and thus do not suffer from gimbal locking, these can still be described using Euler rotation. These can easily be converted into quaternions. Let $\alpha$, $\beta$, and $\gamma$ be the consecutive rotation about the global $x$, $y$, and $z$-axes, respectively. Equation 2.4 shows the conversion between the Euler angles and quaternion representation [30].

$$\boldsymbol{q}(\alpha,\beta,\gamma) = \begin{pmatrix} \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) - \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) + \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) + \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) \end{pmatrix} \tag{2.4}$$

### Rotation matrix

In order to determine the locations of an object's vertices, or the inertia tensor in global coordinates, a rotation matrix is required. Equation 2.5 shows how to determine the rotation matrix for an element from global to local coordinate system, making use of a unit quaternion [30]. Since the rotation matrix is orthogonal, its transpose can be used for the opposite operation. It is noted that this is the same as substituting $-\omega$ for $\omega$.

$$\boldsymbol{R} = 2\begin{pmatrix} x^2+\omega^2 & xy-z\omega & xz+y\omega \\ xy+z\omega & y^2+\omega^2 & yz-x\omega \\ xz-y\omega & yz+x\omega & z^2+\omega^2 \end{pmatrix} - \boldsymbol{I} \tag{2.5}$$

*with **I** the identity tensor*

**Quaternion multiplication**
If an additional rotation is desired to an existing orientation, the new orientation can be found using 'quaternion multiplication'. Let $\boldsymbol{q}$ be the orientation of an object, and $\boldsymbol{\Delta q}$ the additional rotation. Equation 2.6 shows how the new orientation can be found [30].

$$\boldsymbol{q}\cdot\boldsymbol{\Delta q} = \begin{pmatrix} x \\ y \\ z \\ \omega \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\omega \end{pmatrix} = \begin{pmatrix} \omega\cdot\Delta x + z\cdot\Delta y - y\cdot\Delta z + x\cdot\Delta\omega \\ -z\cdot\Delta x + \omega\cdot\Delta y + x\cdot\Delta z + y\cdot\Delta\omega \\ y\cdot\Delta x - x\cdot\Delta y + \omega\cdot\Delta z + z\cdot\Delta\omega \\ -x\cdot\Delta x - y\cdot\Delta y - z\cdot\Delta z + \omega\cdot\Delta\omega \end{pmatrix} \tag{2.6}$$

As can be seen from the result above, adding two quaternion rotations is remarkably simple, and avoids singularities such as gimbal locking. It is noted that quaternion multiplication is noncommutative, thus the order of multiplication matters [30].

## *2.7 Relevance of inertia in overdamped systems*

In most numerical integration techniques, the computation time and stability heavily depend on the inertia terms. In case of asphalt compaction, the mass of the aggregates is extremely low in comparison to the viscosity of the binder. Thus, this might lead to very long computation times and instability. A possible work around method is to increase the mass, such that it does not influence the results too much, which leads to a reasonable computation time. However, this solution is not ideal. Bullet Physics [17] is an impulse based engine, where this effect is reasonably mitigated, but could still lead to instability in case of large external forces. A possible way to overcome this, is by neglecting the inertia term in the differential equation.

Steven H. Strogatz [33] used a dimensionless analysis to suggest that in case of an overdamped system, a second order differential equation can very well be approximated by a much simpler first order one. This means that in case of an extremely viscous environment, where the system is supercritically damped, the inertia term could in some cases be neglected. What is of importance here, is the time scale. Rapid oscillations over short periods of time will be neglected. Furthermore, reducing the order of the system has the inevitable consequence that not all initial conditions can be satisfied. The theory only holds in case initial velocities are close to zero.

To demonstrate the potential application in a classical numerical integrator or physics engine, Equation 2.7 gives the governing equation of a classical system. Equation 2.8 gives an approximation of the differential equation for a supercritically damped system. Both of these solve for the highest order derivative, allowing for numerical integration.

$$m\ddot{u} + c\dot{u} + ku = F_{ext} \leftrightarrow \ddot{u} = \frac{F_{ext} - c\dot{u} - ku}{m} \tag{2.7}$$

$$c\dot{u} + ku \approx F_{ext} \leftrightarrow \dot{u} \approx \frac{F_{ext} - ku}{c} \tag{2.8}$$

$where$
$v_0 \approx 0$
$c \gg m$

It is noted that neglecting the inertia does not necessarily mean neglecting gravity. This can still be considered in the equation of motion.

To further investigate the validity of this theory, two tests have been performed using a damped mass-spring system with an initial displacement, and external force. A first and second order systems are compared with each other. Figure 2.4 shows the displacement of three dynamical systems with an initial displacement. Figure 2.5 shows the displacement of another three systems with a constant external force. Both Figures vary the damping-mass ratio of the second order systems, and compare these with a first order equation, where the inertia term is neglected. Table 2.3 presents a comparison of the results and shows the relative error at $t = 5\ s$.

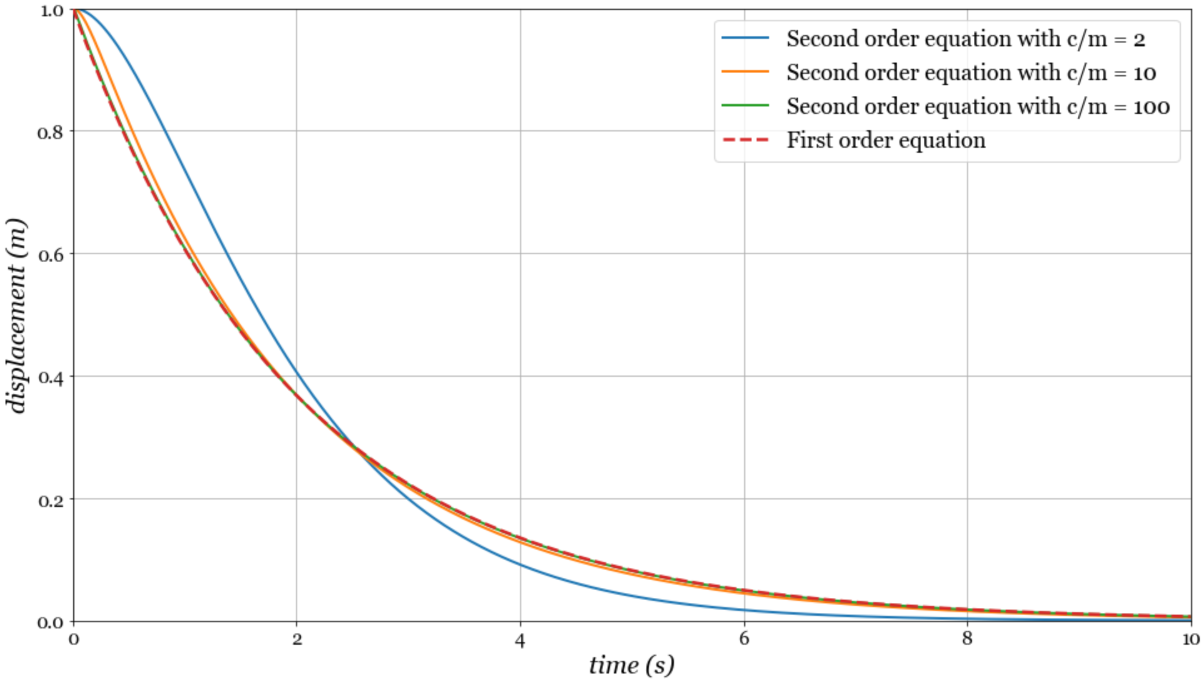


Figure 2.4: Comparison of a first and second order systems with an initial displacement

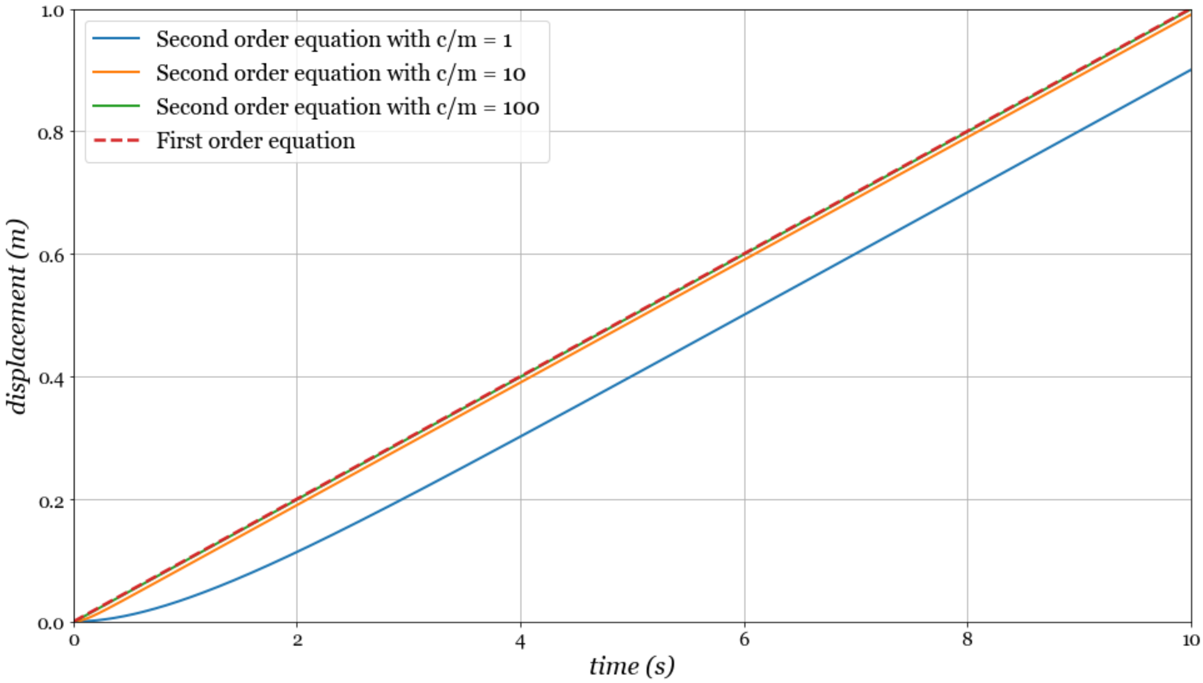


Figure 2.5: Comparison of a first and second order systems with an external force

Table 2.3: relative error of the different tests at $t = 5\ s$

| $c/m$-value | Initial displacement test (%) | External force test (%) |
|---|---|---|
| 2, 1 | 50.4 % | 20.0 % |
| 10 | 7.2 % | 2.0 % |
| 100 | 0.0 % | 0.2 % |

Figures 2.4, 2.5, and Table 2.3 show that a first order system can very well approximate a second order system for supercritically damped systems. The higher the damping/mass ratio, the better the second order system approaches the first order system. Based on the above observations, it could be concluded that the inertia term can indeed be neglected in some cases. In chapter 6 this is further investigated.

# 3 Implementation of Burgers' contact model

Burgers' contact model could be considered as the core of this research, as this is the main addition to the physics engine. In the proposed model, Burgers' contact model enables researchers to compute contact forces between colliding elements. A detailed explanation of the contact model is described in this chapter. The following paragraphs describe the proposed contact model, as well as a derivation of the equation of motion, and shows numerical solutions as well as an analytical solution.

## 3.1 Equation of motion

As shown in Figure 3.1, Burgers' contact model consists of a Maxwell and a Kelvin-Voigt part. The Maxwell part consists of a spring $k_M$, and damper $c_M$, in series, which represents the elastic $u_e$, and viscous deformation $u_v$, respectively. The Kelvin-Voigt part consists of a parallel spring $k_K$, and damper $c_K$, which represents the viscoelastic deformation $u_{ve}$. The total deformation is denoted as $u$.

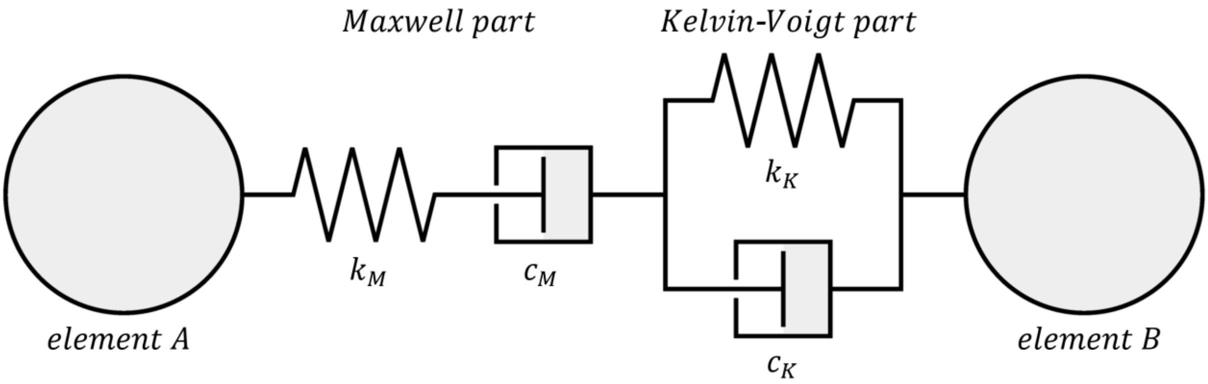


Figure 3.1: Mechanical scheme of Burgers' contact model (Kelvin representation)

Table 3.1 presents the values for the mortar as obtained elsewhere [8]. Following the equation for the axial stiffness of a rod, all parameters should be multiplied with the contact area, and divided by the thickness of the mortar layer, before these can be implemented in a discrete element method or physics engine [8].

Table 3.1: Burgers' parameters at different temperatures [8]

| T [°C] | 100 | 125 | 150 |
|---|---|---|---|
| $k_M$ $[MPa]$ | 40.27 | 23.45 | 15.12 |
| $c_M$ $[MPa\ s]$ | 613.50 | 558.73 | 530.09 |
| $k_K$ $[MPa]$ | 15.02 | 11.63 | 10.52 |
| $c_K$ $[MPa\ s]$ | 3.33 | 2.43 | 2.05 |

In order to implement this custom contact model in a physics engine, its equation of motion is derived. It is noted that in the derivation, inertia of the elements is neglected. Equations 3.1 and 3.2 give the constitutive relations between the forces in the Maxwell and Kelvin-Voigt part, $F_M$ and $F_K$ respectively, and the displacements, $u_e$, $u_v$, and $u_{ve}$ for this model.

$$F_M = k_M\, u_e = c_M\, \dot{u}_v \tag{3.1}$$

$$F_K = k_K\, u_{ve} + c_K\, \dot{u}_{ve} \tag{3.2}$$

Equations 3.3 and 3.4 describe the equilibrium and kinematic relations, respectively. Since the springs and dampers are massless, the contact force in each element is equal. Also, the sum of the deformations of the individual parts forms the total deformation.

$$F = F_M = F_K \tag{3.3}$$

$$u = u_e + u_v + u_{ve} \tag{3.4}$$

This leads to the derivation of the equation of motion. Equation 3.5 presents the substitution of the viscoelastic deformation $u_{ve}$ in Equation 3.2, using Equation 3.4.

$$F = k_K(u - u_e - u_v) + c_K(\dot{u} - \dot{u}_e - \dot{u}_v) \tag{3.5}$$

Equation 3.6 presents the derivation with respect to time, which allows for the substitution of the elastic and viscous deformation.

$$\dot{F} = k_K(\dot{u} - \dot{u}_e - \dot{u}_v) + c_K(\ddot{u} - \ddot{u}_e - \ddot{u}_v) \tag{3.6}$$

Equation 3.7 shows how the elastic and viscous deformation, $u_e$ and $u_v$ respectively, are substituted, using Equation 3.1.

$$\dot{F} = k_K\left(\dot{u} - \frac{\dot{F}}{k_M} - \frac{F}{c_M}\right) + c_K\left(\ddot{u} - \frac{\ddot{F}}{k_M} - \frac{\dot{F}}{c_M}\right) \tag{3.7}$$

The contact force and displacement are separated in Equation 3.8, resulting in the differential equation.

$$\frac{c_K c_M}{k_K k_M}\ddot{F} + \frac{c_K k_M + c_M k_K + c_M k_M}{k_K k_M}\dot{F} + F = \frac{c_K c_M}{k_K}\ddot{u} + c_M\dot{u} \tag{3.8}$$

The differential equation is rewritten in Equation 3.9, resulting in the general form.

$$\ddot{F} + 2\zeta\omega\dot{F} + \omega^2 F = \kappa\ddot{u} + s\dot{u} \tag{3.9}$$

$$with$$

$$\omega^2 = \frac{k_K k_M}{c_K c_M} \qquad \kappa = k_M$$

$$2\zeta\omega = \frac{k_M}{c_M} + \frac{k_K + k_M}{c_K} \qquad s = \frac{k_K k_M}{c_K}$$

$$\zeta = \frac{c_K k_M + c_M(k_K + k_M)}{2\sqrt{c_K c_M k_K k_M}}$$

## 3.2 Numerical solution

This paragraph describes the theoretical implementation of Burgers' contact model in a physics engine. Since the numerical model consists of many interacting discrete elements, a numerical solution is required. Various numerical integration methods exist for solving initial value problems.

Since the forces, as well as the displacements, are present in Equation 3.9, it is a partial differential equation, which can be solved numerically. The following approach has been taken to obtain the solution:

1. In every discretised time step, the Burgers' contact force and displacements, as well as its first derivatives are known. The accelerations are then obtained using Newton's second law of motion.
2. The second derivatives of the Burgers' contact force in that same time step are determined using Equation 3.9.
3. A numerical integration method is then applied for determining the contact forces and the displacements in the next time step.

This process is repeated until predefined iterations.

Equation 3.10 gives the acceleration of an element under a force at a discrete time step $n$. It is noted that in 3D with multiple elements, the sum of the forces should be considered. Equation 3.11 presents the second derivative of Burgers' contact force at the $n^{\text{th}}$ time step.

$$\ddot{u}_n = \frac{F_n}{m} \tag{3.10}$$

$$\ddot{F}_n = -2\zeta\omega\dot{F}_n - \omega^2 F_n + \kappa\ddot{u}_n + s\dot{u}_n \tag{3.11}$$

**Euler Forward**

A possible way to numerically integrate is the Euler Forward method [34]. This will also form the basis for the Newmark-beta integration method [35] as explained on the next page. Equation 3.12 shows a way to numerically integrate the displacements for the next time step, using Euler Forward. Equation 3.13 numerically integrates Burgers' contact forces.

$$\begin{aligned} \dot{u}_{n+1} &= \dot{u}_n + \Delta t\, \ddot{u}_n \\ u_{n+1} &= u_n + \Delta t\, \dot{u}_n + \frac{1}{2}\Delta t^2\, \ddot{u}_n \end{aligned} \tag{3.12}$$

$$\begin{aligned} \dot{F}_{n+1} &= \dot{F}_n + \Delta t\, \ddot{F}_n \\ F_{n+1} &= F_n + \Delta t\, \dot{F}_n + \frac{1}{2}\Delta t^2\, \ddot{F}_n \end{aligned} \tag{3.13}$$

From the resulting quantities in Equations 3.12 and 3.13, the acceleration is obtained at the next time step, and the process repeats itself.

### Newmark-beta

Newmark-beta is a widely used second order implicit numerical integration technique [35], which might be an improvement over the previously discussed method. The key element of this method is that it uses the accelerations of the new time step as well. It is highlighted that since the accelerations in the next time step are unknown, these have to be initially estimated using Euler Forward as described above, which can then be reiterated.

Equation 3.14 presents the Newmark-beta integration method for determining the positions and velocities in the next time step. Equation 3.15 shows how to numerically integrate the Burgers' contact forces.

$$\begin{aligned} \dot{u}_{n+1} &= \dot{u}_n + \Delta t\, \ddot{u}_\gamma \\ u_{n+1} &= u_n + \Delta t\, \dot{u}_n + \frac{1}{2}\Delta t^2\, \ddot{u}_\beta \end{aligned} \tag{3.14}$$

$$\begin{aligned} \dot{F}_{n+1} &= \dot{F}_n + \Delta t\, \ddot{F}_\gamma \\ F_{n+1} &= F_n + \Delta t\, \dot{F}_n + \frac{1}{2}\Delta t^2\, \ddot{F}_\beta \end{aligned} \tag{3.15}$$

$$\begin{aligned} &with \\ &\ddot{u}_\gamma = (1-\gamma)\, \ddot{u}_n + \gamma\, \ddot{u}_{n+1} \\ &\ddot{F}_\gamma = (1-\gamma)\, \ddot{F}_n + \gamma\, \ddot{F}_{n+1} \\ &\ddot{u}_\beta = (1-2\beta)\, \ddot{u}_n + 2\beta\, \ddot{u}_{n+1} \\ &\ddot{F}_\beta = (1-2\beta)\, \ddot{F}_n + 2\beta\, \ddot{F}_{n+1} \end{aligned}$$

The boundaries for $\gamma$ and $\beta$ are presented in Equation 3.16. This equation also presents the values in which the middle point rule applies, where the integration method is indeed of the second order [35].

$$\begin{aligned} 0 \leq\ \gamma\ \leq 1 \quad &\rightarrow \quad \gamma = 1/2 \\ 0 \leq 2\beta \leq 1 \quad &\rightarrow \quad \beta = 1/4 \end{aligned} \tag{3.16}$$

Since the accelerations in the next time step are both in the input and output, a more precise answer, or better approximation, is generally found after multiple iterations.

### Runge Kutta

A third alternative is the use of the Runge Kutta integration method [36], which is implemented in the standard SciPy [37] package, and can thus be used. This fourth order method is not explained here in detail, as it exceeds the scope of this research. A detailed description can be found elsewhere [36].

### 3.3 Analytical solution

In order to verify the accuracy of the numerical model, an analytical solution can be used. Burgers' differential equation (Equation 3.9) can be solved analytically for either a described force, or velocity. In the present study, SymPy [38] was used to obtain the solution.

Using Equation 3.9, the initial value problem is obtained under a constant external force, as presented in Equation 3.17. The initial displacement follows from a static analysis, where only the undamped spring is activated. The initial velocity is calculated likewise, where the effective damping of the two dampers in series is determined. The analytical solution is presented in Equation 3.18.

$$\kappa\ddot{u} + s\dot{u} = \omega^2 F \tag{3.17}$$

$$\begin{aligned}
&with\\
&u(0) = \frac{F}{k_M}\\
&\dot{u}(0) = F\left(\frac{1}{c_M} + \frac{1}{c_K}\right)
\end{aligned}$$

$$u(t) = F\left(\frac{1}{k_M} + \frac{t}{c_M} + \frac{1}{k_K}\left(1 - e^{-\frac{k_K}{c_K}t}\right)\right) \tag{3.18}$$

Using Equation 3.9, the initial value problem could also be obtained with a constant velocity, which is presented in Equation 3.19. The initial conditions can be chosen arbitrarily. Equation 3.20 presents the analytical solution for the supercritically damped system.

$$\ddot{F} + 2\zeta\omega\dot{F} + \omega^2 F = s\dot{u} \tag{3.19}$$

$$\begin{aligned}
&with\\
&F(0) = F_0\\
&\dot{F}(0) = \dot{F}_0\\
&\zeta > 1
\end{aligned}$$

$$F(t) = c_M\dot{u} - \frac{\dot{F}_0 + \mathcal{Z}^-(F_0 - c_M\dot{u})}{\mathcal{Z}_\omega}e^{-\mathcal{Z}^+t} + \frac{\dot{F}_0 + \mathcal{Z}^+(F_0 - c_M\dot{u})}{\mathcal{Z}_\omega}e^{-\mathcal{Z}^-t} \tag{3.20}$$

$$\begin{aligned}
&with\\
&\mathcal{Z}^+ = \omega\left(\zeta + \sqrt{\zeta^2 - 1}\right)\\
&\mathcal{Z}^- = \omega\left(\zeta - \sqrt{\zeta^2 - 1}\right)\\
&\mathcal{Z}_\omega = 2\omega\sqrt{\zeta^2 - 1}
\end{aligned}$$

To verify the numerical models with the analytical solution, a simple experiment was conducted. Two coated aggregates are compressed with a constant external force $F_0$. After the aggregates come into contact, they are pulled apart with a constant velocity $-\dot{u}_3$. This case is derived as shown in Equations 3.21. At first, due to the non-penetration condition of the aggregates, the time at which the aggregates come into contact $t_1$, is determined. Equation 3.22 presents the solution, where $W$ represents the Lambert $W$ function.

$$u(t_1) = F_0\left(\frac{1}{k_M} + \frac{t_1}{c_M} + \frac{1}{k_K}\left(1 - e^{-\frac{k_K}{c_K}t_1}\right)\right) = 2\tau_{mor} \tag{3.21}$$

$$t_1 = \frac{c_K}{k_K}W - \frac{c_M}{k_M} - \frac{c_M}{k_K} + \frac{2c_M\tau_{mor}}{F_0} \tag{3.22}$$

$$\begin{aligned}&with\\&W = W_0\left(\frac{c_M}{c_K}\exp\left(\frac{c_M k_K}{c_K k_M} + \frac{c_M}{c_K} - \frac{2c_M k_K \tau_{mor}}{c_K F_0}\right)\right)\end{aligned}$$

Following Equation 3.18, Equation 3.23 presents the analytical solution for the displacements for the experiment as described above. In the first step ($t < t_1$), the elements are subjected to the contact force. In the second and third step, the solution is obtained in a velocity controlled mode. It is noted that a few conditions must be met: the initial elastic deformation may not exceed the mortar thickness, and the simulation is terminated before the elements completely separate.

$$u(t) = \begin{cases} F_0\left(\frac{1}{k_M} + \frac{t}{c_M} + \frac{1}{k_K}\left(1 - e^{-\frac{k_K}{c_K}t}\right)\right) & for \quad t \le t_1 \\ 2\tau_{mor} & for \quad t_1 < t \le t_2 \\ 2\tau_{mor} + \dot{u}_3(t - t_2) & for \quad t_2 < t \le t_3 \end{cases} \tag{3.23}$$

$$\begin{aligned}&with\\&t_1 < t_2 < t_3 < t_2 - 2\tau_{mor}/\dot{u}_3\\&F_0/k_M < 2\tau_{mor}\\&\dot{u}_3 < 0\end{aligned}$$

Following Equation 3.20, Equation 3.24 presents the analytical solution for the Burgers' contact forces in the same situation. The boundary conditions at the start of last step follow the results from the previous step.

$$F(t) = \begin{cases} F_0 & for \quad t \leq t_1 \\ \dfrac{F_0}{\mathcal{Z}_\omega}\left(\mathcal{Z}^- e^{-\mathcal{Z}^+(t-t_1)} + \mathcal{Z}^+ e^{-\mathcal{Z}^-(t-t_1)}\right) & for \quad t_1 < t \leq t_2 \\ c_M\dot{u}_3 - \dfrac{\dot{F}_2 + \mathcal{Z}^-(F_2 - c_M\dot{u}_3)}{\mathcal{Z}_\omega} e^{-\mathcal{Z}^+(t-t_2)} & \\ \quad + \dfrac{\dot{F}_2 + \mathcal{Z}^+(F_2 - c_M\dot{u}_3)}{\mathcal{Z}_\omega} e^{-\mathcal{Z}^-(t-t_2)} & for \quad t_2 < t \leq t_3 \end{cases} \tag{3.24}$$

$$with$$

$$F_2 = F_0 \frac{\mathcal{Z}^+ e^{-\mathcal{Z}^-(t_2-t_1)} - \mathcal{Z}^- e^{-\mathcal{Z}^+(t_2-t_1)}}{\mathcal{Z}_\omega}$$

$$\dot{F}_2 = F_0\omega \frac{e^{-\mathcal{Z}^+(t_2-t_1)} - e^{-\mathcal{Z}^-(t_2-t_1)}}{\mathcal{Z}_\omega}$$

$$\mathcal{Z}^+ = \omega\left(\zeta + \sqrt{\zeta^2 - 1}\right)$$

$$\mathcal{Z}^- = \omega\left(\zeta - \sqrt{\zeta^2 - 1}\right)$$

$$\mathcal{Z}_\omega = 2\omega\sqrt{\zeta^2 - 1}$$

## 3.4 Semi-analytical solution

Alternatively to the numerical solution, the contact forces can also be solved analytically over a small time step. The advantage being that this method might be more precise. Therefore, the analytical method is explored in the research work.

Following Equation 3.9, Equations 3.25 and 3.26 give the semi-analytical solution for the Burgers' contact force over a small time step $\Delta t$, assuming a near constant velocity and acceleration. It is noted that under the assumption of a constant velocity, the particular solution is the same as the right-hand side of Equation 3.8. The velocity and acceleration are determined using the finite difference method, as these are not determined by PyBullet [18]. The solutions were obtained using SymPy [38].

$$F_n = \gamma_{F_0} F_{n-1} + \gamma_{F_1} \dot{F}_{n-1} + \gamma_{u_1} \dot{u}_{n-1} + \gamma_{u_2} \ddot{u}_{n-1} \tag{3.25}$$

$$\dot{F}_n = \delta_{F_0} F_{n-1} + \delta_{F_1} \dot{F}_{n-1} + \delta_{u_1} \dot{u}_{n-1} + \delta_{u_2} \ddot{u}_{n-1} \tag{3.26}$$

*with*

$$\gamma_{F_0} = \frac{\mathcal{Z}^+ e^{-\mathcal{Z}^- t} - \mathcal{Z}^- e^{-\mathcal{Z}^+ t}}{\mathcal{Z}_\omega} \qquad \delta_{F_0} = \omega^2 \frac{e^{-\mathcal{Z}^+ t} - e^{-\mathcal{Z}^- t}}{\mathcal{Z}_\omega}$$

$$\gamma_{F_1} = \frac{e^{-\mathcal{Z}^- t} - e^{-\mathcal{Z}^+ t}}{\mathcal{Z}_\omega} \qquad \delta_{F_1} = \frac{\mathcal{Z}^+ e^{-\mathcal{Z}^+ t} - \mathcal{Z}^- e^{-\mathcal{Z}^- t}}{\mathcal{Z}_\omega}$$

$$\gamma_{u_1} = c_M \ (1 - \gamma_{F_0}) \qquad \delta_{u_1} = -c_M \ \delta_{F_0}$$

$$\gamma_{u_2} = \frac{c_K c_M}{k_K} (1 - \gamma_{F_0}) \qquad \delta_{u_2} = -\frac{c_K c_M}{k_K} \delta_{F_0}$$

$$\dot{u}_{n-1} = \frac{u_n - u_{n-1}}{\Delta t} \qquad \ddot{u}_{n-1} = \frac{u_n - 2u_{n-1} + u_{n-2}}{\Delta t^2}$$

$$\mathcal{Z}^+ = \omega\left(\zeta + \sqrt{\zeta^2 - 1}\right)$$
$$\mathcal{Z}^- = \omega\left(\zeta - \sqrt{\zeta^2 - 1}\right)$$
$$\mathcal{Z}_\omega = 2\omega\sqrt{\zeta^2 - 1}$$

The $\gamma$ and $\delta$ factors from Equations 3.25 and 3.26 respectively, are determined from the Burgers' quantities at specific temperatures from Table 3.1 [8]. Table 3.2 gives the obtained values for a time step of $1/240\,s$, which is the predefined step size for Bullet Physics [17].

Table 3.2: Geometrical properties of aggregates (unit size)

| Factor | T = 100 °C | T = 125 °C | T = 150 °C |
|---|---|---|---|
| $\gamma_{F_0}$ | 0.999997 | 0.999998 | 0.999999 |
| $\gamma_{F_1}$ | 0.004025 | 0.004043 | 0.004060 |
| $\gamma_{u_1}$ | 0.001541 | 0.000955 | 0.000662 |
| $\gamma_{u_2}$ | 0.000342 | 0.000200 | 0.000129 |
| $\delta_{F_0}$ | -0.001192 | -0.000812 | -0.000594 |
| $\delta_{F_1}$ | 0.932899 | 0.941456 | 0.949107 |
| $\delta_{u_1}$ | 0.731141 | 0.453807 | 0.314999 |
| $\delta_{u_2}$ | 0.162097 | 0.094819 | 0.061383 |

Table 3.2 shows that $\gamma_{F_0}$ and $\delta_{F_1}$ are very close to 1, and $\gamma_{F_1}$ approaches $\Delta t$, which is in line with the Euler Forward theory [34]. It is noted that since this is an analytical solution, these values are assumed to be more precise, which could be an improvement over conventional numerical time integration methods. The negative values of $\delta_{F_0}$ indicates that the contact force tends to go to zero over time, which is expected due to the Maxwell damper from Figure 3.1. Furthermore, $\gamma_{u_1} > \gamma_{u_2}$ and $\delta_{u_1} > \delta_{u_2}$ indicate that the velocity has a larger influence than the accelerations on the Burgers' contact force.

## 3.5 Implementation in Bullet Physics

### Semi-analytical model

A way of implementing the Burgers' contact forces in PyBullet, is by applying these as external forces at the beginning of each time step. As shown in Figure 3.2, a 3-step iterative procedure was followed to implement the semi-analytical model in Bullet Physics. For every element, velocities and accelerations were calculated at different positions using the finite difference method (see Section 3.4). Whereas, Burgers' contact forces (Equation 3.25) were determined using the semi-analytical solution. Subsequently, the obtained contact forces were applied on the elements. The element positions in the next step are determined using Bullet's projected Gauss Seidel solver [28, 29].

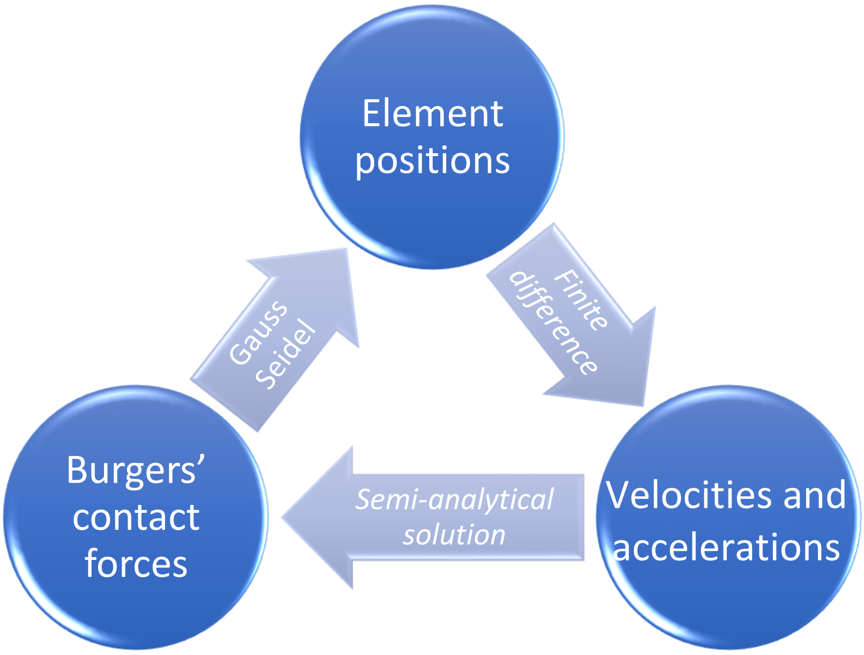


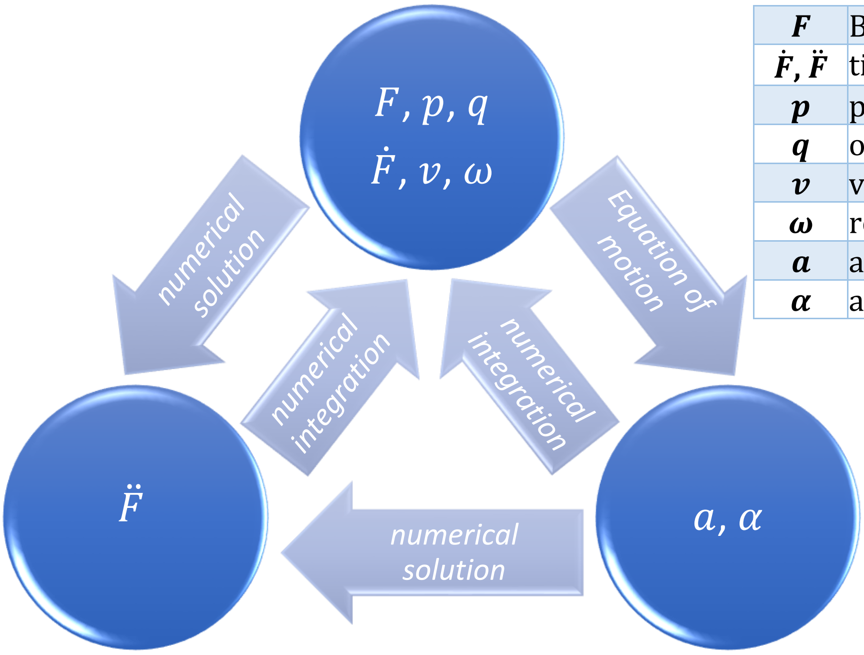
Figure 3.2: Simulation flow chart

It is highlighted that the displacements and contact forces between elements in the previous time steps are stored in arrays. Using the finite difference method, the relative velocity and acceleration are determined. Since Bullet Physics is conservative with its collision detection, the contact distance is also determined when two elements are close, but not touching. This provides an opportunity to determine the relative velocity and acceleration at the first moment of contact. The contact forces, as well as their derivatives are known from the previous time step, allowing for the determination of the forces in the current time step.

### Numerical model

Alternatively to the method described above, numerical integration could be performed separately. It is noted that the collision directions are still calculated using Bullet Physics. This procedure allows for the implementation of complex contact models. This method is slightly more challenging, as rotational accelerations and orientations have to be computed as well.

As shown in Figure 3.2, a 3-step iterative procedure was followed to implement the numerical model in Bullet Physics. For every element, positions and velocities were determined by Bullet itself. Whereas, Burgers' contact forces (Equation 3.15) were determined using a numerical solution. Subsequently, the accelerations and second derivatives of the Burgers' contact forces were determined using the second law of Newton and the equation of motion (Equation 3.9) respectively. The element positions in the next step are determined using a numerical integration method.



| | |
|---|---|
| $\boldsymbol{F}$ | Burgers' contact force |
| $\dot{\boldsymbol{F}}, \ddot{\boldsymbol{F}}$ | time derivatives to BCF |
| $\boldsymbol{p}$ | position |
| $\boldsymbol{q}$ | orientation |
| $\boldsymbol{v}$ | velocity |
| $\boldsymbol{\omega}$ | rotational velocity |
| $\boldsymbol{a}$ | acceleration |
| $\boldsymbol{\alpha}$ | angular acceleration |

Figure 3.3: Simulation flow chart

The Equation of motion is presented in state space representation [39], meaning that it inputs the locations $x$ and velocities $v$ of the particles as well as the current time step $t$, and outputs the derivatives with respect to time. It is noted that the accelerations and second derivatives to the contact forces are the key quantities, as these follow from the equation of motion and the numerical solution, respectively. Using state space representation allows for the initial value problem to be solved numerically. Equation 3.27 shows a typical state space function. For ease of understanding, an example of the numerical implementation process in Python is presented in Appendix A.

$$input: \; t, x, v$$

$$a = \frac{1}{m} \sum F(t, x, v) \tag{3.27}$$

$$output: \; v, a$$

It is highlighted that in 3 dimensions with multiple elements, the function for $F$ can be complex, and rotational accelerations should also be considered. The equation of motion is captured by the following steps:

1. Positions and rotations of all the elements were prescribed in Bullet Physics. It is noted that the collision could be detected from the physics engine.
2. Bullet Physics returns the collision information, such as interacting elements, contact distance, contact normal and position of contact.
3. Based on the information as described above, contact forces and moments can be determined. Usually this is done automatically in PyBullet. However, here it is done manually, such that a complex contact model can be implemented.

Considering $\boldsymbol{\Delta p}_A$ and $\boldsymbol{\Delta p}_B$ as the contact position tensor with respect to their centres of masses for element A and B, respectively. Considering $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ as the first order linear velocities tensors of both elements. Considering $\boldsymbol{\omega}_A$ and $\boldsymbol{\omega}_B$ as the first order angular velocity tensors. Equation 3.28 shows how to determine the first order relative velocity tensor at the position of contact.

$$\boldsymbol{\Delta v} = \boldsymbol{v}_B + \boldsymbol{\omega}_B \times \boldsymbol{\Delta p}_B - \boldsymbol{v}_A - \boldsymbol{\omega}_A \times \boldsymbol{\Delta p}_A \tag{3.28}$$

Considering $\boldsymbol{n}$ as the first order contact normal tensor between the elements. Equation 3.29 shows how to determine the relative normal velocity scalar between element $A$ and $B$, positive for compression.

$$\Delta v_n = \boldsymbol{\Delta v} \cdot \boldsymbol{n} \tag{3.29}$$

Equation 3.30 shows how to determine the normalized first order velocity tensor, perpendicular to the normal, in the direction of the relative velocity.

$$\boldsymbol{\Delta v_p} = \frac{\boldsymbol{\Delta v} - \Delta v_n\, \boldsymbol{n}}{|\boldsymbol{\Delta v} - \Delta v_n\, \boldsymbol{n}|} \tag{3.30}$$

Considering $k$ and $c$ respectively as the stiffness and damping between the elements. Equation 3.31 shows how to determine the normal force scalar between the elements. It is noted that a more complex function can be implemented here, and is described in Paragraph 3.2.

$$F_N = k\, d + c\, \Delta v_n \tag{3.31}$$

Considering $\mu$ as the friction coefficient between element $A$ and $B$. Equation 3.32 shows how to determine the frictional force scalar between the elements.

$$F_f = \mu\, F_N \tag{3.32}$$

Equation 3.33 shows how to determine the first order total force tensors, working on elements $A$ and $B$.

$$\begin{aligned} \boldsymbol{F}_A &= F_N\, \boldsymbol{n} - F_f\, \boldsymbol{\Delta v_p} \\ \boldsymbol{F}_B &= -\boldsymbol{F}_A \end{aligned} \tag{3.33}$$

Equation 3.34 shows how to determine the first order moments tensor, applied on elements $A$ and $B$.

$$\begin{aligned} \boldsymbol{M}_A &= \boldsymbol{\Delta p}_A \times \boldsymbol{F}_A \\ \boldsymbol{M}_B &= \boldsymbol{\Delta p}_B \times \boldsymbol{F}_B \end{aligned} \tag{3.34}$$

Considering $m_A$ and $m_B$ as the masses of elements $A$ and $B$ respectively. Equation 3.35 shows how to determine the linear accelerations, following from the force tensor. Since multiple elements may interact with the element of interest, it is noted that the prime symbol $'$ has been used to indicate that this variable is being updated, instead of assigned.

$$\begin{aligned} \boldsymbol{a}_A{}' &= \boldsymbol{F}_A / m_A \\ \boldsymbol{a}_B{}' &= \boldsymbol{F}_B / m_B \end{aligned} \tag{3.35}$$

Considering $\boldsymbol{J}_A$ and $\boldsymbol{J}_B$ as the second order rotational inertia tensor in global coordinates for elements A and B, respectively. Equation 3.36 shows how to determine the angular accelerations for this object in local coordinate system. Since multiple elements may interact with the element of interest, the prime symbol $'$ has been used to indicate that this variable is being updated, instead of being assigned.

$$\begin{aligned} \boldsymbol{\alpha}_A{}' &= \boldsymbol{J}_A{}^{-1} \cdot \boldsymbol{M}_A \\ \boldsymbol{\alpha}_B{}' &= \boldsymbol{J}_B{}^{-1} \cdot \boldsymbol{M}_B \end{aligned} \tag{3.36}$$

With the known derivatives of input varies, the state space representation of the equation of motion can be used for numerical integration in Python. Additional considerations for regulating the stability for the friction and non-penetration condition of the aggregates, as well as the determination of the inertia tensor in global coordinates, can be found in appendices C, D and E.

# 4 Model description

## 4.1 Aggregates

This chapter describes the geometrical model, such that the superpave gyratory compaction process can be digitally analysed. In order to accurately describe the shape of the elements, ten different aggregates have been 3D scanned. The resulting meshes were imported in Bullet Physics [17]. Figure 4.1 shows the ten different aggregate shapes which have been used in the model. A video is also presented which shows the aggregates from all angles.

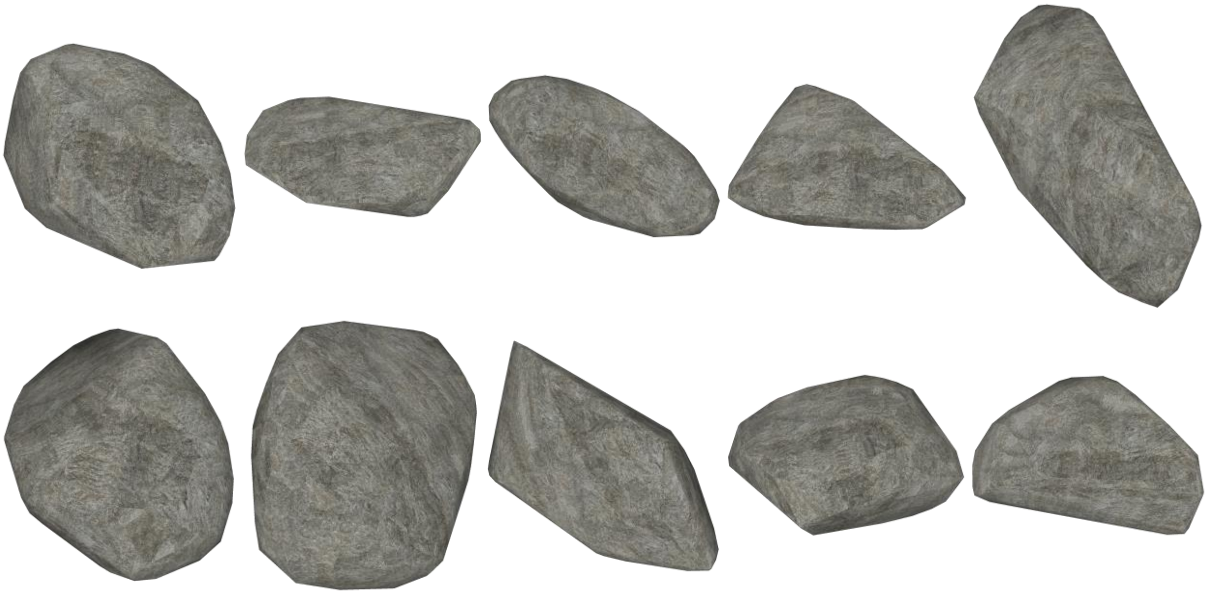


Figure 4.1: Aggregate shapes
(a video is also available [19])

The aggregate volume ranges from 0.6 to 1.0 $mm^3$ for a unit size element with sieve of size 1 $mm$. The surface area ranges from 3.8 to 5.8 $mm^2$. The masses can easily be calculated using the density of aggregates and mortar, which is 2400 to 2900 $kg/m^3$ [40], and 1030 to 1040 $kg/m^3$ [41], respectively. Although it is noted that due to the high viscosity of the bitumen and the relatively low mass of the coarse aggregates, the inertia terms might be neglectable. The centre of gravity, as well as the surface area, volume and inertia tensor were determined using MeshLab [42]. A rock texture was applied to make the aggregates visually more appealing.

The sieve size of the scanned aggregates was initially unknown. Therefore, these were to be determined. This way, the digital shapes can be rescaled to a unit sieve size, and can later be implemented with a custom size, such that any degradation can be chosen. In order to determine the sieve size of a digital aggregate, it is loaded in PyBullet multiple times at different sizes. A digital sieve is loaded as well. The aggregates are then subjected to gravity. The maximum object size that passes through is determined to be the sieve size. Multiple iterations have been performed per aggregate, where the range of the element sizes was reduced such that the actual sieve size can be determined with great accuracy. This method has been verified using a unit size sphere, which resulted in a mesh size very close to 1. Figure 4.2 shows the sieve size determination for each mesh in PyBullet. A video is also presented.

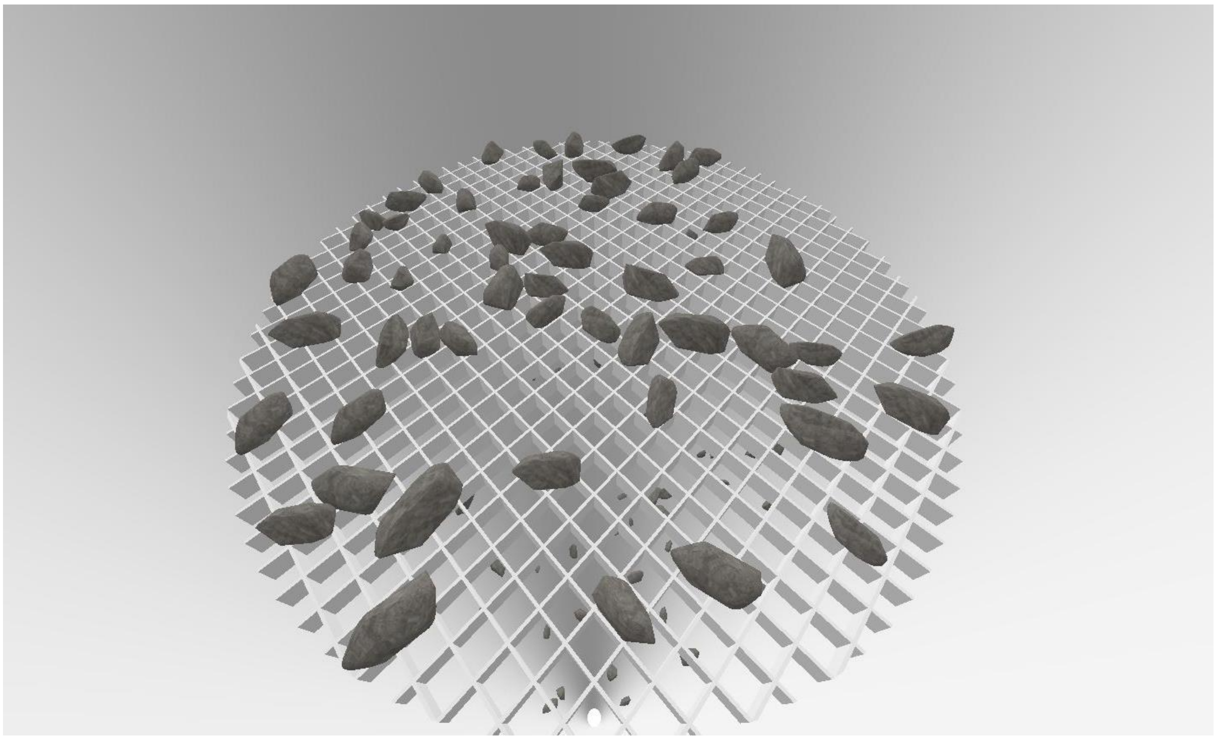


Figure 4.2: Sieve size analysis in Bullet Physics
(a video of this simulation is also available [19])

### 4.2 Mortar layer

Asphalt consists of aggregates, coated with a bituminous layer. In order to determine the thickness of the mortar layer, Al Khateeb et al. [8] regarded a sample mixture and multiplied the size all aggregates with a factor of 1.087. In this research, a constant value was chosen for all sieve sizes using their sample data. The total volume of the mortar was divided by the total surface area of the aggregates, resulting in a uniform thickness for the mortar layer of 0.44 $mm$.

Table 4.1 presents the sample data [8] of a lab mixture, as well as the number of elements needed to approximate this volume. It is noted that since the aggregate shapes differ from Al Khateeb's research, the number of elements per sieve size also differs. The number of elements per sieve size was chosen such that the volume accurately matches the given data. Equation 4.1 shows how the average thickness of the mortar layer is determined.

Table 4.1: Example data of a lab mixture

| Sieve Size (mm) | | Aggregates | Number of | Elements | Surface |
|---|---|---|---|---|---|
| Min | Max | Volume (cm$^3$) | elements | Volume ($cm^3$) | Area ($cm^2$) |
| 2.0 | 5.6 | 46.37 | 870 | 46.41 | 663.49 |
| 5.6 | 8.0 | 141.88 | 552 | 142.05 | 1267.11 |
| 8.0 | 11.2 | 218.35 | 303 | 218.70 | 1384.74 |
| 11.2 | 16.0 | 136.48 | 67 | 138.05 | 615.36 |
| 16.0 | 22.4 | 11.05 | 2 | 12.17 | 37.23 |
| Mortar | | 174.83 | - | - | - |
| Total | | 728.95 | 1794 | 557.38 | 3967.93 |

$$\tau_{mor} = \frac{V_{mor}}{\sum A_{agg}} = \frac{174\,830}{396\,793} \approx 0.44\ mm \tag{4.1}$$

Although this gives an indication of the average mortar thickness, it is noted that this value cannot be interpreted as a standard value, but needs to be determined per mixture and differs with degradation, aggregate shape and amount of bituminous binder.

In order to accommodate for the different properties for the mortar and the aggregates, it was chosen to work with a so called ‘multibody’. First, the aggregate shape was implemented with the desired sieve size. Second, a slightly bigger element was implemented, and rigidly connected, to represent the mortar layer. This allows for mortar–mortar interactions to be modelled with Burgers’ contact model, and aggregate–aggregate interactions using a non-penetration condition. Figure 4.3 shows an example of the implementation of the mortar coating on an aggregate.

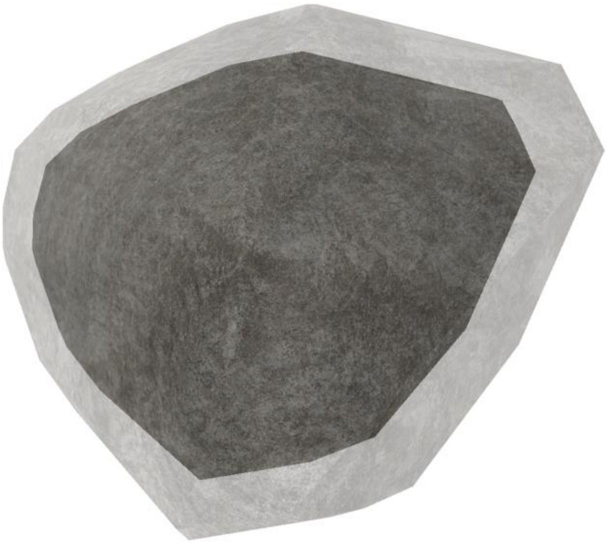


Figure 4.3: Implementation of the mortar layer (transparent) on an aggregate

One might argue that this does not lead to a uniform mortar thickness for non-spherical shapes, which is indeed the case. In fact, the mortar layer thickness is overestimated at the elongated parts. This effect is somewhat mitigated by the relatively small contact area at these places.

For consistency it was chosen to let aggregates and the mortar layers of different elements never interact with each other. So, mortar – mortar leads to Burgers' forces, and aggregate – aggregate leads to a non-penetration condition. Any mortar – aggregate collision was ignored. This could easily be changed with more complex contact conditions.

## 4.3 Degradation

Al Khateeb's [8] degradation has been used for this research as well. With the only difference being that the maximum aggregate size was chosen to be 16 $mm$. The volume of the aggregates larger than that is added to the 11.2–16.0 $mm$ sieve size. This is done to avoid a small number of elements in the largest sieve size, and so create a more accurate volume percentage in all groups. Also, there was chosen for a mould diameter of 100 $mm$, which does not allow for aggregates larger than 16 mm [2]. A linear element size distribution was chosen for the sieves. Table 4.2 gives the degradation of the digital specimen used during the simulations.

Table 4.2: Degradation digital specimen

| Sieve Size (mm) | | Aggregates volume (%) | Number of elements |
|---|---|---|---|
| Min | Max | | |
| 2.0 | 5.6 | 8.4 | 726 |
| 5.6 | 8.0 | 25.6 | 461 |
| 8.0 | 11.2 | 39.3 | 253 |
| 11.2 | 16.0 | 26.7 | 60 |
| **Total** | | **100** | **1500** |

A mould diameter of 100 $mm$, together with 1500 elements, allows for reasonable computation time and meets the requirements for $h_{min}$ [2], as described in Equation 2.2. The actual volume per sieve changes slightly with the randomly selected aggregates. Since the number of elements per sieve is large, the volume remains considerably consistent.

## 4.4 Contact area

Burgers' contact forces depend on the contact area, and since PyBullet does not assess the contact area or volume, this has to be done manually. Both can be accurately determined using ray tracing, a technique for modelling light transport.

Figure 4.4a shows an example of an aggregate, which interacts with another element during the simulation. Figure 4.4b removes the part of the aggregate which overlaps, revealing the contact area in white. The black dot represents the contact point, which is the point where the overlap is the largest. Figure 4.4c shows how the contact area can be estimated using ray tracing. Eight rays have been casted in different directions from the contact point. The distance of these rays can be measured. The lower limit for the contact area is presented in blue.

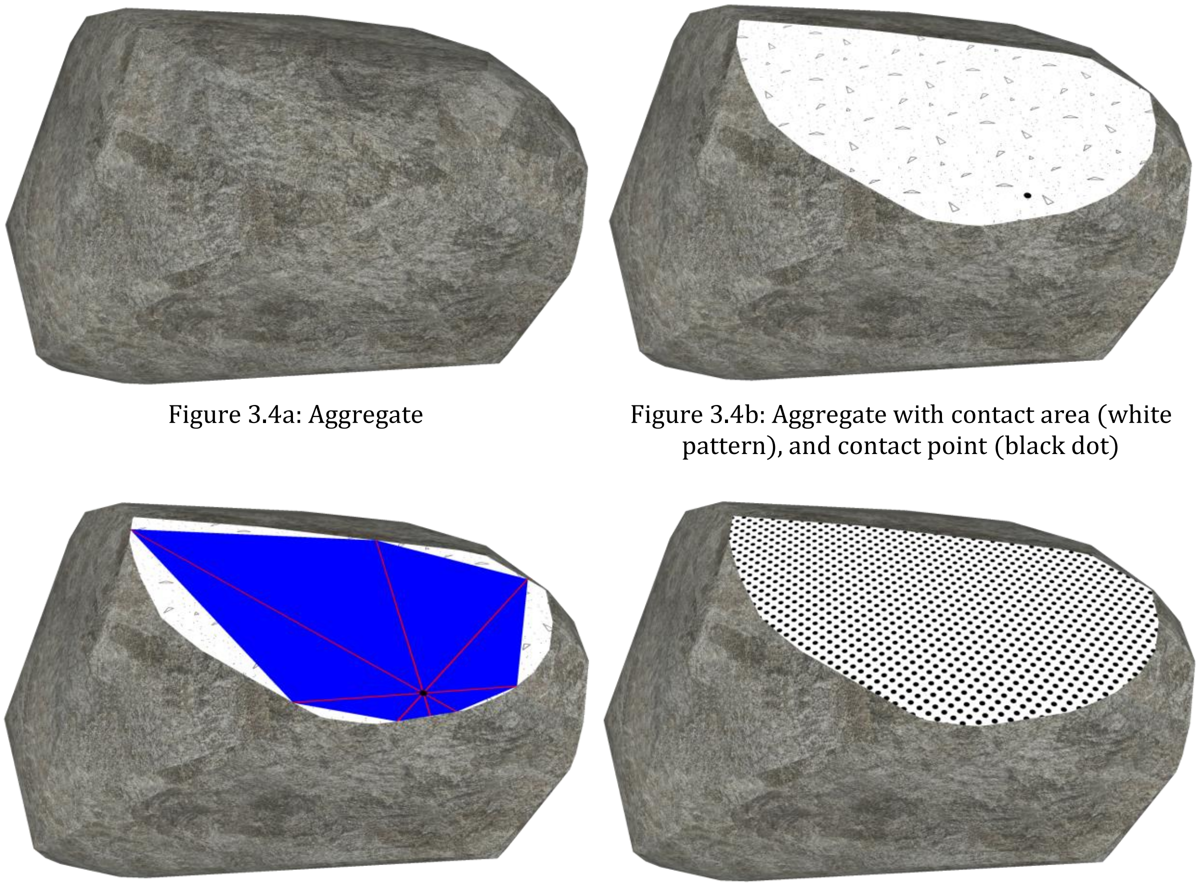


Figure 3.4a: Aggregate

Figure 3.4b: Aggregate with contact area (white pattern), and contact point (black dot)

Figure 3.4c: Estimated contact area (blue), using perpendicular ray tracing (red)

Figure 3.4d: Estimated contact volume, using parallel ray tracing (black dots)

Figure 4.4: Determining the contact area and volume

Since all elements in PyBullet are convex, any concave shape cannot exist in the contact surface. Therefore, Equation 4.2 gives the lower limit for the contact area $A_c$, based on the number of rays $n_r$, and length of the rays $r_i$.

$$A_c \geq \frac{1}{2} \sin\left(\frac{2\pi}{n_r}\right) \sum_{i=1}^{n} r_i \, r_{i+1} \tag{4.2}$$

In the case of two colliding elements, the contact area of both are to be determined, and the lowest value should be chosen. To enhance precision, more rays could be casted. Also, the shape of the contact area could be more accurately described by utilizing the angle of contact. The latter involving some more advanced mathematics. Both would come at the cost of additional computation time. However, this method does demonstrate the possibility of determining the contact area.

Ray tracing also allows for determining the contact volume, which is useful for determining the total volume loss. Figure 3.4d presents the same aggregate, where again the part of the aggregate which overlaps has been removed for visibility. The black dots represent rays parallel to the contact normal. These rays give the height of the particle at every location. This allows for the calculation of the contact volume. Equation 4.3 gives the total contact volume, as a function of the distance between rays, $\Delta r$, and the length of the rays $r_i$.

$$V_c \approx \Delta r^2 \sum_{i=1}^{n_r} r_i \tag{4.3}$$

Alternatively, more sophisticated numerical integration techniques can be used [43]. Also, by taking into consideration the angle at which the rays are reflected, the contact volume could be determined very precisely. Since many rays are casted, determining the volume does come at the price of additional computation time. Therefore, this method is only useful for post-processing in case volume losses are to be determined.

## 4.5 Boundary conditions

As described in Paragraph 2.3, the asphalt specimen is subjected to specific boundary conditions during gyratory compaction. The key properties are: the gyratory motion of the cylindrical mould, and the compressive force on the top end plate, which must remain horizontal [2]. This is achieved by the the implementation of specific objects in the Bullet Physcis engine, and carefully setting the correct position, orientation, and velocity for each element in time.

**Implementation of the mould**

The mould restricts the elements from sideways motion, and causes the gyration of the asphalt specimen. The mould is velocity controlled, meaning that it is not influenced by other objects or gravity. Instead, the position and velocity in each time step are fully determined by the programming code. For every time step the correct position and velocity needs to be determined in order to let Bullet correctly assess the contact forces and friction.

The mould consists of lamellae, just like a circle can be built up from multiple linear segments. These lamellae can be rotated in any orientation to simulate the angle of gyration and rotational velocity. In order to correctly implement these segments, three sequential operations have to be performed. This is done using quaternion multiplication. One lamella consists of a vertical panel at location $[1/2\,, 0, 0]$. The diameter of the mould can easily be changed by resizing the object. By implementing multiple panels at different orientations, the entire mould is created. Considering $n_{mld}$ as the number of mould lamellae. Equation 4.4 shows the rotation of all segments $i$ over the $z$-axis to form the cylinder itself.

$$\boldsymbol{q}_{\boldsymbol{1},\boldsymbol{i}}\left(\alpha = 0\,;\, \beta = 0\,;\, \gamma = 2\pi \frac{i}{n_{mld}}\right) \tag{4.4}$$

$$with \quad i = 0, 1, 2, \dots, n_{mld} - 1$$

Next, the segments are rotated, such that the angle of gyration $\phi$, is satisfied. Equation 4.5 presents the rotation over the $x$-axis.

$$\boldsymbol{q_2}(\alpha = \phi \; ; \; \beta = 0 \; ; \; \gamma = 0) \tag{4.5}$$

The rotations also need to be correctly described over time. Equation 4.6 presents the rotation over a time segment $j$ over the $z$-axis.

$$\boldsymbol{q_{3,j}}\left(\alpha = 0 \; ; \; \beta = 0 \; ; \; \gamma = \frac{2\pi \, j}{T_{mld}/\Delta t}\right) \tag{4.6}$$

$$with \;\; j = 0, 1, 2, \ldots, T_{mld}/\Delta t - 1$$

In order to convert Euler rotations to quaternions, Equation 2.4 can be used. Alternatively, PyBullet offers a built-in function 'getQuaternionFromEuler'. It is noted that the number of time steps $j$ is not limited, but simply continues at 0 after the last value is reached.

In order to get the final orientations, Equation 4.7 quaternion multiplies the three rotations to get the correct rotation for every segment $i$, at every time step $j$.

$$\boldsymbol{q_{ij}} = \boldsymbol{q_{1,i}} \cdot \boldsymbol{q_2} \cdot \boldsymbol{q_{3,j}} \tag{4.7}$$

This way, the correct orientation at every discretised time step is determined. It is noted that only the orientation is described, which does not lead to a velocity in the simulation. In order the ensure the correct direction for the friction, the velocity of the mould is determined as well. Equation 4.8 gives the rotational velocity of the mould around the $z$-axis.

$$\omega_z = \frac{2\pi}{T_{mld}} \tag{4.8}$$

Equation 4.9 gives the velocity of the mould at a point with height $h$.

$$v = \omega_z h \sin(\phi) \tag{4.9}$$

Equation 4.10 shows how to separate the velocity in $x$ and $y$ direction. Where $\phi$ is the angle of gyration, and $\gamma$ is the current rotation around the $z$-axis.

$$\begin{aligned} v_x &= v \sin(\gamma) = 2\pi \frac{h}{T_{mld}} \sin(\phi) \sin(\gamma) \\ v_y &= v \cos(\gamma) = 2\pi \frac{h}{T_{mld}} \sin(\phi) \cos(\gamma) \end{aligned} \tag{4.10}$$

Equation 4.11 show how the rotational velocity around the $x$ and $y$ axes is determined.

$$\begin{aligned} \omega_x &= -\frac{v_y}{h} = -\frac{2\pi}{T_{mld}} \sin(\phi) \sin(\gamma) \\ \omega_y &= \frac{v_x}{h} = \frac{2\pi}{T_{mld}} \sin(\phi) \cos(\gamma) \end{aligned} \tag{4.11}$$

This final rotational velocity is implemented in every discretised time step for the mould, such that is accurately describes the linear velocity at every point of the mould. Alternatively to the lamellae, the mould could be created using a single concave cylindrical element. This improves the computation time, but runs the risk of losing elements in case of relatively large overlaps, which might happen in the case of very small elements or at relatively large angles of gyration.

**Implementation of the end plates**

The end plates on top and the bottom refrain the particles from up- and downward motion. The compressive force is applied on the top end plate. Preferably, the end plates remains horizontally [2]. The bottom end plate is a static object. The top end plate needs to receive the correct position and velocity. This object also exerts the compressive force on top. Two methods exist for modelling the end plate, where the end plate's horizontal position is velocity controlled, and the height is unrestrained. This could be achieved by creating a contact link between the dynamic and a static object, or by implementing velocity controlled objects around the end plate. The latter was chosen. The end plate is modelled as an unrestrained octagonal prism, which is wider than the mould such that there are no gaps. The horizontal location, as well as its rotation are restricted by place controlled tetrahedrons. This way, the only degree of freedom is the height of the end plate. Figure 4.5 gives a visual representation of the collision shape, where the bottom of the object functions as the top end plate.

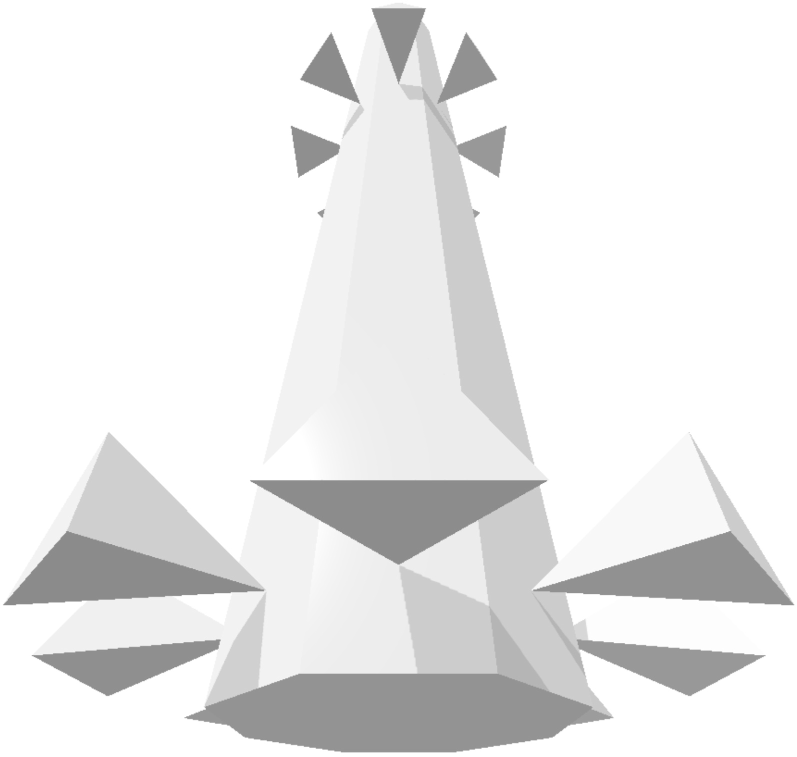


Figure 4.5: End plate's collision shape

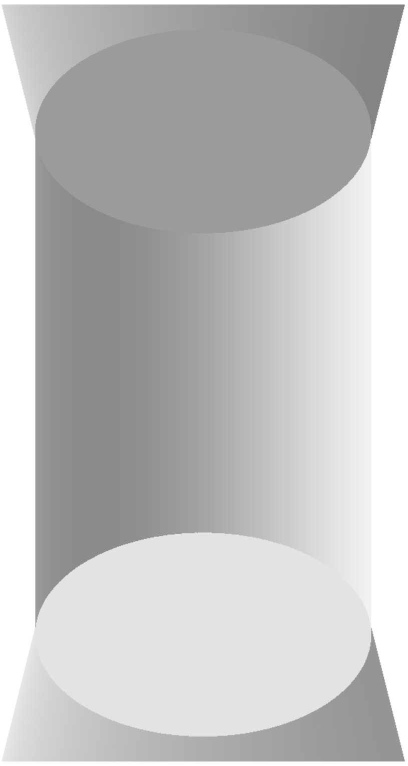


Figure 4.6: Mould's visual shape

The horizontal motion has to be precisely described in order to make the simulation valid, so that friction can be correctly modelled. Equation 4.10 from gives the correct horizontal velocities. The bottom end plate was modelled as a static element, which does not move.

Of course, the word 'plate' does not accurately describe this object. However, at the bottom it acts as a horizontal plate with horizontal locations prescribed, and free vertical motion. During post-processing, these objects are replaced by a visually more appealing end plate. Figure 4.6 shows what the final mould looks like.

## 4.6 Simulation processes

The simulation consists of three phases: placement, compaction, and analysis. It is useful to separate the placement and the gyratory compaction process, as the initial positions are re-used many times for parametric study. The analysis can only take place after the simulation is completed, as the simulation time is well beyond the compaction time. The latter also allowing for the creation of a video.

### Placement process

An important aspect of performing a simulation with granular material is the placement of particles. The elements must be arranged in a realistic manner, lie still, and may not overlap. Also, in the case of hot asphalt mixtures, the top surface of the specimen must be relatively flat before compaction.

Although this is a study in itself, a simple tool has been developed to meet these requirements. Figures 4.7, on the next page, gives a visual representation of the placement process. A video is also presented. Figures 4.7a shows how all elements are first placed in circles with sufficient spacing between them such that these do not overlap. Figures 4.7b-d show how they are then subjected to gravity only, so that the particles can arrange themselves. Figures 4.7e shows how the mould is consecutively rotated slightly such that the angle of gyration is satisfied. Figures 4.7f shows how the elements are finally pre-compressed resulting in a relatively flat surface of the specimen. Finally, the initial positions, orientations, sizes and shapes are stored in text files, such that the elements can easily be loaded during the simulation. During this positioning, the mortar layer is treated as impenetrable, as to avoid initial overlaps.

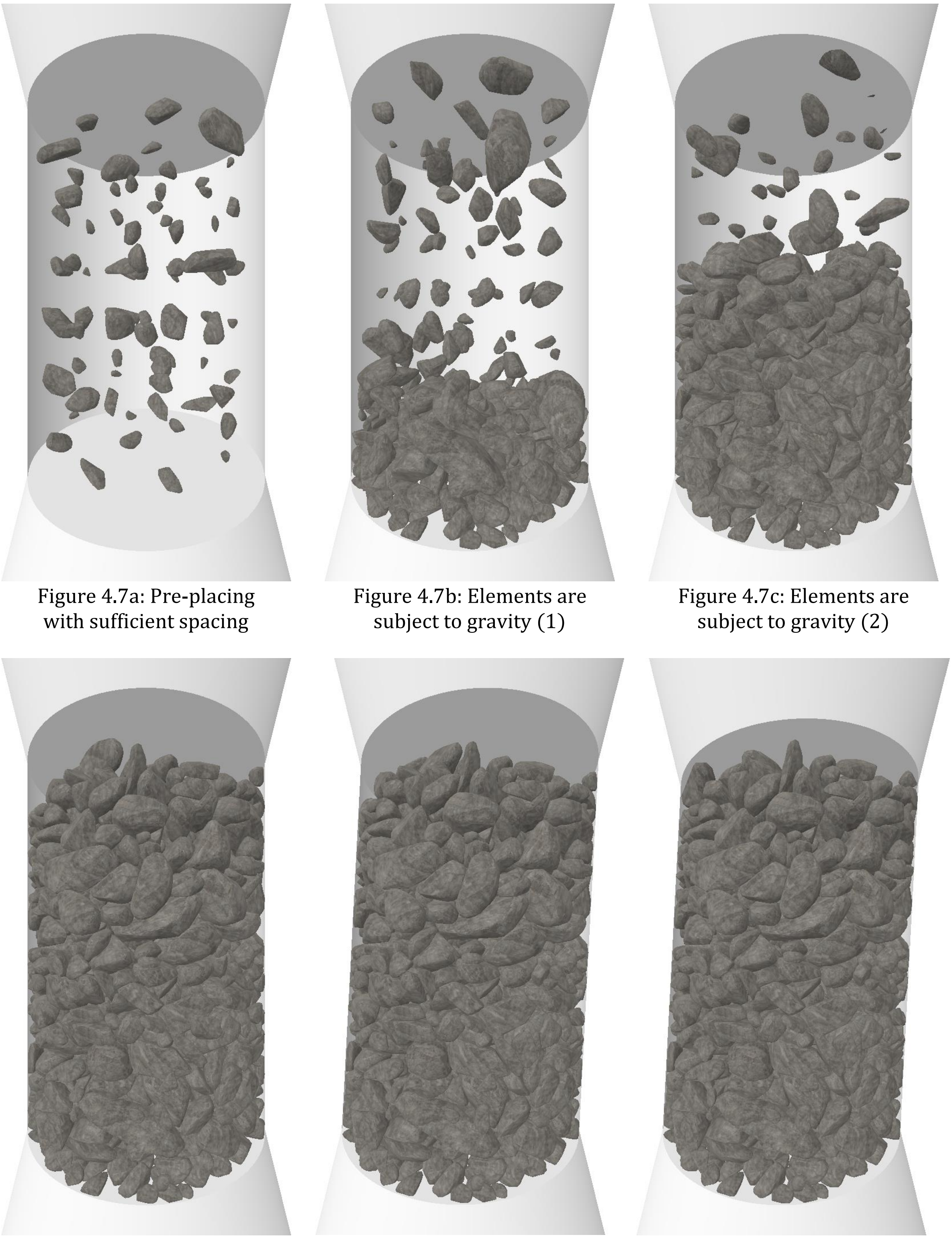


Figure 4.7a: Pre-placing with sufficient spacing

Figure 4.7b: Elements are subject to gravity (1)

Figure 4.7c: Elements are subject to gravity (2)

Figure 4.7d: All elements are stationary

Figure 4.7e: Rotation of the mould to the desired angle of gyration

Figure 4.7f: Pre-compression of the specimen

Figure 4.7: Particle placement process
(a video of this simulation is also available [19])

The NEN norms do not specify specific criteria for pre-compression. Therefore it is chosen to compress the specimen until the top end plate is semi-static. In practice, this means until it reaches a certain velocity. A practical value is 0.25 $mm/s$. This results in a specimen which is relatively flat on the top, and gives an acceptable initial compression rate during the simulation.

### Gyratory compaction process

The gyratory compaction process forms the main body of the simulation. The process is described in Paragraphs 2.3 and 4.5. The initial positions of the elements follow from the placement process, as described above. The number of gyrations is set to 100 for all simulations. With 1500 aggregates and mortar shapes this results in a computation time of 12 to 14 hours on an i5 desktop processor. Multiple simulations can be run in parallel simply by creating multiple instances. Memory usage is well below 2 GB for a simulation. This means that it can easily be run on any desktop pc. Parametric study can be performed by changing specific properties, while all other properties remain identical, including initial positions and exact degradation of the aggregates. Finally, the positions, and orientations over time, as well as the height of the top end plate, are stored to a text file, such that these can be used during the post-processing analysis.

### Post-processing

The main goal of post-processing is the analysis of the results. This includes creating the void curve, a video of the simulation, and visualizing trajectories of the aggregates inside the mould. The height of the end plate over time allows for the determination of the mean air void content over time. Screen shots and videos are made visually more appealing by adding rock textures, and creating attractive visual shapes for the mould and end plates as described in Paragraph 4.5. When analysing aggregate trajectories, this initially resulted in oscillatory movement, as the mould is gyrating. An easy method to overcome this is by only presenting the data at one specific orientation of the mould, or positioning the aggregates relatively to the centre of the mould.

# 5 Verification of the model

## 5.1 Verification questions

This section verifies the numerical contact model as described in chapter 3. In order to verify Burgers' equation of motion, the numerical model is compared with the analytical solution. The four research questions are as follows.

1. Are the displacements over time determined correctly during compression under a constant external force?
2. Is the non-penetration condition in order for the aggregates?
3. Is the contact force over time determined correctly in the case of a constant velocity?
4. What is the influence of the mass on the results?

## 5.2 Experiment to verify the numerical model

To answer these verification questions, a simple experiment is performed. The experiment runs as follows: two unit-size elements with a mortar layer of 0.44 $mm$ are 1) compressed using a constant external force, 2) held into place by the non-penetration condition of the aggregates, and 3) pulled apart from each other with a constant velocity. The process is repeated with a different mass for the aggregates. Figure 5.1 gives a visual representation of the simulation. Table 5.1 gives the properties of the simulation.

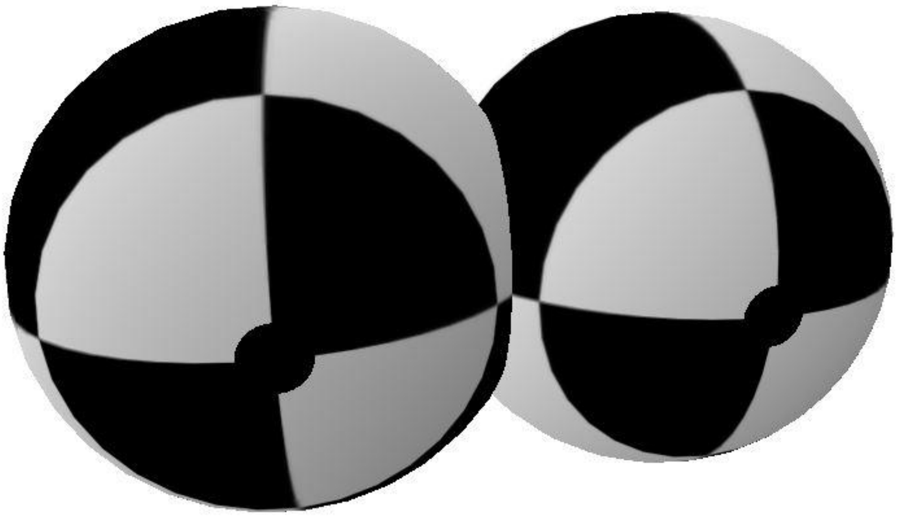


Figure 5.1: Verification experiment

Table 5.1: Properties of the verification experiment

| Variable | Symbol | Value | Unit |
|---|---|---|---|
| Thickness of the mortar layer | $\tau_{mor}$ | 0.44 | $mm$ |
| Temperature | $T$ | 150 | $°C$ |
| Contact area (constant) | $A_c$ | 0.88 | $mm^2$ |
| Force (phase 1 & 2) | $F_0$ | 1.0 | $N$ |
| Velocity (phase 3) | $u_3$ | $-2.0 \cdot 10^{-3}$ | $mm/s$ |
| effective mass aggregates | $m_{eff}$ | $0.1 \leftrightarrow 10$ | $kg$ |
| effective stiffness aggregates | $k_{eff}$ | $1.0 \cdot 10^3$ | $N/mm$ |
| Critical damping aggregates | $c_{cr}$ | $2\sqrt{k_{eff} m_{eff}}$ | $Ns/mm$ |

Equation 5.1 shows how the effective mass of the aggregates is calculated. The equation is comparable to the effective stiffness of two springs in series.

$$m_{eff} = \frac{m_A m_B}{m_A + m_B} \tag{5.1}$$

## 5.3 Comparison between the numerical and analytical results

First, the two particles with an effective mass of 0.1 $kg$ are being analysed. Figure 5.2 gives the displacement of the particles over time. Both the analytical and numerical solution are presented. Three segments are distinguished. In the first segment, the simulation is force controlled while the elements are drawing nearer. In the second segment, the aggregates touch and the non-penetration condition is activated, while the compressive forces are still being active. In the third segment, the simulation is velocity controlled while the elements move apart from each other.

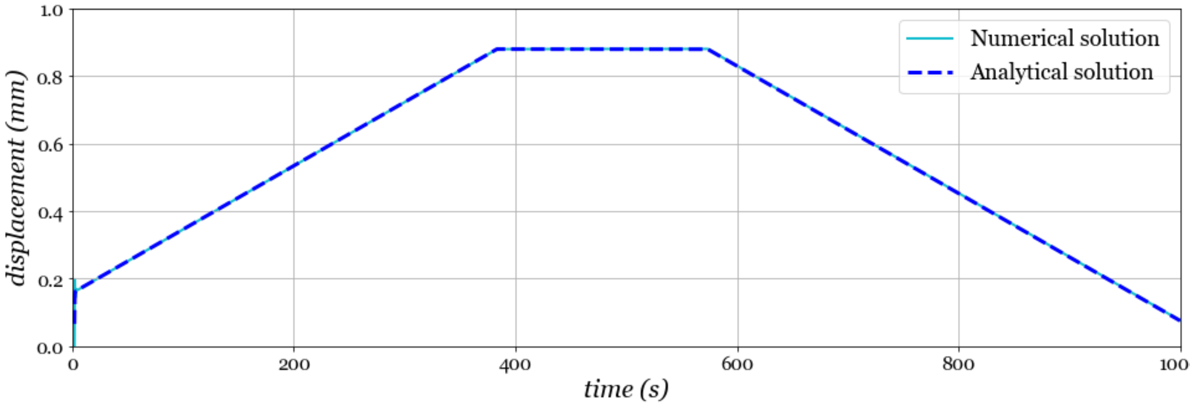


Figure 5.2: Displacements over time

Figure 5.3 shows the Burgers' contact force over time. In the first segment, the simulation is force-driven, resulting in a constant force. In the second segment, the displacement remains constant, resulting in the Burgers' forces going to zero over time. It is noted that in the second segment, the actual contact forces are bigger due to the non-penetration condition. In the third part, where the elements separate, the contact forces become negative, meaning a tensional force is present between the elements.

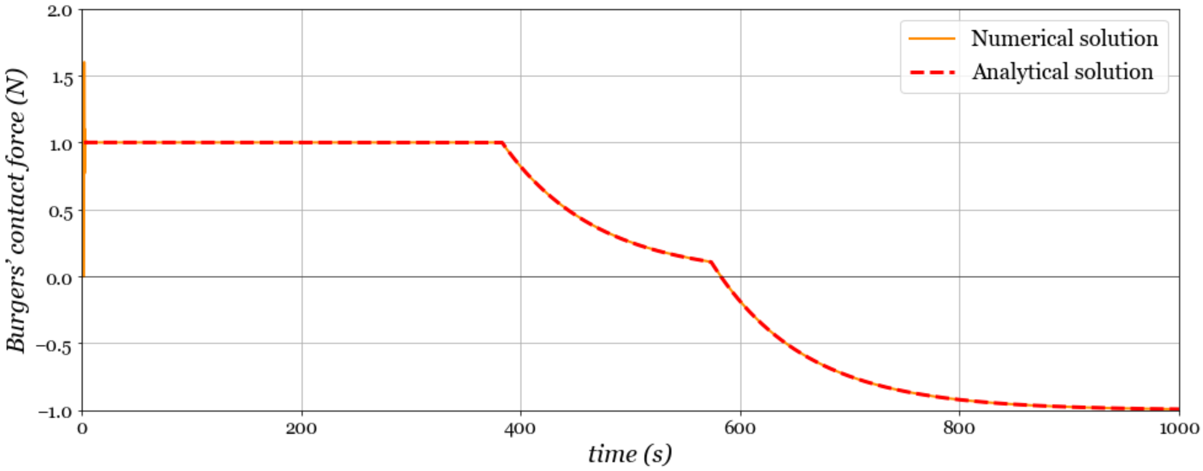


Figure 5.3: Burgers' contact forces over time

The first three validation questions can be answered in the affirmative, as Figures 5.2 and 5.3 show that the numerical and analytical solution are well in agreement with each other. However, also a few differences can be observed. Figure 5.2 shows that at the start of the simulation, the numerical solution starts with zero initial overlap, whereas the analytical solution starts with an initial elastic deformation. Figure 5.3 shows that the same phenomenon occurs in case of the Burgers' contact force. This is caused by neglecting inertia in the theoretical model, such that instantaneous deformations can occur. What is more, the numerical solution seems to make oscillations at the start, which is discussed further in this paragraph. Despite the initial differences, the solutions are well in agreement with each other.

To investigate this issue further, two particles with an effective mass of 10 $kg$ are being analysed. Figure 5.4 gives the relative displacement of the particles. Figure 5.5 compares the Burgers' contact force.

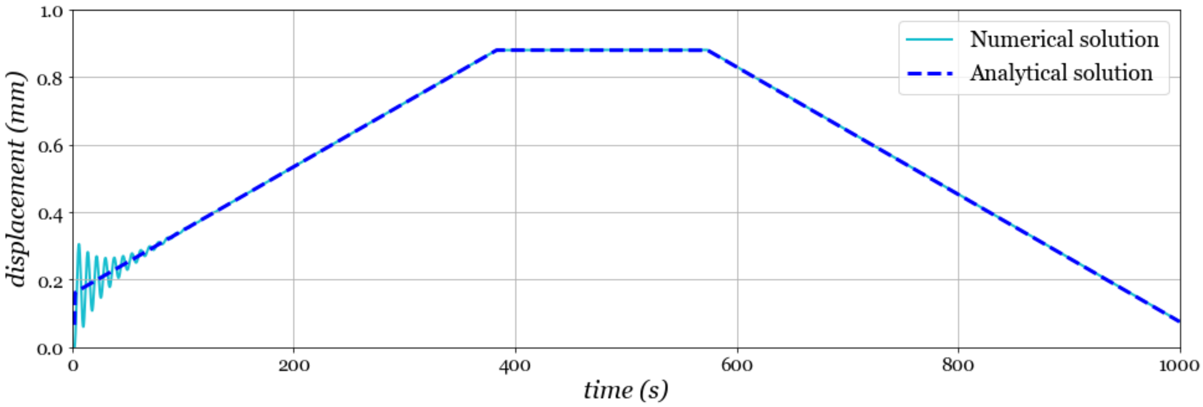


Figure 5.4: Displacements over time

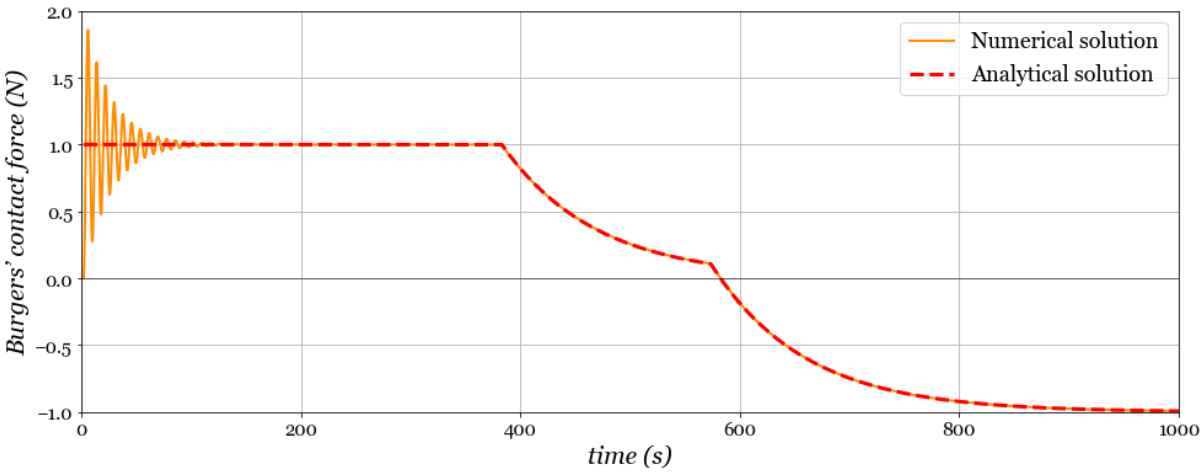


Figure 5.5: Burgers' contact forces over time

Figures 5.4 and 5.5 show an increase in the oscillatory motion. Thus, it becomes clear that the initial oscillations are indeed caused by the inertia of the elements, and become more pronounced in the case of larger masses. After some time, the oscillations die out, and the numerical solution converges towards the analytical one. After that, the results are correct.

This allows for the answering of the fourth validation question. When comparing the results with the two different masses, it becomes clear that the initial oscillations die out more quickly with a smaller mass. What is more, the magnitude of the oscillations in the displacement are smaller in case of the smaller mass, whereas the magnitude of the oscillations in the Burgers' contact forces is roughly the same. The oscillations in the displacement are of much greater importance, since two interacting elements may lose contact too early in case of an external pulling force.

This experiment shows that there is a maximum value for the mass, for which the numerical solution can be considered correct. A realistic mass would be in the order of milligrams, instead of kilograms. It is useful to know that a much larger mass can still lead to correct results. However, the problem with implementing realistic masses with this method is that the computation time increases drastically, and so becomes impractical.

## 5.4 Verification of the semi-analytical solution

The implementation of the semi-analytical solution in a stand alone tool gave very similar results as displayed in Paragraph 5.3, and the method itself is therefore verified. There is no need to present these almost identical graphs here again.

However, after implementing this method in PyBullet, instability was observed, where rapid motion occurs. These oscillations are much bigger and more rapid than what was seen in Figure 5.4. The cause for this is in the integration method, where Bullet makes use multiple iterations per time step to ensure stability. When applying Burgers' contact force as an external force, these are determined only once per time step, and are not considered during the iterative process. Thus, this can quickly lead to instability. Therefore, it proves impractical to directly implement a custom contact model in PyBullet.

## 5.5 Improved Burgers' contact model

During the simulations with Burgers' contact model, oscillations were observed, as is shown in Figure 5.4. This does not accurately describe the material behaviour. Since the magnitude of the oscillations depends on the mass, this is likely to be caused by the neglected inertia terms. Therefore this paragraph proposes two improved contact models.

Figure 3.1 presented the currently existing Burgers' contact model. It showed that the Maxwell spring is undamped, which is likely to cause the oscillations. This spring was intended for instantaneous elastic deformation, but unfortunately resulted in unrealistic oscillations. Therefore, two improved models are proposed: one where this spring receives an additional damper, and one where it is removed altogether. The values for the springs and dampers change as well.

Figure 5.6 presents the first proposal for an improved contact model. Parallel to the Maxwell spring on the left side, a critical damper is added. Equation 5.2 presents the value for the added Maxwell damper. Also the Kelvin-Voigt damper on the right side must be critically or supercritically damped. Equation 5.3 gives the new value for this damper. Equation 5.4 presents the equation of motion. A derivation can be found in appendix F.

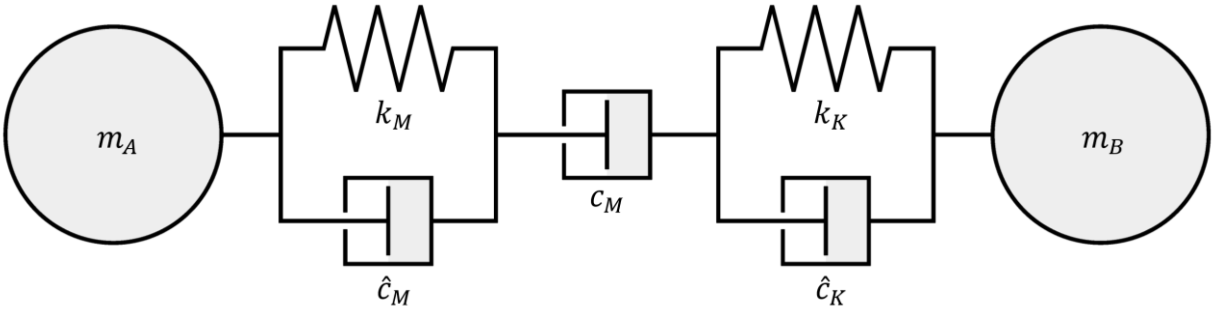


Figure 5.6: Improved Burgers' contact model – expanded version

$$\hat{c}_M = 2\sqrt{k_M m_{eff}} \tag{5.2}$$

$$\hat{c}_K = \max\left(c_K\ ;\ 2\sqrt{k_K m_{eff}}\right) \tag{5.3}$$

$$\begin{gathered} with \\ m_{eff} = \frac{m_A m_B}{m_A + m_B} \end{gathered}$$

$$\begin{aligned} (c_M\hat{c}_K + c_M\hat{c}_M + \hat{c}_K\hat{c}_M)\ddot{F} + (c_M k_K + c_M k_M + \hat{c}_M k_K + \hat{c}_K k_M)\dot{F} + k_K k_M\, F \\ = c_M\hat{c}_K\hat{c}_M\, \dddot{u} + (c_M\hat{c}_K k_M + c_M\hat{c}_M k_K)\ddot{u} + c_M k_K k_M\, \dot{u} \end{aligned} \tag{5.4}$$

It is noted that this model can only be applied if long term deformations are of interest, as the instantaneous elastic deformations are replaced by short term deformations.

To further improve and simplify the model, the Maxwell and Kelvin-Voigt springs could also be merged into a single spring, combined with critical damping. This way, the Maxwell part is only left with its damper. This significantly simplifies the contact model, but robs the user of adding supercritical damping to the Kelvin-Voigt part, and can thus only be used in case the viscoelastic and elastic displacements are considered short-term and only the long-term deformations are of importance. Figure 5.7 presents this improved model. Equations 5.5 and 5.6 present the values for the Kelvin-Voigt components. Equation 5.7 gives the equation of motion for this contact model, which can simply be obtained by setting $k_M \rightarrow \infty$ in Equation 3.8.

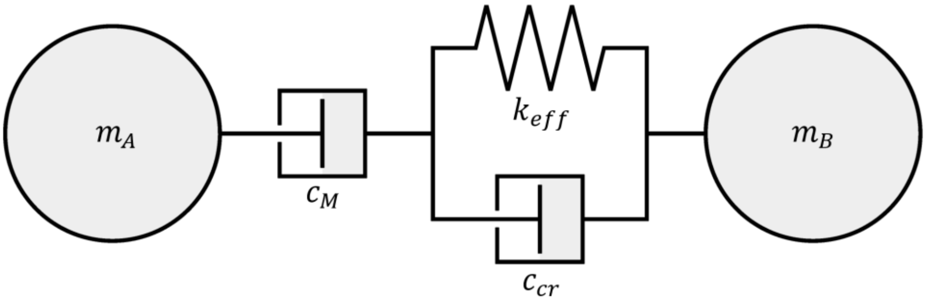


Figure 5.7: Improved Burgers' contact model – reduced version

$$k_{eff} = \frac{k_K k_M}{k_K + k_M} \tag{5.5}$$

$$c_{cr} = 2\sqrt{k_{eff} m_{eff}} \tag{5.6}$$

$$\frac{c_{cr} + c_M}{k_{eff}} \dot{F} + F = \frac{c_{cr} c_M}{k_{eff}} \ddot{u} + c_M \dot{u} \tag{5.7}$$

So, the first proposed contact model still contains elastic, viscous, as well as viscoelastic deformation. The second model only contains two: the short term and long term deformation, where the short term deformation contains both the elastic and viscoelastic part, and the long term deformation contains the viscous part. A downside to these new contact models is the additional energy dissipation for the elastic part. Therefore, in case a new material is being analysed, the values for the springs and dampers could be obtained by means of fitting an analytical solution on a test experiment using the least squares method.

In order to analyse the effects of the improved models, the same two experiments have been done, where again the masses of the elements differ. First, Figure 5.8 presents the deformation with a relatively large mass. Second, Figure 5.9 shows the deformation for a relatively small mass. All three numerical models, as well as the analytical solution are presented. Since the oscillations only occur in the beginning of the verification experiment, only this part is considered here.

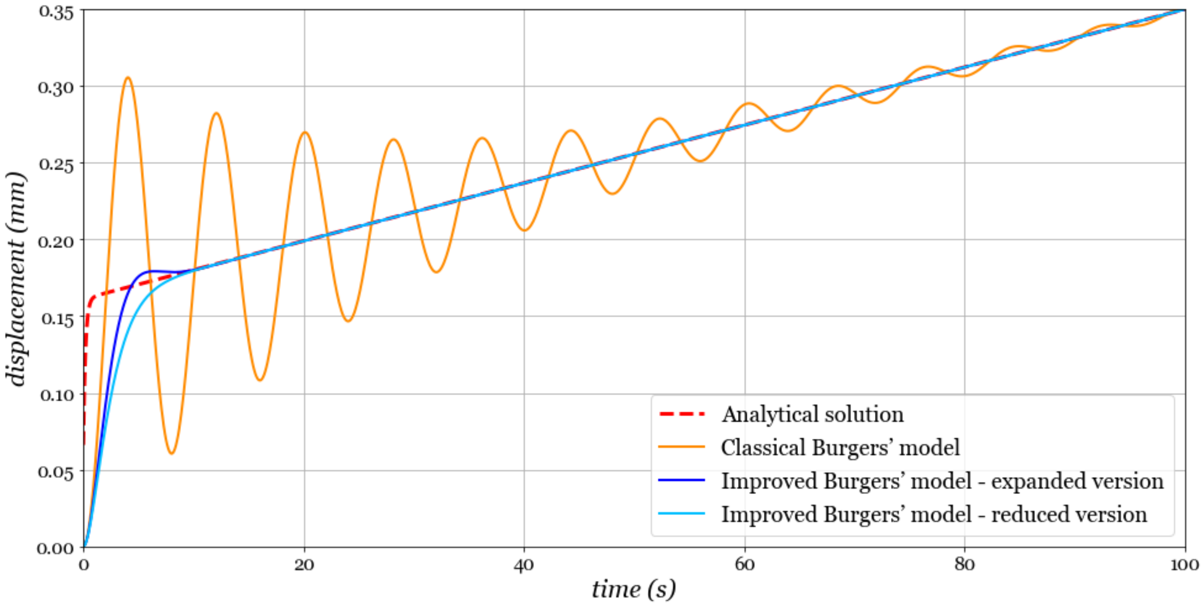


Figure 5.8: Behaviour of the improved Burgers' contact models with a relatively large mass

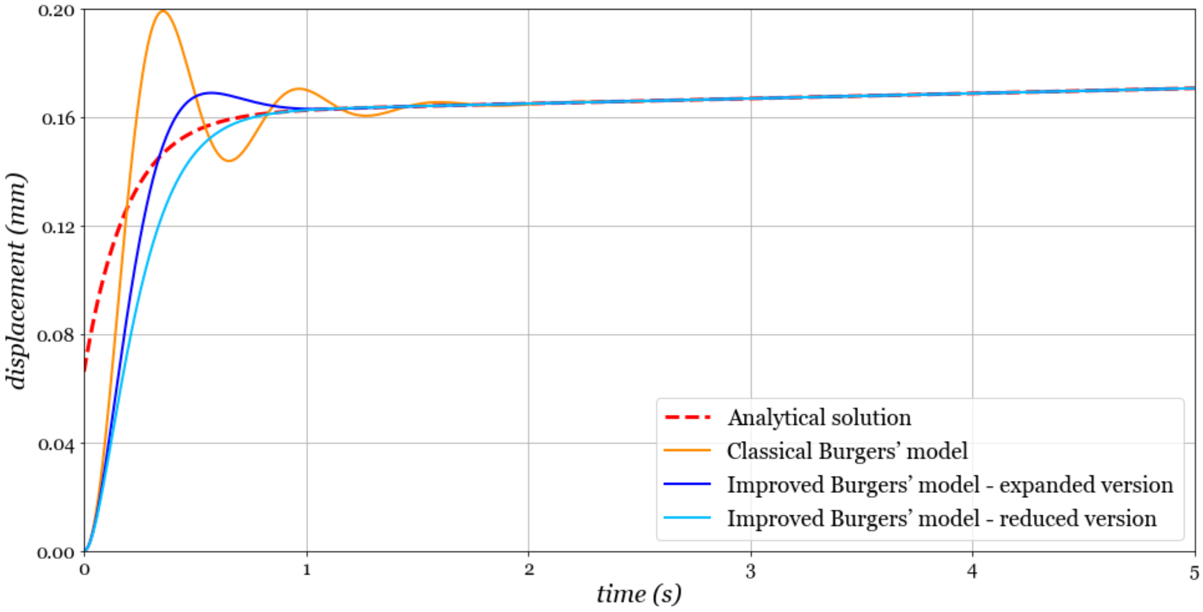


Figure 5.9: Behaviour of the improved Burgers' contact models with a relatively small mass

Although all three numerical solutions are correct on the long term under constant pressure, it becomes clear that both the improved models approximate the analytical solution better on the short term. Therefore, it is recommended to use one of these. In case the viscoelastic deformation is considered short term and only long term behaviour is of importance, the simplified method is to be preferred, as this one is the easiest to implement in a numerical model.

# 6 Results and applications

## 6.1 Compaction process

It proved impractical to implement Burgers' contact model in PyBullet. Still, it is useful to know if the software itself is capable of running large scale simulations. To illustrate that Bullet Physics is a viable physics engine for performing digital superpave gyratory compaction, this chapter describes various simulations, where the mortar layer was modelled with a damper, such that it reasonably resembles the mortar dynamics. This way, the non-penetration condition of the aggregates and the viscous deformation of the mortar layer can be modelled. The elastic and viscoelastic deformation are not present. Thus, the results could be interpreted as an upper bound for the air void content of the specimen under a constant pressure. Figure 6.1 gives a visual representation of the digital superpave gyratory compaction process. A video is also presented.

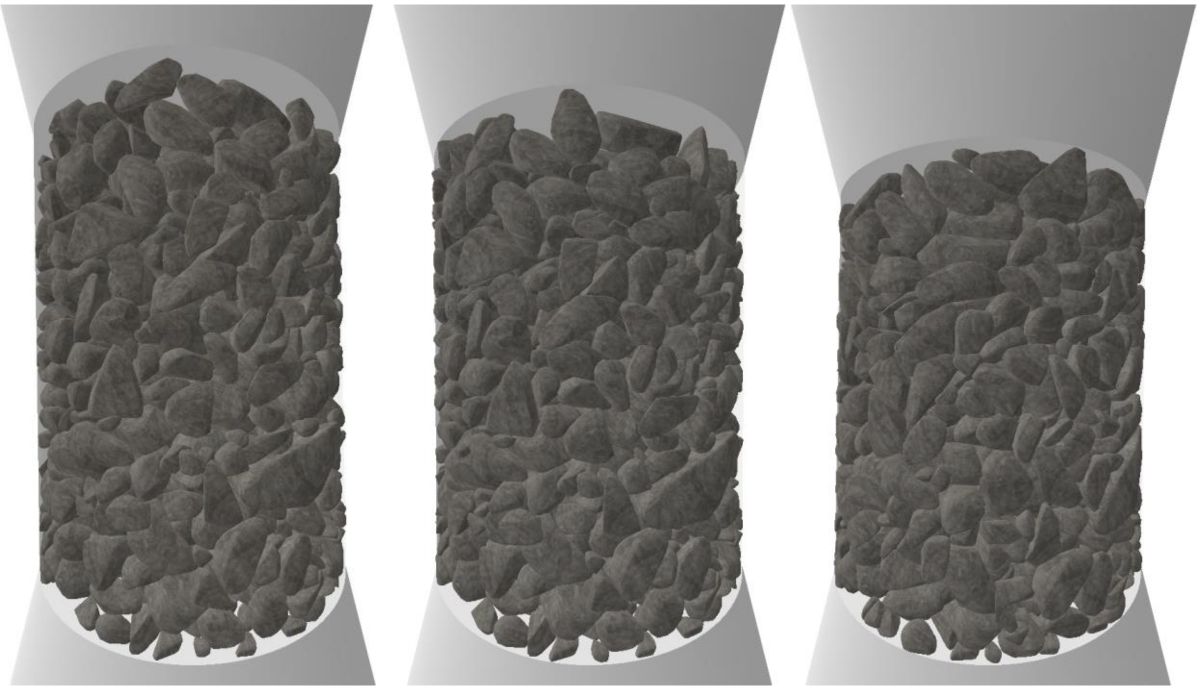


Figure 6.1a: Initial positions

Figure 6.1b: Positions after 7 gyrations

Figure 6.1c: Positions after 100 gyrations

Figure 6.1: Superpave gyratory compaction simulation in Bullet Physics
(a video of this simulation is also available [19])

First of all, it is important to research the consistency of the simulations. Therefore, three tests have been performed with the same target degradation, but different initial positions and randomly selected shapes for the aggregates. Figure 6.2 presents the mean air void content over the number of gyrations for three different simulations. Figure 6.3 shows the coefficient of variation over time, and the 1.5% threshold which may not be exceeded after 20 gyrations [2].

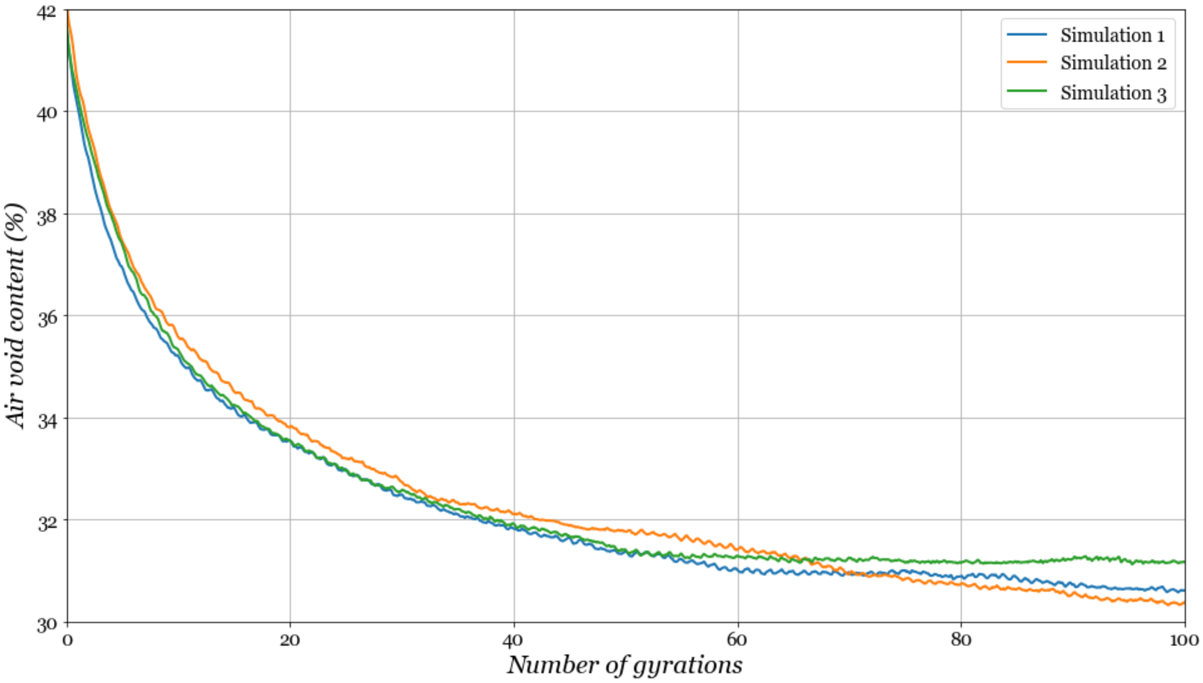


Figure 6.2: Consistency test of three simulations

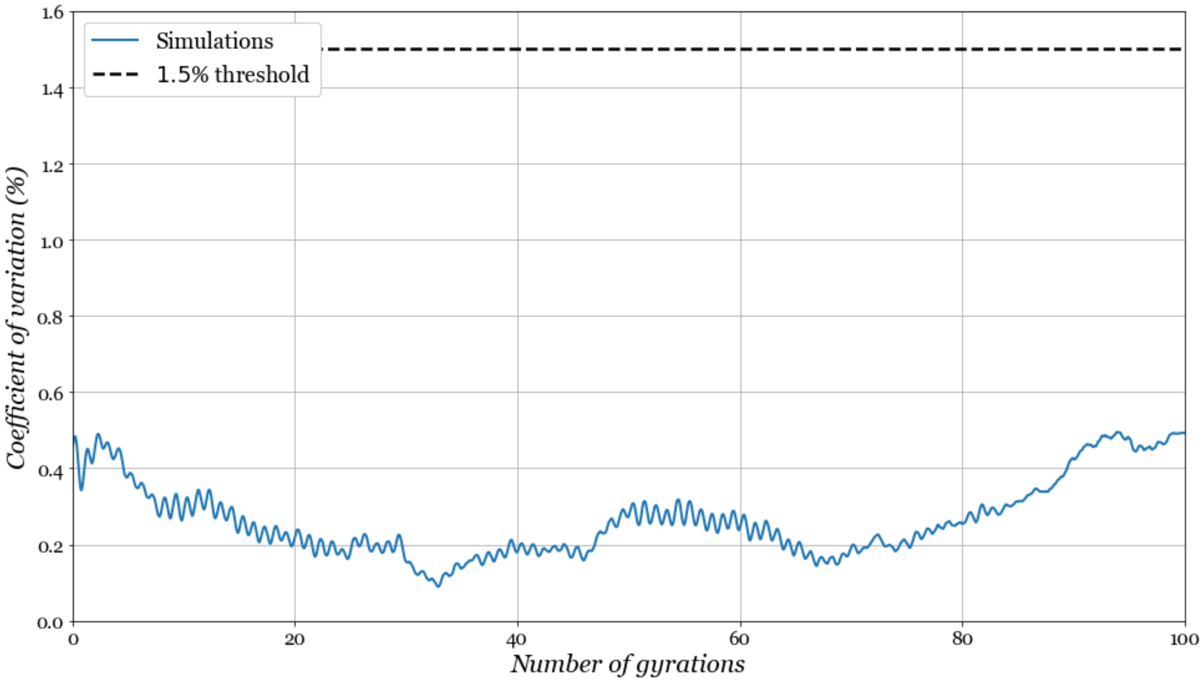


Figure 6.3: Coefficient of variation of the three simulations

Figures 6.2 and 6.3 show that the simulations in Bullet Physics are very consistent. The initial positions, and randomly selected shapes of the aggregates do not significantly influence the result and are well within the norms. The coefficient of variation is well below the threshold, even before 20 gyrations.

## 6.2 Relevance of inertia

A useful advantage of a simulation versus an experiment is that specific properties can be changed, which cannot easily be altered during a laboratory test, while all other conditions can remain identical. This way, the influence of these specific properties can be investigated. For example, inertia is a variable that could potentially be influenced during an experiment by varying materials, but this does not nearly give the range or precision of parameterized simulations, as is demonstrated in this paragraph.

Strogatz [33] has stated that in highly viscous environments, inertia could in some cases be neglected. In order to validate whether that is true in the case of asphalt compaction, Figure 6.4 shows the results of three different simulations: one where the mass was lowered, and one where the mass was increased. These two are compared to an earlier obtained compaction result. Figure 6.5 presents the coefficient of variation for these samples. In order to ensure that all other parameters are the same, gravity was turned off for this simulation.

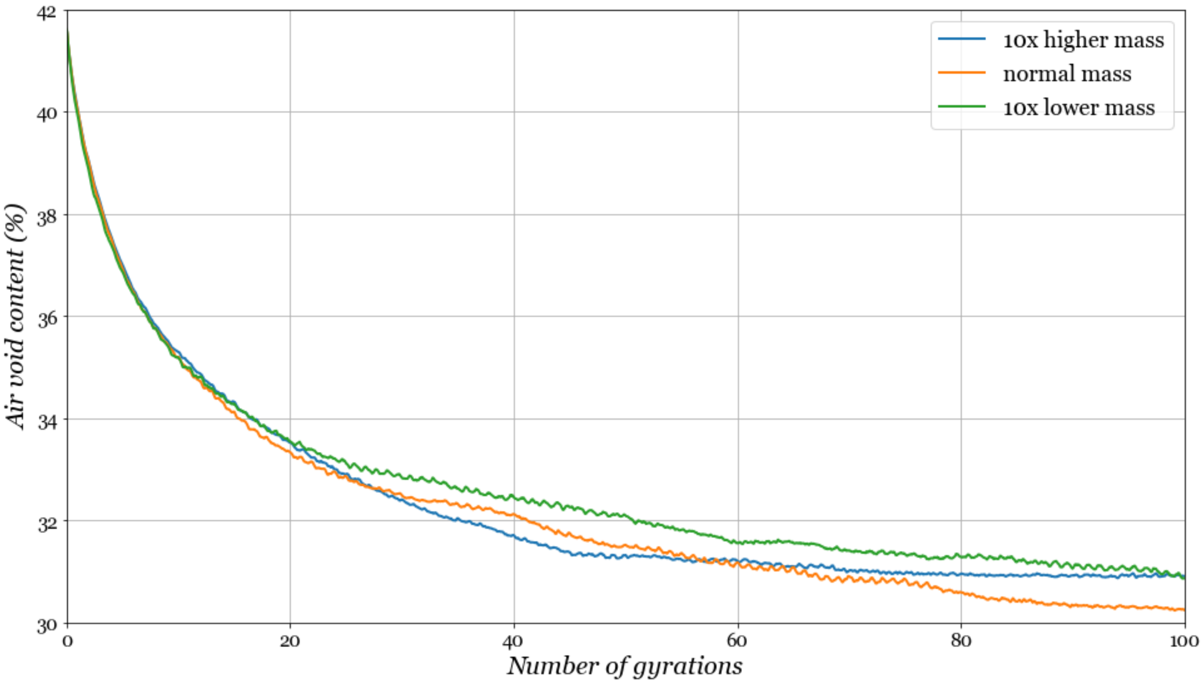


Figure 6.4: Void graph with different inertial properties

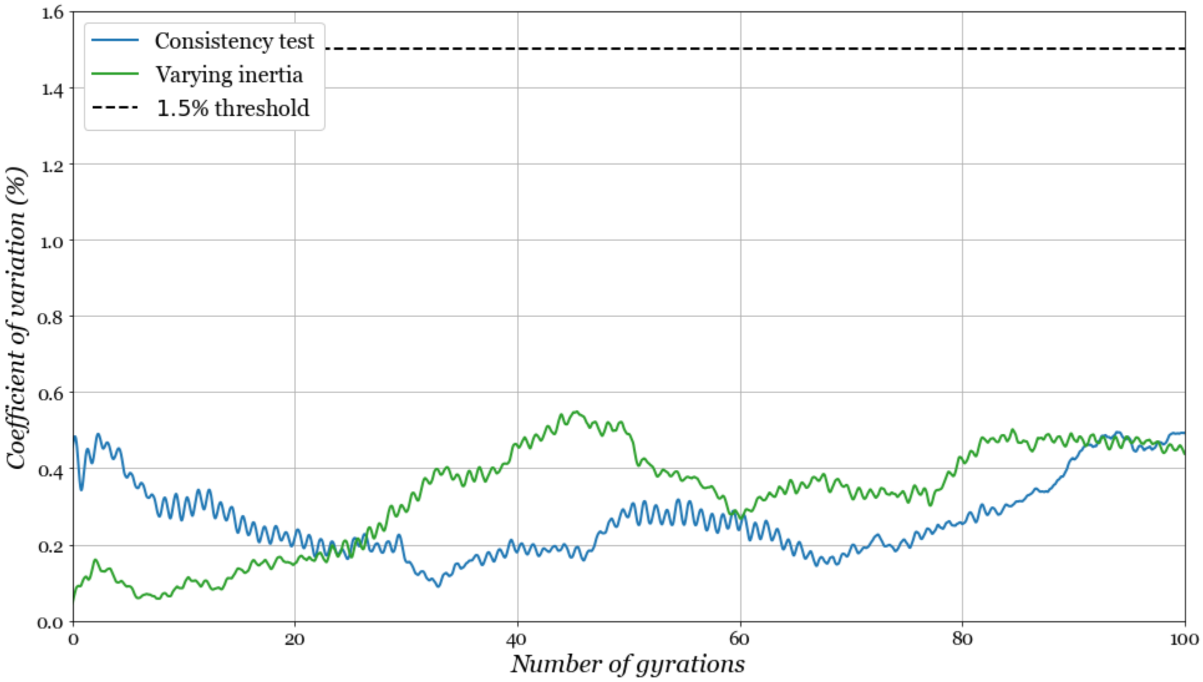


Figure 6.5: Comparison of the coefficient of variation

Figures 6.4 and 6.5 show that the compaction results with varying masses are indistinguishable compared to the noise caused by varying initial conditions. The coefficient of variation is of the same order of magnitude and does not exceed the 1.5% threshold. Therefore, it can be concluded that inertia can indeed be neglected in the case of hot asphalt compaction. Only the viscosity of the binder, and stiffness of the aggregates are of real importance.

## 6.3 Revolving aggregates

Another advantage of simulations over experiments is the possibility to better analyse and understand the behaviour of the aggregates within the mould. Not only can the sample be observed from all possible angles, and can elements be tracked, but the simulation can also be sped up or slowed down, resulting in observations that cannot easily be seen in real experiments.

During the compaction simulations, it was observed that the aggregates revolve around the vertical axis within the mould, in the opposite direction of the rotation of the mould. These revolutionary velocities of the aggregates increases with the height. The phenomenon of aggregate revolution inside the gyrating mould has been observed before [16]. Figure 6.6 gives a top view of the mould, and shows the trajectories of the aggregates during the gyratory compaction. Figure 6.7 presents the revolution of the aggregates over the height of the mould. Negative values correspond with clockwise rotation, whereas the mould rotates counter-clockwise. A video is also presented.

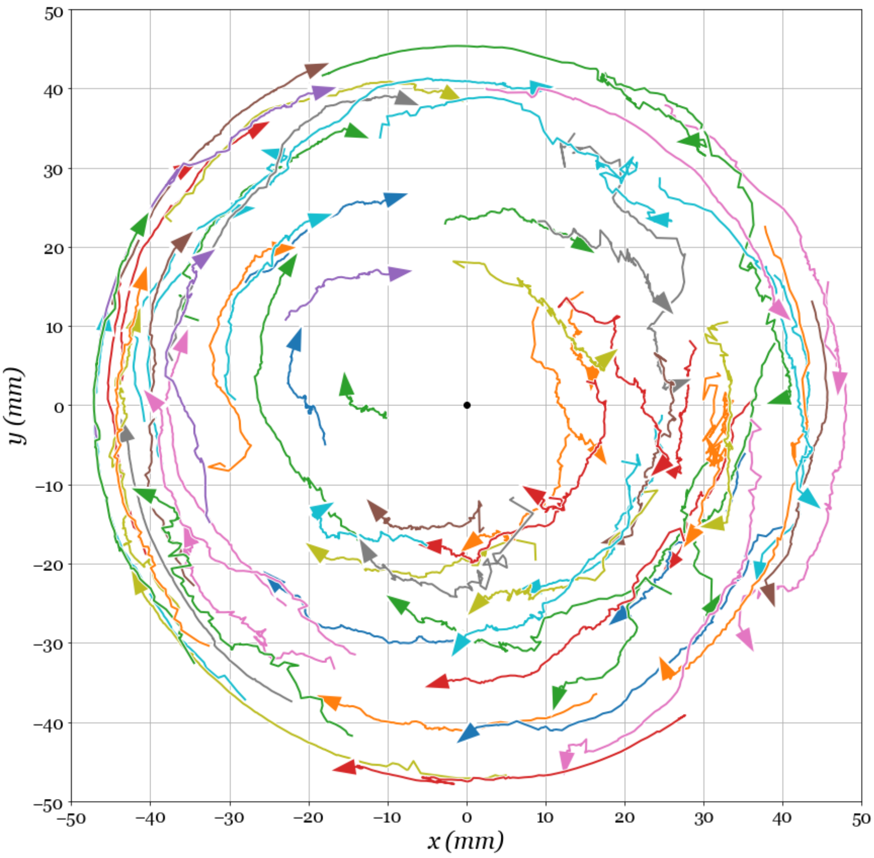


Figure 6.6: Aggregate revolution after 100 gyrations of the top 75 elements

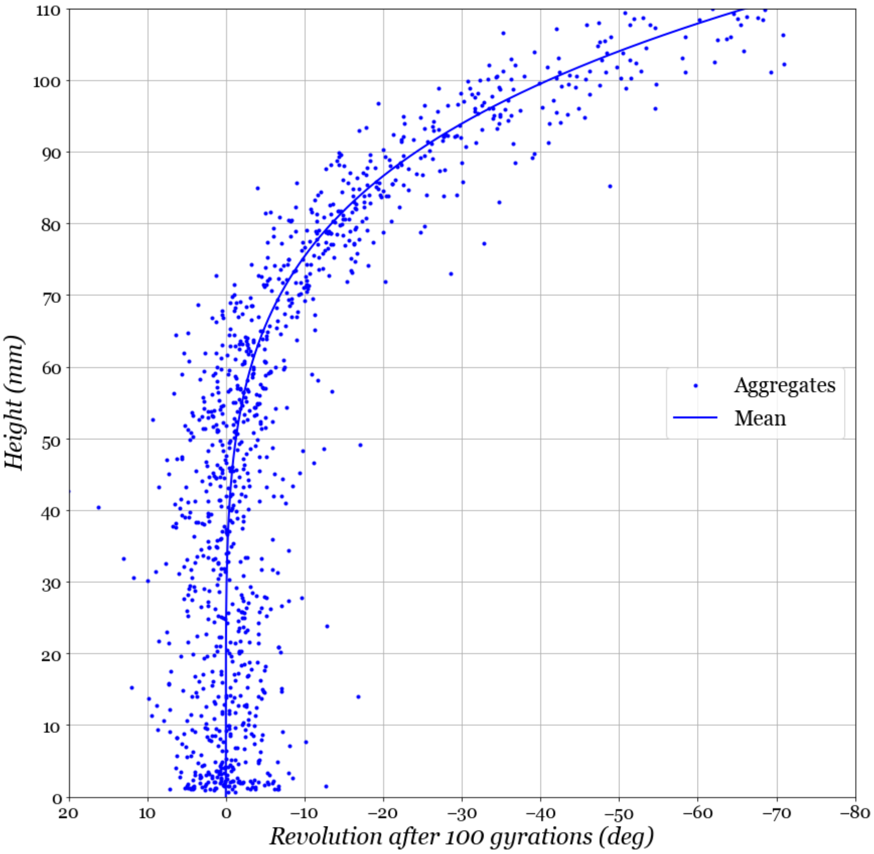


Figure 6.7: Aggregate revolution over the height
(a video of this phenomenon is also available [19])

Figure 6.7 shows a couple of remarkable things: the aggregates barely revolve at the bottom half of the mould. At the top half, the aggregate's velocity depends on the height. The higher an aggregate is located, the faster it moves. Also, at the bottom of the graph a relatively dense agglomeration of aggregates can be found. At the top, the opposite can be observed. This indicates that the smaller particles have the tendency to be in lower locations. Due to gravity, smaller elements tend to fill up the voids.

## 6.4 Influence of friction

Another property that can be well investigated by a simulation, is the influence of the friction parameters. During a simulation, any value can be chosen, or friction can be completely turned off. Completely identical initial positions were selected to further rule out any noise caused by this. Figure 6.8 gives the mean air void content over the number of gyrations for different frictional properties. Friction has been set to zero for both the aggregates and the mould, to better understand the influence of both of these on the results.

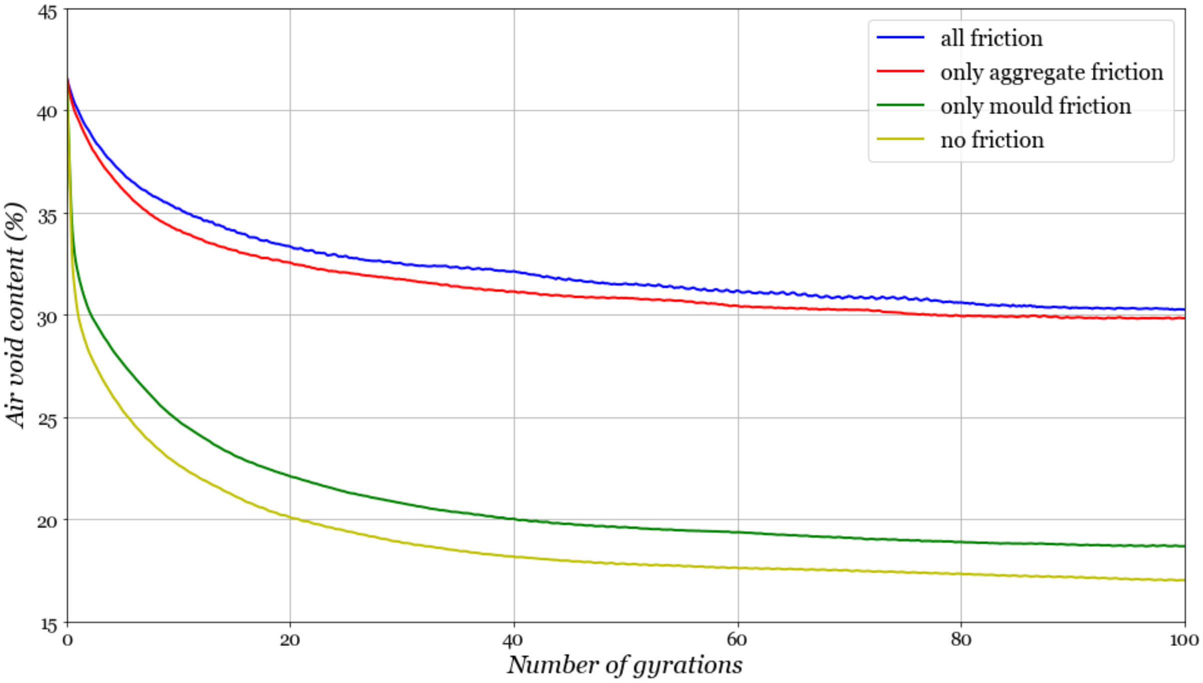


Figure 6.8: Mean air void graph with different friction parameters

Figure 6.8 shows that friction is of great importance when it comes to asphalt compaction. Larger values for the friction lead to a slowing down of the compaction process. It also shows that the friction between the aggregates is most important. The friction between the aggregates and the mould seems to have no significant role in the density curve.

Also the influence of friction on the revolution of the aggregates can be investigated. Figure 6.9 presents the revolution of the aggregates over the height of the mould, and compares the influence of the different friction parameters.

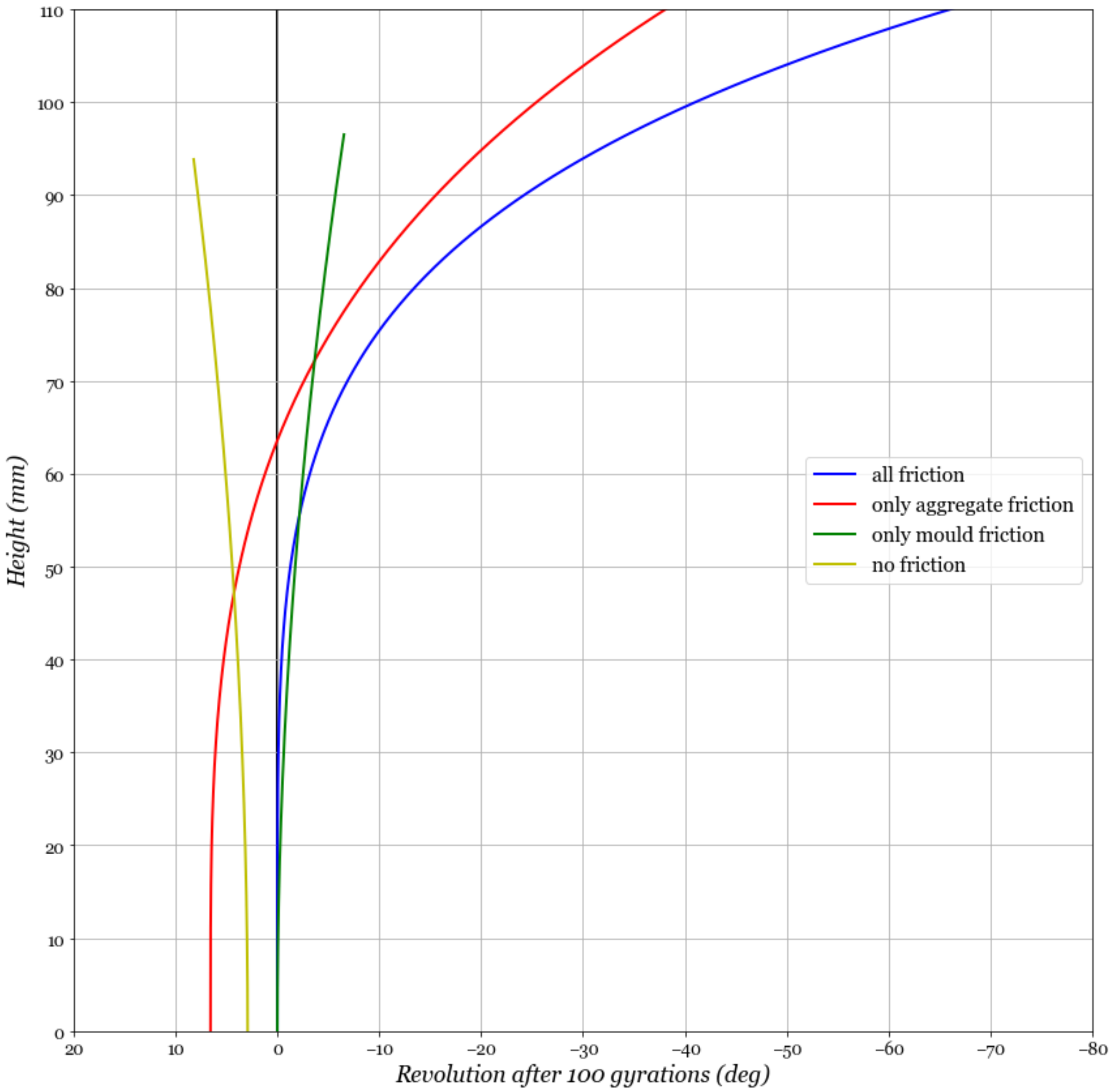


Figure 6.9: Aggregate revolution over the height with different friction parameters

Figure 6.9 shows that the revolutionary motion of the aggregates is not only caused by the gyration of the mould, but also by the lateral contact model. Especially the friction between the aggregates is of importance. In case the friction of the mould is set to zero, the aggregates rotate in the same direction as the mould at the bottom half.

## 6.5 Contact Area

In earlier research [16] it was estimated that the average contact area of two colliding elements equals half the surface area of the smallest of the two. However, this does not accurately reflect the true contact area. To illustrate this, Figure 6.10 shows the calculated contact area using ray tracing (see Section 4.4) versus the element size for a typical asphalt sample (see Table 4.2). For every collision, two dots are presented: one for each element size.

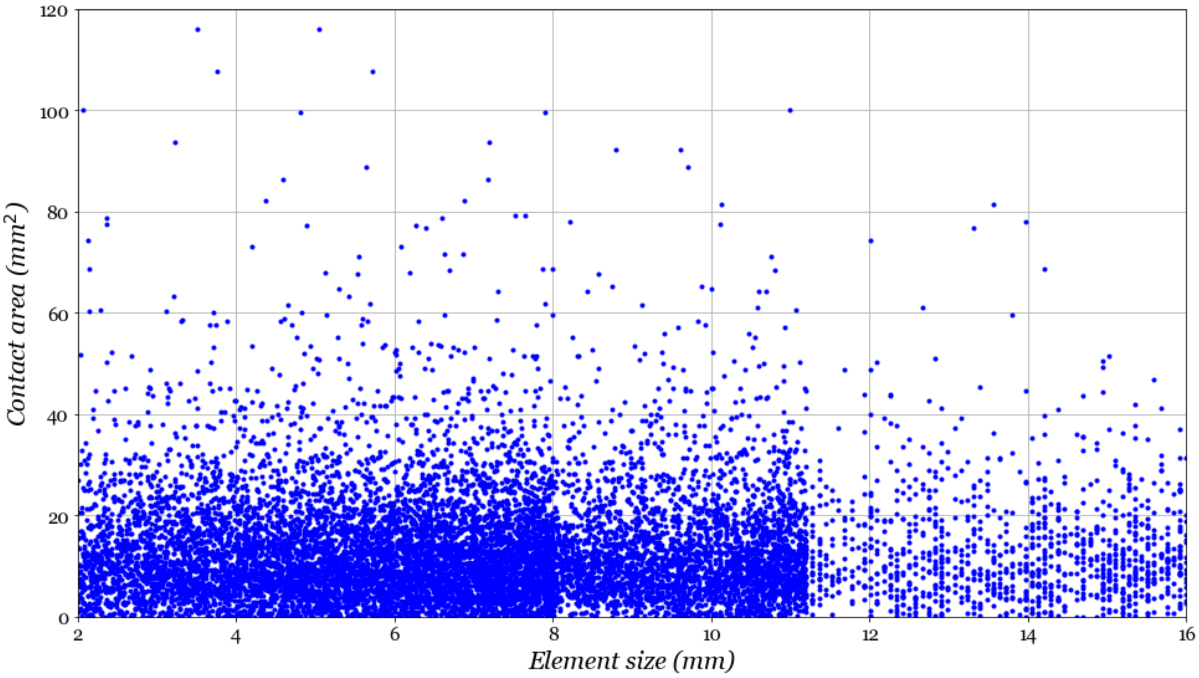


Figure 6.10: Contact area vs. element size for a compacted specimen

Figure 6.10 shows that the average contact area remains somewhat constant over the element size. Also, a large variance can be found. Due to the large number of smaller elements in the degradation, it is hypothesised that most of the contacts are with these smaller elements which could explain the constant average value. Figure 6.11 shows the same graph, but with Komaragiri's [16] estimation added in red. This shows that the contact area is estimated correctly for small particles, but is hugely overestimated for larger elements.

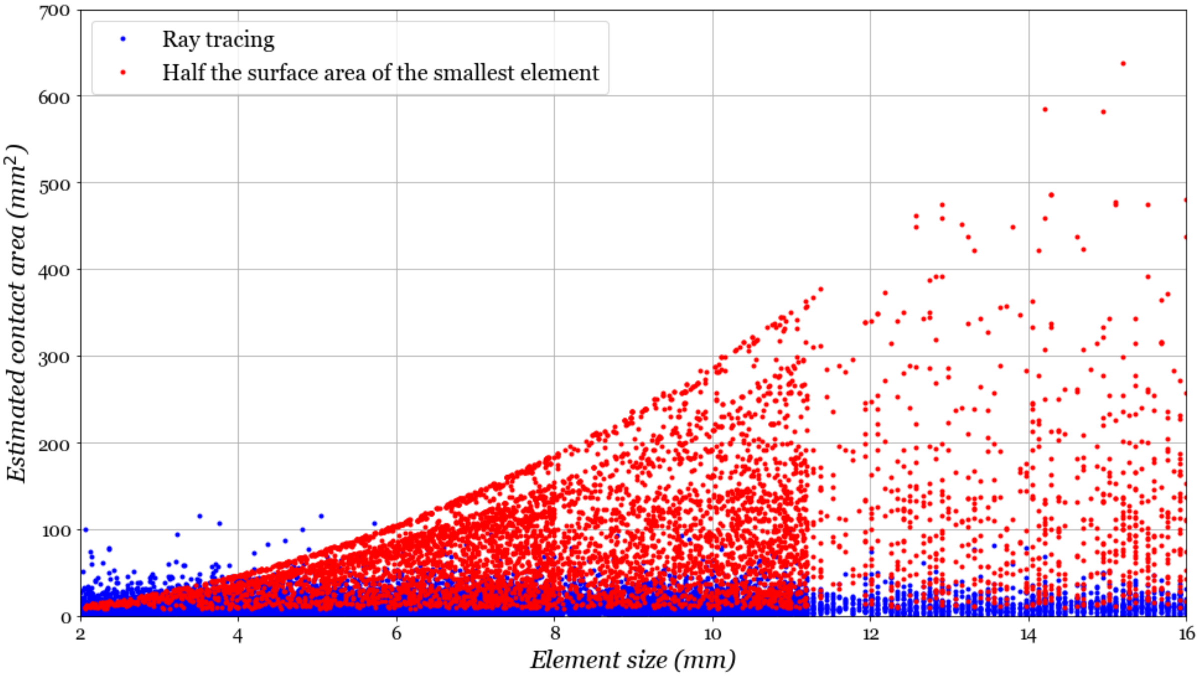


Figure 6.11: Validation of the contact area assumption

In conclusion, if the contact area is to be estimated, it is recommended to determine this once for a compacted specimen using ray tracing. This average contact area will most likely depend on the degradation, aggregate shape and thickness of the mortar layer. In the case of the sample's properties as described in this report (see Paragraphs 4.1-3), the average contact area could be estimated as 15 $mm^2$.

## 6.6 Computation time and memory usage

Since research time and resources are limited, computation time is of importance in order to assess the utility of a simulation. Therefore, additionally to the previous experiments, an additional simulation has been performed by adding four times the number of elements to the simulation. This way, results regarding computation time and memory usage can be compared and extrapolated for any number of particles.

All simulations were performed on a desktop pc with an i5 6600K processor, clocked at 4.2 GHz. All relevant specifications of the simulation can be found in Table 6.1. The processor is the main factor which determines the computational speed [44]. The amount of memory should be sufficient to perform the simulation. Since the mortar layers are implemented as separate elements, the number of dynamic elements is twice the number of aggregates. It is noted that PyBullet does not allow for multithreading, and the number of cores of a CPU is only relevant for running parallel simulations. Python does allow for running multiple simulations at the same time by initializing multiple instances, which can simply be done by opening multiple Notebooks. This can be useful for parametrical study or stochastic analysis. Multithreaded simulations are however implemented in the C++ module of Bullet Physics, allowing for marginal performance improvements [44].

Table 6.1: Simulation properties

| Variable | Simulation 1 | Simulation 2 |
|---|---|---|
| simulation time (seconds) | 200 | |
| step size (seconds) | 1/240 | |
| iterations per step | 50 | |
| number of dynamic elements | $2 \times 1500$ | $2 \times 6000$ |
| memory usage (GB) | 1.2 | 3.6 |
| computation time (hours) | 14.14 | 63.06 |

Table 6.1 shows that with four times as many elements, both the computation time and memory usage increase. For further research, it is beneficial to know what number of elements leads to a reasonable computation time and memory usage. Therefore, the influence of the number of elements on computation time and memory usage is investigated. It is assumed that the computation time grows exponentially with the number of elements. The exponent may differ per simulation. Equation 6.1 shows how this time order can be determined. This number gives information about the efficiency of the physics engine. So, if there are twice as many elements, the computation time takes $2^O$ as long. $f_t$ is the increase factor in calculation time, and $f_e$ is the increase in number of elements.

$$O = \log_{f_e}(f_t) = 1.078 \tag{6.1}$$

$$\begin{aligned}
&with \\
&f_t = \frac{t_{c_2}}{t_{c_1}} = \frac{63.06}{14.14} = 4.46 \\
&f_e = \frac{n_{el_2}}{n_{el_1}} = \frac{6000}{1500} = 4
\end{aligned}$$

The time order is very close to 1, meaning that it manages collisions very efficiently. This opens the door for simulations with an extraordinary large number of elements. Equation 6.2 shows how to determine the estimated computation time for a simulation of 100 gyrations with 100 000 coated aggregates.

$$t_{c_3} = \left(\frac{100\ 000}{n_{el_1}}\right)^{O} \cdot t_{c_1} = 55\ days \tag{6.2}$$

This shows that a very large number of elements can still be modelled within a reasonable amount of time, assuming that Bullet Physics can manage the number of elements. It is noted that these calculations are simulation specific, and could vary based on the degradation of the specimen, complexity of the contact model and aggregate shapes. This is therefore a rough estimate. If one takes into consideration that this is an older CPU model from 2015, and that the C++ version of Bullet Physics is faster in itself, which also allows for multithreading, the computation time might decrease drastically.

Further, it can be observed that Bullet Physics requires a very low amount of memory. Only 1.2 GB for 1500 elements, and 3.6 GB for 6000 elements. Using linear interpolation, this would result in 52 GB for a simulation with 100 000 elements. In case collision shapes are re-used, this number might be significantly reduced. However, during the simulations it was observed that Bullet suffers from memory leakage. Meaning that over time, more and more memory is being used, which might become problematic in case of a long term simulation.

# 7 Conclusions and future research recommendations

## 7.1 Answers to the research questions

The main research goal of this report is to assess the utility of Bullet Physics for analysing hot asphalt compaction, and more specifically, for analysing the superpave gyratory compaction process. In order to describe the contact behaviour of the bituminous mixture, Burgers' contact model has been implemented. Furthermore, a geometrical model of the superpave gyratory compaction process has been created, and the advantages of a simulation are mapped out. This allows for the answering of the research questions as follows.

1. *Can a complex contact model, such as the Burgers' contact model, be implemented directly in PyBullet?*

PyBullet is a free-to-use open source physics engine with great potential. It was, however, not initially designed for simulations with a complex contact model. It proved cumbersome to implement a custom contact model in PyBullet. Two attempts have been made. In the first model, Bullet Physics was used for collision detection only, which was numerically integrated by a custom Python code, allowing for the implementation of any contact model. This accurately described the motion of the aggregates, but resulted in a computation time that do not allow for large scale simulations. A second model was therefore created, where the Burgers' contact forces were implemented as external forces, and the numerical integration was done by Bullet itself. Also this model proves impractical, as the simulation becomes unstable, resulting in rapid oscillations. A standalone tool has shown that the model itself is correct, however. In conclusion, it proves impractical to implement a custom contact model in PyBullet. The only way to achieve this, is by altering the source code of Bullet Physics itself.

2. *Is Bullet Physics able to simulate the gyratory compaction process, and hot asphalt compaction in general?*

Despite this limitation, Bullet Physics has shown great capability for modelling the superpave gyratory compaction process. Aggregate shapes could easily be implemented, sieve sizes could be determined using a digital sieve, and the mortar layer could be modelled by slightly increasing the size of the aggregate meshes, and rigidly connecting it to the aggregate elements. The latter allows for a different contact model for the mortar layers and a non-penetration condition for the aggregates. By means of rescaling the elements, a precise degradation could be chosen. The shape of the mould, and end plates accurately resemble the actual situation. Furthermore, it proves possible to apply a large compressive force on the top end plate, while maintaining its correct position and orientation. This demonstrates the ability of Bullet Physics to model the gyratory compaction process, and provides a sold basis for field compaction as well.

3. *Does Bullet Physics hold potential for future research when analysing hot asphalt compaction?*

It certainly does, and advantages of simulations are many. The simulations prove to be far more consistent than the requirements. It is demonstrated that inertia of the aggregates can be neglected. It is also observed that the aggregates revolve inside the mould during compaction, in opposite direction as the rotation of the mould. It can be demonstrated that the friction of the mould is of marginal influence, and the behaviour of the aggregates inside the mould strongly depends on friction between the aggregates themselves. Also, the contact area remains roughly constant over the element size. Most of these conclusions could not have been made using an actual gyratory compactor. An analysis of the computation time has shown that large scale simulations are possible within a reasonable amount of time. Apart from that, if a reliable model could be developed, this would be very cost-effective. These findings demonstrate the utility and advantages of Bullet Physics.

## 7.2 Recommendations for future work

Following these conclusions, in combination with the results and other experiences with Bullet Physics, this leads to recommendations and opportunities for future research.

### Implementing the custom contact model

What cannot be done in PyBullet, is possible in Bullet Physics. Implementing Burgers' contact model using Python proved possible, but very cumbersome. The only correct way of handling a custom contact model is by implementing it directly into the C++ version of Bullet Physics. This way the contact forces are not only determined at the beginning of each time step, but can be considered during the iterative process as well. This allows for more stability and faster computation.

### Mortar layer

The mortar layer was modelled as a single element, slightly bigger than the aggregate, thus restricting the mortar from moving relatively to its aggregate. Whereas in reality, the mortar can deform, as well as relocate to any other position within the mould. This has four mayor consequences: first, deformation of the mortar layer might fill up air voids, resulting in higher contact forces at specific locations. Second, in the case of crushing, the digital layer simply overlapped with the other object, resulting in a volume loss, which might potentially lead to a negative air void content. Third, the deformation of the mortar layer leads to a higher contact area. Fourth, due to the possible displacement, an additional degree of freedom is added, and the contact forces might in reality be significantly lower compared to the rod model. In fact, due to the low thickness, it is doubtful whether the mortar layer plays a role at all when calculating the normal forces between aggregates. The instantaneous crushing of a mortar layer between two compressed aggregates is therefore not unlikely. Instead, the mortar layer might act like a glue or viscous liquid, resulting in viscous lateral damping and the elimination of friction. The properties of which depends on the temperature, allowing the aggregates to position themselves. The exact behaviour of the mortar layer during the compaction of a hot asphalt mixture is to be further investigated.

Furthermore, after initial contact and later separation between two elements, in reality the bitumen elongates holding the two elements together. During the simulation this was not considered, and after separation of the mortar layers, there was no force present anymore. To enhance the model, pulling forces could be added which are still active even after two elements have separated. A practical way to achieve this is by implementing an additional thicker mortar layer, which is only activated in case of initial contact, thus allowing for contact forces over longer distances.

### Aggregate shapes

In this model, ten different aggregate shapes have been 3D scanned. These shapes have been used for all sieve sizes. As the shapes might differ per sieve size, it is advised to scan individual aggregates per sieve size.

Furthermore, during the verification of the numerical model, a larger mass was chosen such that acceptable computation times were achieved. In reality, the mass of the elements is in the range of milligrams, instead of kilograms. It has been clearly demonstrated that inertia can be neglected. This might significantly improve the performance and stability of hot asphalt simulations. Further study might prove the validity and utility of this concept.

### Placement process

The results of the gyratory compaction process strongly depends on the packing arrangements. During this research, a simple tool has been created to facilitate decent initial positioning of the aggregates. However, during simulation it shows that smaller particles tend to place themselves in lower positions, in between the larger elements. This has the effect of lowering the air void content in this section of the specimen. The air void content in the upper half increases this way, which did not match earlier test results either [8]. Therefore, developing a tool to model aggregate packing could be extremely helpful in facilitating further studies.

### Contact model

Contact forces contain damping components that reflect the kinetic energy dissipated through microscopic actions such as the wave scattering and internal friction, and bring the simulation, if necessary, to a quasi-static state. In addition to viscous local damping acting at each contact, damping vectors should be proposed to consider the energy dissipation.

This research focussed on laboratory compaction, where the environment is carefully controlled. For further research it is recommended to simulate and validate field compaction as well. This leads to increasing complexity in the contact model, since temperature and viscosity of the binder might vary greatly over the compaction time and volume.

A useful finding is the semi-analytical solution of Burgers' contact model. Which solves the contact forces analytically over a small time step, instead of numerically. This might lead to enhanced precision, and is therefore recommended for further research.

**Improvements for Bullet Physics**

Bullet Physics proves to be a very efficient and reliable physics engine. The software was not initially designed for scientific research in the field of Pavement Engineering. Therefore, a few adaptations to the source code might prove to be useful.

To enhance the accuracy and realism of the Bullet Physics engine, it is recommended to give the user the possibility of setting the dynamical properties for specific pairs of objects, instead of setting the properties for only the individual elements. This includes: contact stiffness and damping, friction coefficient, lateral viscous damping and restitution. Another valuable addition to Bullet Physics is to let contact forces depend on the contact area. Whereas the contact forces can currently only be modelled with a spring and damper, having these values multiplied with the contact area would yield more realistic contact forces. Furthermore, to specialize in the field of engineering, Bullet Physics could give the user the possibility to determine the contact forces between objects, based on a user-defined function. Making these changes would require the cooperation of the Bullet Physics developers, or advanced C++ programmers.

It is also strongly recommended that researchers and Bullet Physics developers establish a better line of connection, since both suffer from a lack of knowledge. Researchers on the field of mechanics and material science lack full knowledge and potential of the Bullet software, resulting in sometimes clumsy models or lack of resources available already implemented in the Bullet software of which they are simply unaware. Also, useful modifications to the software are currently not always communicated. The same is true for Bullet developers. Some approximations and errors exist within the Bullet software that an expert in the field of physics could easily point out. A cooperation between the two would therefore be of mutual benefit.

Furthermore, in Bullet Physics, interacting concave elements do not exist, which greatly simplifies the aggregate's shape and limits the researcher's ability to research aggregate interlocking. This is mainly due to the emphasis on computational speed. This could potentially be solved by rigidly connecting multiple convex parts to create concavity, but this workaround method can be cumbersome. A tool in Bullet physics could be developed to give the user the possibility of creating concave elements.

Also, the mortar layer has been implemented by slightly increasing the aggregate's shape. This does not lead to a uniform thickness in case of a non-spherical elements. Bullet Physics might add the possibility of creating a layer around an object with a constant thickness.

**Improvement for the NEN norms**

NEN-EN 12697-31 [2] describes the superpave gyratory compaction process in detail, which is a standardized tool for determining the void curve. One of the instructions is to precompress the sample. However, this process leaves room for interpretation, which is disadvantageous for performing a standardized digital simulation. Therefore it is recommended that this subprocess is specifically described.

**Computation time**

Bullet Physics provides the possibility of running simulations with many elements while maintaining a reasonable computation time. For example, 100 000 coated aggregates could be modelled for 50 gyrations within 4 weeks. This gives researchers the possibility of better studying hot asphalt compaction, as this allows for the smallest of aggregates to be modelled. Alternatively, the bitumen could be modelled as separate elements, thus allowing for better studying the displacement and behaviour of this part of the mixture.

**Beauty of simplicity**

Although it is more accurate to model the behaviour of the mortar layer with Burgers' contact model and viscous lateral damping, this report has shown that it is also possible to model superpave gyratory compaction using a damper and friction, which can easily be implemented. Herein lies the beauty of simplicity, which does not require further adaptations of the Bullet Physics software. It has been demonstrated that the compaction strongly depends on the friction, among other things. Perhaps an accurate description of the compaction process could be achieved by carefully tweaking the currently existing parameters. This would significantly reduce the amount of work required to perform simulations.

# Bibliography

[1] H. Ter Huerne, “Compaction of asphalt road pavements: Using finite elements and critical state theory,” University of Twente, 2004.

[2] Normcommissie 353 058 "Wegenbouwmaterialen", “NEN-EN 12697-31: Bituminous mixtures - Test methods for hot mix asphalt - Part 31: Specimen preparation by gyratory compactor,” April 2007. [Online]. Available: connect.nen.nl.

[3] K. Mollenhauer and M. P. Wistuba, “Influence of asphalt compaction procedure on 3D deformation properties,” *International Journal of Pavement Engineering,* vol. 17, no. 1, pp. 5-12, 2016.

[4] G. Seleridis, “Evaluation Of The Current Test Methods Of Water Sensitivity And Permanent Deformation,” TU Delft, 2017.

[5] J. Chen, B. Huang, X. Shu and C. Hu, “DEM Simulation of Laboratory Compaction of Asphalt Mixtures Using an Open Source Code,” *Journal of Materials in Civil Engineering,* vol. 27, no. 3, 2015.

[6] L. Tashman, E. Masad, B. Peterson and H. Saleh, “Internal structure analysis of asphalt mixes to improve the simulation of superpave gyratory compaction to field conditions,” *Proceedings of the Association of Asphalt Paving Technologists,* vol. 70, pp. 605-645, 2001.

[7] F. Gong, Y. Liu, X. Zhou and Z. You, “Lab assessment and discrete element modeling of asphalt mixture during compaction with elongated and flat coarse aggregates,” *Construction and Building Materials,* vol. 182, pp. 573-579, 2018.

[8] L. Al Khateeb, K. Anupam, C. Kasbergen, S. Erkens and A. Scarpas, “Micromechanical simulation of porous asphalt mixture compaction using discrete element method (DEM),” 2021. [Online]. Available: https://doi.org/10.1016/j.conbuildmat.2021.124305.

[9] L. Wang, B. Zhang, D. Wang and Z. Yue, “Fundamental Mechanics of Asphalt Compaction through FEM and DEM Modeling,” *Geotechnical Special Publication,* no. 176, pp. 45-63, 2007.

[10] S. Koneru, E. Masad and K. Rajagopal, “A thermomechanical framework for modeling the compaction of asphalt mixes,” *Mechanics of Materials,* vol. 40, no. 10, pp. 846-864, 2008.

[11] J. Zheng, “ANSYS research on vibratory compacting process of hot asphalt mixture pavement,” *Engineering mechanics,* vol. 25, no. 10, pp. 200-206, 2008.

[12] E. Masad, A. Scarpas, K. R. Rajagopal, E. Kassem, S. Koneru and C. Kasbergen, “Finite element modelling of field compaction of hot mix asphalt. Part II: Applications,” *International Journal of Pavement Engineering,* vol. 17, no. 1, pp. 24-38, 2016.

[13] E. Izadi and A. Bezuijen, “Simulating direct shear tests with the Bullet physics library: A validation study,” *Plos One,* vol. 13, no. 4, pp. 1-28, 2018.

[14] E. Izadi and A. Bezuijen, “Simulation of granular soil behaviour using the bullet physics library,” in *Geomechanics from Micro to Macro, Vols I and II*, London, Taylor and Francis Group, 2015, pp. 1565-1570.

[15] H. He, J. Zheng, Q. Sun and Z. Li, "Simulation of Realistic Particles with Bullet Physics Engine," *E3S Web Conf.,* vol. 92, p. 14004, 2019.

[16] S. Komaragiri, A. Gigliotti and A. Bhasin, "Feasibility of using a physics engine to virtually compact asphalt mixtures in a gyratory compactor," *Construction and Building Materials,* vol. 308, no. 124977, 2021.

[17] E. Coumans, "Bullet Physics SDK," Bullet Physics development team, 2016-2022. [Online]. Available: github.com/bulletphysics. [Accessed 10 March 2022].

[18] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," 2016-2021. [Online]. Available: http://pybullet.org.

[19] M. Jol, "Supporting media for the master thesis of Mark Jol," 5 May 2022. [Online]. Available: https://www.youtube.com/channel/UCH4-3f95zA7ZDmM4Or6rBmw.

[20] Normcommissie 353058 "Wegenbouwmaterialen", "NEN-EN 12697-35: Bituminous mixtures - Test methods for hot mix asphalt - Part 35: Laboratory mixing," April 2016. [Online]. Available: connect.nen.nl.

[21] Normcommissie 353058 "Wegenbouwmaterialen" , "NEN-EN 13108-1: Bituminous mixtures - Material specifications - Part 1: Asphalt Concrete," June 2016. [Online]. Available: connect.nen.nl.

[22] Trimble, "SketchUp," [Online]. Available: sketchup.com.

[23] The Blender Foundation, "Blender," [Online]. Available: www.blender.org.

[24] Academy of Motion Picture Arts and Sciences, "21 scientific and technical achievements to be honored with academy awards®," 13 January 2015. [Online]. Available: https://www.oscars.org/news/21-scientific-and-technical-achievements-be-honored-academy-awardsr. [Accessed 10 March 2022].

[25] I. Bzhikhatlov and S. Perepelkina, "Research of robot model behaviour depending on model parameters using physic engines bullet physics and ODE," in *International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2017.

[26] T. Erez, Y. Tassa and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[27] M. Busy and M. Caniot, "qiBullet, a Bullet-based simulator for the Pepper and {NAO} robots," *CoRR,* vol. 1909.00779, 2019.

[28] E. Coumans, "Exploring MLCP solvers and Featherstone," in *Game Developers Conference*, San Francisco, California, 2014.

[29] M. Silcowitz, S. Niebe and K. Erleben, "Projected Gauss-Seidel Subspace Minimization Method for Interactive Rigid Body Dynamics - Improving Animation Quality using a Projected Gauss-Seidel Subspace Minimization Method," *GRAPP 2010 - Proceedings of the International Conference on Computer Graphics Theory and Applications,* no. 229, pp. 38-45, 2010.

[30] J. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, 2002.

[31] Code Project, “Kinect Joint Rotation – The Definitive Guide,” 27 January 2019. [Online]. Available: www.codeproject.com/Articles/1275578/Kinect-Joint-Rotation-The-Definitive-Guide-2. [Accessed 7 December 2021].

[32] Wikipedia, “Sphere rotation,” 5 July 2014. [Online]. Available: commons.wikimedia.org/wiki/File:Sphere_rotation_qtl1.svg. [Accessed 6 December 2021].

[33] S. H. Strogatz, in *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, Perseus Books Publishing, 1994, pp. 61-69.

[34] B. N. Biswas, S. Chatterjee, S. P. Mukherjee and S. Pal, “A discussion on Euler method: a review,” *Electronic Journal of Mathematical Analysis and Applications,* vol. 1, no. 2, pp. 294-317, 2013.

[35] W. Huang and M. Jawed, “Newmark-beta method in discrete elastic rods algorithm to avoid energy dissipation,” *Journal of Applied Mechanics,* vol. 86, no. 8, p. 084501, 2019.

[36] J. Butcher, “Implicit Runge-Kutta Processes,” *Mathematics of Computation,* vol. 85, no. 18, pp. 50-64, 1964.

[37] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods,* vol. 17, no. 3, pp. 261-272, 2020.

[38] A. Meurer et al., “SymPy: symbolic computing in Python,” *PeerJ Computer Science,* vol. 3, p. e103, 2017.

[39] J. Trigeassou and N. Maamri, “6th International Multi-Conference on Systems, Signals and Devices,” in *State space modeling of fractional differential equations and the initial condition problem*, 2009.

[40] S. Shahriar, “Density of Aggregate - Bulk and Relative Density,” [Online]. Available: https://civiltoday.com/civil-engineering-materials/aggregate/198-density-of-aggregate. [Accessed 15 September 2021].

[41] A. Jamali, “Unconsolidated Oil Sands: Vertical Single Well SAGD Optimization,” May 2014. [Online]. Available: hdl.handle.net/2346/58600.

[42] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia, “MeshLab: an Open-Source Mesh Processing Tool,” in *Sixth Eurographics Italian Chapter Conference*, pp. 129-136, 2008.

[43] P. Davis and P. Rabinowitz, Methods of Numerical Integration. 2nd Edition, New York City: Dover Publishers, 2007.

[44] OpenBenchMarking.org, “Bullet Physics Engine,” 14 April 2022. [Online]. Available: https://openbenchmarking.org/test/pts/bullet. [Accessed 8 July 2022].

## *Appendix A: DEM exercise in Python*

What is fun about the Discrete Element Method (DEM), is that you can easily program it yourself in Python. In the link below, an example code for a DEM simulation in Python is presented. This could be interpreted as a bunch of ping pong balls in a box. Study it, then try to answer the questions, and see if you can obtain the objectives.

The Python Notebook can be found here. It imports all relevant packages, sets the parameters, and makes a plot of the initial conditions. Also, the equation of motion is defined, and the initial value problem is numerically solved. Finally, an animation is created which can be stored as an mp4 file.

Questions:

1. Why are the contact forces in the equation of motion divided by 2?
2. In what way could the equation of motion be optimized in terms of computation time?
3. Imagine that, after the initial value problem has been solved, the accelerations over time are required. How can these best be obtained?

Objectives:

a. Add contact damping to the 1D DEM model.
b. Add a custom force on an element as a function of time.
c. Create elements with varying masses / sizes / contact stiffnesses (choose one).
d. Create your own 2D DEM model.
   Tip: use atan2 for determining the contact force in both directions.
e. Add gravity to the 2D model.
f. Add varying masses / sizes / contact stiffnesses to the 2D model (choose one).

## Appendix B: Gyratory compaction in PyBullet

This appendix shows how the reader can simulate the gyratory compaction process in PyBullet. First, follow the three steps below to install the software.

1. Install Anaconda Navigator with Jupiter Notebook, most Python 3.X versions should work.
2. Install PyBullet using `!pip install PyBullet`.
3. Download all the files from here.

Three files in this link are of importance: 1. Placing, 2. Simulation and 3. Video. These are the files the reader will be using. Also a PyBullet Quickstart guide is added for those who are interested in the software. It is assumed that the reader is familiar with the superpave gyratory compaction process. If not, a NEN norm file is added, which explains the process in detail in chapter 4. Furthermore, the aggregate and environment folders contain files that are needed to run the simulation. The results of the simulations are stored in the text files folder.

The gyratory compaction process consists of three steps: first, placing the aggregates in their initial positions. Second, the compacting the sample. Third, analysing the results. The three files are described.

**Placing**

First, open the Placing file. This file places the aggregates in their initial positions before compacting. In the second cell, after importing the packages, all variables can be found. These are explained here step-by-step. It is recommended to experiment with these to become more familiar with them. When the simulation becomes visible, the user can use the 's' and 'g' keys to change the way it is visualized. The 'alt' key in combination with the mouse can be used to change the camera position and angle. Table B.1 shows and explains all variables in the placement file.

Table B.1: Placement variables

| | |
|---|---|
| VISIBLE | This parameter determines whether the simulation is visible or not. For setting the parameters it is best to set this to 1, for running simulations in the background it is best to set this to 0. |
| g | Gravitational constant |
| n_dynamic | Number of dynamic elements. It is noted that this has a huge impact on computational speed. |
| t_mor | Thickness of the mortar layer |
| d_mould | Diameter of the mould |
| angle_of_gyration | Angle of the mould |
| pressure | pre-pressure during placing process |
| spacing | Spacing in between the aggregates, relative to their size. This parameter allows the user to avoid initial overlaps. |
| n_mould | Number of mould lamella, comparable with segments of a circle. |
| f_elel | Friction parameter element-element |

| | |
|---|---|
| f_elmo | Friction parameter element-mould |
| f_roll | Rolling friction parameter aggregates |
| f_spin | Spinning friction parameter aggregates |
| sieve_size | sieve sizes of the aggregates (see example how to use it) |
| volume_per_sieve | volume percentage per sieve (see example how to use it) |

The rest of the cells perform the calculations, and run the simulation. The end of cell 4 prints the h_min/d_mould ratio. This value depends on the size, and number of elements, as well as the diameter of the mould. This value should ideally be between 0.66 and 1.05. The user may change the variables to their liking, and then run the simulation.

The simulation finishes by storing initial positions of the aggregates to text files. These can later be loaded in the Simulation file. It is noted that certain parameters have to be the same in this file in order to work properly.

**Compaction**

When the placing is finished, the user can start working with the Simulation file. Like in the previous file, it consists of variables which the user can change, and calculations which should be left into place. Table B.2 shows and explains all variables in the compaction file.

Table B.2: Compaction variables

| | |
|---|---|
| VISIBLE | This parameter determines whether the simulation is visible or not. For setting the parameters it is best to set this to 1, for running simulations in the background it is best to set this to 0. |
| g | Gravitational constant |
| t_max | Simulation time in seconds |
| angle_of_gyration* | Angle of the mould |
| d_mould* | Diameter of the mould |
| n_mould* | Number of mould lamella, comparable with segments of a circle. |
| t_mor* | Thickness of the mortar layer |
| c_mor | Damping term of the mortar |
| k_mor | Stiffness term of the mortar |
| pressure | Pre-pressure during placing process |
| f_elel | Friction parameter element-element |
| f_elmo | Friction parameter element-mould |
| f_roll | Rolling friction parameter aggregates |
| f_spin | Spinning friction parameter aggregates |

*These variables must be the same as the placing file.

Again, it is recommended to experiment with the variables to see what effect this has on the results. It is noted that the compaction process might take more time than the placing process.

Once this process finishes, the positions and orientations of the aggregates over time, as well as the height of the end plate are stored in text files. These can later be used in the video file.

### Video

The third file allows the user to see the results in real time. Only three initial parameters exist in this file, all of which must be the same as previously chosen. Table B.3 shows and explains all variables in the video file.

Table B.3: Analysing variables

| | |
|---|---|
| t_mor* | Thickness of the mortar layer |
| d_mould* | Diameter of the mould |
| angle_of_gyration* | Angle of the mould |
| pace | Speed of the simulation, standard is 1 |

A fourth parameter is the pace, this value can be chosen later. This parameter lets the user speed up or slow down the simulation.

It is noted that in case the user are working with thousands of elements, they might want to remove or rename the texture file in the aggregate folder to reduce memory usage. Also, when more than 2000 elements are being analysed, some computers aren't able to produce real time videos. A way to overcome this is by setting the pace to a value lower than 1, and recording the delayed simulation. Later, it can be sped up using video editing software.

## *Appendix C: Determining the inverse of the global inertia tensor*

This section derives the inverse of the global inertia tensor, making use of the local inertia tensor $\boldsymbol{J}_L$, and the rotation matrix $\boldsymbol{R}$, from global to local coordinate system. It shows that the inverse of the local inertia tensor can directly be rotated to form the inverse of the inertia tensor in global coordinates. The advantage of this is that the inertia tensor has to be inverted only once.

Considering $\boldsymbol{M}_L$ as the first order external moment tensor acting on a rigid body in local coordinates. Considering $\boldsymbol{\alpha}_L$ as the first order angular acceleration, also in local coordinates. Equation C. 1 shows the relation between the two.

$$\boldsymbol{M}_L = \boldsymbol{J}_L \cdot \boldsymbol{\alpha}_L \tag{C.1}$$

Equation C. 2 shows how to convert the local quantities to global quantities, making use of the second order rotation tensor.

$$(\boldsymbol{R} \cdot \boldsymbol{M}_G) = \boldsymbol{J}_L \cdot (\boldsymbol{R} \cdot \boldsymbol{\alpha}_G) \tag{C.2}$$

$$\begin{aligned} &with \\ &\boldsymbol{M}_L = \ \boldsymbol{R} \cdot \boldsymbol{M}_G \\ &\boldsymbol{\alpha}_L = \boldsymbol{R} \cdot \boldsymbol{\alpha}_G \end{aligned}$$

Equation C. 3 shows the result after pre-multiplying with $\boldsymbol{R}^T$.

$$\boldsymbol{M}_G = (\boldsymbol{R}^T \cdot \boldsymbol{J}_L \cdot \boldsymbol{R}) \cdot \boldsymbol{\alpha}_G \tag{C.3}$$

Equation C. 4 gives the relation between the moments and angular accelerations in global coordinates.

$$\boldsymbol{M}_G = \boldsymbol{J}_G \cdot \boldsymbol{\alpha}_G \tag{C.4}$$

following Equations C. 3 and C. 4, Equation C. 5 gives the relation between the second order inertia tensor in global and local coordinates.

$$\boldsymbol{J}_G = \boldsymbol{R}^T \cdot \boldsymbol{J}_L \cdot \boldsymbol{R} \tag{C.5}$$

Since the accelerations are to be determined, and the external moments are known, the Equation is rewritten for the accelerations. Equation C. 6 gives the relation between the accelerations in global coordinates, and external moments in global coordinates. Since the rotation matrix is orthogonal, it’s transpose equals the inverse.

$$\boldsymbol{J}_G{}^{-1} = (\boldsymbol{R}^T \cdot \boldsymbol{J}_L \cdot \boldsymbol{R})^{-1} = \boldsymbol{R}^T \cdot \boldsymbol{J}_L{}^{-1} \cdot \boldsymbol{R} \tag{C.6}$$

It becomes clear that the local inertia tensor has to be inverted only once, and can be used in every time step, reducing the computation time.

## Appendix D: Frictional stability

This appendix derives a stability requirement for the friction. Equation D.1 gives the equation for the friction force at the point of contact.

$$F_f \leq \mu F_N \tag{D.1}$$

In numerical simulations this value correctly describes the motion. However, in the case of low velocities the friction might 'overshoot', and thus result in unrealistic oscillating movement. This paragraph aims to mitigate that effect.

Consider the sliding element on a surface in Figure D.1 with relative orthogonal contact velocity $\Delta v \ll 1$ at the point of contact. This object has mass $m$ and rotational inertia $J$. A frictional force $F_f$ acts on the object over a time span $\Delta t$. In order to avoid oscillations, the frictional force must be reduced to ensure that the relative contact velocity remains positive semi-definite.

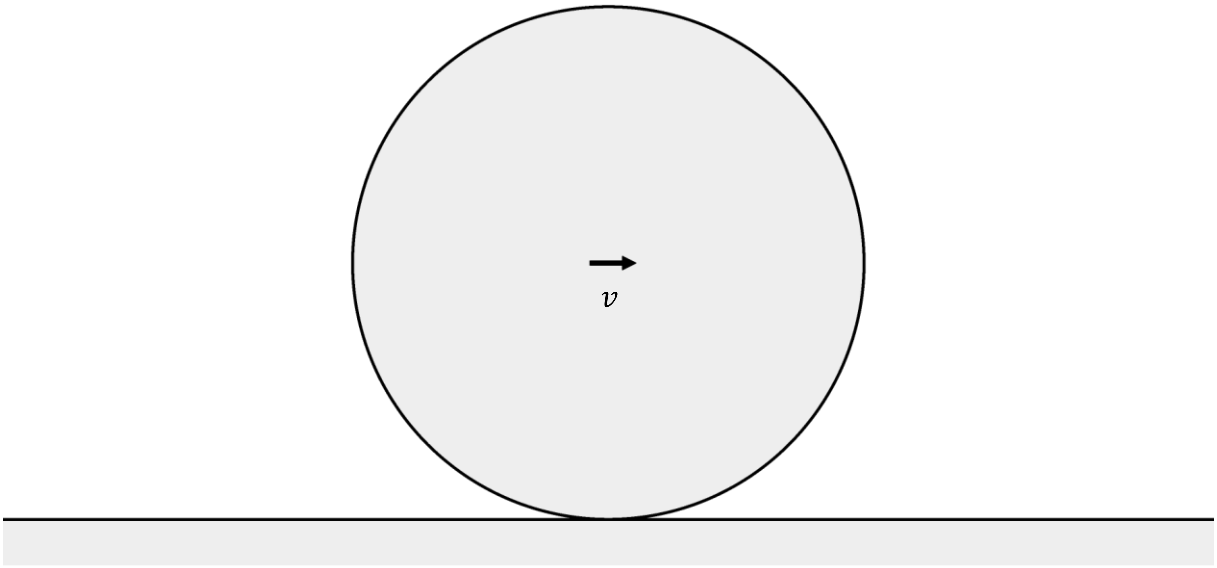


Figure D.1: Sliding/rotating element over a surface

Equation D.2 gives the difference in linear velocity at the contact position, $\Delta v_l$, due to the friction over a single time step.

$$F_f = ma \leftrightarrow \Delta v_l = \frac{\Delta t}{m} F_f \tag{D.2}$$

$$with \;\; a = \frac{\Delta v_l}{\Delta t}$$

Equation D.3 gives the difference in velocity at the contact position, $\Delta v_a$, due to the frictional torque over a single time step.

$$M_f = J\alpha \ \leftrightarrow \ \Delta v_a = \frac{\Delta p^2 \Delta t}{J} F_f \tag{D.3}$$

$$\begin{aligned}
&with\\
&M_f = \Delta p F_f\\
&\alpha = \frac{\Delta\omega}{\Delta t}\\
&\Delta v_a = \Delta\omega\, \Delta p
\end{aligned}$$

Equation D.4 gives the stability condition. Which results in the maximum frictional force, where the relative velocity becomes zero in a single time step.

$$\Delta v \geq \Delta v_l + \Delta v_a = \left(\frac{1}{m} + \frac{\Delta p^2}{J}\right) \Delta t\, F_f \ \leftrightarrow \ F_f \leq \frac{mJ}{\Delta p^2 m + J} \frac{\Delta v}{\Delta t} \tag{D.4}$$

Equation D.5 gives the final solution for the frictional force, which concludes the derivation of the maximum friction acting on a single element in 1D.

$$F_f = \min\left(\mu F_N\, ; \, \frac{mJ}{\Delta p^2 m + J} \frac{\Delta v}{\Delta t}\right) \tag{D.5}$$

In 3D, the same derivation can be done, making use of tensors. Considering the same element in Figure D.1, Equation D.6 gives the first order linear velocity tensor at the contact position, $\boldsymbol{\Delta v_l}$, due to the friction over a single time step.

$$\boldsymbol{F_f} = m\boldsymbol{a} \ \leftrightarrow \ \boldsymbol{\Delta v_l} = \frac{\Delta t}{m} \boldsymbol{F_f} \tag{D.6}$$

$$with \ \ \boldsymbol{a} = \frac{\boldsymbol{\Delta v_l}}{\Delta t}$$

Equation D.7 gives the first order velocity tensor at the contact position, $\boldsymbol{\Delta v_a}$, due to the frictional torque over a single time step.

$$\boldsymbol{M_f} = \boldsymbol{J} \cdot \boldsymbol{\alpha} \ \leftrightarrow \ \boldsymbol{\Delta v_a} = \Delta t\, \boldsymbol{J}^{-1} \cdot \left(\boldsymbol{\Delta p} \times \boldsymbol{F_f}\right) \times \boldsymbol{\Delta p} \tag{D.7}$$

$$\begin{aligned}
&with\\
&\boldsymbol{M_f} = \boldsymbol{\Delta p} \times \boldsymbol{F_f}\\
&\boldsymbol{\alpha} = \frac{\boldsymbol{\Delta\omega}}{\Delta t}\\
&\boldsymbol{\Delta v_a} = \boldsymbol{\Delta\omega} \times \boldsymbol{\Delta p}
\end{aligned}$$

Equation D.8 gives the first order critical velocity tensor, at which the contact velocity difference nullifies in a single time step.

$$\boldsymbol{\Delta v_{crit}} = \boldsymbol{\Delta v_l} + \boldsymbol{\Delta v_a} = \Delta t \left(\frac{\boldsymbol{F_f}}{m} + \boldsymbol{J}^{-1} \cdot \left(\boldsymbol{\Delta p} \times \boldsymbol{F_f}\right) \times \boldsymbol{\Delta p}\right) \tag{D.8}$$

Since the cross product does not know an inverse operation for two non-orthogonal vectors, this final equation cannot be solved analytically for $\boldsymbol{F_f}$. However, since the equation is linear, the initial value, $\mu F_N$, can easily be reduced to satisfy the stability condition, as shown in Equation D.9.

$$\|\boldsymbol{F_f}\| = \begin{cases} \mu F_N & for \quad \Delta v \geq \Delta v_{lim} \\ \gamma \mu F_N & for \quad \Delta v < \Delta v_{lim} \end{cases} \tag{D.9}$$

$$\begin{aligned} & with \\ & \gamma = \frac{\Delta v}{\Delta v_{lim}} \\ & \Delta v_{lim} \geq \|\boldsymbol{\Delta v_{crit}}\| \end{aligned}$$

In case the friction is reduced more than $\gamma$, the velocity difference is damped slowly, which is comparable with supercritical damping. In case the frictional force is reduced less, oscillations occur, which is comparable with subcritical damping. Therefore the term 'critical' was chosen for this velocity. It is noted that to avoid oscillations, the limit velocity must be at least equal to the critical velocity.

Next, let us consider two dynamic elements $A$ and $B$, which interact with each other and thus exert frictional forces. Equation D.10 gives the combined critical velocity in this case. The rest of the computations remain identical as shown above.

$$\boldsymbol{\Delta v_{crit}} = (\boldsymbol{\Delta v_l} + \boldsymbol{\Delta v_a})_{\boldsymbol{A}} + (\boldsymbol{\Delta v_l} + \boldsymbol{\Delta v_a})_{\boldsymbol{B}} \tag{D.10}$$

When considering multiple objects, interacting with each other, one might argue that this method might still overestimate the friction, as multiple frictional forces might work in the same direction. However, this effect is mitigated by the fact that confined non-spherical objects have severely restricted rotational freedom, effectively losing a degree of freedom and so compensating for this. To be conservative, the limit velocity could be set to $\Delta v_{lim} = 3\ \|\boldsymbol{\Delta v_{crit}}\|$. It is noted that the time step must be small enough to minimize the error. In conclusion, the frictional forces are calculated correctly, while the 'overshooting' effect is considerably mitigated.

## Appendix E: Non-penetration condition

This appendix provides a possible non-penetration condition for the aggregates. In reality, aggregates have an extremely high elastic modulus and sub-critical damping. Thus, in a discretized simulation, the non-deformable bodies should barely overlap. This would quickly lead to instability. Therefore, a more efficient method has been developed.

In order to avoid large overlaps, the elements are modelled with a high stiffness and critical damping. Naturally, the contact stiffness of the aggregates should be chosen as high as the step size allows. In order to ensure stability, interacting elements may not increase their total kinetic energy. This is achieved by implementing critical damping when two elements diverge. In order for this damping term to come into effect, multiple time steps are required while the aggregates overlap. Therefore, the contact force may not lead to the separation of elements in a single time step. This leads to limitations for the contact force and the contact stiffness.

Equation E.1 presents the impulse equation, which is integrated over a time step $\Delta t$. This results in the limit for the contact force $F_{lim}$, at which the elements separate in a single time step, assuming a first order integration.

$$\int_0^{\Delta t} Ft\,dt = \int_0^{\Delta t} mv\,dt \leftrightarrow \frac{1}{2}F\Delta t^2 = mu \leftrightarrow F_{lim} = \frac{2mu}{\Delta t^2} \tag{E.1}$$

Equation E.2 presents the equation for the spring force, which must be much lower than the limiting value. Solving this equation for the spring stiffness $k$ gives the limit value, for which the simulation is stable.

$$F = ku \ll F_{lim} \leftrightarrow k \ll \frac{2m}{\Delta t^2} \tag{E.2}$$

Equation E.3 presents the value for the critical damping term.

$$c = 2\sqrt{km} \tag{E.3}$$

It is noted that this method eliminates all restitution, which is not a huge issue for the purpose at hand.

## Appendix F: Equation of motion for the expanded contact model

This paragraph derives the equation of motion for the expanded Burgers' contact model, where an additional damper is added parallel to the maxwell spring. Figure F.1 presents the mechanical scheme.

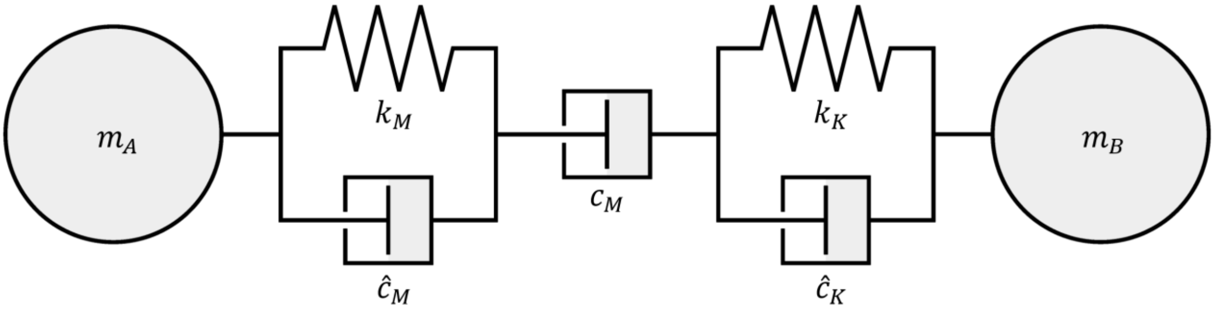


Figure F.1: Improved Burgers' contact model – expanded version

Equation F.1 presents the elastic force in the left part. Equation F.2 presents the viscous force in the middle part. Equation F.3 presents the viscoelastic force in the right part. All three are solved for their relative displacements, where the time derivative is temporarily treated as a constant. This way, the equation of motion can easily be derived.

$$F_e = k_M u_e + \hat{c}_M \dot{u}_e \qquad \leftrightarrow \qquad u_e = \frac{F_e}{k_M + \hat{c}_M \frac{d}{dt}} \tag{F.1}$$

$$F_v = c_M \dot{u}_v \qquad \leftrightarrow \qquad u_v = \frac{F_v}{c_M \frac{d}{dt}} \tag{F.2}$$

$$F_{ve} = k_K u_{ve} + \hat{c}_K \dot{u}_{ve} \qquad \leftrightarrow \qquad u_{ve} = \frac{F_{ve}}{k_K + \hat{c}_K \frac{d}{dt}} \tag{F.3}$$

Equation F.4 shows how to determine the total displacement. Since all springs and dampers are massless, the contact force is the same in all three parts.

$$u = u_e + u_v + u_{ve} = \frac{F}{k_M + \hat{c}_M \frac{d}{dt}} + \frac{F}{c_M \frac{d}{dt}} + \frac{F}{k_K + \hat{c}_K \frac{d}{dt}} \tag{F.4}$$

The equation of motion is rewritten in Equation F.5. This is achieved by first rewriting the right hand side as a single fraction, and consecutively bringing the denominator to the left hand side, as to allow for the correct implementation of the time derivatives. This results in the following equation of motion for the system.

$$\begin{aligned}(c_M \hat{c}_K + c_M \hat{c}_M + \hat{c}_K \hat{c}_M)\ddot{F} + (c_M k_K + c_M k_M + \hat{c}_M k_K + \hat{c}_K k_M)\dot{F} + k_K k_M\, F \\ = c_M \hat{c}_K \hat{c}_M\, \dddot{u} + (c_M \hat{c}_K k_M + c_M \hat{c}_M k_K)\ddot{u} + c_M k_K k_M\, \dot{u}\end{aligned} \tag{F.5}$$