PhD-FSTC-2012-04
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 13/02/2012 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN SCIENCES DE L'INGENIEUR

by

KASRA SAMIEI

Born on 21 April 1977 in Tehran (Iran)

ASSESSMENT OF IMPLICIT AND EXPLICIT ALGORITHMS IN NUMERICAL SIMULATION OF GRANULAR MATTER

Dissertation defense committee

Dr.-Ing. B. Peters, dissertation supervisor
Professor, Université du Luxembourg

Dr.-Ing. M. Beckmann
Professor, Technische Universität Dresden

Dr.-Ing. S. Maas, Chairman
Professor, Université du Luxembourg

Dr. A. Džiugys
Lithuanian Energy Institute

Dr. A. Frommer, Vice Chairman
Professor, Bergische Universität Wuppertal

Abstract

The objectives of this dissertation are to investigate and demonstrate the potentials of implicit integration methods in predicting the dynamics of granular media and to describe the granular dynamics on forward and backward acting grates by discrete element method.

The software used is the Discrete Particle Method (DPM) which is developed based on the principles of the Discrete Element Method. Newton's equations of motion for translation and rotation are solved at each time step and for each particle in the system. The positions and orientations along the trajectory of the particles are then predicted by numerical integration. The weight of the particles, the impact force from other particles and the external forces are taken into account. The impact forces are modelled based on the developed impact models in the literature.

Traditionally, explicit integration methods are employed within the context of Discrete Element Method. Generally, explicit equations are simpler to solve than the implicit ones but they require a small time step to be utilized. Implicit integration methods enable larger time steps to be used. However a system of non-linear equations need to be solved at each time step.

In this study, an implicit Numerov integration scheme is employed to integrate the equations of motion. The Newton-Raphson method is used to solve the system and the Gauss-Seidel iterative method is employed to solve the linear system inherent in the Newton-Raphson method. The Jacobian and right hand side matrices are updated at each iteration which is required for a fully implicit implementation. The implicit method is verified in different test cases starting from simple cases to more complicated cases including hundreds of particles. Comparing the results with the results of the explicit method, it is shown that the implicit method exhibits a distinguished advantage only at very large time steps. Taking into account the overhead of solving non-linear equations at each time step, it is concluded that implicit methods are computationally too expensive for their limited gains. Admitting the current state of the art where the computational costs of force computation is a major obstacle in large scale Discrete Element Method analysis, any integration method which requires more than one force computation per time step is not recommended. Assessing different methods, semi-implicit methods which employ predictor-corrector schemes are suggested. These schemes only require one force evaluation per time step while partly conserving the advantages of the implicit integration.

Addressing the second objective of this study, the residence time distribution of granular particles on forward and backward acting grates are numerically predicted. Reciprocating grates are commonly used in waste to energy plants where combustible solid waste and biomass particles are converted to heat and electricity. In this study, the analysis is performed using model fuels on a cold grate where heating and combustion is excluded and only the motion of the granular media is studied. The residence time distribution in forward and backward acting grates are compared with each other which indicates better mixing behaviour in backward acting grates. The results of the numerical prediction are compared to the available experimental results. Consistent set of boundary conditions are used in the predictions and experiments and the results are compared quantitatively based on direct comparison of absolute values. Several cases are studied which differ in the mass flow rate and/or particles' material. Very good agreement between the predictions and experimental results is achieved. It can be concluded that the Discrete Element Method is highly capable of predicting the dynamics of solid fuel particles on grate systems. Future work shall include coupling of the method with computational fluid dynamics in order to account for thermal conversion of the fuel particles.

Acknowledgement

First of all, I would like to thank my PhD supervisor, B. Peters, for his continuous support and supervision throughout the whole period of my PhD study. His commitment and firm management approach have strongly contributed to my qualifications as a PhD graduate.

I also would like to thank my supervisors from the University of Wuppertal, M. Bolten and A. Frommer, for their support, supervision and useful comments while hosting me at their university. What would we, engineers, do without you mathematicians?!

I am very much grateful to the cooperation of M. Beckmann from the University of Dresden who shared their experimental data with me. Without this, a main part of this thesis was not possible. I am also grateful to the personnel of the CUTEC institute in Clausthal-Zellerfeld, in particular M. Davidovic, who facilitated a visit to their experimental device.

The simulation of the backward acting grate in chapter 3 is carried out in collaboration with my great colleague and friend, G. Berhe. His contribution is very much acknowledged. Working with him was easy, fun and a great learning experience. How much can we, engineers, do in modern societies without the support of you computer scientists?!

I am also thankful to S. Maas for the helpful discussions and comments on my thesis. The acceptance of A. Džiugys from the Lithuanian Energy Institute to be a member of my defense jury is also acknowledged.

Thanks also to my great colleagues F. Hoffmann, L. Breyer, S. K. Kannan and M. Michael for their support, friendship and the interesting discussions we had. It was a lot of fun to work with such great people.

Experiments presented in this paper were carried out using the HPC facility of the University of Luxembourg.

Contents

List of Figures

List of Tables

List of Symbols

Latin symbols

A	normal dissipation coefficient
a	acceleration
c	dissipation coefficient
CR	contact resolution
$C(t)$	concentration of the tracer material at the outlet
D	diameter
d	distance
E	Young modulus
e	coefficient of restitution
$E(t)$	residence time distribution function
F	force
$F(t)$	cumulative distribution function
G	shear modulus
g	earth's standard gravity
h	packed bed height
I	moment of inertia
J	Jacobian matrix
k	spring stiffness
l	length of the grate
m	mass
$\hat{n}$	normal direction of impact
q	quaternion
R	radius
r	position
T	torque
t	time
$\hat{t}$	tangential direction of impact
V	volume
v	velocity
w	width of the grate

Greek symbols

α	exponent for impact law
Δt	time step
δ	overlap depth
θ	Euler angle
μ	friction coefficient
ν	Poisson ratio
ρ	density
ϕ	Euler angle
ψ	Euler angle
ω	angular velocity

Chapter 1

Introduction

1.1 Motivation and Scientific Context

Granular materials consist of a large number of particles whose typical size ranges from micrometers to centimetres [72]. Such materials can be found in enormous variety of places in nature and industry ranging from the powders used to make vitamin pills to the rings of Saturn. In nature examples of granular flow include avalanches, landslides, river sedimentation, dune formation, planetary ring dynamics, soil liquefaction and ice flow [49]. The industries that handle granular materials include food preparation, pharmaceutical, consumer products, agricultural products, metal powders, mechanical, geo-technical, chemical, nuclear and green industries [2].

The prominent presence of granular material in our culture gives the subject great economic importance. The worldwide annual production of grains and aggregates of various kinds is gigantic, reaching approximately ten billion metric tons. Coal accounts for about 3.5 billion tons of that total, cements and ordinary construction materials for about one billion tons, to which we can add equal amounts of sand and gravel. The processing of granular media and aggregates consumes roughly 10 percent of all the energy produced on this planet [24]. As it turns out, this class of materials ranks second, immediately behind water on the scale of priorities of human activity [86]. As such any advance in improving the technologies which process or transport granular media is bound to have a major economic impact. However, this has not been given enough attention until very recently. For instance, a US government agency report in 1993 stressed such obsolescence of the techniques used to process granular materials while admitting at least 40 percent (roughly 61 billion US dollars) of all money investments of the US chemical industry were linked to particle technology [27]. Another report points out that particulate and multiphase processing rarely reach more than 60% of the design capacity because of inadequate understanding of the fundamentals [68].

Modelling of granular assembly is not straightforward. Granular materials are complex systems of a large number of particles of various sizes, shapes and materials. Under different circumstances they reveal different behaviour such as solid, liquid or gas state behaviour. In fact, the science of granular flow is not yet well understood and well developed as other class of materials [84, 15]. In order to predict and optimize the behaviour and motion of granular matter in engineering devices, numerical simulation tools are increasingly employed [16]. Simulation is popular especially because experiments with engineering devices are frequently expensive, time-consuming and sometimes even dangerous [84]. The continuous increase in computing power is now enabling researchers to implement numerical methods that deduce its global characteristics from analysing the individual behaviour of each grain. However, demands for the extension of computer simulation methods to granular systems of increasing size and complexity cannot be met solely by developing better hardware and consuming more computer time. Needs for conceptual improvement of the existing algorithms are becoming more and more obvious. The objective of this dissertation is two-fold. The first objective is to present an implicit algorithm, which is a potential improvement over existing explicit methods, to predict the motion of granular material. The second objective is numerical simulation of the motion of granular material on forward and backward acting grates.

1.2 Disposition

The present dissertation is organized as follows.

Chapter 2 introduces the Discrete particle Method (DPM) which is the tool derived from the Discrete Element Method and is employed in this dissertation. The theoretical foundation of this tool including the governing equations, the force models, the integration schemes and the methods to model particles of different shapes and geometries are introduced. This is followed by introducing the software's features and examples of industrial applications.

The subject of Chapter 3 is the implicit method. The chapter's opening is a brief description of the theory of the matter followed by a review of the status of the implicit approach in the literature. The implicit algorithm is presented and the governing equations and the methods to solve them are explained. Verification of the implicit method is performed in different test cases starting from simple cases and moving towards more complex settings. The robustness of the results are also discussed and the merits and the bottlenecks of the implicit method are described.

Chapter 4 presents the computational model applied to forward and backward acting grates. The boundary conditions and the computational model used for each case are explained. The comparison of the results with experimental findings are presented. The focus of the moving grates simulation is the residence time behaviour of the particles on forward and backward acting grates and the simulation results are compared with the experiments. The residence time's dependency on various factors such as particles's material and the flow rate of the particles is also studied.

Finally chapter 5 is a summary of the whole dissertation and presents the main conclusions and recommendations for the future work.

Chapter 2

Discrete Element Method

2.1 Introduction

Computer simulations have turned out to be a powerful tool to investigate the physics of the granular flow. This is especially valuable as there is no comprehensive theory on granular materials which reliably predicts the behaviour of such materials so far [84]. To date the Discrete Element Method (also called Distinct Element Method) is the leading approach to simulate the dynamics of granular media. The DEM is a numerical approach where statistical measures of the global behaviour of a phenomenon are computed from the individual motion and mutual interactions of a large population of elements [19]. Modelling is straightforward: the grains are the elements, they interact through local, pairwise contacts, yet are also subject to external factors such as gravitation or contacts with surrounding objects, and they otherwise obey Newtons laws of motion [55]. In contrast to the continuum approach, DEM analysis accounts for inter-particle contacts.

This chapter introduces the Discrete Particle Method (DPM), derived from Discrete Element method (DEM), and its software architecture. This is an advanced numerical simulation tool which deals with both motion and chemical conversion of particulate material such as coal or biomass in furnaces in conjunction with Computational Fluid Dynamics (CFD). However, predictions of solely motion or conversion in a de-coupled mode are also applicable. The focus of this dissertation is only the dynamics of the granular particles.

The Discrete Particle Method uses object oriented techniques which enables representation of real world entities as objects in software. The real world consists of concrete and conceptual entities, including things, relationships and occurrences that have a purpose, structural and behavioural characteristics [89]. In general granular material consists of a finite number of particles with different sizes, shapes and material properties. The first step in the object-oriented approach is to identify common characteristics, which are grouped into classes. From these classes objects are created e.g. an object is instantiated from a particular class to represent a real world object in software. Therefore, an obvious choice is a particle class, which contains all the attributes such as shape and size of a particle. Thus a particle of spherical size and given diameter can be created as an object of the class particle. Elliptical particles are other objects instantiated from the same class. Methods are responsible to pass messages between various objects and thus establishing the desired information between objects. In the current application, the interaction between particles consists of forces exerted by inter-particle contacts. Passing these information between particles causes them to move such as observed in real world applications. As a result, the motion of granular material is represented by a well-defined and transparent hierarchy of classes. A more detailed discussion on the possibilities offered by object-orientation methodologies in simulation of granular dynamics is presented by Peters et al. [81].

2.2 Time Driven Method

In classical Newtonian DEM, there are basically two methods to simulate granular material: event driven and time driven methods. Event driven method is used when the duration of collisions are small compared to the free motion of particles. This could be the case when the particles are very hard and their deformations due to contacts with other bodies are very small. This method is, therefore, also called the hard particle approach. In contrast when the duration of collisions is comparable to the free motion of particles, the time driven method or soft particle approach is better suited [13]. DPM uses the time driven method which is well suited for many granular applications when particles are packed and in close contact.

The Lagrangian time driven method is applied to the discrete particles of a moving ensemble which is regarded as a system of a finite number of particles with a given shape and material properties. The state of particles is obtained by time integration of the dynamics equations derived from the classical Newtonian mechanics approach based on the Newton's second law for translation and rotation of each particle in the particle ensemble. All the forces and moments acting on each particle are evaluated at every time step.

The particles are modelled as visco-elastic bodies. In such a model when two bodies get in contact with each other, they experience small deformations which causes a repulsive force to counter the deformation. This is analogous to the repulsive force in a displaced spring as illustrated in fig. 2.1.

Figure 2.1: Demonstrating the spring model for repulsive force between a particle and a wall before collision (left) and after collision (right)

The deformation is, however, modelled as overlap between the two shapes as shown in fig. 2.2. Contact forces depend on the overlap geometry, material properties and dynamics of the particles. Both the normal and tangential components of the repulsion force with visco-elastic models for energy dissipation and friction are included in the contact force. The change of mass and/or shape of particles due to combustion and chemical reaction, though part of the DPM software, is not considered in this dissertation.

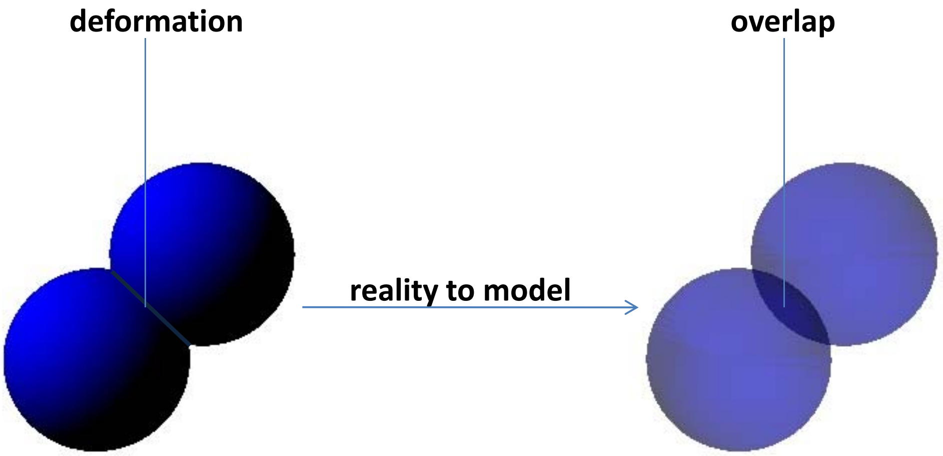


Figure 2.2: Deformation modelled as overlap

2.3 Equations of Motion

2.3.1 Translational Motion

The translational motion of a particle is described by the equation of Newton's second law. It is expressed by the forces acting at the centre of gravity of a particle i and can be written as follows:

$$m_i d\vec{v}_i/dt = m_i\vec{a}_i = \vec{F}_i \quad \text{with} \quad \vec{v}_i = d\vec{r}_i/dt \tag{2.1}$$

where m_i, $\vec{v}_i$, $\vec{a}_i$, $\vec{F}_i$ and $\vec{r}_i$ denote the particle 's mass, velocity, acceleration, force and position, respectively. The force on a particle in general can be from different sources such as the contact forces, the gravity, the Archimedes buoyancy force from the surrounding fluid, the fluid drag and lift forces or other external forces. Though many of these forces are part of the DPM software, this dissertation only takes into account the contact force, the gravity and the external forces from fixed boundary walls:

$$\vec{F}_i = \vec{F}_{i,contact} + \vec{F}_{i,gravity} + \vec{F}_{i,external} \tag{2.2}$$

The contact force is the sum of the forces acting on the particle from other bodies in contact with it:

$$\vec{F}_{i,Contact} = \sum_{j=1,j\neq i}^{N} \vec{F}_{ij} \tag{2.3}$$

where N is the number of bodies including particles and boundary shapes in the system.

2.3.2 Rotational Motion

The orientation of a rigid body in space can be described by three angles known as the Euler angles. There are different definitions for the Euler angles depending on the order of rotation

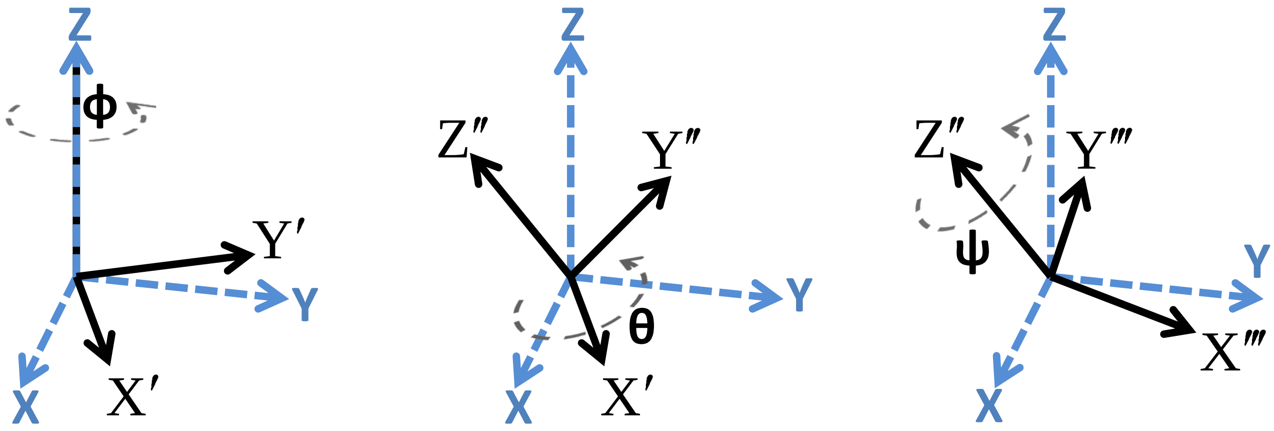


Figure 2.3: Definition of the Euler angles ϕ, θ, ψ, based on rotation around the axes z, x', z'', respectively.

about the axes. The definition used in the DPM is the x-covention i.e. rotation around the axes z, x', z'', respectively [36], as shown in fig. 2.3.

The change in the orientation and consequently the Euler angles is caused by torques acting on particles. Since the gravitational force can be described as a force acting at the centre of mass of a particle, it does not generate any torque and the total torque on the particle can be deduced from eq. 2.2 to be:

$$\vec{T}_i = \vec{T}_{i,contact} + \vec{T}_{i,external} \tag{2.4}$$

where T_i denotes the total torque acting on particle i. The contact torque is derived from the contact force acting on the particle and is expressed as follows:

$$\vec{T}_{i,Contact} = \sum_{j=1,j\neq i}^{N} \vec{T}_{ij} = \sum_{j=1,j\neq i}^{N} \vec{d}_{ij} \times \vec{F}_{ij} \tag{2.5}$$

Where $\vec{d}_{ij}$ is the position of the contact relative to centre of mass of particle i in the contact between particles i and j. Fig. 2.4 illustrates the contact forces and torques on a particle from two other particles.

The relationship between total torque acting on a particle and the evolution of the Euler angles in three dimensional space is not as simple as the translational motion. Before describing the complexity and the solution, first several co-ordinate systems need to be distinguished. Different coordinate systems are illustrated in fig. 2.5 and defined as follows:

- The space-fixed coordinate system or also called the laboratory system is fixed in space and time and is independent of the particles motion. It is the reference system for the simulation and the equations of motion are formulated in this coordinate system.
- In order to simplify the understanding of rotation, the rotational motion can be described separately from the translational motion. To this end, the co-moving coordinate system is introduced as the system where its origin lies at the body's centre of gravity and the axes

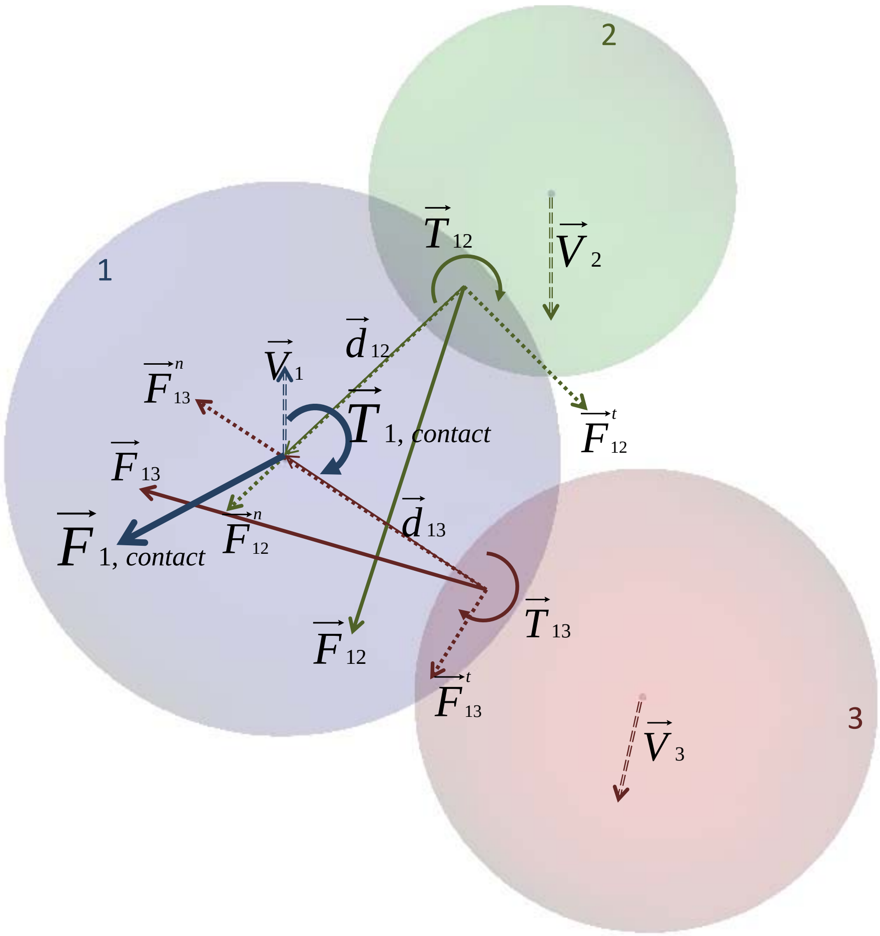


Figure 2.4: Contact forces and torques on a particle which collides with two other particles at the same time

are in the same direction as the space-fixed coordinate system. The relationship between this coordinate system and the space-fixed coordinate system describes pure translation of the body.

- The body-fixed coordinate system is here defined as the system where its origin is the same as the body's centre of gravity and their axes are in the same direction as the body's principal axes at time t. This coordinate system is not fixed in time and is moving with the body. The relationship between this coordinate system and the co-moving coordinate system describes pure rotation of the body.

The forces and torques acting on a particle are generally formulated in the space-fixed coordinate system where the moment of inertia tensor is in general time-dependent and contains off-diagonal elements. In order to avoid this complication, the moment of inertia can be calculated around the principal axes of the shape in which case it is constant and diagonal. This does not cause a problem for spherical shapes which are symmetric around all axes passing through the centre of gravity. The space-fixed coordinate axes can also represent the sphere's principal axes independent of the sphere's orientation. This is not the case for non-spherical shapes though and the contact torque in eq. 2.5 needs to be converted into the body-fixed coordinate system so that

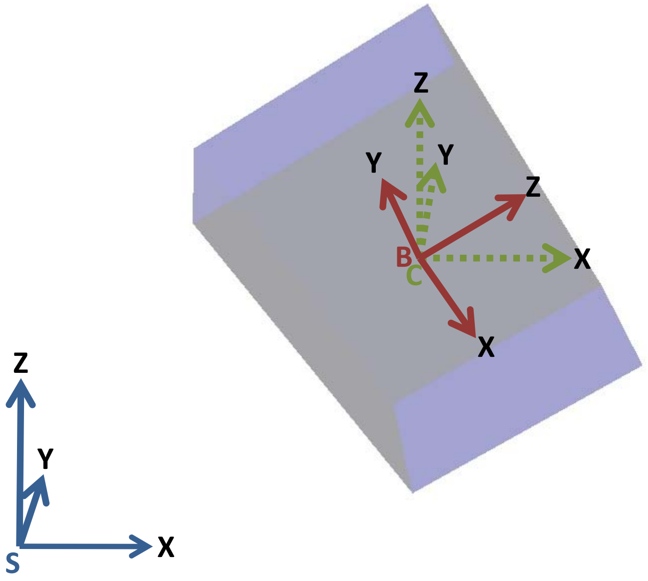


Figure 2.5: Different coordinate systems: Space-fixed (S), Co-moving (C) and Body-fixed (B)

the simple diagonal form of the moment of inertia could be used. The rotational equations of motion in the body fixed co-ordinate system can be described by the so called Euler equations:

$$\begin{bmatrix} T_x^B \\ \\ T_y^B \\ \\ T_z^B \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{\omega}_x^B - \omega_y^B\omega_z^B(I_{yy} - I_{zz}) \\ \\ I_{yy}\dot{\omega}_y^B - \omega_z^B\omega_x^B(I_{zz} - I_{xx}) \\ \\ I_{zz}\dot{\omega}_z^B - \omega_x^B\omega_y^B(I_{xx} - I_{yy}) \end{bmatrix} \tag{2.6}$$

where T^B, I and ω^B are the torque, the moment of inertia and the angular velocity of the bodies in the body-fixed coordinate system, respectively. The Euler equations can be derived from Newton equations [84, 118] or the Lagrange equation of motion [102]. The superscript B in eq. 2.6 indicates that the variables should be converted to the body-fixed coordinate system before inserting into the equation.

The relationship between the angular velocities in the body-fixed coordinate system and the change in the Euler angles ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) can be written as [36]:

$$\begin{bmatrix} \dot{\phi} \\ \\ \dot{\theta} \\ \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{\omega_x^B \sin\psi + \omega_y^B \cos\psi}{\sin\theta} \\ \\ \omega_x^B \cos\psi - \omega_y^B \sin\psi \\ \\ \omega_z^B - (\omega_x^B \sin\psi + \omega_y^B \cos\psi)\frac{\cos\theta}{\sin\theta} \end{bmatrix} \tag{2.7}$$

However, eqs. 2.7 contains singularities when the Euler angle approaches 0, π, 2π, etc. In order to avoid this problem, quaternions are used to describe the rotation of bodies in DPM. Quaternions and Euler angles, together with some other methods, are simply different ways of representing the orientation of bodies in three dimensional space. Diebel, 2006 [23] elaborately describes and compares different methods to represent the orientation of rigid bodies.

There are different equivalent definitions of the quaternion in the literature [1, 109]. The definition used in the DPM relates to the Euler angles by the following equation:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2}\cos\frac{\phi+\psi}{2} \\ \sin\frac{\theta}{2}\cos\frac{\phi-\psi}{2} \\ \sin\frac{\theta}{2}\sin\frac{\phi-\psi}{2} \\ \cos\frac{\theta}{2}\sin\frac{\phi+\psi}{2} \end{bmatrix} \tag{2.8}$$

where q_0 to q_3 represent the four components of the quaternion which should satisfy

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{2.9}$$

Calculating the time derivatives of the quaternions in eq. 2.8 and replacing the derivatives of the Euler angles from eq. 2.7 and after some algebraic manipulation, the evolution of quaternions can be written in terms of quaternions and angular velocities as follows:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} \omega_x^B \\ \omega_y^B \\ \omega_z^B \end{bmatrix} \tag{2.10}$$

The details of the derivations are not presented here and the interested reader can refer to Pöschel et al., 2004 [84] and Džiugys et al., 2001 [118] among other sources for more elaborated discussions.

2.4 Particle Shapes

2.4.1 Spheres

Contact geometry is greatly simplified in case of spherical particles which also explains their common utilization in DEM simulations. Contact detection could be identified in a simple way: two particles are in contact if the sum of their radii exceeds the distance between their centres. The overlap depth can be written as:

$$\delta_{ij} = R_i + R_j - |\vec{r}_i - \vec{r}_j| \tag{2.11}$$

where δ_{ij} stands for the overlap depth between the two spheres and R and $\vec{r}$ represent the radius and the position vector of the particles respectively. The normal direction of contact is simply the unit vector connecting the centre of the two spheres as illustrated in fig. 2.6.

The tangential direction of contact is the normal vector in the direction of the tangential velocity:

$$\hat{t}_{ij} = \frac{\vec{v}_{ij}^{t}}{\left|\vec{v}_{ij}^{t}\right|} = \frac{\vec{v}_{ij} - \vec{v}_{ij}^{n}}{\left|\vec{v}_{ij} - \vec{v}_{ij}^{n}\right|} \tag{2.12}$$

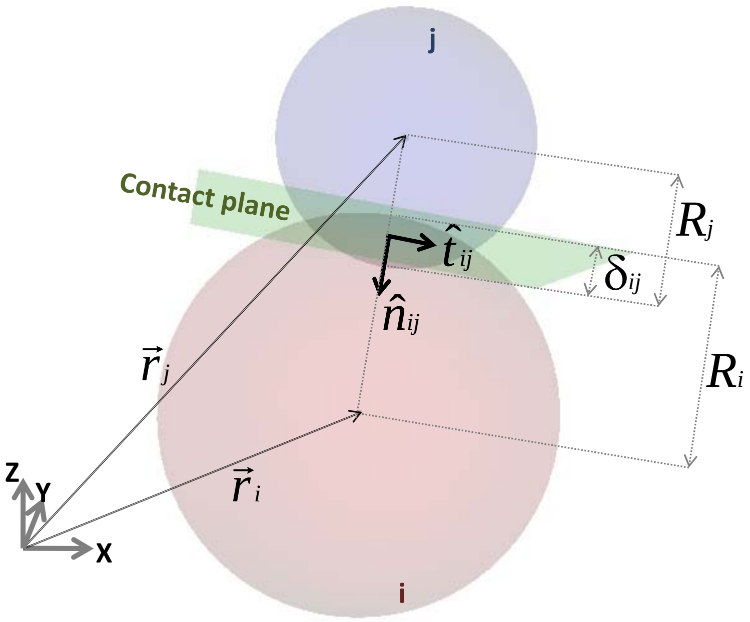


Figure 2.6: Geometry of contact between spherical particles

If the tangential velocity equals zero, the tangential direction is then zero too.

The rotation of particles is also simplified significantly for spheres since the moment of inertia is the same about any axis passing through the centre of gravity.

2.4.2 Shapes other than Spheres

Approximation of granular particles' shape as sphere might be justified for molecular dynamics but not for many industrial practices which include granular grains of more complex shapes or even often irregular arbitrary shapes. DPM therefore provides different particle shapes to be utilized depending on the application. Modelling of non-spherical shapes are in general not as simple as spheres and require larger computational efforts. Mathematical description of contact is developed for some other shapes for example ellipsoids but some experience suggests this method reduces the numerical efficiency significantly especially for more complicated shapes[104, 78]. There are few methods in the literature to model non-spherical shapes, some developed only for certain analytical shapes [118, 41, 61, 56]. The method employed in DPM is a generic method principally based on the work of Nolan et al., 1995 [75]. The shape in this method is replaced by a number of spherical sub-shapes which approximately fill the space the original shape occupies as demonstrated in fig. 2.7. The spherical sub-shapes employed to construct the shape can differ in size and can also overlap each other. The whole shape is considered a rigid body which means the distance between the sub-shapes always remains the same.

The contact between shapes is solved in terms of the contacts between their sub-shapes. Therefore this method retains the simplicities of spherical contact evaluation while eventually solving the dynamics for the desired shape which can be far from sphere. Contact is detected between two bodies if any of their spherical sub-shapes are in contact with each other. The total force and torque acting on a shape is the sum of all the forces and torques acting on its sub-shapes. Based on the rigid body assumption, the forces and torques are calculated for the centre of gravity of the

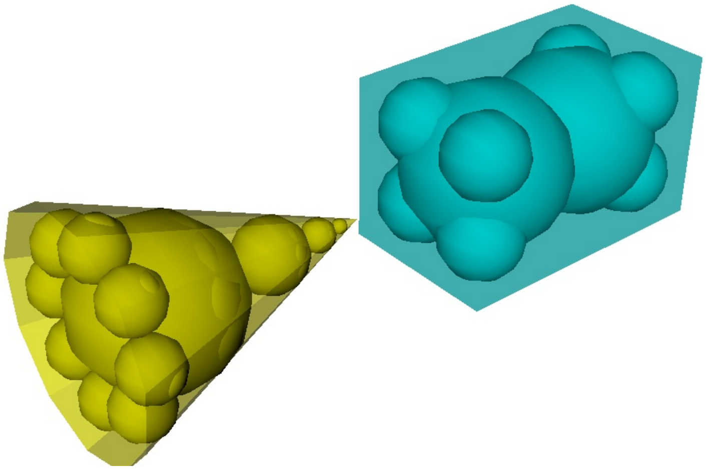


Figure 2.7: Constructing non-spherical shapes from spherical sub-shapes

composite shape.

The approach is illustrated in fig. 2.8 where a particle in shape of a torus is in contact with two perpendicular walls. The torus is composed of seven spherical sub-shapes in this example. Three of these sub-shapes are in contact with the walls. The walls exert forces on these sub-shapes which are the sum of normal and tangential forces. The total force on the torus is the sum of the forces on its sub-shapes:

$$\vec{F}_{external} = \vec{F}_1 + \vec{F}_4 + \vec{F}_5 \tag{2.13}$$

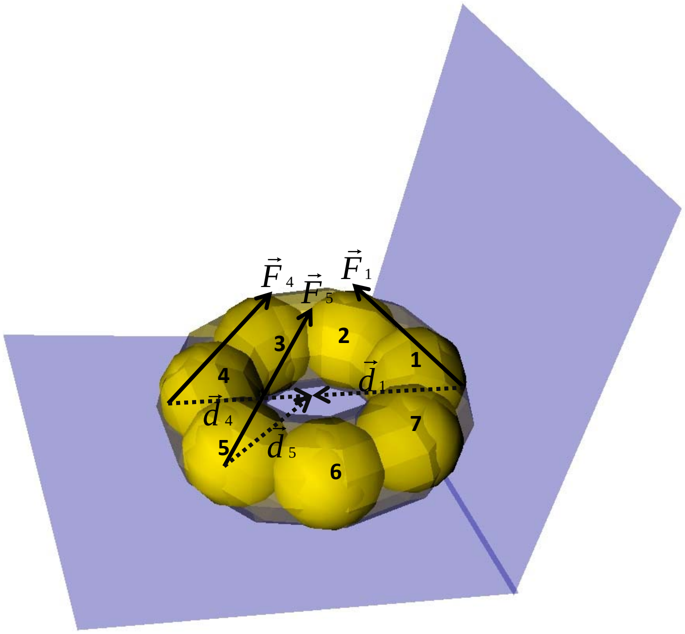


Figure 2.8: A torus in contact with two perpendicular walls at several points

The total torque on the torus is, similarly, the sum of the torques each of these forces exert on the torus based on the position of the impact point in relation to the centre of the torus:

$$\vec{T}_{external} = \vec{d}_1 \times \vec{F}_1 + \vec{d}_4 \times \vec{F}_4 + \vec{d}_5 \times \vec{F}_5 \tag{2.14}$$

Sub-shapes relative position

Solving for the orientation of the shapes does not conclude the problem of rotation. Rotation of a shape in general affects how the sub-shapes are positioned relative to the shape's centre of gravity. To clarify this, the relative position vector of a sub-shape is defined as a vector connecting the shape's centre of gravity to the sub-shape's centre. While the rigid body assumption implies that the relative distance between the sub-shapes and the shape's centre of gravity does not change, their relative position vector changes with shape's rotation as demonstrated in fig. 2.9. Therefore, the last step in solving the rotational motion of non-spherical shapes is updating of the sub-shapes' relative position vector according to the new orientation. Of course, this step is specific to the above introduced method of using sub-shapes to construct a non-spherical shape.

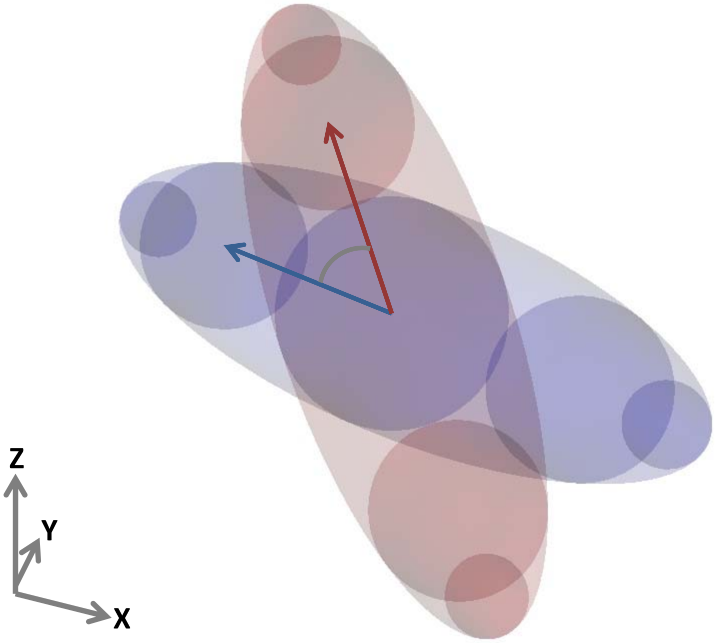


Figure 2.9: Change of the relative position of sub-shapes during rotation

Sub-shape Packing Density

As stated above, the spherical sub-shapes can fill the space the non-spherical shape occupies only approximately. However, the approximation could be improved and even approach absolute precision at the limit by increasing the number of sub-shapes. Such an improvement, however, comes at the cost of higher computational effort. This is not only because the number of contact evaluations would increase but also because the improvement might be the result of using smaller sub-shapes especially at the edges which generally imposes a smaller time step to be used. The balance between the desired accuracy and the efficiency could be adjusted in the DPM code by

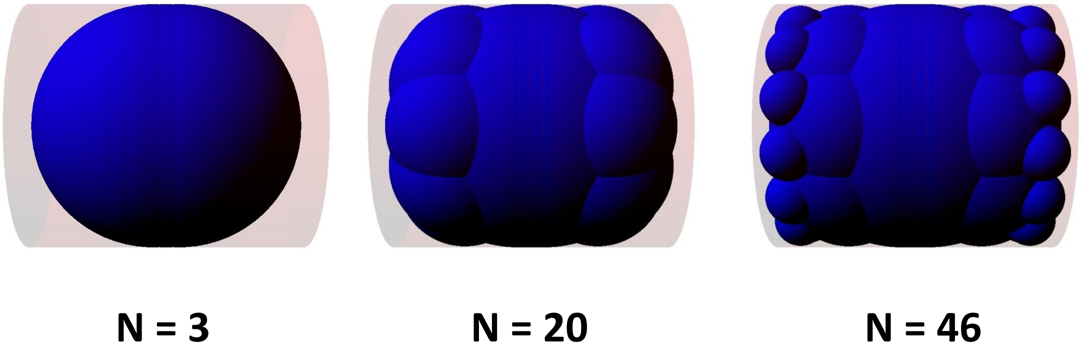


Figure 2.10: Different sub-shape packing density used to construct a cylinder, N = number of sub-shapes.

defining a variable called "sub-shape packing density" which determines the resolution of the shape. Fig. 2.10 shows the assembly of a cylindrical shape by different sub-shape packing densities.

It is worth reminding that real life practices rarely involve perfect mathematical shapes and to take the granular particles as certain exact geometries is already an assumption in itself. Although the presented method in its efficient implementation does not construct the defined analytical geometry with absolute precision, it provides an acceptable approximation for most applications.

2.5 Flowchart

Following the introduction to the basic theory and the governing equations of DEM, the outline of the method is introduced in this section. The principle behind the time-driven method is to discretize time into small time steps where the motion of the system of particles and boundaries is computed for each time step. Summing up the computations for all time steps simulates the motion of the system in the specified duration of time. As already stated, the Newton's second law is employed to calculate the motion of particles. The basic flowchart of the method is sketched in fig. 2.11 and is described below.

The program first reads the input data provided by the user which define the global parameters and boundary conditions of the simulation. The global parameters include the time span of the simulation, the time step, the simulation space, the type of force model and the integration scheme, how often the results are to be saved and so on. Input data also determine the initial number of particles and boundary shapes, their shape, size, material, initial positions, orientations, velocities and angular velocities. If particles are meant to be generated during the simulation, their data plus the generation rate need to be specified too. The program uses the input data to initialize solvers, define the domain space and in case of parallelization, to define the sub-domain space for each processor.

After the initialization step, the time loop starts. First, the collision partners for each particle need to be detected based on eq. 2.11. However, to search and check for contacts with all other particles in the simulation domain is computationally costly and in general unnecessary. For an efficient contact detection, it would be sufficient to check contacts with particles which are near to the particle or the so called neighbouring particles. To this end, neighbour particles need to be identified and since the particles are moving the neighbours need to be updated at each time step. Contacts are checked relying on the shapes' positions, orientations, size and geometry. For

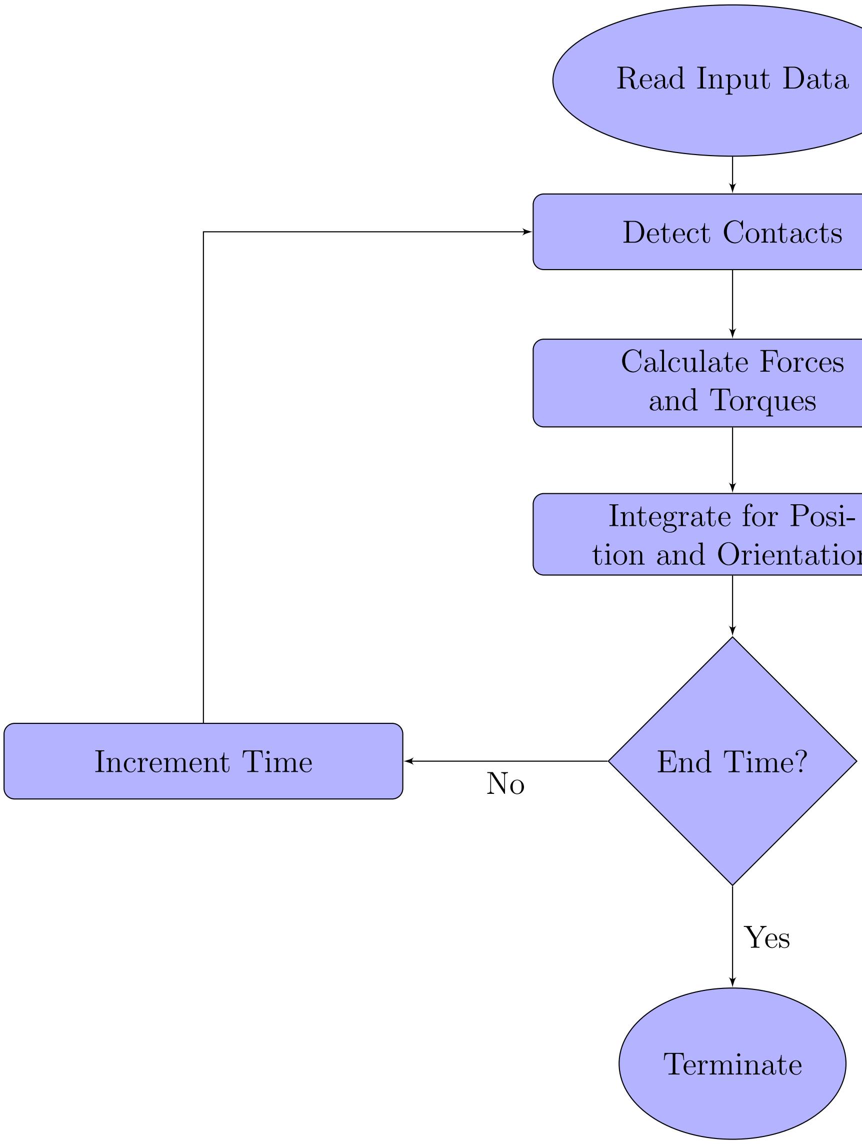


Figure 2.11: The basic flowchart for explicit time-driven method

non-spherical shapes, the contacts are detected based on the contacts between their sub-shapes, as described in section 2.4.2. Every sub-shape of each shape is checked against collisions with every sub-shape of each neighbour shape.

Once two particles are identified as collision partners, their contact force is calculated. The contact force, in general, depends on the mechanical and interaction properties, the overlap, impact velocity and size of the contacting pair. The exact formula to calculate the impact force is dependent on the force model which are described in section 2.6. The torque applied on each particle is then calculated based on its contact force and the position of impact.

The contact forces and torques are then used to calculate the acceleration and angular acceleration of each particle based on Newton's second law for translation and rotation as already described in eqs. 2.1 and 2.6. Numerical integration leads to the acceleration's first and second integrals: velocities and positions. The details of the integration step and different integration methods are presented in section 2.7. Each time step concludes once the positions, orientations, velocities and angular velocities of all particles in the domain space are updated. The time loop continues until the defined end time of the simulation has reached at which point the program safely terminates.

At frequent time steps, the results are saved and written to output files for post-processing. The main results are the shapes' linear and angular positions, velocities and acceleration but may also include collision partners, contact forces and torques. Application specific information might be saved as well such as residence time or discharge time of the particles.

2.6 Impact Models

The impact force between two colliding shapes could be modelled based on different established force schemes in the literature ranging from simple linear force models to non-linear and models with hysteresis. The force models available in the DPM are:

- Linear Spring-Dashpot
- Maw Linear Spring
- Hooke's Impact Model
- Hertz-Brilliantov
- Hertz-Mindlin
- Walton-Braun

These models are going to be described in this section. A user interface allows extending the DPM software by adding user-defined impact models to the already existing ones. For further review and discussions on different force models, one is referred to the works by Ristow [87], Zhu et al. [117], Peters[77], Schafer[92] and Kruggel-Emden et al. [57, 59]. The force equations in this section only represent the magnitude of the normal and tangential forces. The direction of these forces are already discussed in section 2.4. The graphs presented in this section are based on normalized values in order to represent general trends independent of any specific case. The values are normalized over the maximum point or the maximum point in one of the presented series to allow a meaningful comparison.

2.6.1 Linear Spring-Dashpot

The damped linear spring contact force model is based on the work by Cundall and Strack [19]. The normal force in this model is composed of a linear spring force and a dashpot force where the spring produces an elastic repulsive force and the dashpot contributes to the damping as sketched in fig. 2.12. This is mathematically described in eq. 2.15.

$$F_n = m_{ij}\ddot{\delta} = -(k_n\delta + c_n\dot{\delta}) \tag{2.15}$$

where δ is the overlap depth between the contacting pair, m_{ij}, k_n and c_n are the reduced mass, the normal spring stiffness and the normal damping (dissipation) coefficients, respectively. Referring back to eq. 2.11 which defines the overlap depth between two contacting spheres, the first derivative of the overlap ($\dot{\delta}$) corresponds to the relative normal velocity in physical sense. Subsequently, the second derivative of the overlap ($\ddot{\delta}$) corresponds to the relative normal acceleration.

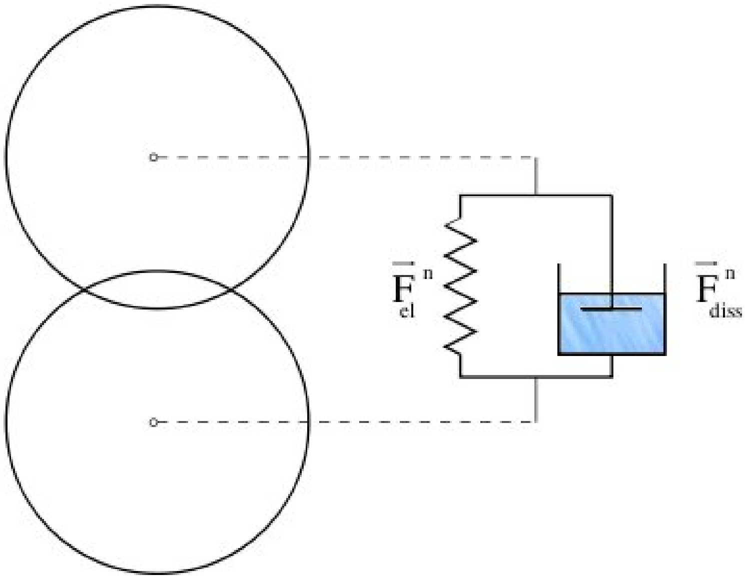


Figure 2.12: Visualisation of spring-dashpot impact model

The reduced mass can be derived from Newton's third law of motion affirming the forces of each particle on the other are of equal magnitude and in the opposite direction:

$$m_{ij}\ddot{\delta} = -m_{ij}(\ddot{r}_i - \ddot{r}_j) = -m_i\ddot{r}_i = m_j\ddot{r}_j \tag{2.16}$$

where m and r represent the mass and position of the contacting particle. The reduced mass can then easily be derived as:

$$\frac{1}{m_{ij}} = \frac{1}{m_i} + \frac{1}{m_j} \tag{2.17}$$

The force differential equation presented in eq. 2.15 has an analytical solution which with boundary conditions $\delta(0) = 0$ and $\dot{\delta}(0) = v_0$, could be written as follows:

$$\delta(t) = e^{-\frac{c_n t}{2m_{ij}}} \cdot \frac{2v_0}{\lambda} \cdot \sin(\frac{\lambda t}{2}) \quad \text{with} \quad \lambda = \frac{\sqrt{4m_{ij}k_n - {c_n}^2}}{m_{ij}} \tag{2.18}$$

Fig. 2.13 plots the analytical solution demonstrating the effect of initial impact velocity on the overlap. The plot shows that the impact velocity does not have any effect on the total duration of contact while the maximum overlap increases linearly by the increase of the impact velocity.

The relative normal velocity $\dot{\delta}$ can be derived from the analytical solution as follows:

$$\dot{\delta}(t) = e^{-\frac{c_n t}{2m_{ij}}} \cdot \frac{v_0}{\lambda}[-\frac{c_n}{m_{ij}}\sin(\frac{\lambda t}{2}) + \lambda\cos(\frac{\lambda t}{2})] \tag{2.19}$$

Employing the analytical solution, the normal dissipation coefficient can be derived in terms of the coefficient of restitution. Coefficient of restitution is an important characteristic of the impact which determines the effect of energy dissipation. It is defined as the ratio between the normal component of the relative velocity of the contacting pair after and before the impact:

$$e = \left|\frac{\dot{\delta}_{final}}{\dot{\delta}_{initial}}\right| \tag{2.20}$$

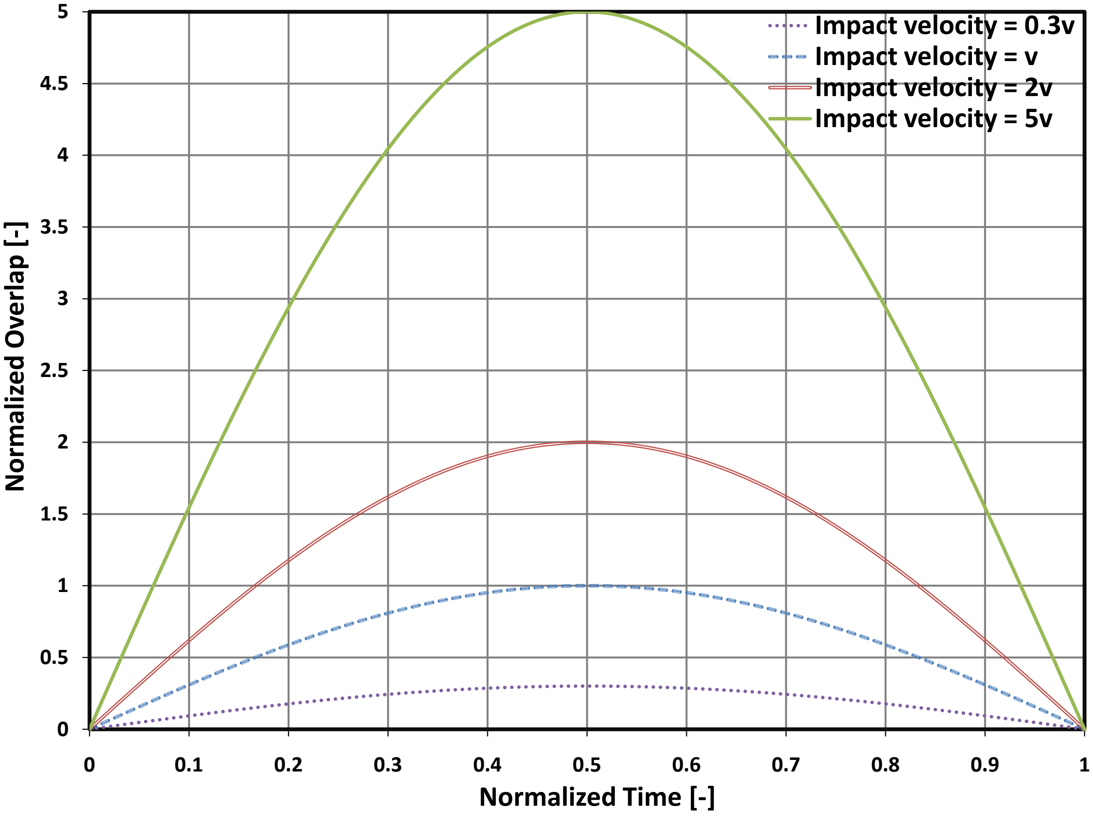


Figure 2.13: Effect of initial impact velocity on the overlap in the linear force model

where e denotes the coefficient of restitution. A coefficient of restitution of 1 corresponds to no damping at all and a value of 0 means 100 percent dissipation of the impact energy. Noting that $\dot{\delta}_{final}$ is when $\delta(t)$ in eq. 2.18 re-approaches zero, the normal dissipation coefficient can then be derived from eqs. 2.19 and 2.20 as follows:

$$c_n = \ln e \sqrt{\frac{4 m_{ij} k_n}{\pi^2 + \ln e^2}} \tag{2.21}$$

The total duration of contact can also be derived from the analytical solution, eq. 2.18:

$$T_{contact} = \sqrt{\frac{m_{ij}(\pi^2 + \ln e^2)}{k_n}} \tag{2.22}$$

As already observed in fig. 2.13, the total duration of contact is independent of the impact velocity in linear force models and it is therefore expressed entirely in terms of material properties. This is helpful in selecting a suitable time step because time step in the DEM simulations should be selected to be as small as a fraction of the contact duration so that each single collision could be solved in several time steps.

The contact duration and coefficient of restitution calculated above in eqs. 2.21 and 2.22 are based on the assumption that the contact is over when the overlap is over. However, the end of the contact is not only dependant on the overlap but also if the normal force becomes attractive force. This might happen during unloading phase if the dissipative term becomes bigger than the repulsive term. The physical explanation is that at some stage during detachment, the particles might separate from each other faster than their deformation is returning back. Based on this, it is necessary to include a cut-off criteria when the contact force is becoming attractive force. The force equation in eq. 2.15 is therefore modified to:

$$F_n = m_{ij}\ddot{\delta} = min[0, -(k_n\delta + c_n\dot{\delta})] \tag{2.23}$$

As a consequence, the real duration of contact is slightly smaller than what is calculated from the differential equation in eq. 2.22. The modified duration of contact can be derived by accounting for the time when the contact force is changing direction:

$$-(k_n\delta_{t_{end}} + c_n\dot{\delta}_{t_{end}}) = 0 \tag{2.24}$$

Replacing δ_t and $\dot{\delta}_t$ from eqs. 2.18 and 2.19 into eq. 2.24, the total duration of contact for the linear spring-dashpot model is modified to:

$$T_{contact} = \frac{2}{\lambda}[\pi + \arctan(\frac{c_n\lambda m_{ij}}{{C_n}^2 - 2m_{ij}K_n})] \tag{2.25}$$

Fig. 2.14 shows the effect of the coefficient of restitution on the overlap. It shows that as the coefficient of restitution declines, or in other words energy dissipation increases, the particles deformations become smaller. Also observed in fig. 2.14 is the transformation of the shape of the overlap curve by varying the coefficient of restitution. The curve deviates from its symmetric shape around the centre with increasing dissipation. In particular, increasing dissipation shortens the duration of the loading phase of the spring. It is also seen that the collision is cut off at some point during unloading before the overlap returns to zero. This cut-off point is already explained above in eq. 2.23 and is the point when the force becomes attraction due to high relative velocity. The loading is defined as the phase of the collision when particles are approaching each other while the unloading phase is when the contacting particles are departing one another. The loading and unloading phases of a collision are illustrated in fig. 2.15.

It is worth stating that the coefficient of restitution, in reality, does not only depend on the material properties but also the impact velocity [10, 112, 65, 84]. However, such an effect is only taken into consideration in the Hertz-Brilliantov force model described in section 2.6.4. The other force models discussed in this chapter including the linear spring-dashpot model lead to a coefficient of restitution independent of the impact velocity.

The maximum overlap of the contact could also be calculated from the analytical solution:

$$\delta_{max} = e^{-\frac{c\pi}{2\sqrt{4m_{ij}k_n - c^2}}}(\frac{2m_{ij}v_0}{\sqrt{4m_{ij}k_n - c^2}}) \tag{2.26}$$

which in case of no damping will reduce to $\delta_{max} = v_0\sqrt{\frac{m_{ij}}{k_n}}$. The value of maximum overlap, which models the maximum deformation as explained in fig. 2.2, can be used to check whether the objects can withstand the impacts with the chosen stiffness. At the extreme case of the overlap exceeding the radius of the particle, the particles start to pass one another or the boundaries because the impact direction starts to change to the opposite direction. In reality, this could be thought of particles breaking or permanently deforming. The maximum overlap in DEM simulations is meant to be small relative to the particle's radius.

The tangential force is calculated from the static friction force and is limited by dynamic friction force:

$$F_t = min(\text{static friction, dynamic friction}) \tag{2.27}$$

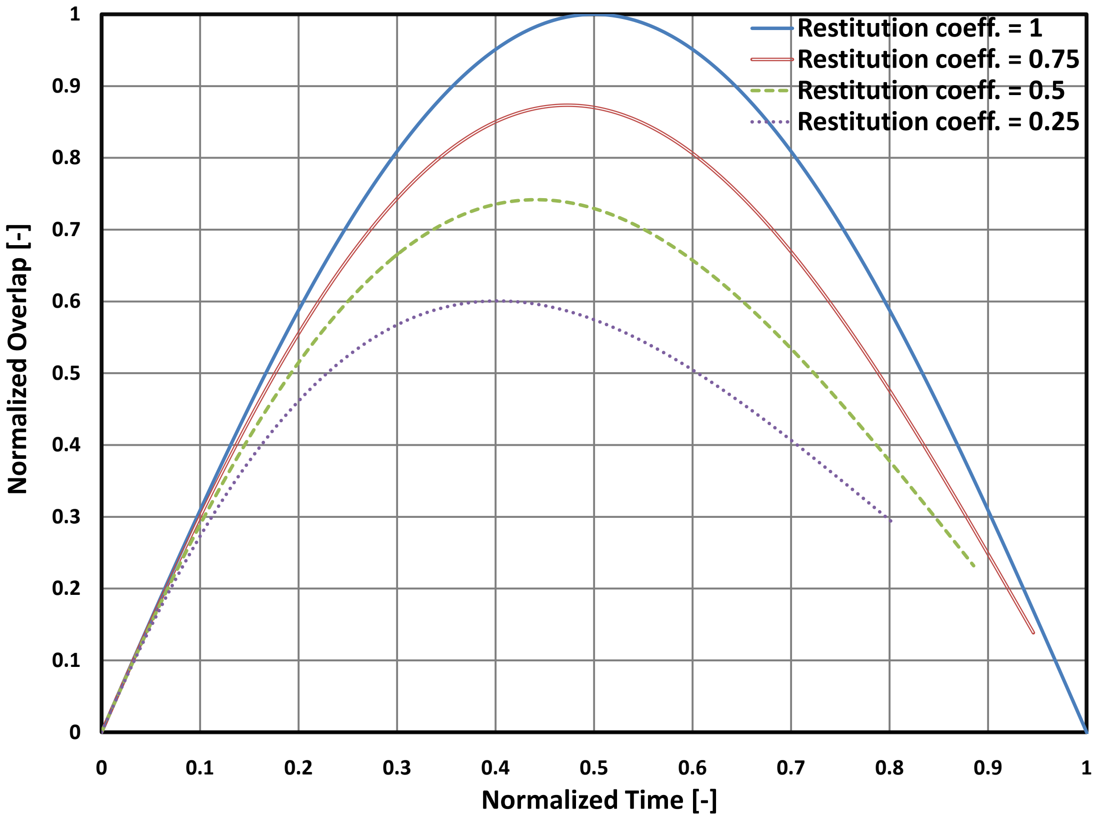


Figure 2.14: Effect of the coefficient of restitution on the overlap in the linear force model

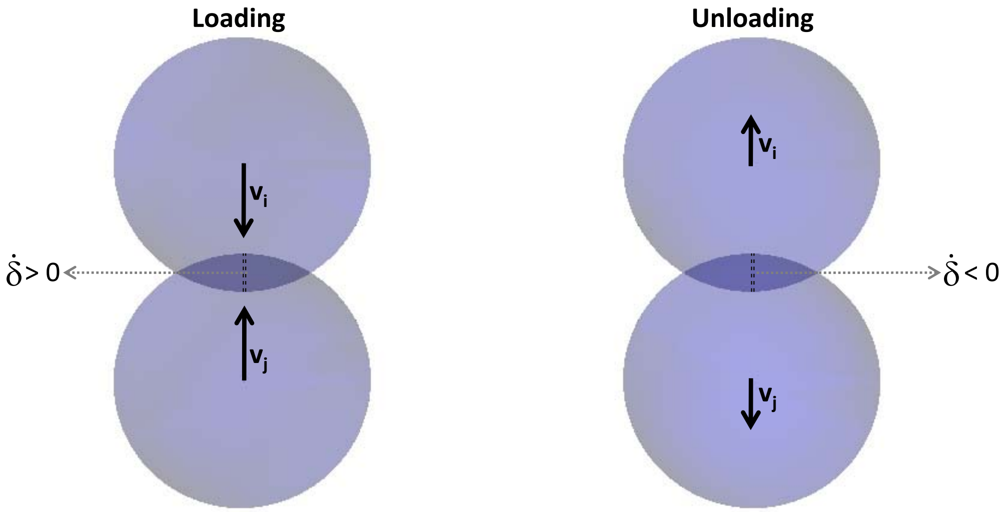


Figure 2.15: Loading and unloading phases of a collision

The static friction in this model is a viscous damping force and the dynamic friction is based on the Coulomb's law of friction:

$$F_t = min(c_t\dot{\delta}_t, \mu F_n) \tag{2.28}$$

where c_t is the tangential damping coefficient and μ is the Coulomb's friction coefficient and $\dot{\delta}_t$ is the relative tangential velocity. If same stiffness and same coefficient of restitution in normal and tangential directions were assumed, the tangential damping coefficient would be chosen equal to the normal damping coefficient.

2.6.2 Maw Linear Spring

This force model is in principle the same as Linear Spring-Dashpot force model. The difference is that the spring constant in this model is calculated by equating the maximum strain energy in a purely Hertzian contact with the maximum strain energy of the existing contact [22, 67]:

$$k_n = \frac{16}{15}R_{ij}^{\frac{1}{2}}E_{ij}(\frac{15m_{ij}v^2}{16R_{ij}^{\frac{1}{2}}E_{ij}})^{\frac{1}{5}} \tag{2.29}$$

where R_{IJ}, E_{ij} and v are the reduced radius, effective young modulus and the impact velocity respectively. Reduced radius is defined as:

$$\frac{1}{R_{ij}} = \frac{1}{R_i} + \frac{1}{R_j} \tag{2.30}$$

The effective young modulus is:

$$\frac{1}{E_{ij}} = \frac{1-{\nu_i}^2}{E_i} + \frac{1-{\nu_j}^2}{E_j} \tag{2.31}$$

where ν denotes the Poisson ratio. The impact velocity could be estimated as the maximum velocity of a particle in the system.

Therefore, the general formula for the magnitude of the normal force could be written as:

$$F_n = m_{ij}\ddot{\delta} = -(\frac{16}{15}R_{ij}^{\frac{1}{2}}E_{ij}(\frac{15m_{ij}v^2}{16R_{ij}^{\frac{1}{2}}E_{ij}})^{\frac{1}{5}}\delta + c_n\dot{\delta}) \tag{2.32}$$

The tangential force is again a function of static and dynamic friction but the static friction in this model is based on a spring-dashpot model:

$$F_t = min(k_t\delta_t + c_t\dot{\delta}_t, \mu F_n) \tag{2.33}$$

where δ_t is the tangential spring displacement or the tangential slip. It is defined as the total displacement in the tangential direction which took place since the beginning of the contact, as shown in fig. 2.16.

The tangential dissipation coefficient can be derived from the tangential stiffness analogous to the normal dissipation coefficient:

$$c_t = \ln e_t\sqrt{\frac{4m_{ij}k_t}{\pi^2 + \ln {e_t}^2}} \tag{2.34}$$

where e_t is the coefficient of tangential restitution. It is defined as the ratio between the tangential component of the relative velocity of the two contacting bodies after and before the impact:

$$e_t = \left|\frac{\dot{\delta}_{t,final}}{\dot{\delta}_{t,initial}}\right| \tag{2.35}$$

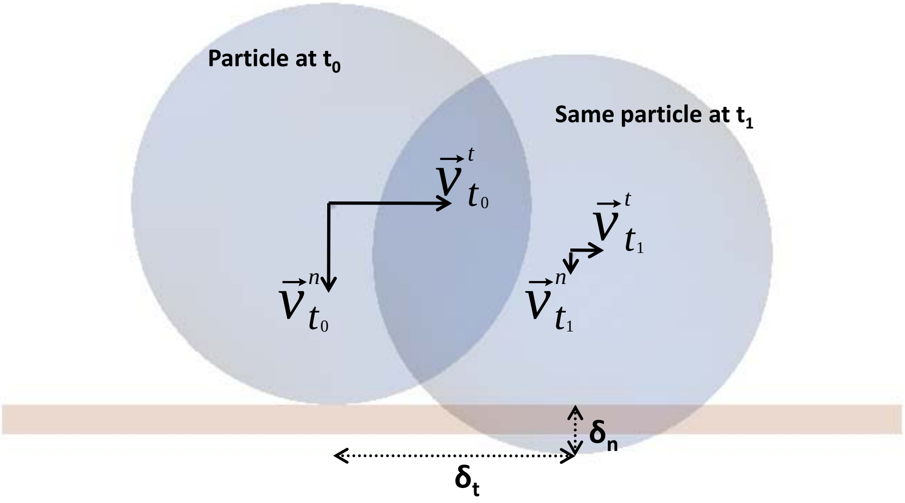


Figure 2.16: An oblique collision between a particle and a wall where the tangential slip δ_t is distinguished from the normal overlap δ_n. t_0 is the beginning of the collision while t_1 represents an arbitrary point in time during the collision.

2.6.3 Hooke's Law

The equation for Hooke's impact model can be written as

$$F_n = m_{ij}\ddot{\delta} = -(\frac{4}{3}E_{ij}R_{ij}\delta + c_n\dot{\delta}) \tag{2.36}$$

The general formula for Hooke's force model is in fact the same as the simple linear spring-dashpot. The only difference is that the spring stiffness constant in Hooke's model is calculated from the material properties and geometry, $k_n = \frac{4}{3}E_{ij}R_{ij}$. Other parameters such as the damping coefficient, c_n, the duration of contact and the magnitude of maximum overlap can be calculated based on this stiffness according to eqs. 2.21, 2.22 and 2.26.

The tangential force is modelled in the same way as the Linear-Spring-Dashpot impact model according to eq. 2.28.

2.6.4 Hertz-Brilliantov

The Hertz model is a non-linear force model based on the elastic solution derived by Hertz [40] for the contact between bodies and is given in eq. 2.37.

$$F_n = m_{ij}\ddot{\delta} = -\frac{4}{3}E_{ij}\sqrt{R_{ij}}(\delta^{\frac{3}{2}} + A_{ij}\delta^{\frac{1}{2}}\dot{\delta}) \tag{2.37}$$

where A_{ij} is the normal dissipation coefficient as a materiel constant. The dissipative term is based on the model proposed by Brilliantov et al. [12] and leads to a coefficient of restitution dependent on the impact velocity. Therefore, this force model models dissipation in a fundamentally different way than other models where the coefficient of restitution is assumed to be a material property. However, the challenge in this model is to assign a suitable normal dissipation coefficient A_{ij} for the materials chosen though some experimental approach to calculate A_{ij} is suggested by the authors [12, 11, 84].

The elastic repulsive force driven by spring displacement in Hooke and Hertz model can be written in one general equation:

$$F_{n,spring} = -\frac{4}{3}E_{ij}R_{ij}^{2-\alpha}\delta^{\alpha} \tag{2.38}$$

where $\alpha = 1$ in the Hooke's model and $\alpha = \frac{3}{2}$ in the Hertz model.

In case of no damping, an energy balance equation could be written between the start of the collision and an arbitrary point during the contact.

$$\frac{1}{2}m_{ij}v_0^2 = \frac{1}{2}m_{ij}v^2 + \frac{2}{5}k_n\delta^{\frac{5}{2}} \tag{2.39}$$

where $k_n = \frac{4}{3}E_{ij}\sqrt{R_{ij}}$ and v is the relative velocity at the specific time. The maximum overlap happens when the relative velocity v is zero and therefore can be derived from the energy balance equation:

$$\delta_{max} = 1.09(\frac{m_{ij}}{k_n})^{\frac{2}{5}}v_0^{\frac{4}{5}} \tag{2.40}$$

Fig. 2.17 compares the Hooke's and Hertz's impact models in terms of the effect of impact velocity on the maximum overlap. Generally larger overlaps are produced with the linear Hooke's model comparing with the non-linear Hertz's model.

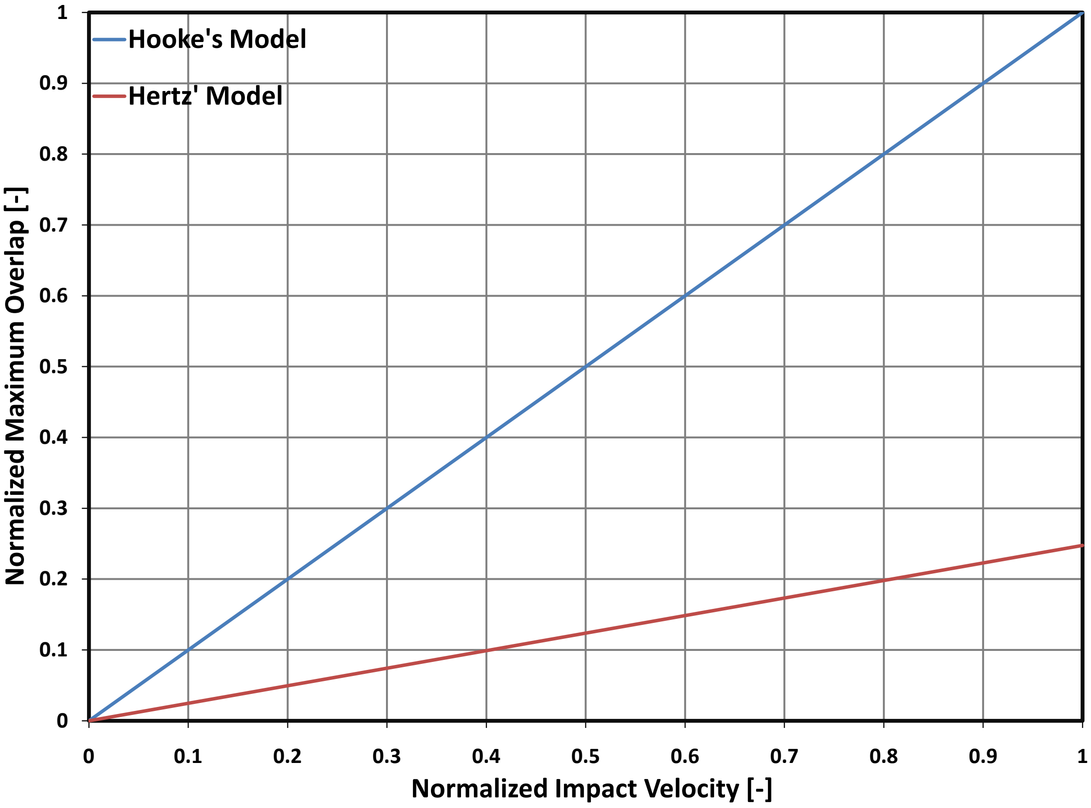


Figure 2.17: Effect of impact velocity on the maximum overlap, comparison of the Hooke's and Hertz' impact models

Fig. 2.18 illustrates the influence of the effective Young modulus on the maximum overlap in Hooke's and Hertz' impact models. Obviously the result of increasing the Young modulus which corresponds to a harder material is producing smaller deformation.

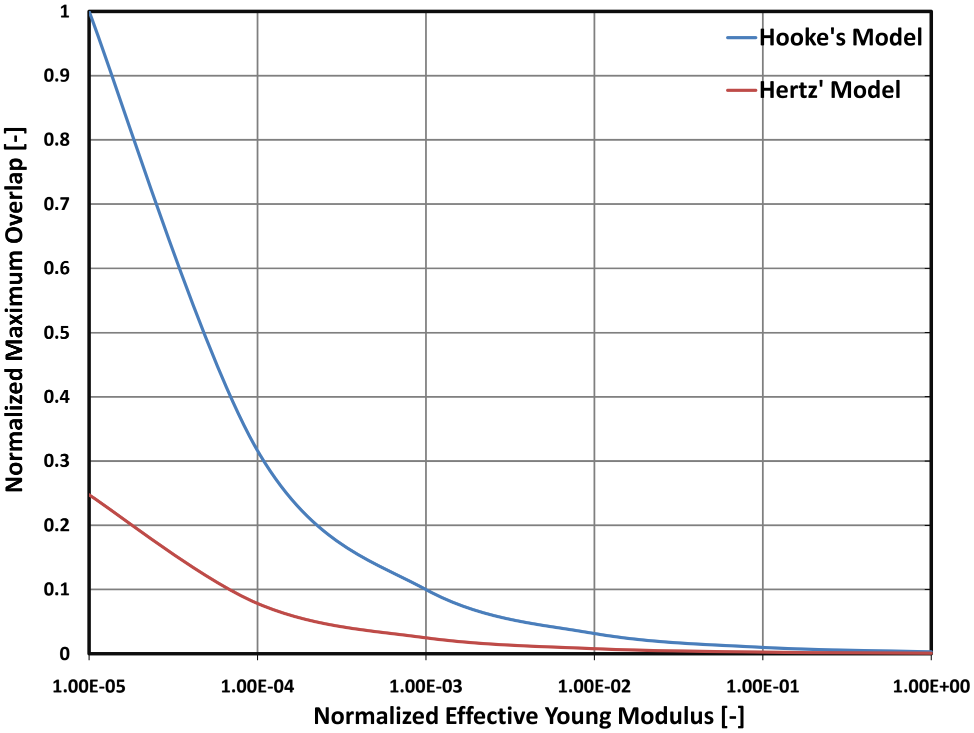


Figure 2.18: Effect of the effective young modulus on the maximum overlap, comparison of the Hooke's and Hertz' impact models

The total duration of contact can also be derived from the energy balance, eq. 2.39, in the absence of damping. This can be done by integrating the time of contact over the loading and unloading phases:

$$T_{contact} = 2\int_0^{\delta_{max}} \frac{\partial\delta}{v} = 2\int_0^{\delta_{max}} \frac{\partial\delta}{\sqrt{v_0^2 - \frac{4k_n}{5m_{ij}}\delta^{\frac{5}{2}}}} = 3.21(\frac{m_{ij}^2}{k_n^2 v_0})^{\frac{1}{5}} \tag{2.41}$$

The details of the integration can be found in [60]. Fig. 2.19 Compares the effect of the impact velocity on the duration of contact in Hooke's and Hertz' impact models. The Hertz model predicts the decrease of the duration of collision as the impact velocity increases which is in agreement with the experimental findings [97]. The Hooke's model prediction of constant duration of collision is, on the other hand, not consistent with the experimental results.

The influence of the effective Young modulus on the duration of collision is presented in fig. 2.20. As expected, under the same impact velocity the duration of collision is shorter for harder materials.

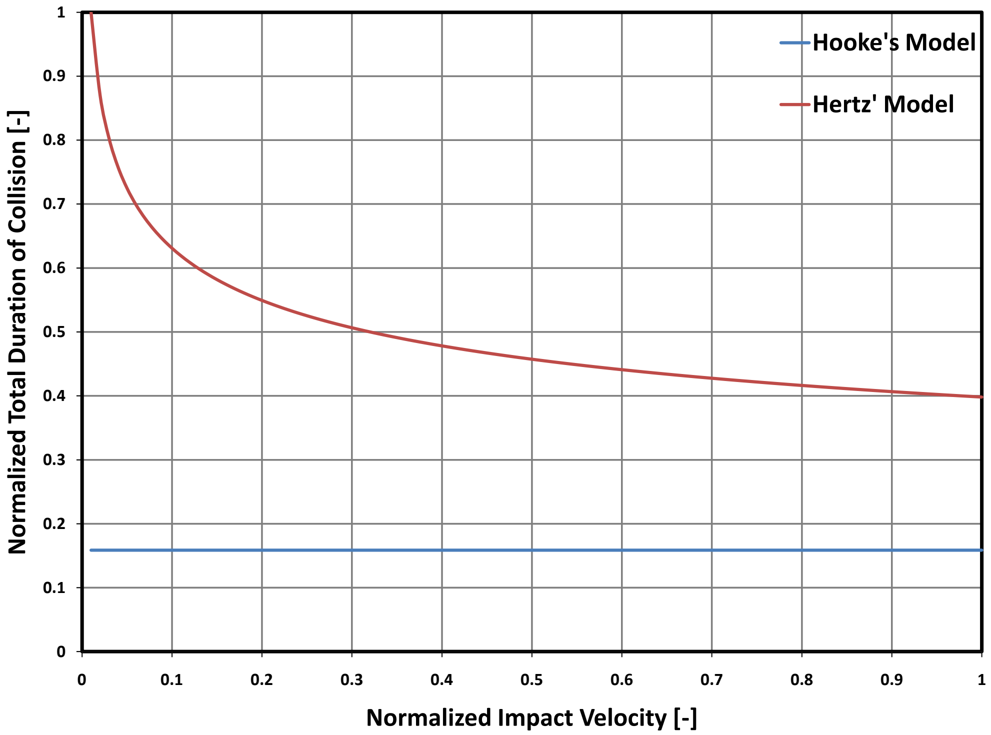


Figure 2.19: Effect of the impact velocity on the total duration of collision, comparison of the Hooke's and Hertz' impact models

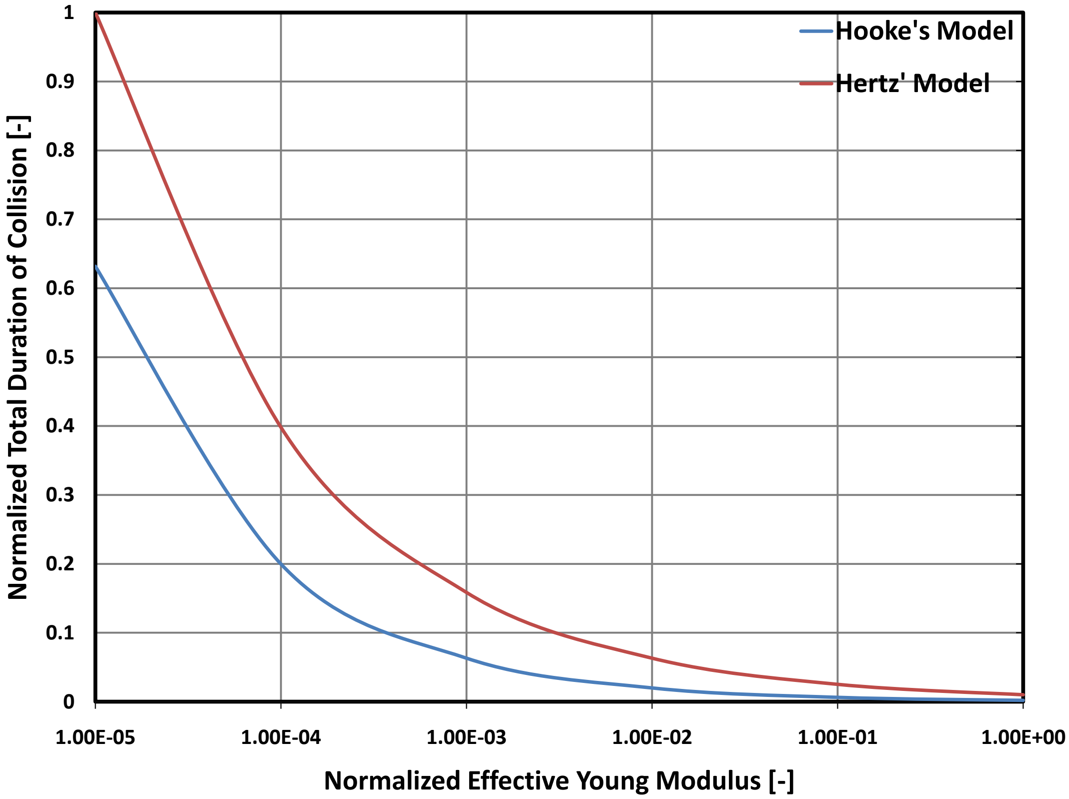


Figure 2.20: Effect of the effective Young modulus on the total duration of collision, comparison of the Hooke's and Hertz' impact models

The tangential force is modelled in the same way as the Linear-Spring-Dashpot impact model according to eq. 2.28.

2.6.5 Hertz-Mindlin

The Hertz-Mindlin model is based on the Hertz theory [40] to model the normal elastic force and the normal energy dissipation is based on the theory proposed by Mindlin [69]. The formula for the normal force in this model is:

$$F_n = m_{ij}\ddot{\delta} = -(\frac{4}{3}E_{ij}\sqrt{R_{ij}}\delta^{\frac{3}{2}} + c_n\delta^{\frac{1}{4}}\dot{\delta}) \tag{2.42}$$

The normal dissipation coefficient, c_n, is based on the model proposed by Tsuji et al., 1992 [105] and Zhang and Whitten, 1996 [115]:

$$c_n \quad = \quad \ln e\sqrt{\frac{5m_{ij}k_n}{\pi^2 + \ln e^2}} \quad \text{with} \quad k_n = \frac{4}{3}E_{ij}\sqrt{R_{ij}} \tag{2.43}$$

Since the normal elastic force in this model is the same as the Hertz model, eqs. 2.40 and 2.41 are also valid to predict the maximum overlap and the duration of contact in the absence of energy dissipation.

The tangential force is again a function of static and dynamic friction and the static friction in this model is based on a spring-dashpot model, eq. 2.33.

The tangential stiffness is estimated by eq. 2.44.

$$k_t \quad = \quad 8G_{ij}\sqrt{R_{ij}\delta} \tag{2.44}$$

where G_{ij} is the effective shear modulus defined in eq. 2.45.

$$\frac{1}{G_{ij}} \quad = \quad \frac{2-\nu_i}{G_i} + \frac{2-\nu_j}{G_j} \tag{2.45}$$

The tangential dissipation coefficient can be derived from the tangential stiffness using eq. 2.46.

$$c_t \quad = \quad \ln e\sqrt{\frac{5(4m_{ij}k_t)}{6(\pi^2 + \ln e^2)}} \tag{2.46}$$

2.6.6 Walton-Braun

Walton and Braun [110] developed a linear hysteretic model which accounts for the effect of plasticity. The deformed particles, in this model, do not return fully to their original shape by the end of the collision. In hysteretic models, different spring constants are used in loading and unloading phases as demonstrated in fig. 2.21.

The Walton-Braun force model could be written as follows,

$$F_n = m_{ij}\ddot{\delta} = \begin{cases} -k_{n,l}\delta & \dot{\delta} \geq 0 \\ -k_{n,ul}(\delta - \delta_0) & (\dot{\delta} < 0 \text{ and } \delta \geq \delta_0) \end{cases} \tag{2.47}$$

where $k_{n,l}$ and $k_{n,ul}$ are the normal loading and unloading stiffness constants and δ_0 is the value of the overlap when the unloading curve approaches zero due to plastic deformation. This model does

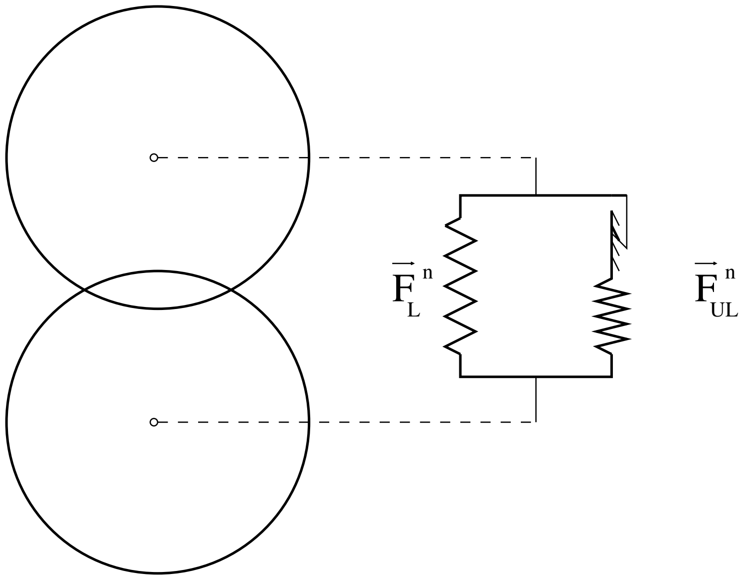


Figure 2.21: Visualisation of Walton-Braun impact model

not require inclusion of a velocity dependant damping term because damping is already applied by separating the loading and unloading phases. Due to its simplicity, the Walton-Braun model has an analytical solution. Considering the boundary conditions $\delta(0) = 0$, $\dot{\delta}(0) = v_0$ for the loading phase and $\delta(0) = v_0\sqrt{\frac{m_{ij}}{k_{n,l}}}$ and $\dot{\delta}(0) = 0$ for the unloading phase, the analytical solution can be written as:

$$\delta(t) = \begin{cases} v_0\sqrt{\frac{m_{ij}}{k_{n,l}}}sin(\sqrt{\frac{k_{n,l}}{m_{ij}}}t) & \dot{\delta} \geq 0 \\ (v_0\sqrt{\frac{m_{ij}}{k_{n,l}}} - \delta_0)cos(\sqrt{\frac{k_{n,ul}}{m_{ij}}}t) + \delta_0 & (\dot{\delta} < 0 \text{ and } \delta \geq \delta_0) \end{cases} \tag{2.48}$$

The maximum overlap when the relative velocity is zero, $\dot{\delta}(t) = 0$, can be easily calculated from the loading phase:

$$\delta_{max} = v_0\sqrt{\frac{m_{ij}}{k_{n,l}}} \tag{2.49}$$

The loading and unloading forces are equal at maximum overlap when the loading of the spring completes and the unloading starts which could be used to calculate the magnitude of the plastic deformation:

$$\delta_0 = \frac{k_{n,ul} - k_{n,l}}{k_{n,ul}}\delta_{max} \tag{2.50}$$

which shows that the plastic deformation is only a fraction of the maximum overlap experienced by the particles. The coefficient of restitution can also be derived from the analytical solution, eq. 2.48:

$$e = \left|\frac{\dot{\delta}_{final}}{\dot{\delta}_{initial}}\right| = \sqrt{\frac{k_{n,l}}{k_{n,ul}}} \tag{2.51}$$

The total duration of collision is the sum of the loading and unloading durations

$$T_{contact} = \frac{\pi}{2}\sqrt{\frac{m_{ij}}{k_{n,l}}} + \frac{\pi}{2}\sqrt{\frac{m_{ij}}{k_{n,ul}}} = \frac{\pi}{2}\sqrt{\frac{m_{ij}}{k_{n,ul}}}(\frac{e+1}{e}) \tag{2.52}$$

In the absence of energy dissipation, the Walton-Braun model is obviously reduced to a linear spring force model. The basic difference between the Walton-Braun model and linear spring-dashpot model is on the way they model energy dissipation in the normal direction of contact. Fig. 2.22 plots the overlap in both models and compares the maximum overlap, the total duration of contact and the duration of the loading phase.

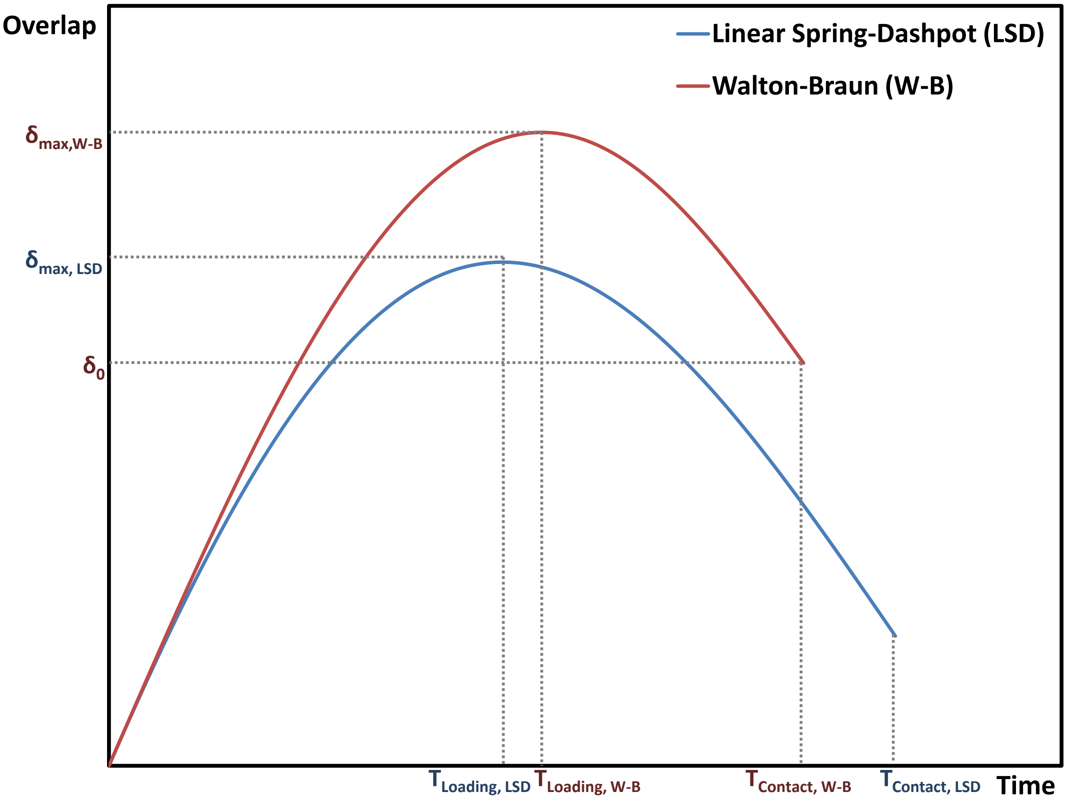


Figure 2.22: Overlap vs. time, comparison of Linear Spring-Dashpot and Walton-Braun impact models

The tangential force is modelled in the same way as the Linear-Spring-Dashpot impact model according to eq. 2.28.

2.6.7 Rolling Friction

In addition to the sliding friction in the tangential direction based on the Coulomb friction theory, a rolling friction or rolling resistance torque is also included in DPM for all force models as follows:

$$\vec{T}_{rolling-friction} = -\mu_r F_n R_i \vec{\omega}_i \tag{2.53}$$

where μ_r is the rolling friction coefficient which is much smaller than the sliding friction coefficient. The direction of this torque is in the opposite direction of the angular velocity which means it resists rotation. For more detailed review and discussions of the rolling friction, one may refer to the works by Tabor, 1955 [101] and Zheng et al. [116].

2.6.8 Summary of the Presented Impact Models

It should be emphasized that there are several more impact models in the literature and it is not the objective nor the claim of this section to present a complete review of all existing models. The impact models introduced in this section are the ones available in the DPM code. They are summarized in Table 2.1.

Table 2.1: Impact models

	Normal force	Tangential force
Linear models		
linear spring-dashpot	$F_n = -(k_n\delta_n + c_n\dot{\delta}_n)$	$F_t = min(c_t\dot{\delta}_t, \mu F_n)$
Maw linear spring-dashpot	$F_n = -(\frac{16}{15}E_{ij}\sqrt{R_{ij}}(\frac{15m_{ij}V^2}{16E_{ij}\sqrt{R_{ij}}})^{\frac{1}{5}}\delta_n + c_n\dot{\delta}_n)$	$F_t = min(k_t\delta_t + c_t\dot{\delta}_t, \mu F_n)$
Hooke	$F_n = -(\frac{4}{3}E_{ij}\sqrt{R_{ij}}\delta_n + c_n\dot{\delta}_n)$	$F_t = min(c_t\dot{\delta}_t, \mu F_n)$
Non-linear models		
Hertz-Brilliantov	$F_n = -\frac{4}{3}E_{ij}\sqrt{R_{ij}}(\delta_n^{\frac{3}{2}} + A_{ij}\delta_n^{\frac{1}{2}}\dot{\delta}_n)$	$F_t = min(c_t\dot{\delta}_t, \mu F_n)$
Hertz-Mindlin	$F_n = -(\frac{4}{3}E_{ij}\sqrt{R_{ij}}\delta_n^{\frac{3}{2}} + c_n\delta_n^{\frac{1}{4}}\dot{\delta}_n)$	$F_t = min(k_t\delta_t + c_t\delta_n^{\frac{1}{4}}\dot{\delta}_t, \mu F_n)$
Hysteretic models		
Walton-Braun	$F_n = -\begin{cases} k_{n,l}\delta & \dot{\delta} \geq 0 \\ k_{n,ul}(\delta - \delta_0) & \dot{\delta} < 0 \end{cases}$	$F_t = min(c_t\dot{\delta}_t, \mu F_n)$

2.7 Integration Schemes

Following the calculation of the forces and subsequently the accelerations based on the Newton's second law, the velocities and positions can be computed by numerical integration. The general sense of this step can be expressed as follows:

$$\ddot{r} \longrightarrow \dot{r} \longrightarrow r$$

Different algorithms are available in the code to perform the integration of the particles accelerations which are presented in this section. All the presented schemes can be derived by approximating the derivatives of a function by Taylor series:

$$f(t+\Delta t) \quad = \quad f(t) + \frac{f'(t)}{1!}\Delta t + \frac{f''(t)}{2!}\Delta t^2 + \frac{f^{(3)}(t)}{3!}\Delta t^3 + ... \qquad (2.54)$$

The description in this section is geared towards solving the Newton's Equations of motion and specifically the examples are given for integration of the elastic spring force. For more review and discussion on different integration schemes in molecular dynamics and discrete element method, one may refer to Press et al., 1988 [85], Allen et al., 1987 [1], Džiugys et al., 2001 [118], van Gunsteren et al., 1977 [107], Satoh, 1995 [90] and 1997 [91], Tuley et al., 2009 [106], Fraige et al., 2004 [32] and Rougier et al., 2004 [88].

The graphs presented in this section are based on normalized values in order to represent general trends independent of any specific case. The values are normalized over the maximum point of the analytical solution in the absence of dissipation in order to allow a meaningful comparison.

2.7.1 Symplectic Euler

A first degree Taylor approximation can be used to derive the first derivative of the function. This will lead to the equation for forward difference approximation:

$$f'(t) \quad = \quad \frac{f(t+\Delta t) - f(t)}{\Delta t} \qquad (2.55)$$

The standard Euler method is based on the forward difference approximation for the first derivative of a function. So the velocity at the new time step can be written as follows:

$$\vec{v}_{t+1} \quad = \quad \vec{v}_t + \vec{a}_t \Delta t \qquad (2.56)$$

If the Taylor series is written for the function at the previous time step, $f(t-\Delta t)$, the backward difference approximation for the first derivative is derived:

$$f'(t) \quad = \quad \frac{f(t) - f(t-\Delta t)}{\Delta t} \qquad (2.57)$$

The symplectic Euler method is different from the standard Euler in calculation of the position because it uses the backward difference approximation. The new position is then can be written as:

$$\vec{r}_{t+1} \quad = \quad \vec{r}_t + \vec{v}_{t+1} \Delta t \qquad (2.58)$$

This is a modification to standard Euler formula where only forward difference approximation is used for both velocity and position [37].

2.7.2 Taylor

In this method, the Taylor polynomial of degree one is used to calculate the velocity:

$$\vec{v}_{t+1} \quad = \quad \vec{v}_t + \vec{a}_t \Delta t \qquad (2.59)$$

The position is derived from the Taylor polynomial of degree two:

$$\vec{r}_{t+1} = \vec{r}_t + \vec{v}_t \Delta t + \frac{1}{2}\vec{a}_t \Delta t^2 \tag{2.60}$$

This method can be derived from constant acceleration motion, too. In other words, this integration method assumes the acceleration during the time step to be equal to the acceleration at the beginning of that time step.

2.7.3 Position Verlet

The basic Verlet method relies on the central difference approximation, in contrast to the forward or backward approximations of a function's derivatives. The central difference approximation for the first derivative can be derived from the Taylor polynomial of degree two written for the two opposite time directions as follows:

$$f(t+\Delta t) = f(t) + \Delta t f'(t) + \frac{1}{2}\Delta t^2 f''(t) \tag{2.61}$$

$$f(t-\Delta t) = f(t) - \Delta t f'(t) + \frac{1}{2}\Delta t^2 f''(t) \tag{2.62}$$

Subtracting eq. 2.62 from eq. 2.61 gives the central difference approximation for the first derivative of a function:

$$f'(t) = \frac{f(t+\Delta t) - f(t-\Delta t)}{2\Delta t} \tag{2.63}$$

The second derivative can be derived from the Taylor polynomial of degree three written for the two opposite time directions as follows:

$$f(t+\Delta t) = f(t) + \Delta t f'(t) + \frac{1}{2}\Delta t^2 f''(t) + \frac{1}{6}\Delta t^3 f^{(3)}(t) \tag{2.64}$$

$$f(t-\Delta t) = f(t) - \Delta t f'(t) + \frac{1}{2}\Delta t^2 f''(t) - \frac{1}{6}\Delta t^3 f^{(3)}(t) \tag{2.65}$$

Summing up the two Taylor polynomials gives the central difference formula for the second derivative of the function:

$$f''(t) = \frac{f(t+\Delta t) - 2f(t) + f(t-\Delta t)}{\Delta t^2} \tag{2.66}$$

The position Verlet method uses eq. 2.66 to compute the position:

$$\vec{r}_{t+1} = 2\vec{r}_t - \vec{r}_{t-1} + \vec{a}_t \Delta t^2 \tag{2.67}$$

The velocity, based on eq. 2.63, can be written as:

$$\vec{v}_t = \frac{\vec{r}_{t+1} - \vec{r}_{t-1}}{2\Delta t} \tag{2.68}$$

However, the velocity in eq. 2.68 is one step behind the position. The solution to this problem is discussed in section 2.7.4.

The acceleration is assumed constant during a time step in this method but the value at the middle of the time step is used to approximate the constant acceleration rather than the beginning or end of the time step in forward or backward approximations. Fig. 2.23 illustrates the forward, backward and central difference approximations of the new velocity $v_{t+\Delta t}$ during loading of a linear spring. As shown in the figure, the central difference approximation is generally more accurate than the forward and backward approximations. The backward difference approximation is more accurate than the forward difference in the loading phase as shown in fig. 2.23. On the contrary, the forward difference is a better approximation than the backward difference in the unloading phase when the acceleration is declining. The position verlet method is a second order integrator, it is time reversible and is generally more stable than the Euler method [76].

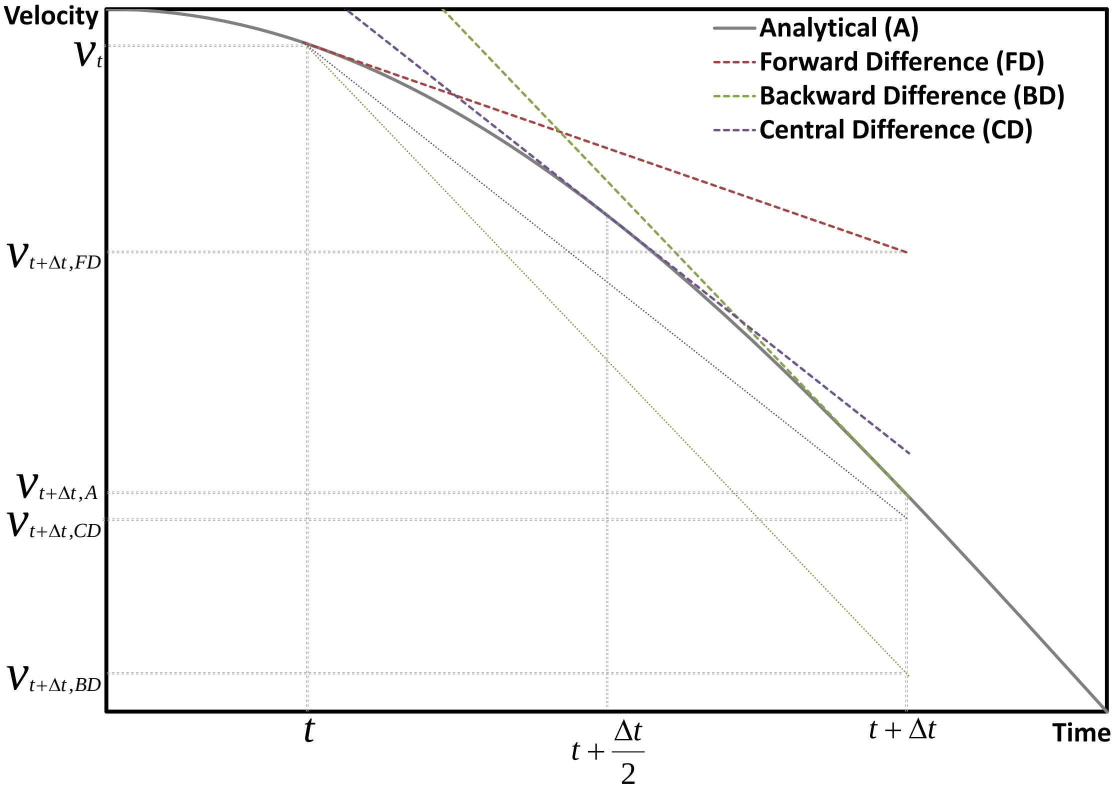


Figure 2.23: Forward, backward and central difference approximation applied to the velocity curve during the loading phase of a linear spring.

2.7.4 Velocity Verlet

As noted above, the basic position Verlet method does not provide velocities at the next time step. To overcome this problem, a modification to the basic Verlet method is widely used [37]. The position in the velocity Verlet method is computed based on the central difference approximation similarly to the basic Verlet method. Replacing $\vec{r}_{t-1}$ from eq. 2.68 into eq. 2.67, the position is derived as:

$$\vec{r}_{t+1} = \vec{r}_t + \vec{v}_t\Delta t + \frac{1}{2}\vec{a}_t\Delta t^2 \tag{2.69}$$

The velocity is approximated as follows:

$$\vec{v}_{t+1} = \vec{v}_t + \frac{1}{2}(\vec{a}_t + \vec{a}_{t+1})\Delta t \tag{2.70}$$

To implement this, a predictor-corrector scheme is developed. The predictor, calculates the position from previous values according to eq. 2.69 and predicts the velocity as:

$$\vec{v}_{t+1}^{\,pr} = \vec{v}_t + \frac{1}{2}\vec{a}_t\Delta t \tag{2.71}$$

At the next step, the force calculation function is called and accordingly the new acceleration is calculated. The corrector, then, corrects the new velocity according to eq. 2.70. This implementation does not require more memory storage than the basic Verlet method but the calculation of the new acceleration is based on the predicted velocity value instead of the corrected value.

2.7.5 Gear

The Gear algorithm is developed by Gear [34, 33] and is known for its numerical stability. The Gear algorithm is implemented in two steps: predictor and corrector.

At the first step the Gear-predictor predicts the new position, velocity, acceleration by extrapolating the current values using the Taylor expansion. The degree of the Taylor polynomial depends on the order of the Gear algorithm. For the forth order Gear scheme, the time derivatives up to $\frac{\partial^3 \vec{r}_t}{\partial t^3}$ are required:

$$\vec{r}_{t+1}^{\,pr} = \vec{r}_t + \vec{v}_t\Delta t + \frac{1}{2}\vec{a}_t\Delta t^2 + \frac{1}{6}\frac{\partial^3 \vec{r}_t}{\partial t^3}\Delta t^3 \tag{2.72}$$

$$\vec{v}_{t+1}^{\,pr} = \vec{v}_t + \vec{a}_t\Delta t + \frac{1}{2}\frac{\partial^3 \vec{r}_t}{\partial t^3}\Delta t^2 \tag{2.73}$$

$$\vec{a}_{t+1}^{\,pr} = \vec{a}_t + \frac{\partial^3 \vec{r}_t}{\partial t^3}\Delta t \tag{2.74}$$

At the next step, the force calculation function is called and accordingly the new acceleration based on the predicted values for positions and velocities is estimated. Then the Gear-corrector corrects the positions, velocities and accelerations based on the difference between the acceleration estimate and the acceleration prediction: $\Delta\vec{a} = \vec{a}^{\,est} - \vec{a}^{\,pr}$. The correction equations for the forth order Gear are shown in eqs. 2.75 to 2.78.

$$\vec{r}_{t+1}^{\,corr} = \vec{r}_{t+1}^{\,pr} + \frac{1}{12}\Delta t^2\Delta\vec{a} \tag{2.75}$$

$$\vec{v}_{t+1}^{\,corr} = \vec{v}_{t+1}^{\,pr} + \frac{5}{12}\Delta t\Delta\vec{a} \tag{2.76}$$

$$\vec{a}_{t+1}^{\,corr} = \vec{a}_{t+1}^{\,pr} + \Delta\vec{a} \tag{2.77}$$

$$\frac{\partial^3 \vec{r}_{t+1}}{\partial t^3}^{corr} = \frac{\partial^3 \vec{r}_{t+1}}{\partial t^3}^{pr} + \frac{\Delta\vec{a}}{\Delta t} \tag{2.78}$$

2.7.6 Discussion of the Presented Integration Schemes

The response of the discussed integration methods to solve the Newton's equation of motion are compared in this section. The motion of particles can be studied in three different situations:

- during the free motion when gravity is the only constant force applied on the particles
- during a collision with another object
- during the interface between the two situations above when the commencement or termination of the contact is first detected

The performance of different integration schemes is not much different during the free motion of particles. The challenging cases are the two last situations. The third case will be discussed in depth in chapter 3. This section utilizes different integration methods to solve the Newton's equations of motion during a normal collision of a particle with a wall. The impact model chosen is the linear spring-dashpot model where the analytical solution is available to validate the results.

Accuracy of any integration scheme obviously depends on the size of the utilized time step. Therefore, in order to carry out a fair comparison, different integration schemes need to be compared using the same time step. The time step chosen here is four times smaller than the total duration of contact. In other words, the collision is resolved in four discrete time steps. Contact resolution is defined as follows:

$$CR = \frac{T_{contact}}{\Delta t} \tag{2.79}$$

The position Verlet method needs the position at two previous time steps in order to predict the next position. Therefore and again for a fair comparison, the initial positions for the first two time steps are provided to all the integration schemes.

Initially, energy dissipation is excluded and only the elastic repulsive force is considered. Fig. 2.24 shows how different integration methods predict the velocity of a particle during a collision with a wall in the absence of energy dissipation. Similarly, fig. 2.25 shows the prediction of the position of the particle by different integration methods.

The results show that the Taylor method's prediction of position and velocity is less accurate than the other methods. This is expected as the Taylor method is a first order integrator. The Symplectic Euler method is also a first order integrator but it is more accurate because it is a modified version of the standard Euler formula using both backward and forward approximations. The position Verlet is predicting the position in the same manner as the velocity Verlet but does not produce a good prediction of the velocity, as expected. Considering both velocity and position, the velocity Verlet and the Gear integration schemes provide quite accurate predictions in the absence of energy dissipation.

Next, energy dissipation in the normal direction of contact is taken into account as a velocity dependent dashpot force. The whole contact is again solved in four discrete time steps. Figs. 2.26 and 2.27 show the velocity and position's predictions respectively when the coefficient of restitution is 0.5.

The results show that by adding dissipation, the integration methods deviate from the analytical solution. This is because the dissipation term in eq. 2.15 increases the loading force while the change to the unloading force becomes more sharp. Fig. 2.26 shows that in all integration schemes, the first prediction of the velocity after impact and also the magnitude of the velocity at the end of the contact are over-estimated. The over-estimation is less in case of the velocity Verlet and Gear

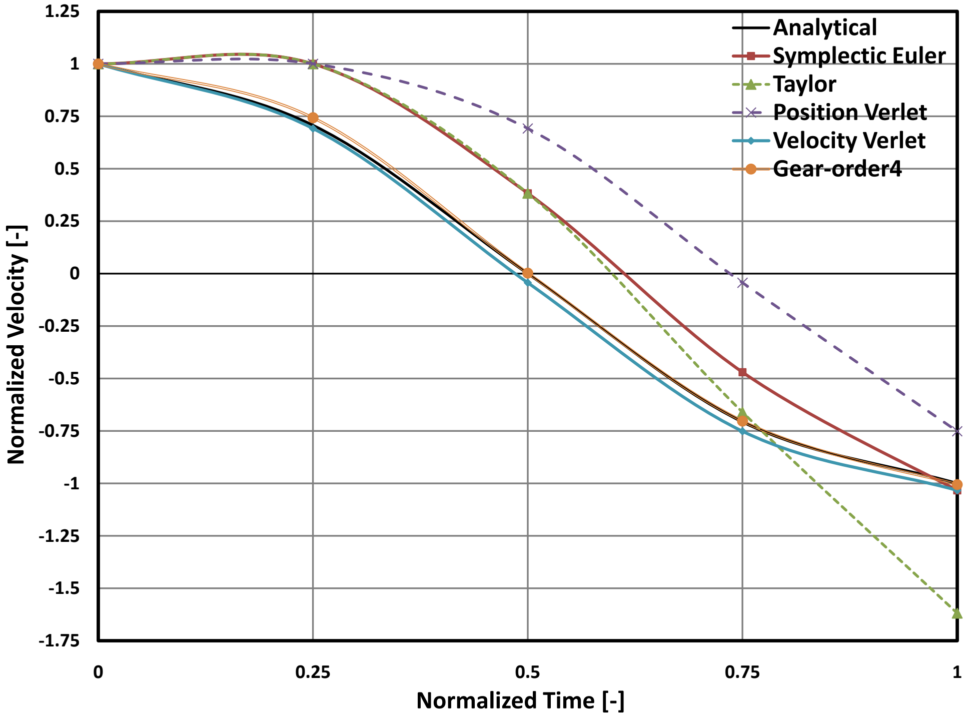


Figure 2.24: Prediction of the velocity of a particle during a collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 1$

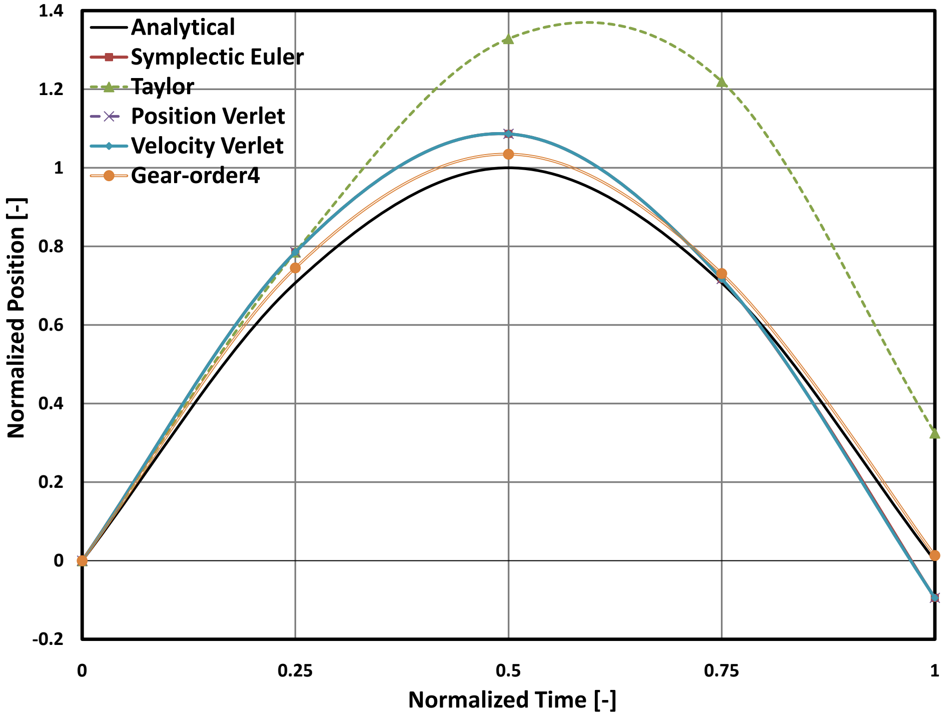


Figure 2.25: Prediction of the position of a particle during collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 1$

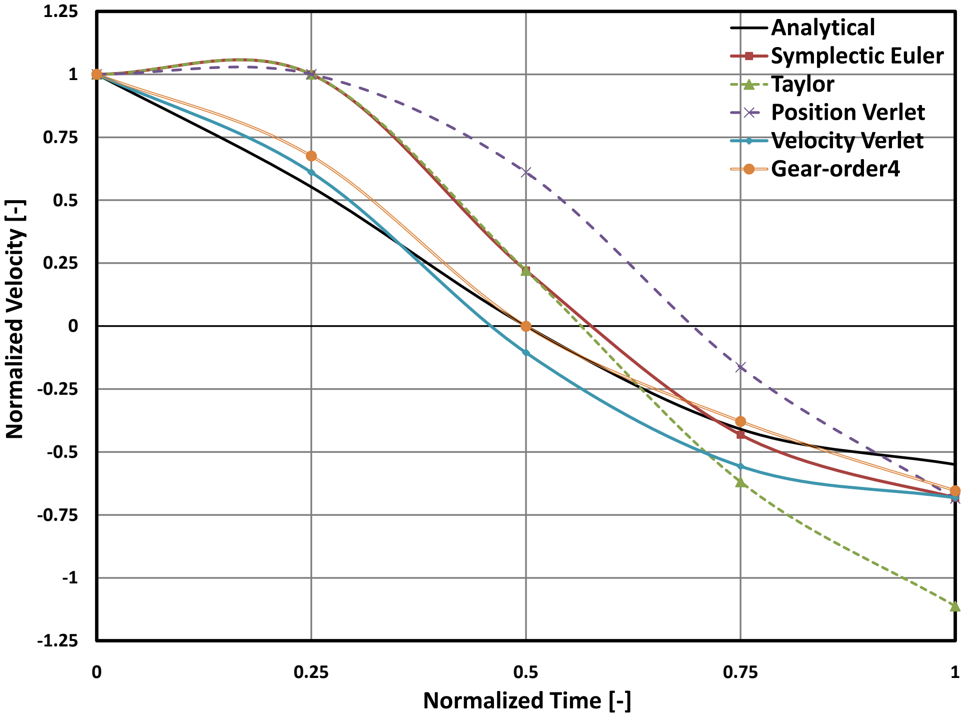


Figure 2.26: Prediction of the velocity of a particle during collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 0.5$

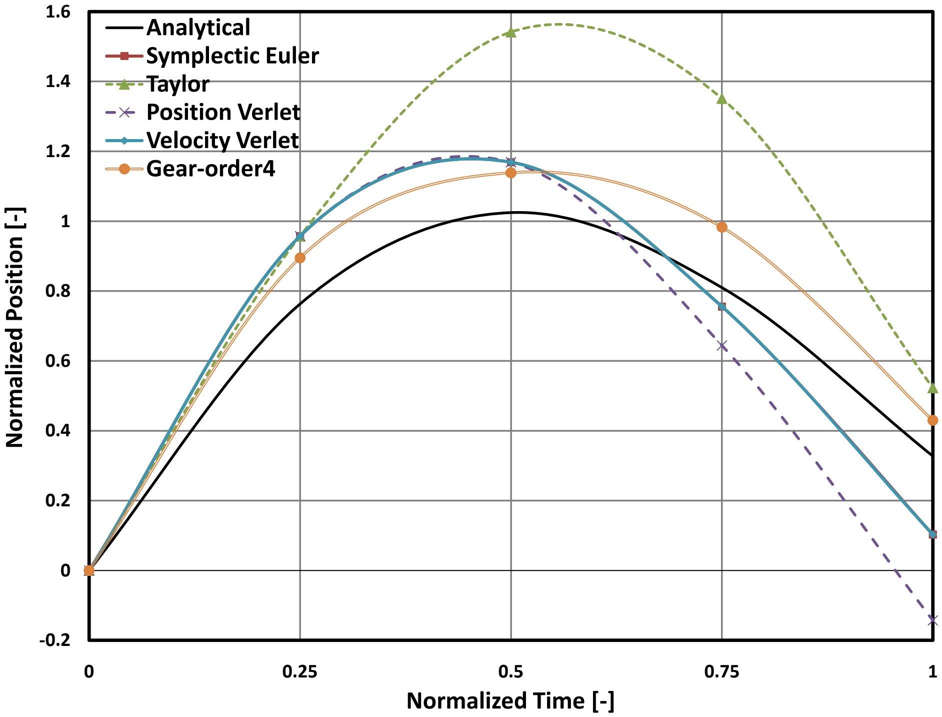


Figure 2.27: Prediction of the position of a particle during collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 0.5$

integration schemes because they utilize a predictor-corrector approach to calculate the velocity which is a semi-implicit approach. Fig. 2.27 also shows that the presented integration schemes exaggerate in predicting the initial position after contact and consequently the maximum overlap. This is because an explicit integration scheme detects a collision at most one time step after it happens. Contact detection with explicit and implicit methods is the subject of chapter 3 where a fully implicit algorithm is presented and elaborately discussed.

Admittedly, the deviations will diminish if smaller time steps are utilized but the relatively large time step chosen here is to magnify and analyse the differences. Such deviations will still exist with smaller time steps but to a less degree, as shown in figs. 2.28 to 2.31 which illustrate the same plots of figs. 2.24 to 2.27 but with the contact resolution of 8. The presented integration schemes are summarized in table. 2.2.

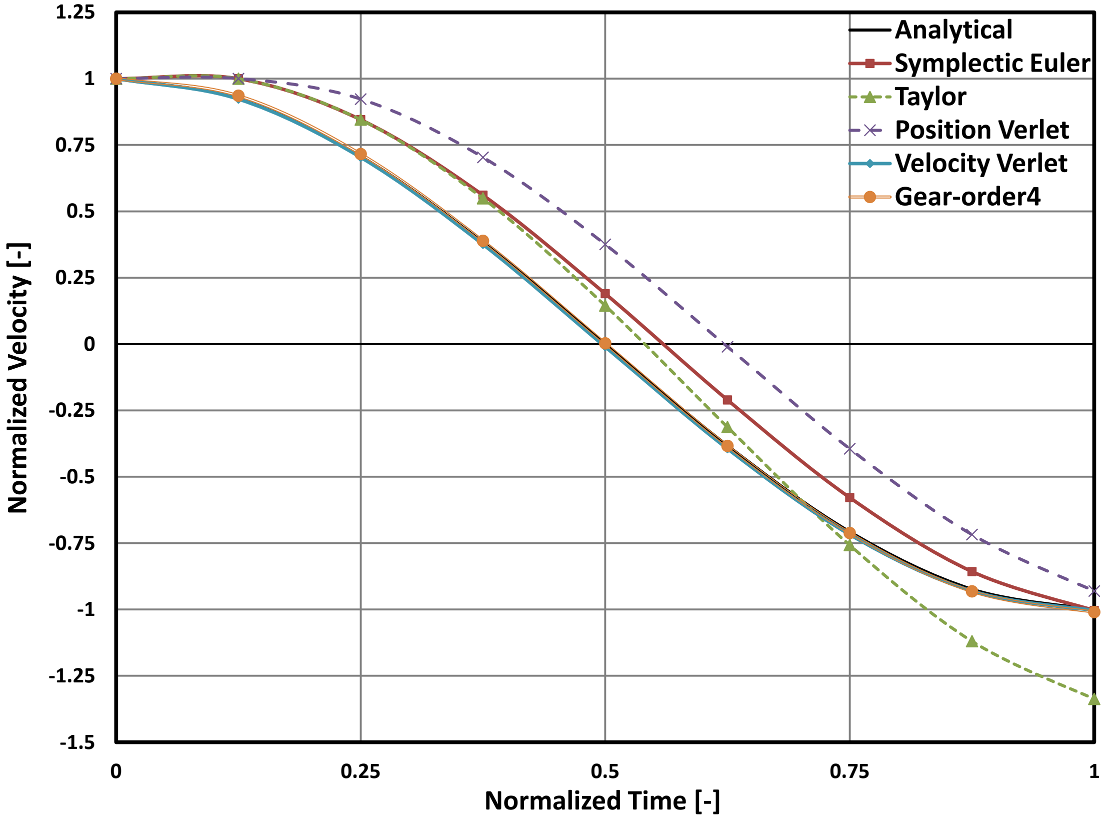


Figure 2.28: Prediction of the velocity of a particle during collision with a wall, contact-resolution= 8, $e = 1$

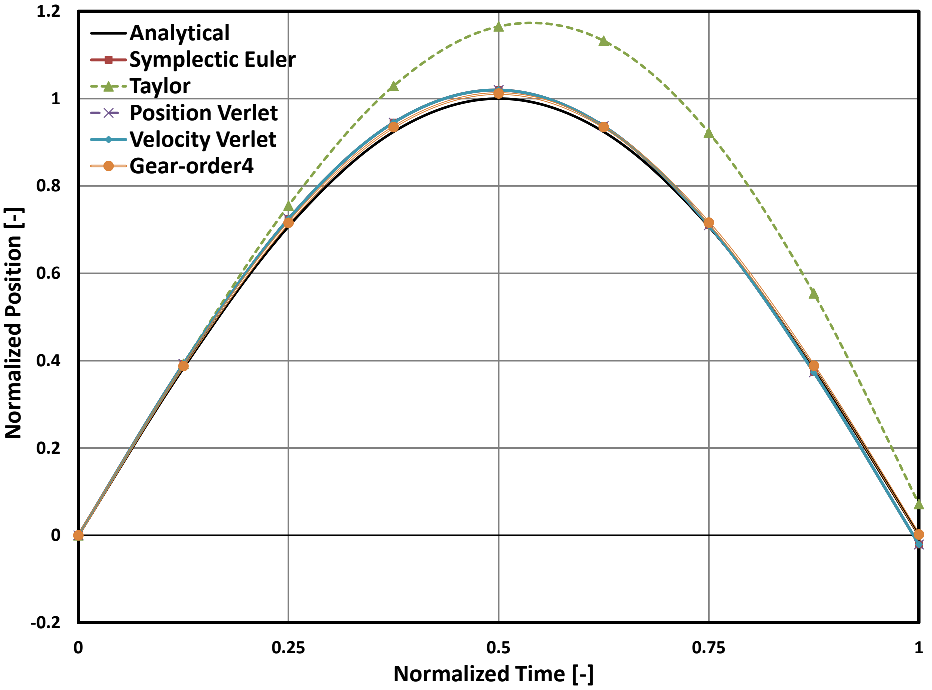


Figure 2.29: Prediction of the position of a particle during collision with a wall, contact-resolution= 8, $e = 1$

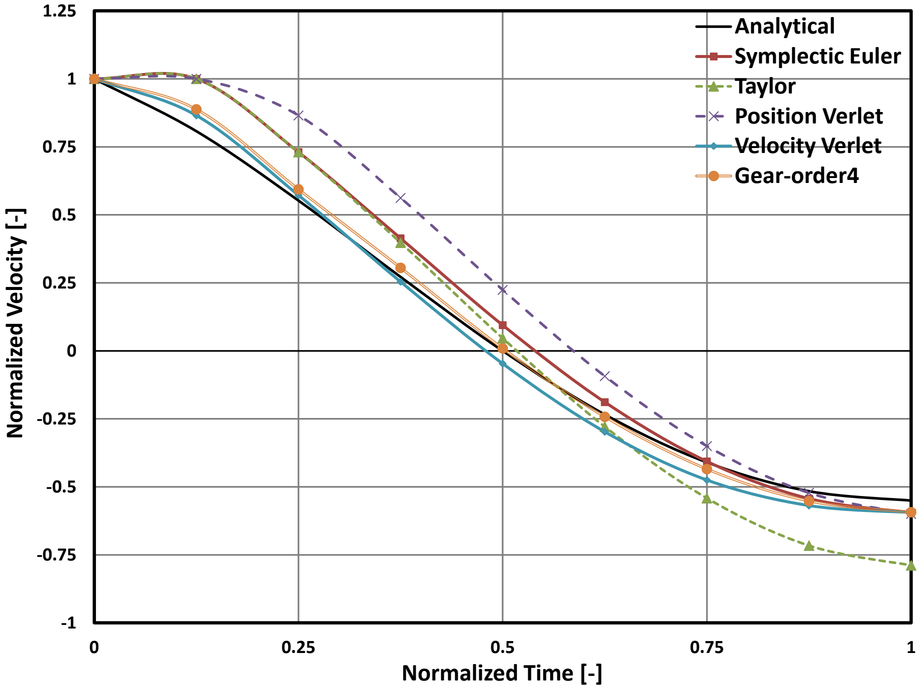


Figure 2.30: Prediction of the velocity of a particle during collision with a wall, contact-resolution= 8, $e = 0.5$

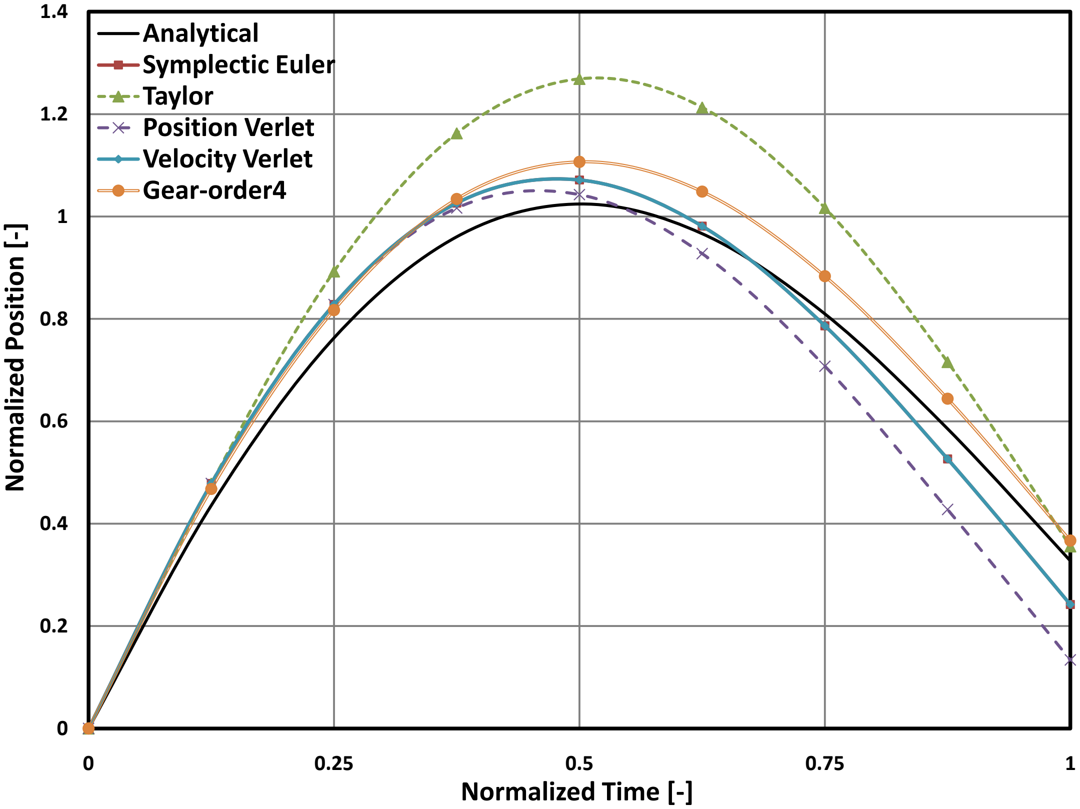


Figure 2.31: Prediction of the position of a particle during collision with a wall, contact-resolution= 8, $e = 0.5$

Table 2.2: Integration schemes

Integration method	**Velocity, $\vec{v}_{t+1}$**	**Position, $\vec{r}_{t+1}$**
Symplectic Euler	$\vec{v}_t + \Delta t \vec{a}_t$	$\vec{r}_t + \Delta t \vec{v}_{t+1}$
Taylor	$\vec{v}_t + \Delta t \vec{a}_t$	$\vec{r}_t + \Delta t \vec{v}_t + \frac{\Delta t^2}{2} \vec{a}_t$
Position Verlet	$\frac{\vec{r}_{t+1} - \vec{r}_{t-1}}{2\Delta t}$	$2\vec{r}_t - \vec{r}_{t-1} + \Delta t^2 \vec{a}_t$
Velocity Verlet	$\vec{v}_t + \frac{\Delta t}{2}(\vec{a}_t + \vec{a}_{t+1})$	$\vec{r}_t + \Delta t \vec{v}_t + \frac{\Delta t^2}{2} \vec{a}_t$
Gear 4th order	$\vec{v}_t + \frac{7\Delta t}{12}\vec{a}_t + \frac{5\Delta t}{12}\vec{a}_{t+1} + \frac{11\Delta t^2}{12}\frac{\partial^3 \vec{r}_t}{\partial t^3}$	$\vec{r}_t + \vec{v}_t \Delta t + \frac{5\Delta t^2}{12}\vec{a}_t + \frac{\Delta t^2}{12}\vec{a}_{t+1} + \frac{\Delta t^3}{12}\frac{\partial^3 \vec{r}_t}{\partial t^3}$

2.8 Methods to Improve Computational Efficiency

One major obstacle in applying the DEM to real world problems is the efficiency of the simulations. One problem is that DEM is only a valid approach if very small time steps are utilized. It will be shown in chapter 3 that applying an implicit algorithm enables larger time steps to be employed but the size of the time step even in the implicit integration is limited too. The other obstacle in DEM is that the number of particles involved could be huge reaching millions or even billions of particles in some applications. There are different approaches to increase the efficiency of the computational efforts which are discussed in this section.

2.8.1 Link Cell Algorithm

One of the computationally costly part of the DEM is detecting collision partners. At each time step, all the collision partners for each particle need to be identified. To identify the collision partners of a certain particle, one obvious approach would be to loop over all other particles in the system and check collision based on their distance from the particle. This requires $n(n-1)/2$ evaluations of overlap at each time step with n being the total number of particles in the domain. This is a very inefficient method especially for large systems. It is not necessary to loop over all other particles because the potential collision partners of a particle are the ones in the neighbourhood of the particle. Therefore, it would be sufficient to check collisions with these potential partners. This can reduce contact evaluations significantly.

In order to implement this idea, different methods are proposed and discussed by Walizer et al., 2011 [108]. The method employed in DPM is the Link cell algorithm. This method is chosen because of its compatibility with the idea of parallelization. The basic principle in this method is to divide the domain into a number of uniform rectangular cells. Each cell is assigned an index which represents its location in the domain. The cell size should be defined large enough to accommodate at least one particle:

$$d_{cell} \geq d_{particle} \tag{2.80}$$

where d_{cell} is the minimum length of the cell and $d_{particle}$ is the maximum diameter of a particle in the domain.

Once the space is divided into cells, the next step is mapping of the particles onto the cells. Given a simulation space starting from $(x_{min},\, y_{min},\, z_{min})$ to $(x_{max},\, y_{max},\, z_{max})$, the number of cells n_x , n_y and n_z in each direction is calculated as follows:

$$n_x = \frac{x_{max} - x_{min}}{d} \tag{2.81}$$

$$n_y = \frac{y_{max} - y_{min}}{d} \tag{2.82}$$

$$n_z = \frac{z_{max} - z_{min}}{d} \tag{2.83}$$

Each particle is assigned a unique integer number and each cell is represented by three integer numbers $(i,\, j,\, k)$ corresponding to the index of the cell. Each particle in the system can be mapped only to one cell. The mapping of particles onto cells is represented by a linked-list associated with each cell. If the centroid of a particle is within a cell, it is included into the list of this cell. For a particle with position $(x,\, y,\, z)$, its corresponding cell $(i,\, j,\, k)$ is calculated using eq. 2.84.

$$i \;=\; int\left(\frac{x - x_{min}}{d}\right)$$

$$j = int\left(\frac{y - y_{min}}{d}\right) \tag{2.84}$$

$$k = int\left(\frac{z - z_{min}}{d}\right)$$

To detect collision partners of a certain particle, it would be sufficient to loop over other particles in the same cell and the immediate neighbour cells. Each cell, being a rectangle, has twenty seven immediate neighbour cells in space except the cells on the border of the domain which have less.

2.8.2 Parallelization

In order to increase the efficiency, parallelisation is increasingly applied. The idea of software parallelisation is based on the fact that the process of solving a problem can be usually divided into smaller parts, which may be executed simultaneously with appropriate co-ordination. The performance and efficiency of the parallel software mainly depends on the way the problem is partitioned into different processors. One of the difficulties in partitioning is to keep the load balance and the communication between the processors at minimum. Therefore, choosing an appropriate decomposition or partitioning strategy is crucial.

Orthogonal Recursive Bisection (ORB)

A domain decomposition method called Orthogonal Recursive Bisection (ORB) [8] is employed in DPM which considers the spatial coordinates of the particles to distribute the particles to different processors. This method assigns particles which are geometrically close to each other on the same processor. To develop the ORB algorithm a cell structure is used.

Two important classes have been developed to implement the ORB algorithm: *domain3D* and *subDomain3D*. The *domain3D* class is used to define the simulation space, the simulation cells, the weight of the cells and the decomposition code. The *subDomain3D* class is used to store the cells belonging to a processor after the decomposition. Both classes are written in C++ programming language to handle any particle or object structure.

The ORB algorithm cell weight corresponds to the number of particles within the cell, assuming that, all the particles have the same computational load. The cell weight approach produces partitions which are usually of higher quality than simple geometric division. Fig. 2.32 illustrates how ORB decomposes a domain into subdomains of similar weight in a simple settings. Partitioning a domain into N subdomains would require N-1 processes.

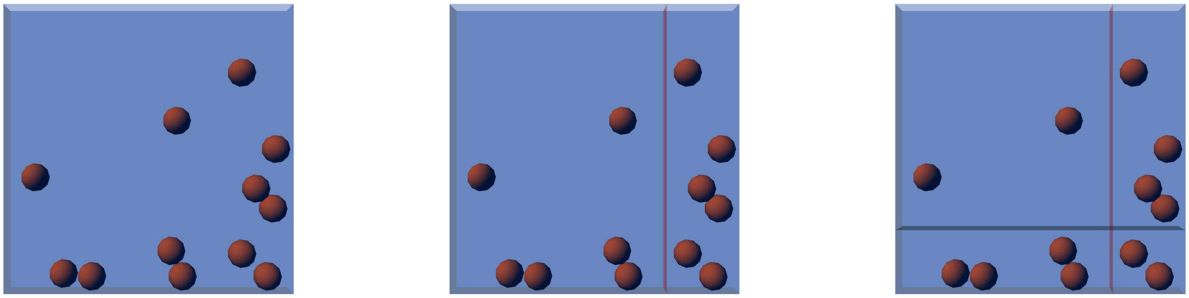


Figure 2.32: A simple settings before domain decomposition (left), after horizontal decomposition (middle) and finally adding a vertical decomposition (right)

Message Exchange

The parallelisation of an application for distributed memory system requires a mechanism for message exchange. The Message Passing Interface (MPI) is chosen for this implementation because it provides low communication overhead and code portability in both Linux and Windows operating systems. Message exchange between processors is required for two reasons:

- When a particle crosses from one processor to another processor
- During updating of the ghost particles.

A ghost particle is a particle in a cell on the border of different sub-domains as shown in fig. 2.33. Such a particle can potentially have contacts with particles in different sub-domains. Therefore, the information of the ghost particles need to be available to all the sub-domains which they are bordering in addition to the sub-domain which they belong to.

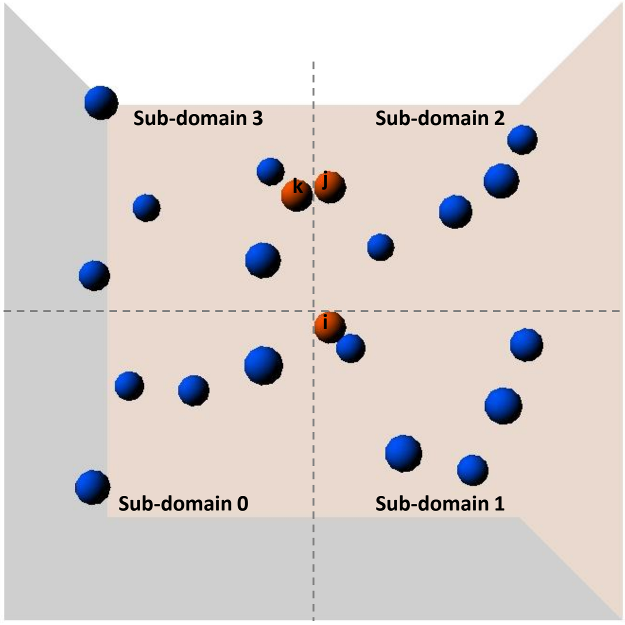


Figure 2.33: A simple box settings where the domain is divided into four sub-domains with particles i, j, k being ghost particles at the specified time step

To maintain buffers for MPI transmission, an MPI interface class called *MPIBuffer* is implemented with MPI package libraries. This class handles all packing/unpacking and sending/receiving functions as well as those required for buffer maintenance such as updating buffer pointers, setting buffer sizes and specifying data packing sizes. The class defines packing/unpacking for all C++ data types. The *MPIBuffer* class encapsulates the low level MPI calls and aims to reduce redundant calls to MPI, e.g. instead of sending data one by one, first all the data are packed and will be sent only with one send call. For data exchange, each object defines packing and unpacking methods. The exchange of objects between two sub-domains is explained as follows. For sending an object, first the sender sub-domain creates an *MPIBuffer* object and calls the packing method of the object to be sent with the *MPIBuffer* object as parameter, the object to be sent in turn calls the packing method of the *MPIBuffer* class for each data to be sent and then the subdomain calls the sending method of the *MPIBuffer* class. For receiving data, first the receiver subdomain creates an *MPIBuffer* object and calls the receiving method of the *MPIBuffer* class and then creates an object and calls the unpacking method of the object with the *MPIBuffer* object as parameter. Fig. 2.34 illustrates the flow of messages between two subdomains/processors.

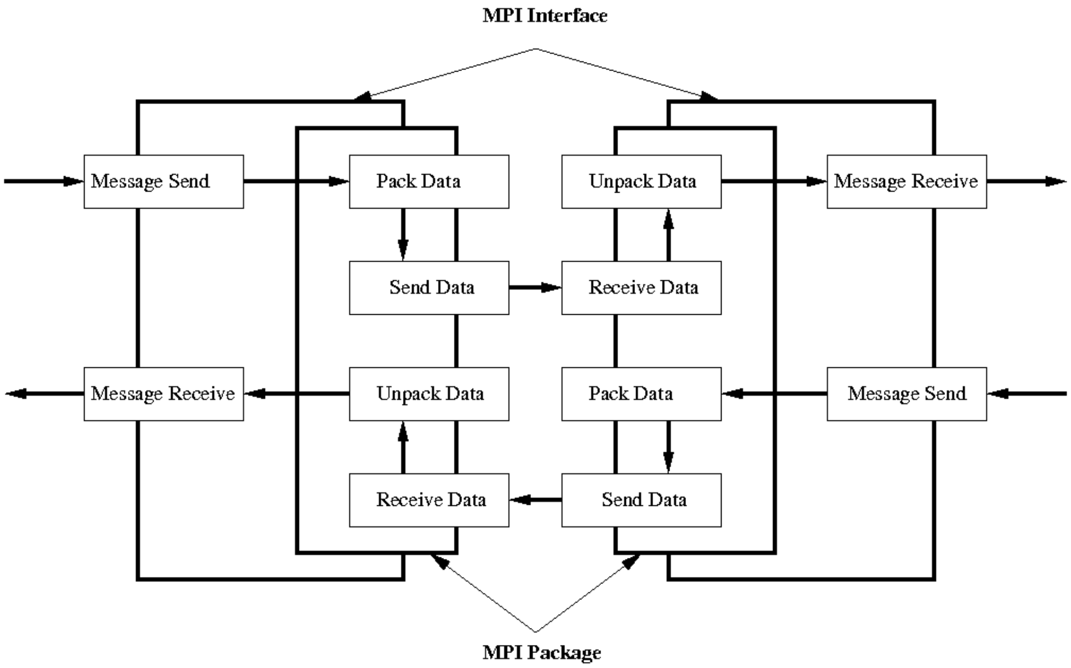


Figure 2.34: Conceptual message flows.

2.8.3 Periodic Boundary Conditions

Another means to save computational costs without jeopardising the accuracy is the utilization of periodic boundary conditions. The basic idea is that in large systems when the behaviour of a system repeats itself in a certain dimension, a part of the system far from its edges would be representative of the whole system. It would be, therefore, sufficient to simulate the representative part rather than the whole domain. This is done by inserting a pair of periodic walls along that certain dimension. An object exiting from one of the periodic walls will be replaced by an identical object with the same velocity at the other periodic wall, as illustrated in fig. 2.35. Periodic walls can be inserted in more than one dimension in space and they can also be utilized in conjunction with parallelisation.

2.9 List of Shapes

The particle and boundary shapes are listed below with their volume and moments of inertia. In all equations, V, I_{xx}, I_{yy} and I_{zz} represent the volume of the shape and the moments of inertia calculated about the principal axes when they meet at the shape's centre of gravity. The figures show the defined geometry in its default position and orientation.

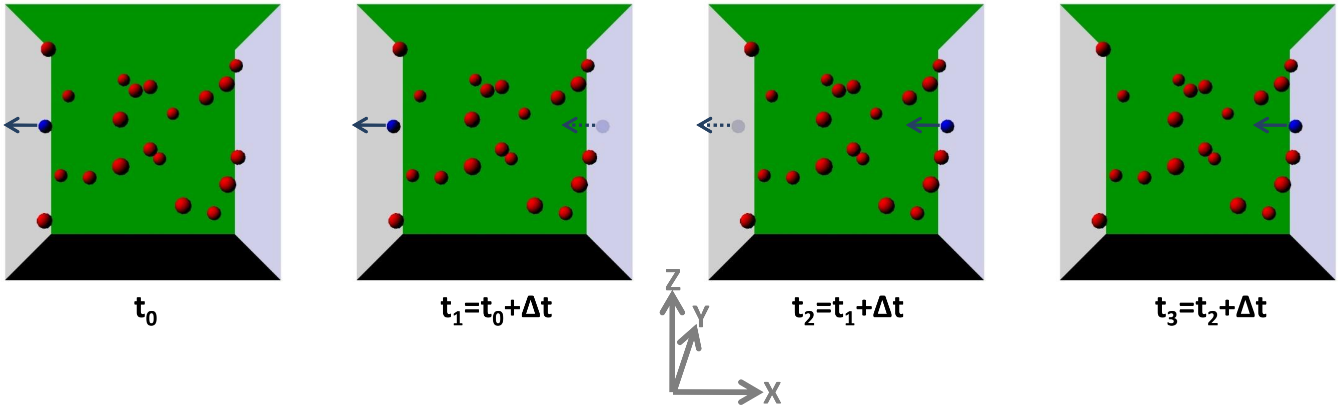


Figure 2.35: Employing periodic boundary conditions along the X-axis, a particle exiting the domain at one end is replaced by an identical particle at the other end.

2.9.1 Particle shapes

Sphere

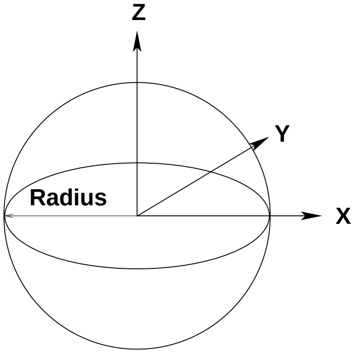


Figure 2.36: Sphere

$$V = \frac{4}{3}\pi r^3$$

$$I_{xx} = I_{yy} = I_{zz} = \frac{8}{15}\pi r^5$$

where r stands for the radius of the sphere.

Ellipsoid

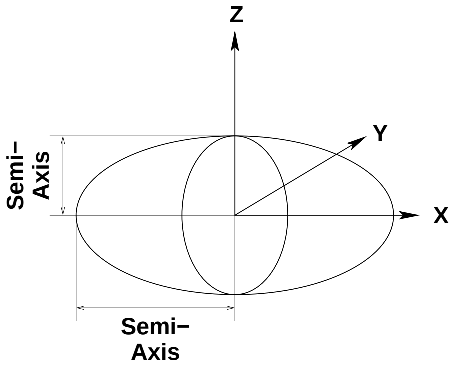


Figure 2.37: Ellipsoid

$$V = \frac{4}{3}\pi abc$$

$$I_{xx} = \frac{4}{15}\pi abc(b^2 + c^2)$$

$$I_{yy} = \frac{4}{15}\pi abc(a^2 + c^2)$$

$$I_{zz} = \frac{4}{15}\pi abc(a^2 + b^2)$$

where a, b and c stand for the radii of the ellipsoid. In DPM the ellipsoid is only defined in the special case of two smaller radii being equal.

Cylinder

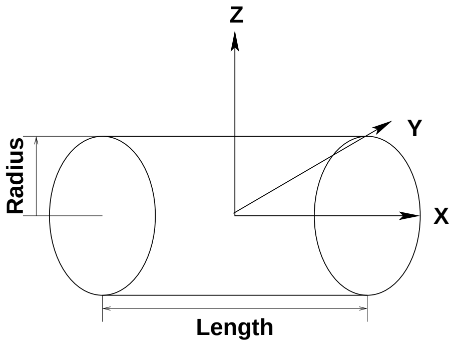


Figure 2.38: Cylinder

$$V = \pi r^2 l$$

$$I_{xx} = \frac{\pi}{2} r^4 l$$

$$I_{yy} = I_{zz} = \frac{\pi}{4} r^2 l(r^2 + \frac{l^2}{3})$$

where r and l stand for the radius and the length of the cylinder respectively.

Cube

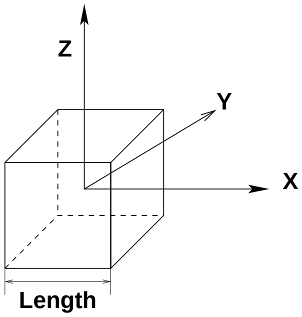


Figure 2.39: Cube

$V = a^3$

$I_{xx} = I_{yy} = I_{zz} = \frac{a^5}{6}$

where a stands for the length of the cube.

Block

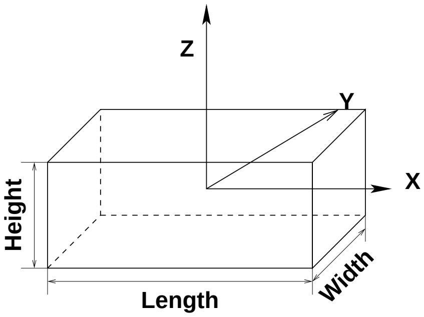


Figure 2.40: Block

$V = abc$

$I_{xx} = \frac{abc}{12}(b^2 + c^2)$

$I_{yy} = \frac{abc}{12}(a^2 + c^2)$

$I_{zz} = \frac{abc}{12}(a^2 + b^2)$

where a, b and c stand for the length, height and width or more simply the three sides of the block.

Parallel-epiped

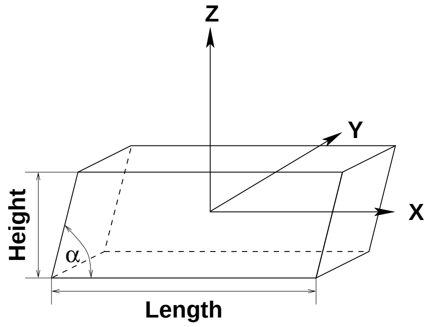


Figure 2.41: Parallel-epiped

$V = abc$

$I_{xx} = \frac{abc}{12}(b^2 + c^2)$

$I_{yy} = \frac{abc}{12}(a^2 + c^2)$

$I_{zz} = \frac{abc}{12}(a^2 + b^2)$

where a, b and c stand for the length, height and width of the parallel-epiped respectively. In DPM, only a special case of the parallel-epiped is considered when the height and width are equal.

Tetrahedron

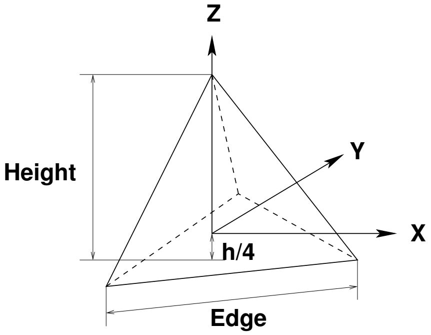


Figure 2.42: Tetrahedron

$V = \frac{\sqrt{2}}{12}a^3$

$I_{xx} = \frac{697}{19440\sqrt{2}}a^5$

$I_{yy} = \frac{533}{14580\sqrt{2}}a^5$

$I_{zz} = \frac{\sqrt{2}}{240}a^5$

where a stands for the edge of the tetrahedron.

Cone

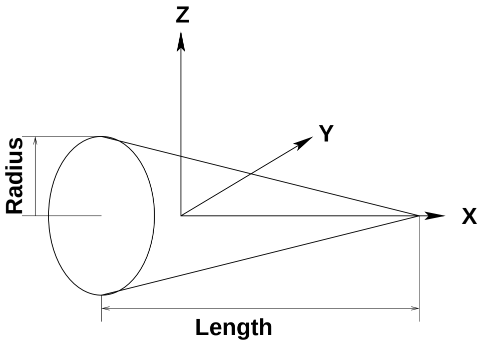


Figure 2.43: Cone

$V = \frac{\pi}{3}r^2l$

$I_{xx} = \frac{\pi}{10}r^4l$

$I_{yy} = I_{zz} = \frac{\pi}{80}r^2l(4r^2 + l^2)$

where r and l stand for the radius and the length of the cone respectively.

Double-Cone

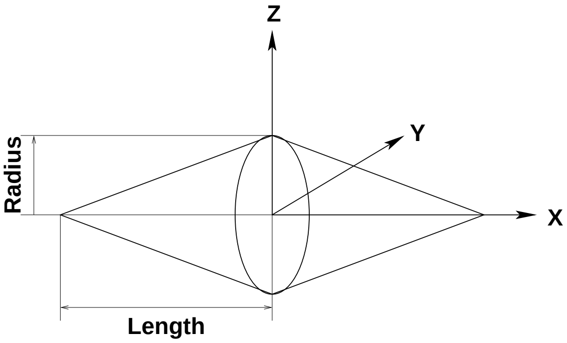


Figure 2.44: Double-Cone

$V = \frac{2}{3}\pi r^2l$

$I_{xx} = \frac{\pi}{5}r^4l$

$$I_{yy} = I_{zz} = \frac{\pi}{10} r^2 l (2r^2 + \frac{4l^2}{3})$$

where r and l stand for the radius and the length of the double-cone respectively.

Hyperboloid

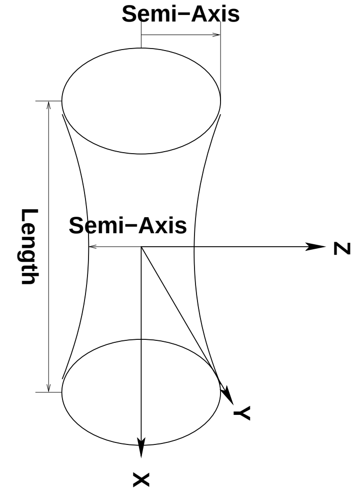


Figure 2.45: Hyperboloid

$$V = \frac{\pi}{3} l ({r_1}^2 + 2{r_2}^2)$$

$$I_{xx} = \pi l b^4 (\frac{4}{15} + \frac{2{r_1}^2}{15{r_2}^2} + \frac{{r_1}^4}{10{r_2}^4})$$

$$I_{yy} = I_{zz} = \pi l^3 {r_2}^2 (\frac{1}{12} - \frac{{r_2}^2 - {r_1}^2}{20{r_2}^2}) + \pi l {r_2}^4 (1 - \frac{{r_2}^2 - {r_1}^2}{6{r_2}^2} + \frac{({r_2}^2 - {r_1}^2)^2}{20{r_2}^4})$$

where r_1, r_2 and l stand for big radius, small radius and the length of the hyperboloid respectively.

Barrel

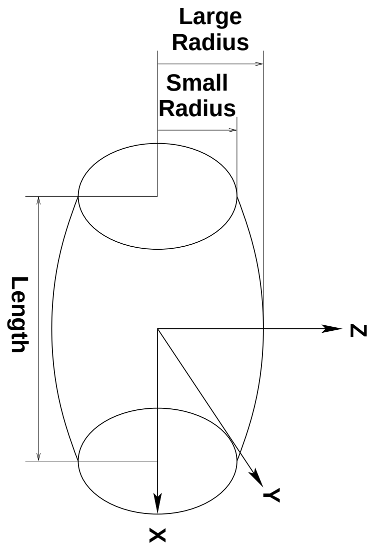


Figure 2.46: Barrel

$$V = \pi l(\frac{{r_1}^2}{5} + \frac{4r_1r_2}{15} + \frac{8{r_2}^2}{15})$$

$$I_{xx} = \pi l(\frac{{r_1}^4}{18} + \frac{4{r_1}^3r_2}{63} + \frac{8{r_1}^2{r_2}^2}{105} + \frac{32r_1{r_2}^3}{315} + \frac{64{r_2}^4}{315})$$

$$I_{yy} = I_{zz} = \pi l^3(\frac{{r_1}^2}{28} + \frac{r_1r_2}{35} + \frac{2{r_2}^2}{105}) + \pi l(\frac{{r_1}^4}{36} + \frac{2{r_1}^3r_2}{63} + \frac{4{r_1}^2{r_2}^2}{105} + \frac{16r_1{r_2}^3}{315} + \frac{32{r_2}^4}{315})$$

where r_1, r_2 and l stand for small radius, big radius and the length of the barrel respectively.

Disc

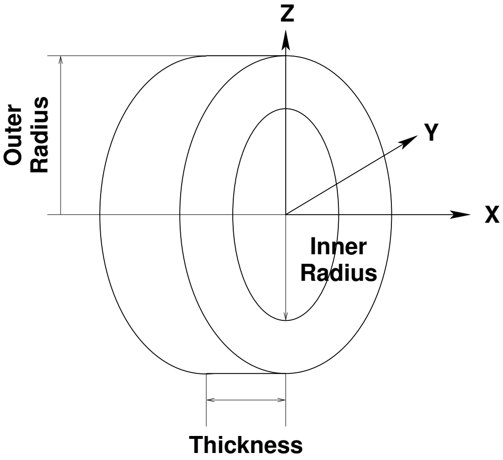


Figure 2.47: Disc

$$V = \pi(r_1^2 - r_2^2)c$$

$$I_{xx} = \frac{\pi}{2}(r_1^4 - r_2^4)c$$

$$I_{yy} = I_{zz} = \frac{\pi}{4}(r_1^4 - r_2^4)c$$

where r_1, r_2 and c stand for outer radius, inner radius and the thickness of the disc respectively.

Torus

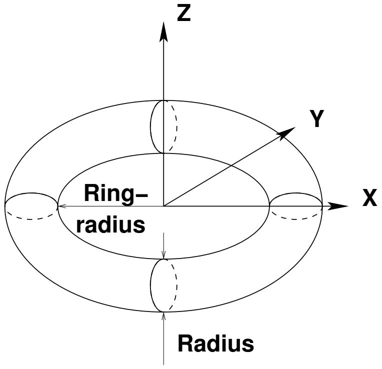


Figure 2.48: Torus

$$V = 2\pi^2 {r_1}^2 r_2$$

$$I_{xx} = I_{yy} = \pi^2 {r_1}^2 r_2 \frac{5{r_1}^2 + 4{r_2}^2}{4}$$

$$I_{zz} = \pi^2 {r_1}^2 r_2 (3{r_1}^2 + 4{r_2}^2)$$

where r_1 and r_2 stand for the tube radius and ring radius of the torus, respectively.

2.9.2 Boundary Shapes

Wall

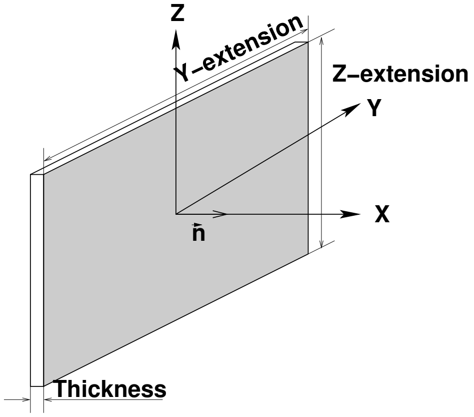


Figure 2.49: Wall

$$V = \delta_1\delta_2 c$$

$$I_{xx} = \frac{\delta_1\delta_2 c}{12}({\delta_1}^2 + {\delta_2}^2)$$

$$I_{yy} = \frac{\delta_1\delta_2 c}{12}({\delta_2}^2 + c^2)$$

$$I_{zz} = \frac{\delta_1\delta_2 c}{12}(c^2 + {\delta_1}^2)$$

where δ_1, δ_2 and c stand for the Y-extension, Z-extension and the thickness of the wall respectively with Y-extension and Z-extension being defined as half of the length of the wall in that certain direction.

Tube

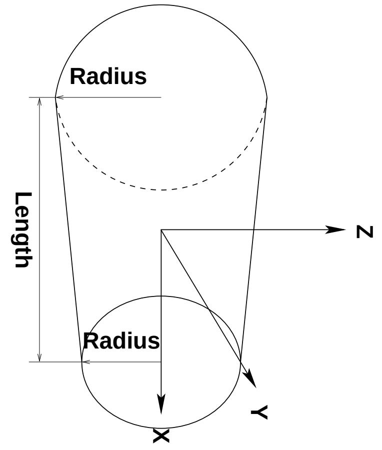


Figure 2.50: Tube

$$V = \frac{\pi}{3}l[(r_1+c)^2 + (r_1+c)(r_2+c) + (r_2+c)^2 - {r_1}^2 - r_1r_2 - {r_2}^2]$$

$$I_{xx} = \frac{\pi}{10}l[(r_1+c)^4 + (r_1+c)^3(r_2+c) + (r_1+c)^2(r_2+c)^2 + (r_1+c)(r_2+c)^3 + (r_2+c)^4 - {r_1}^4 - {r_1}^3r_2 - {r_1}^2{r_2}^2 - r_1{r_2}^3 - {r_2}^4]$$

$$I_{yy} = I_{zz} = \frac{\pi}{2}(r_1+c)^4[l - \frac{2l(r_1+c)^2-(r_2+c)^2}{3(r_1+c)^2} + \frac{l[(r_1+c)^2-(r_2+c)^2)]^2}{5(r_1+c)^4}]$$

where r_1, r_2, l and c stand for small radius, big radius, length and the wall thickness of the tube respectively.

Bend

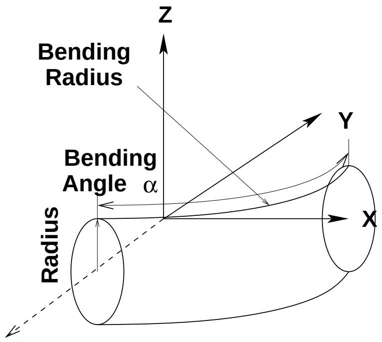


Figure 2.51: Bend

$$V = \frac{\pi^2 R\alpha}{180}[(r+c)^2 - r^2]$$

$$I_{xx} = I_{yy} = \frac{\pi^2\alpha}{4\times 360}[(r+c)^2 R[4R^2 + 5(r+c)^2] - r^2 R(4R^2 + 5r^2)]$$

$$I_{zz} = \frac{\pi^2\alpha}{360}[(r+c)^2 R[4R^2 + 3(r+c)^2] - r^2 R(4R^2 + 3r^2)]$$

where R, r, α and c stand for the bending radius, pipe radius, bending angle and the wall thickness of the bend respectively.

2.10 Applications

The DPM is designed to describe transport processes in a wide range of technical applications. Figs. 2.52 to 2.54 show solid particles in a hopper, on a reverse acting grate and in a rotary kiln respectively as examples of industrial applications.

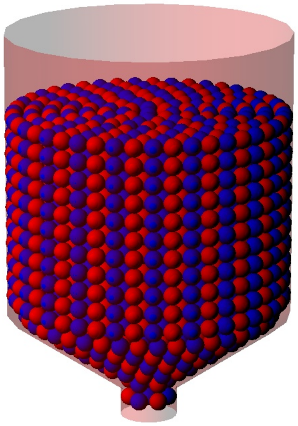


Figure 2.52: Granular particles in a hopper

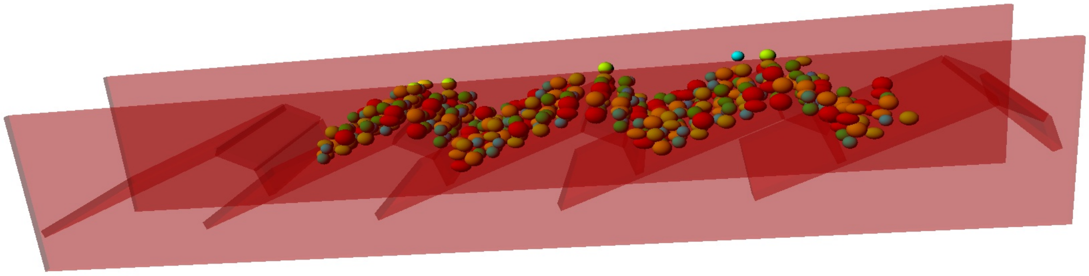


Figure 2.53: Solid fuel particles with different sizes on a reverse acting grate

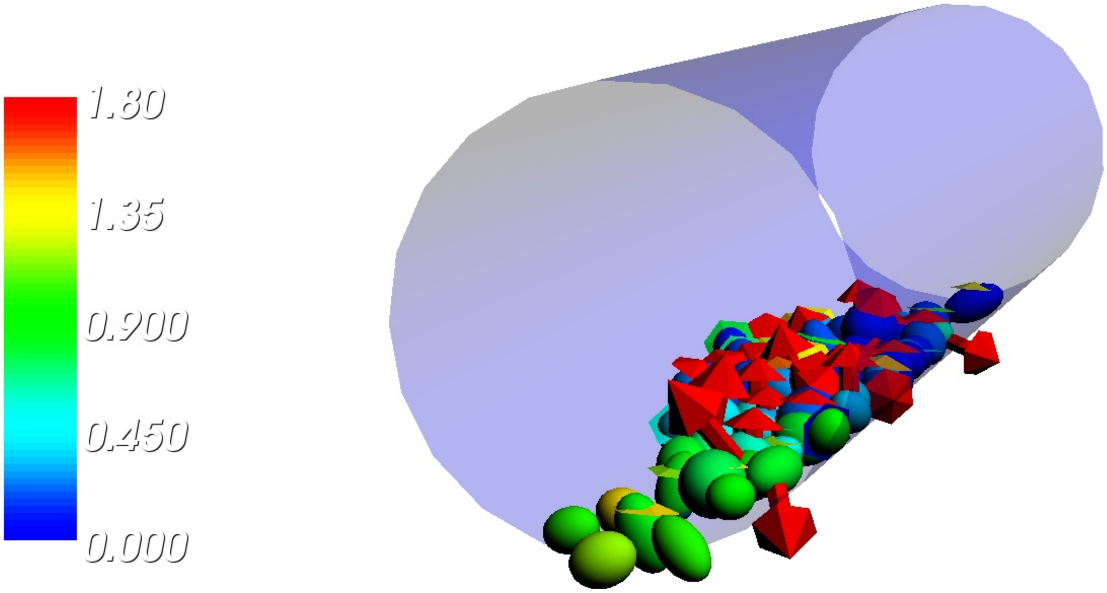


Figure 2.54: Snapshot of an animation of solid particles motion in a rotary kiln simulated by DPM. Particles are colour coded based on their velocities in m/s and arrows visualize the forces exerted on particles.

Chapter 3

Implicit Method

3.1 Introduction

The integration schemes presented in chapter 2 are either fully explicit or in few cases semi-implicit methods. A fully implicit integration method is, however, not presented in section 2.7 because using an implicit method will fundamentally change the DEM algorithm as it will be shown in this chapter. The difference between the explicit and implicit methods is in their formulas for predicting the new state of the particles. The state of particles at the next time step can be calculated from the state of particles at the current and/or previous time steps. This will simplify solving the equation because the new positions could be expressed as explicit functions of the already known values. If the state of particles at the next time step is calculated not only from the current and previous time steps but also from the next time step, the equation of motion will be implicit in new positions. Generally the implicit method is computationally more expensive because it requires a system of equations to be solved at each time step. On the other hand, relatively larger time steps could be used in implicit methods due to higher numerical stability. The common practice in DEM simulations is the explicit updating and the implicit method has been rarely used.

This chapter starts with a literature review of the state of the art of the implicit method in DEM. The implicit formulation is then introduced and the equations are derived for the DEM particle collision problem. The flowchart for the implicit method is then presented and the methods to solve the problem are discussed. The implicit predictions of the position and velocity are compared with the explicit method in a simple collision. The implemented implicit algorithm is then validated with available experimental results in various real cases including hopper discharge and particles motion on backward acting grate. The chapter will conclude by assessing the merits and drawbacks of the implicit method in the framework of DEM.

3.2 Literature Review

Several scientists have discussed and argued for and against the implicit method for molecular dynamics. Yet, the experience is still very limited particularly in the field of granular material. In the following, articles or book chapters reflecting the authors experience with implicit implementation are briefly discussed. The interested reader is recommended to refer to the sources for more detailed discussions.

The first attempt to develop a parallel algorithm for the implicit Runge-Kutta scheme was presented by Janezic et al., 1994 [44]. Their objective was the numerical solution of molecular dynamics equation on the distributed memory computers in the ring topology. In their work, they showed that the time step can be doubled in 24GLRK method for the same performances given by Verlet-type methods. The net enlargement of time complexity, however, compared to Verlet-type methods was negligible.

Ke et al., 1995 [53] presented an implicit method called the discontinuous deformation analysis for two dimensional simulation of particulate media. This method is claimed to solve systems of few thousands of particles in “reasonable times”. No comparison with explicit methods is, however, presented.

Cummins et al., 2002 [18] also presented an implicit-in-time method for granular materials. Their method combines the material point method, a first order contact algorithm and a Newton-Krylov equation solver. In their conclusions, they affirm that even though limits on time step impose constraints on the maximum time step that can be used, the implicit algorithm with time steps that exceed certain stability limit yields more accuracy than can be achieved with the explicit algorithm. They claim their method to be more stable, robust and physical than explicit algorithms and yields higher quality results. However, a clear comment on overall efficiency is missing.

Katsu Yamane, 2004 [113] presents his implementation of implicit integration technique to simulate soft collision model for motions of human figures. He employs backward Euler integration scheme and explains his methods to reduce the computational costs for implicit integration. He then compares his results with two different explicit methods, explicit Euler and 4th order Runge-Kutta for a simple setting where a box falls down onto a flat floor. In this particular case he states that implicit integration allowed 16 times larger time steps than explicit Euler and 4 times larger than 4th order Runge-Kutta method. The total simulation time reduced to 1/8 of both explicit methods. He then applied his method to more complex settings and demonstrated very positive results. For example for simulation of the motion of human figures, he claims to be able to use as large time steps as 1-4 ms which is almost ten times larger than typical time steps for penalty-based contact models with explicit integration.

Schäfer et al., 2008 [94] focused on methods to reduce the computational costs of implicit methods for classical molecular dynamics. They suggest that much of the costs typically cited as reason not using implicit methods can be reduced, and if programmed in parallel, essentially eliminated. Their suggestions include replacing the Jacobian used in Quasi-Newton method with a constant, diagonal mass matrix, evaluating the Jacobian infrequently and finding a better prediction of the system configuration to improve the convergence of the Quasi-Newton method. They report a particular numerical experiment where implicit methods, even in a relatively inefficient implementation, shown to be more efficient than explicit integration. However, they admit that this is not true in general yet.

More recently, Schäfer et al., 2010 [93] evaluated the potential of implicit integration methods in molecular dynamics simulation of biological molecules. Although they report good energy conservation response by the implicit methods, the increase in the time step was limited due to loss of convergence of the iterative method. They refer to the overhead caused by the implicit integration as the main reason limiting the usefulness of this class of methods.

Tuley et al., 2010 [106] investigated the performance of different integration schemes including implicit schemes in solving the motion of a particle during a normal collision with a wall. This work was later advanced by Jasion et al., 2011 [45] including the response of the same integration schemes to a collision in the tangential direction of motion. They state in their work that the order of accuracy of the integration schemes is limited to second order in DEM even in higher order schemes due to discontinuities in the contact force. Taken this into account and considering the high computational costs of DEM in large systems, they conclude that more simple integrators such as a first order Symplectic Euler scheme are more suitable than costly and conditionally more accurate integrators for simulations containing thousands of particles.

Overall, the experience with implicit methods in the context of DEM has been limited and reporting mixed, i.e. both positive and negative, results. Despite several investigatory studies discussing the advantages and disadvantages of the implicit methods, very few have drawn their conclusions out of the results of an implemented implicit method. This could be, among other reasons, due to the fact that implicit methods are in general harder to implement than the explicit integrators.

3.3 Implicit Integration Schemes

As described in section 2.6, the contact force in DPM is in general a function of the positions and velocities of particles in contact. On the other hand, it determines the particle's acceleration according to the Newton's second law of motion. Therefore the differential equation describing the motion of a particle involves the first and second derivatives of the position as expressed in eq. 3.1.

$$\frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{1}{m_i}\vec{F}_i(\vec{r}_i, \frac{\partial \vec{r}_i}{\partial t}) \tag{3.1}$$

Various implicit integration schemes are available to solve such an Ordinary Differential Equation (ODE) [31, 73]. The most common implicit methods to solve the Newton's equations of motion are listed below.

3.3.1 Backward Euler

A first order method is the backward Euler method described in eq. 2.57. Applying the backward Euler method, the equations for the new velocity and position will be:

$$\vec{v}_{t+1} = \vec{v}_t + \vec{a}_{t+1}\Delta t \tag{3.2}$$

$$\vec{r}_{t+1} = \vec{r}_t + \vec{v}_{t+1}\Delta t \tag{3.3}$$

3.3.2 Beeman

The implicit Beeman's method is a second order method developed by Beeman, 1976 [7]. It can be derived from the Taylor polynomial of order three:

$$\vec{r}_{t+1} = \vec{r}_t + \Delta t \vec{r}\,'_t + \frac{1}{2!}\Delta t^2 \vec{r}\,''_t + \frac{1}{3!}\Delta t^3 \vec{r}\,^{(3)}_t \tag{3.4}$$

Applying the implicit Backward Euler method to the third derivative term in eq. 3.4 and rearranging gives the Beeman's implicit formula:

$$\vec{r}_{t+1} = \vec{r}_t + \Delta t \vec{v}_t + \frac{1}{6}\Delta t^2 (2\vec{a}_t + \vec{a}_{t+1}) \tag{3.5}$$

One method to calculate the velocity is, then, by assuming constant average acceleration between the current and next time steps. Applying it to the Taylor polynomial of degree two yields:

$$\vec{v}_{t+1} = \vec{v}_t + \Delta t(\frac{\vec{a}_t + \vec{a}_{t+1}}{2}) \tag{3.6}$$

3.3.3 Newmark-Beta

The Newmark-Beta method is derived by Newmark, 1959 [74] based on the Cauchy's mean value Theorem [46]. Its most commonly used form can be derived by assuming constant average acceleration in the Taylor polynomial of degree two.

$$\vec{v}_{t+1} = \vec{v}_t + \Delta t(\frac{\vec{a}_t + \vec{a}_{t+1}}{2}) \tag{3.7}$$

$$\vec{r}_{t+1} = \vec{r}_t + \Delta t \vec{v}_t + \frac{\Delta t^2}{4}(\vec{a}_t + \vec{a}_{t+1}) \tag{3.8}$$

The method in this case can also be derived by applying the trapezoidal rule of integration [3].

3.3.4 Second Order Backward Difference

The second order backward difference method, also called the three time level method by Ferziger et al., 2002 [31], can be derived from the Taylor polynomial of degree two for the two consecutive time steps as follows:

$$f(t-\Delta t) = f(t) - \Delta t f'(t) + \frac{\Delta t^2}{2} f''(t) \tag{3.9}$$

$$f(t-2\Delta t) = f(t) - 2\Delta t f'(t) + \frac{4\Delta t^2}{2} f''(t) \tag{3.10}$$

Multiplying eq. 3.9 by four and subtracting eq. 3.10 and rearranging gives the second order backward difference approximation for the first derivative of a function:

$$f'(t) = \frac{3f(t) - 4f(t-\Delta t) + f(t-2\Delta t)}{2\Delta t} \tag{3.11}$$

Applying this scheme to the Newton's equations of motion, yields the velocity and position at the new time step as:

$$\vec{v}_{t+1} = \frac{4\vec{v}_t - \vec{v}_{t-1} + 2\vec{a}_{t+1}\Delta t}{3} \tag{3.12}$$

$$\vec{r}_{t+1} = \frac{4\vec{r}_t - \vec{r}_{t-1} + 2\vec{v}_{t+1}\Delta t}{3} \tag{3.13}$$

3.3.5 Forth Order Backward Difference

Similar to the second order backward difference, the forth order method can also be derived from the Taylor polynomial of degree two written for four consecutive time steps. The first two polynomials are given in eqs. 3.9 and 3.10, and the third and forth can be similarly written as:

$$f(t-3\Delta t) = f(t) - 3\Delta t f'(t) + \frac{9\Delta t^2}{2} f''(t) \tag{3.14}$$

$$f(t-4\Delta t) = f(t) - 4\Delta t f'(t) + \frac{16\Delta t^2}{2} f''(t) \tag{3.15}$$

Multiplying the above equations by the right coefficients to even out the second derivative term yields a formula including only the first derivative term. The coefficients in this case are 48 multiplied by eq. 3.9, -36 by eq. 3.10, 16 by eq. 3.14 and -3 by eq. 3.15. Summing up gives the forth order backward difference approximation for the first derivative of a function:

$$f'(t) = \frac{25f(t) - 48f(t-\Delta t) + 36f(t-2\Delta t) - 16f(t-3\Delta t) + 3f(t-4\Delta t)}{12\Delta t} \tag{3.16}$$

Applying this formula to the Newton's equations of motion yields the velocity and position at the new time step as:

$$\vec{v}_{t+1} = \frac{48\vec{v}_t - 36\vec{v}_{t-1} + 16\vec{v}_{t-2} - 3\vec{v}_{t-3} + 12\vec{a}_{t+1}\Delta t}{25} \tag{3.17}$$

$$\vec{r}_{t+1} = \frac{48\vec{r}_t - 36\vec{r}_{t-1} + 16\vec{r}_{t-2} - 3\vec{r}_{t-3} + 12\vec{v}_{t+1}\Delta t}{25} \tag{3.18}$$

3.3.6 Adams-Moulton

The Adams-Moulton implicit methods are produced by polynomial interpolation of a function at different points [38]. The first and second degree methods reduce to the backward Euler and trapezoidal rule respectively. The third degree scheme can be derived from the Taylor polynomial of order three, eq. 3.4 and replacing the second and third derivative terms by the central difference formulas, eqs. 2.63 and 2.66. The new velocity and position can then be written as:

$$\vec{v}_{t+1} = \vec{v}_t + \Delta t\left(\frac{5}{12}\vec{a}_{t+1} + \frac{2}{3}\vec{a}_t - \frac{1}{12}\vec{a}_{t-1}\right) \tag{3.19}$$

$$\vec{r}_{t+1} = \vec{r}_t + \Delta t\left(\frac{5}{12}\vec{v}_{t+1} + \frac{2}{3}\vec{v}_t - \frac{1}{12}\vec{v}_{t-1}\right) \tag{3.20}$$

3.3.7 Runge-Kutta

The family of Runge-Kutta methods are commonly used to solve ordinary differential equations. The backward Euler and the trapezoid method are the most commonly used implicit methods in this series.

3.3.8 Numerov

The Numerov's method, also called Cowel's method or sometimes the implicit Störmer method [96], can be used to solve the Newton's equation of motion. The integration formula can be derived from the Taylor series, eq. 2.54. Writing up the Taylor polynomial of order five for the position at two opposite time directions gives:

$$\vec{r}_{t+1} = \vec{r}_t + \Delta t \vec{r}\,'_t + \frac{1}{2!}\Delta t^2 \vec{r}\,''_t + \frac{1}{3!}\Delta t^3 \vec{r}\,^{(3)}_t + \frac{1}{4!}\Delta t^4 \vec{r}\,^{(4)}_t + \frac{1}{5!}\Delta t^5 \vec{r}\,^{(5)}_t \tag{3.21}$$

$$\vec{r}_{t-1} = \vec{r}_t - \Delta t \vec{r}\,'_t + \frac{1}{2!}\Delta t^2 \vec{r}\,''_t - \frac{1}{3!}\Delta t^3 \vec{r}\,^{(3)}_t + \frac{1}{4!}\Delta t^4 \vec{r}\,^{(4)}_t - \frac{1}{5!}\Delta t^5 \vec{r}\,^{(5)}_t \tag{3.22}$$

Summing up eqs. 3.21 and 3.22 results in:

$$\vec{r}_{t+1} \quad = \quad 2\vec{r}_t - \vec{r}_{t-1} + \Delta t^2 \vec{r}\,''_t + \frac{1}{12}\Delta t^4 \vec{r}\,^{(4)}_t \tag{3.23}$$

Replacing the second derivative of the position by acceleration leads to:

$$\vec{r}_{t+1} \quad = \quad 2\vec{r}_t - \vec{r}_{t-1} + \Delta t^2 \vec{a}_t + \frac{1}{12}\Delta t^4 \vec{a}\,''_t \tag{3.24}$$

Replacing the second derivative of the acceleration by the central difference approximation presented in eq. 2.66 yields the Numerov formula:

$$\vec{r}_{t+1} \quad = \quad 2\vec{r}_t - \vec{r}_{t-1} + \frac{\Delta t^2}{12}(\vec{a}_{t+1} + 10\vec{a}_t + \vec{a}_{t-1}) \tag{3.25}$$

The Numerov's method does not provide the velocity by itself. The backward difference approximation can be used to predict the velocity:

$$\vec{v}_{t+1} \quad = \quad \frac{\vec{r}_{t+1} - \vec{r}_t}{\Delta t} \tag{3.26}$$

3.3.9 The Selected Integration Scheme

The primary objective in this study is to implement an implicit method which is easy to validate and compare with analytical and explicit results. The main idea is to start simple considering primarily only the DEM repulsive force and extending it to energy dissipation and friction forces at a next stage. The Numerov integration scheme is chosen for this purpose for several reasons listed below.

- The Numerov method is best suited to solve ODEs in the following form:

  $$\frac{\partial^2 y}{\partial t^2} \quad = \quad f(y) \tag{3.27}$$

  Neglecting the viscous energy dissipation term, the DEM normal repulsive force, eq. 2.38, leads to an ODE in the same form as eq. 3.27.

  $$\frac{\partial^2 \delta}{\partial t^2} \quad = \quad -\frac{K_n}{m_{ij}}\delta^\alpha \tag{3.28}$$

- In contrast to many other higher order schemes which need several previous values initially, the Numerov method only requires two previous values in order to predict the next value. This makes it suitable for simple validation tests and comparison with conventional explicit schemes. Numerov's position prediction according to eq. 3.25 can be directly compared to the prediction of the position by the Verlet method according to eq. 2.67 which is a widely used explicit scheme [62].

- The Numerov integration method has not been used in the context of DEM previously which makes it an interesting research topic.

3.4 Comparison with Explicit Methods

Section 2.7.6 compared different explicit integration schemes in a single collision between a particle and a wall using linear spring-dashpot force model. The same case is here extended by including the implicit Numerov integration method in the comparison. The objective is to compare the implicit Numerov's prediction with the prediction of the explicit methods. Since the Numerov's method does not yield the velocity by itself, only the particle's position is the subject of the comparison in this section. Similar to the graphs presented in section 2.7.6, the graphs presented in this section are based on normalized values in order to represent general trends independent of any specific case. The values are normalized over the maximum point of the analytical solution in the absence of dissipation in order to allow a meaningful comparison.

3.4.1 Normal Repulsive Force

Initially energy dissipation is excluded and the contact resolution, defined in eq. 2.79, is set to four. Fig. 3.1 shows the results of the prediction of the particle's position.

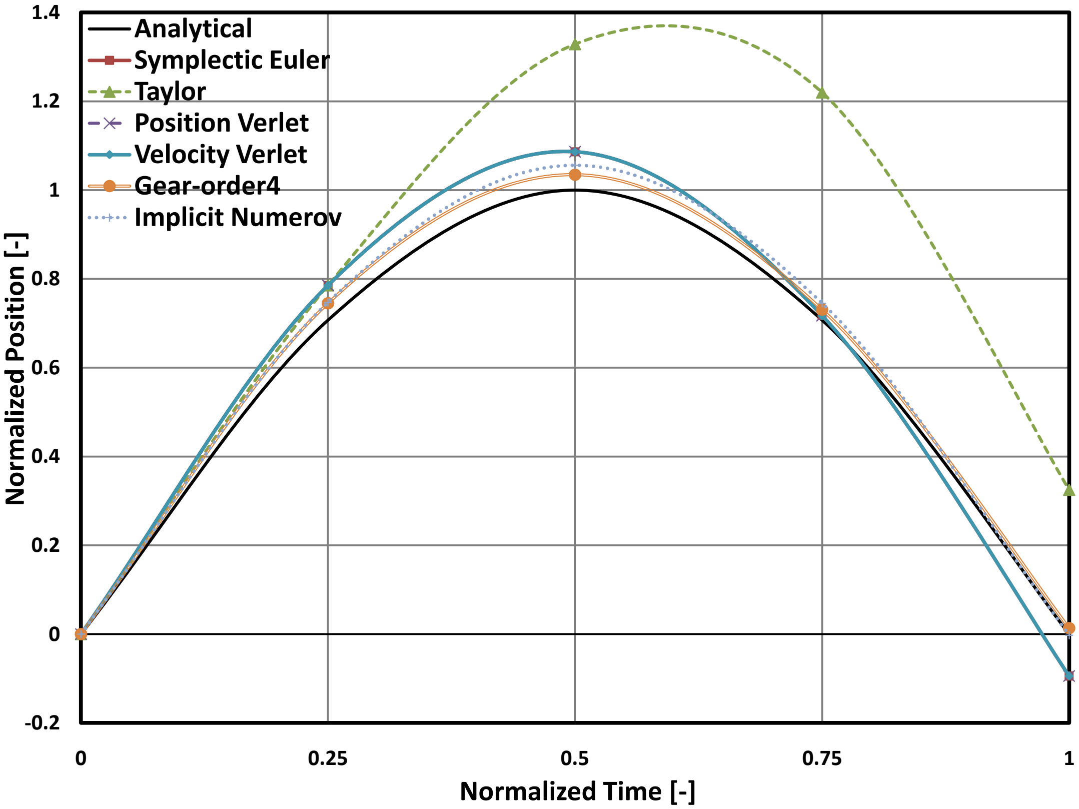


Figure 3.1: Prediction of the position of a particle during collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 1$

Fig. 3.1 shows that the prediction provided by the implicit Numerov's method is very good. Its prediction of the particle's position during contact is in the same range of accuracy as the best explicit prediction produced by the Gear's method. The Gear's method predicts the maximum overlap better than the Numerov's method while on the other hand, the position at the end of the collision is better predicted by the Numerov's method.

3.4.2 Normal Energy Dissipation

The energy dissipation is modelled as a linear dashpot force dependent on the relative normal velocity during contact. There are different ways to integrate the normal energy dissipation into the Numerov's method. Before arriving at comparisons between implicit and explicit methods, the methodology to integrate the energy dissipation in the implicit method needs to be determined.

The normal velocity, being the first derivative of the position, can be approximated by the backward difference formula according to eq. 2.57. The backward difference or backward Euler method incorporated into the Numerov's method leads to eq. 3.29:

$$F_n = -\frac{k_n}{12}(\delta_{t+1} + 10\delta_t + \delta_{t-1}) - c_n \cdot \frac{\delta_t - \delta_{t-1}}{\Delta t} \tag{3.29}$$

Using this approach, the dissipation term would not contain the new position and can be formulated explicitly in terms of the known values. Alternatively, the forward difference approximation defined in eq. 2.55 will introduce the new position in the dissipation term as shown in eq. 3.30:

$$F_n = -\frac{k_n}{12}(\delta_{t+1} + 10\delta_t + \delta_{t-1}) - c_n \cdot \frac{\delta_{t+1} - \delta_t}{\Delta t} \tag{3.30}$$

Finally, the Numerov's method in conjunction with the central difference approximation of the normal velocity leads to eq. 3.31:

$$F_n = -\frac{k_n}{12}(\delta_{t+1} + 10\delta_t + \delta_{t-1}) - c_n \cdot \frac{\delta_{t+1} - \delta_{t-1}}{2\Delta t} \tag{3.31}$$

Fig. 3.2 shows how these different combinations predict the particle's position. It is clear that the backward difference approximation of the velocity does not provide an accurate prediction comparing to the two other methods. The forward difference approximation is a better prediction during the loading phase while the central difference approximation is a better match with the analytical solution during the unloading phase.

Fig. 3.3 shows the comparison of the implicit Numerov method combined with the central difference approximation with the explicit methods. It shows that the prediction of the Numerov implicit scheme is superior to the predictions produced by the explicit methods.

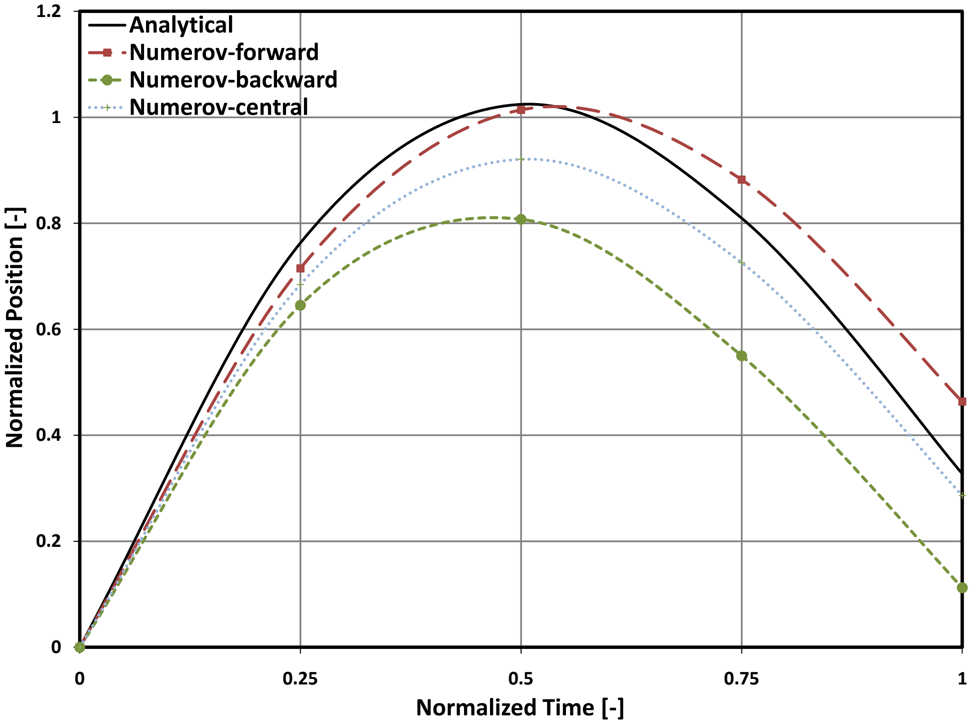


Figure 3.2: Prediction of the position of a particle during collision with a wall, comparison of different methods to integrate energy dissipation, contact-resolution= 4, $e = 0.5$

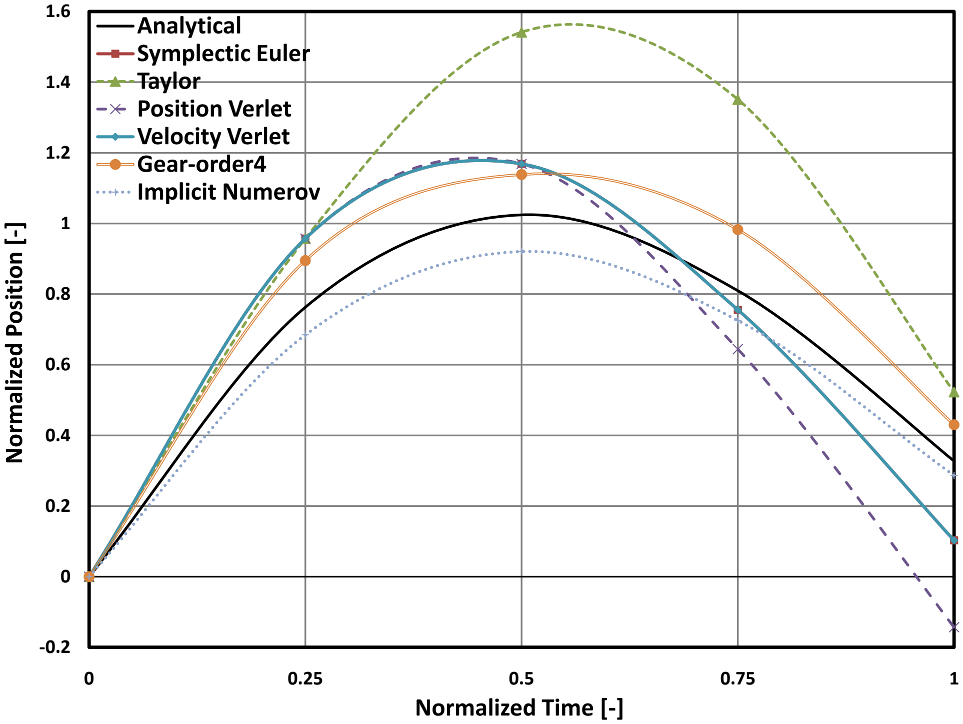


Figure 3.3: Prediction of the position of a particle during collision with a wall, comparison of different integration methods, contact-resolution= 4, $e = 0.5$

3.5 System of Equations

Normal repulsive force between spherical particles is commonly modelled as elastic spring force with linear or non-linear force-displacement relations depending on the force model, as described in section 2.6. The general differential equation describing the motion of particle i due to contact with particle j can then be written as:

$$\frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{k_{ij}}{m_i}(R_i + R_j - |\vec{r}_i - \vec{r}_j|)^{\alpha} \cdot \frac{(\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|} \tag{3.32}$$

where k, m, R and r represent the stiffness, mass, radius and position respectively and α is the coefficient dependent on the force model.

A particle might have collisions with more than one other shape at the same time. Therefore, Eq. 3.32 should be extended to account for all collisions of the specified particle as follows:

$$\frac{\partial^2 \vec{r}_i}{\partial t^2} = \sum_{j=1}^{N} \frac{k_{ij}}{m_i}(R_i + R_j - |\vec{r}_i - \vec{r}_j|)^{\alpha} \cdot \frac{(\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|} \delta_{ij}^{\alpha} \tag{3.33}$$

where N is the total number of bodies including particles and boundary shapes in the system. Of course not all other shapes have contacts with particle i necessarily for which the overlap and consequently the force would be zero.

Another equation can be written for the jth sphere, in the same pattern as eq. 3.33 which is from the ith sphere's standpoint. This could be extended to the total number of spheres in contact:

$$\begin{bmatrix}
\sum_{j=1}^{N} \frac{k_{1j}\delta_{1j}^{\alpha}}{|\vec{r}_1 - \vec{r}_j|} & -\frac{k_{12}\delta_{12}^{\alpha}}{|\vec{r}_1 - \vec{r}_2|} & -\frac{k_{13}\delta_{13}^{\alpha}}{|\vec{r}_1 - \vec{r}_3|} & . \; . \; . & -\frac{k_{1n}\delta_{1n}^{\alpha}}{|\vec{r}_1 - \vec{r}_n|} \\
-\frac{k_{21}\delta_{21}^{\alpha}}{|\vec{r}_2 - \vec{r}_1|} & \sum_{j=1}^{N} \frac{k_{2j}\delta_{2j}^{\alpha}}{|\vec{r}_2 - \vec{r}_j|} & -\frac{k_{23}\delta_{23}^{\alpha}}{|\vec{r}_2 - \vec{r}_3|} & . \; . \; . & -\frac{k_{2n}\delta_{2n}^{\alpha}}{|\vec{r}_2 - \vec{r}_n|} \\
-\frac{k_{31}\delta_{31}^{\alpha}}{|\vec{r}_3 - \vec{r}_1|} & -\frac{k_{32}\delta_{32}^{\alpha}}{|\vec{r}_3 - \vec{r}_2|} & \sum_{j=1}^{N} \frac{k_{3j}\delta_{3j}^{\alpha}}{|\vec{r}_3 - \vec{r}_j|} & . \; . \; . & -\frac{k_{3n}\delta_{3n}^{\alpha}}{|\vec{r}_3 - \vec{r}_n|} \\
. & . & . & . \; . \; . & . \\
. & . & . & . \; . \; . & . \\
. & . & . & . \; . \; . & . \\
-\frac{k_{n1}\delta_{n1}^{\alpha}}{|\vec{r}_n - \vec{r}_1|} & -\frac{k_{n2}\delta_{n2}^{\alpha}}{|\vec{r}_n - \vec{r}_2|} & -\frac{k_{n3}\delta_{n3}^{\alpha}}{|\vec{r}_n - \vec{r}_3|} & . \; . \; . & \sum_{j=1}^{N} \frac{k_{nj}\delta_{nj}^{\alpha}}{|\vec{r}_n - \vec{r}_j|}
\end{bmatrix}
\begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vec{r}_3 \\ . \\ . \\ . \\ \vec{r}_n \end{bmatrix}
=
\begin{bmatrix} m_1\ddot{\vec{r}}_1 \\ m_2\ddot{\vec{r}}_2 \\ m_3\ddot{\vec{r}}_3 \\ . \\ . \\ . \\ m_n\ddot{\vec{r}}_n \end{bmatrix} \tag{3.34}$$

As described in eq. 2.2, in addition to the forces from other particles and boundary walls, the motion of particles is governed by the earth's gravitational force in DPM. Incorporating the gravity force into eq. 3.34, the system of equations can be concluded as:

$$\begin{bmatrix}
\sum_{j=1}^{N} \frac{k_{1j}\delta_{1j}^{\alpha}}{|\vec{r}_1 - \vec{r}_j|} & -\frac{k_{12}\delta_{12}^{\alpha}}{|\vec{r}_1 - \vec{r}_2|} & -\frac{k_{13}\delta_{13}^{\alpha}}{|\vec{r}_1 - \vec{r}_3|} & \cdot\;\cdot\;\cdot & -\frac{k_{1n}\delta_{1n}^{\alpha}}{|\vec{r}_1 - \vec{r}_n|} \\
-\frac{k_{21}\delta_{21}^{\alpha}}{|\vec{r}_2 - \vec{r}_1|} & \sum_{j=1}^{N} \frac{k_{2j}\delta_{2j}^{\alpha}}{|\vec{r}_2 - \vec{r}_j|} & -\frac{k_{23}\delta_{23}^{\alpha}}{|\vec{r}_2 - \vec{r}_3|} & \cdot\;\cdot\;\cdot & -\frac{k_{2n}\delta_{2n}^{\alpha}}{|\vec{r}_2 - \vec{r}_n|} \\
-\frac{k_{31}\delta_{31}^{\alpha}}{|\vec{r}_3 - \vec{r}_1|} & -\frac{k_{32}\delta_{32}^{\alpha}}{|\vec{r}_3 - \vec{r}_2|} & \sum_{j=1}^{N} \frac{k_{3j}\delta_{3j}^{\alpha}}{|\vec{r}_3 - \vec{r}_j|} & \cdot\;\cdot\;\cdot & -\frac{k_{3n}\delta_{3n}^{\alpha}}{|\vec{r}_3 - \vec{r}_n|} \\
\cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\
-\frac{k_{n1}\delta_{n1}^{\alpha}}{|\vec{r}_n - \vec{r}_1|} & -\frac{k_{n2}\delta_{n2}^{\alpha}}{|\vec{r}_n - \vec{r}_2|} & -\frac{k_{n3}\delta_{n3}^{\alpha}}{|\vec{r}_n - \vec{r}_3|} & \cdot\;\cdot\;\cdot & \sum_{j=1}^{N} \frac{k_{nj}\delta_{nj}^{\alpha}}{|\vec{r}_n - \vec{r}_j|}
\end{bmatrix}
\begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \vec{r}_3 \\ \cdot \\ \cdot \\ \cdot \\ \vec{r}_n \end{bmatrix}
=
\begin{bmatrix} m_1(\ddot{\vec{r}}_1 - \vec{g}) \\ m_2(\ddot{\vec{r}}_2 - \vec{g}) \\ m_3(\ddot{\vec{r}}_3 - \vec{g}) \\ \cdot \\ \cdot \\ \cdot \\ m_n(\ddot{\vec{r}}_n - \vec{g}) \end{bmatrix}
\tag{3.35}$$

where g is the earth's standard gravity.

It is worth noting the difference between n and N in eqs. 3.34 and 3.35 and other equations in this chapter. n is the number of moving particles in the system. These are the bodies which move due to the forces from the other objects in the system. N is the total number of moving and non-moving bodies in the system. The non-moving particles are the fixed boundary shapes which do not move due to collision with particles. They only exert force on particles upon contact.

Energy dissipation in the normal direction of contact and friction forces in the tangential direction are excluded from the implicit algorithm. The idea in this dissertation is to implement and validate the implicit algorithm in its basic form i.e. considering normal repulsive forces. Yet, discussion and comparison of different approaches to include energy dissipation are already presented in section 3.4.2.

Eq. 3.35 describes a system of non-linear differential equations with the same number of unknowns as the number of equations. There is a unique answer for such a system which is to be found.

3.6 Implicit Formulation

As already discussed in section 3.3.9, the implicit Numerov method is chosen to solve the system. Applying the Numerov formula, eq. 3.25, to the system of equations given in eq. 3.35 would lead to:

$$\vec{r}_{i,t+1} = \frac{\Delta t^2}{12}(\vec{a}_{i,t+1} + 10\vec{a}_{i,t} + \vec{a}_{i,t-1}) + 2\vec{r}_{i,t} - \vec{r}_{i,t-1} \quad , \quad i = 1, 2, 3, ..., n \tag{3.36}$$

where:

$$\vec{a}_i \;=\; \sum_{j=1}^{N} \frac{k_{ij}(R_i + R_j - |\vec{r}_i - \vec{r}_j|)^{\alpha}(\vec{r}_i - \vec{r}_j)}{m_i\,|\vec{r}_i - \vec{r}_j|} + \vec{g} \quad , \quad i = 1, 2, 3, ..., n \tag{3.37}$$

Eq. 3.36 contains vector quantities which need to be decomposed into their Cartesian components before solving. Decomposing the vectors and rearranging gives:

$$\begin{bmatrix} x_{i,t+1} - 2x_{i,t} + x_{i,t-1} \\ \\ y_{i,t+1} - 2y_{i,t} + y_{i,t-1} \\ \\ z_{i,t+1} - 2z_{i,t} + z_{i,t-1} \end{bmatrix} - \frac{\Delta t^2}{12} \begin{bmatrix} (a^x_{i,t+1} + 10a^x_{i,t} + a^x_{i,t-1}) \\ \\ (a^y_{i,t+1} + 10a^y_{i,t} + a^y_{i,t-1}) \\ \\ (a^z_{i,t+1} + 10a^z_{i,t} + a^z_{i,t-1}) \end{bmatrix} = 0 \quad , \quad i = 1, 2, 3, ..., n \tag{3.38}$$

where:

$$\begin{bmatrix} a^x_i \\ \\ a^y_i \\ \\ a^z_i \end{bmatrix} = \sum_{j=1}^{N} \frac{k_{ij}(R_i + R_j - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2})^{\alpha}}{m_i\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}} \begin{bmatrix} x_i - x_j \\ \\ y_i - y_j \\ \\ z_i - z_j \end{bmatrix} - \begin{bmatrix} 0 \\ \\ 0 \\ \\ g \end{bmatrix} \tag{3.39}$$

where the Cartesian coordinate system is defined in a way that the gravity vector is in opposite Z direction, as it is the case in DPM software.

3.7 Newton-Raphson Method

The system of equations introduced in eqs. 3.38 and 3.39 represents a system of non-linear equations. The unknowns are the new positions, $\vec{r}_{i,t+1}$ while the current and old positions $\vec{r}_{i,t}$, $\vec{r}_{i,t-1}$ are known values. The Newton-Raphson method is a common way to solve such a system [82] and it is also used here.

F_i can be defined as the left hand side function in eq. 3.38:

$$\begin{bmatrix} F^x_i \\ \\ F^y_i \\ \\ F^z_i \end{bmatrix} = \begin{bmatrix} x_{i,t+1} - 2x_{i,t} + x_{i,t-1} \\ \\ y_{i,t+1} - 2y_{i,t} + y_{i,t-1} \\ \\ z_{i,t+1} - 2z_{i,t} + z_{i,t-1} \end{bmatrix} - \frac{\Delta t^2}{12} \begin{bmatrix} (a^x_{i,t+1} + 10a^x_{i,t} + a^x_{i,t-1}) \\ \\ (a^y_{i,t+1} + 10a^y_{i,t} + a^y_{i,t-1}) \\ \\ (a^z_{i,t+1} + 10a^z_{i,t} + a^z_{i,t-1}) \end{bmatrix} = 0 \ , \ i = 1, 2, 3, ..., n \tag{3.40}$$

It is clear from eqs. 3.40 and 3.39 that F_i is a function of positions of the contacting particles:

$$\begin{bmatrix} F_i{}^x(x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n) \\ \\ F_i{}^y(x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n) \\ \\ F_i{}^z(x_1, y_1, z_1, x_2, y_2, z_2, ..., x_n, y_n, z_n) \end{bmatrix} = 0 \quad , \quad i = 1, 2, 3, ..., n \tag{3.41}$$

The objective is to find the roots of the function F_i. In the neighbourhood of the current positions, each of the functions F_i can be approximated by Taylor expansion. By neglecting terms of order

δ^2 and higher in Taylor series, one can write:

$$\begin{bmatrix} F_i{}^x(x_1+\delta x_1, ..., z_n+\delta z_n) \\ \\ F_i{}^y(x_1+\delta x_1, ..., z_n+\delta z_n) \\ \\ F_i{}^z(x_1+\delta x_1, ..., z_n+\delta z_n) \end{bmatrix} = \begin{bmatrix} F_i{}^x(x_1, ..., z_n) \\ \\ F_i{}^y(x_1, ..., z_n) \\ \\ F_i{}^z(x_1, ..., z_n) \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n}(\frac{\partial F_i{}^x}{\partial x_j}\delta x_j + \frac{\partial F_i{}^x}{\partial y_j}\delta y_j + \frac{\partial F_i{}^x}{\partial z_j}\delta z_j) \\ \\ \sum_{j=1}^{n}(\frac{\partial F_i{}^y}{\partial x_j}\delta x_j + \frac{\partial F_i{}^y}{\partial y_j}\delta y_j + \frac{\partial F_i{}^y}{\partial z_j}\delta z_j) \\ \\ \sum_{j=1}^{n}(\frac{\partial F_i{}^z}{\partial x_j}\delta x_j + \frac{\partial F_i{}^z}{\partial y_j}\delta y_j + \frac{\partial F_i{}^z}{\partial z_j}\delta z_j) \end{bmatrix} \tag{3.42}$$

In order to find the roots of the functions (F_i) at the new time step, the right hand sides of eq. 3.42 should be set to zero. The new set of equations can be described in the matrix equation below:

$$\begin{bmatrix} \frac{\partial F_1{}^x}{\partial x_1} & \frac{\partial F_1{}^x}{\partial y_1} & \frac{\partial F_1{}^x}{\partial z_1} & \frac{\partial F_1{}^x}{\partial x_2} & . & . & . & \frac{\partial F_1{}^x}{\partial z_n} \\ \frac{\partial F_1{}^y}{\partial x_1} & \frac{\partial F_1{}^y}{\partial y_1} & \frac{\partial F_1{}^y}{\partial z_1} & \frac{\partial F_1{}^y}{\partial x_2} & . & . & . & \frac{\partial F_1{}^y}{\partial z_n} \\ \frac{\partial F_1{}^z}{\partial x_1} & \frac{\partial F_1{}^z}{\partial y_1} & \frac{\partial F_1{}^z}{\partial z_1} & \frac{\partial F_1{}^z}{\partial x_2} & . & . & . & \frac{\partial F_1{}^z}{\partial z_n} \\ \frac{\partial F_2{}^x}{\partial x_1} & \frac{\partial F_2{}^x}{\partial y_1} & \frac{\partial F_2{}^x}{\partial z_1} & \frac{\partial F_2{}^x}{\partial x_2} & . & . & . & \frac{\partial F_2{}^x}{\partial z_n} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \frac{\partial F_n{}^z}{\partial x_1} & \frac{\partial F_n{}^z}{\partial y_1} & \frac{\partial F_n{}^z}{\partial z_1} & \frac{\partial F_n{}^z}{\partial x_2} & . & . & . & \frac{\partial F_n{}^z}{\partial z_n} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta y_1 \\ \delta z_1 \\ \delta x_2 \\ . \\ . \\ . \\ \delta z_n \end{bmatrix} = - \begin{bmatrix} F_1{}^x \\ F_1{}^y \\ F_1{}^z \\ F_2{}^x \\ . \\ . \\ . \\ F_n{}^z \end{bmatrix} \tag{3.43}$$

The partial derivatives matrix appearing in eq. 3.43 is the Jacobian matrix J. If the whole vector containing δx_1, δy_1, δz_1, δx_2,..., δz_n, is simply called δ and the right hand side vector in eq. 3.43 simply F, then the matrix equation could be written as follows:

$$J \cdot \delta \;\; = \;\; -F \tag{3.44}$$

Eq. 3.44 represents a linear system of equations with δ being the unknown. To solve it, the Newton-Raphson method adopts an iterative strategy. The linear equation is first solved assuming an initial guess for the positions. The initial guess for the new position of a particle is taken as its current position which is a reasonable choice assuming the particles displacement during a time step is small. The guess is then improved according to the following equations:

$$\begin{bmatrix} x_i^{new-guess} \\ \\ y_i^{new-guess} \\ \\ z_i^{new-guess} \end{bmatrix} = \begin{bmatrix} x_i^{old-guess} \\ \\ y_i^{old-guess} \\ \\ z_i^{old-guess} \end{bmatrix} + \begin{bmatrix} \delta x_i \\ \\ \delta y_i \\ \\ \delta z_i \end{bmatrix} \quad , \quad i = 1, 2, 3, ..., n \tag{3.45}$$

The equation is solved and the guesses are improved until δ approaches zero which defines convergence.

3.8 Implicit Algorithm

The solution method explained in section 3.7 forms the basic algorithm for the implicit method which is summarized in the flowchart in fig. 3.4. This can be compared to the flowchart in fig. 2.11 which illustrates the basic explicit algorithm.

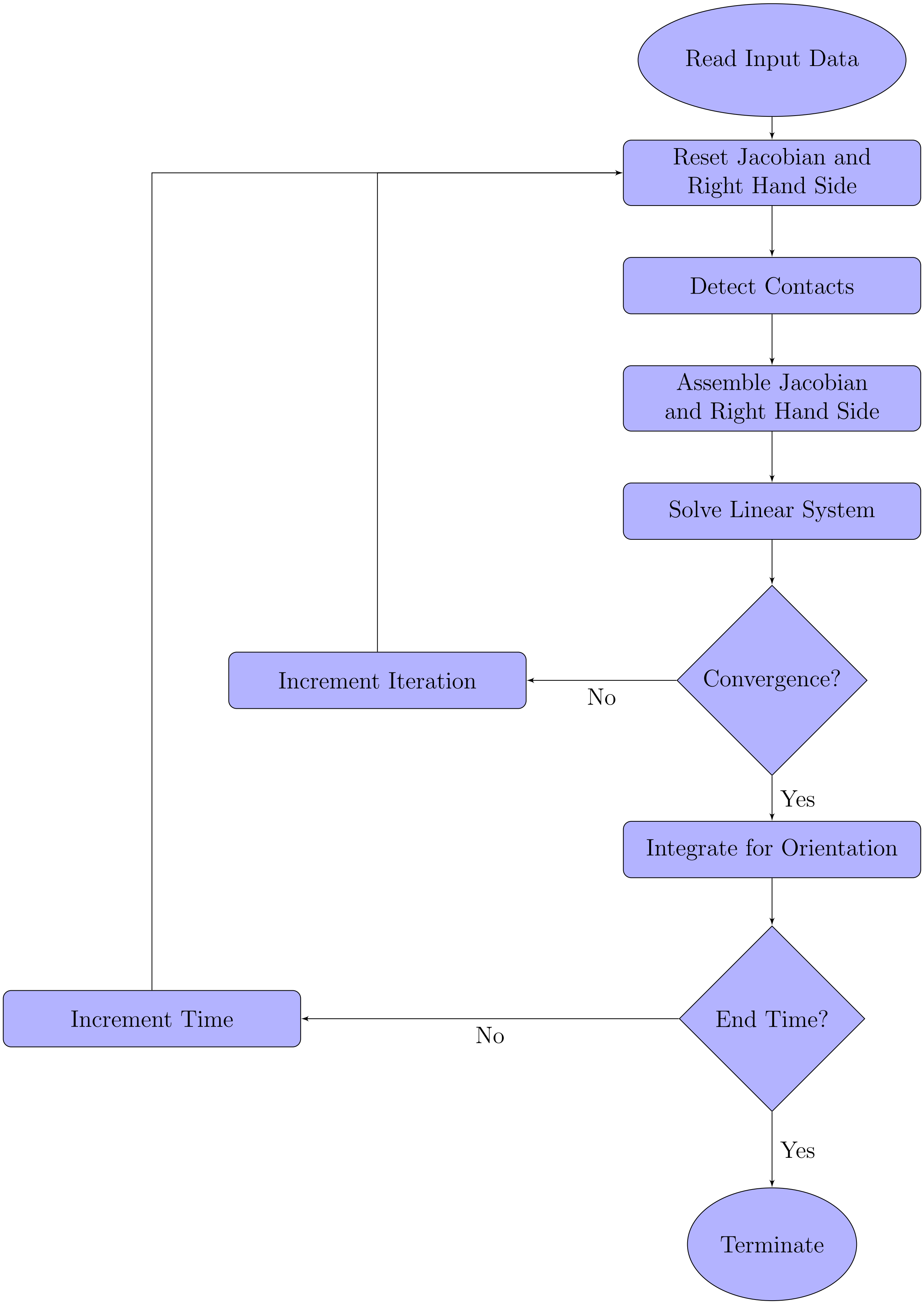


Figure 3.4: The basic flowchart for implicit method

3.8.1 Contact Detection

Based on the same principle as the explicit method illustrated in fig. 2.6, a contact is defined if there is an overlap between the two shapes. Contact detection in the implicit method is performed at every iteration. This can be explained by recalling that the force in an implicit scheme depends on the next overlap in addition to the current and old overlaps as shown in eq. 3.46 for the implicit Numerov formula:

$$F_n = -\frac{k_n}{12}(\delta_{t+1} + 10\delta_t + \delta_{t-1}) \tag{3.46}$$

Therefore, at every iteration the new solved positions need to be used for calculating new overlaps based on eq. 2.11. Identifying contacts at every iteration has its advantages and disadvantages. On one hand contact detection is a costly part of the DEM algorithm and performing this at every iteration adds to the computational burden. On the other hand contacts can be detected even during a time step in the implicit method which is advantageous over explicit method where contacts are detected only at the beginning of the time step. The difference between implicit and explicit methods in contact detection is demonstrated in depth in section 3.9.2.

3.8.2 Matrix Assembly

To set up the linear system inherent in the Newton-Raphson method, eq. 3.43, the right hand side matrix and the Jacobian need to be assembled. The right hand side matrix, defined in eq. 3.40, has $3n$ elements where n is the number of particles in the system. It needs to be updated at every iteration by new positions.

The entries of the Jacobian matrix are basically the partial derivatives of the right hand side matrix. The elements on the main diagonal are given in eq. 3.47.

$$\begin{bmatrix} \dfrac{\partial F_i^x}{\partial x_i} \\[2ex] \dfrac{\partial F_i^y}{\partial y_i} \\[2ex] \dfrac{\partial F_i^z}{\partial z_i} \end{bmatrix} = 1 - \frac{\Delta t^2}{12 m_i} \sum_{j=1}^{N} k_{ij} \begin{bmatrix} \dfrac{-\alpha\delta_{ij}^{\alpha-1}(x_{i,t+1}-x_{j,t+1})^2 d_{ij} + \delta_{ij}^{\alpha}[d_{ij}^2 - (x_{i,t+1}-x_{j,t+1})^2]}{d_{ij}^3} \\[2ex] \dfrac{-\alpha\delta_{ij}^{\alpha-1}(y_{i,t+1}-y_{j,t+1})^2 d_{ij} + \delta_{ij}^{\alpha}[d_{ij}^2 - (y_{i,t+1}-y_{j,t+1})^2]}{d_{ij}^3} \\[2ex] \dfrac{-\alpha\delta_{ij}^{\alpha-1}(z_{i,t+1}-z_{j,t+1})^2 d_{ij} + \delta_{ij}^{\alpha}[d_{ij}^2 - (z_{i,t+1}-z_{j,t+1})^2]}{d_{ij}^3} \end{bmatrix} \tag{3.47}$$

The other elements in the Jacobian matrix are given in eqs. 3.48 to 3.50.

$$\begin{bmatrix} \dfrac{\partial F_i^x}{\partial x_j} \\[2ex] \dfrac{\partial F_i^y}{\partial y_j} \\[2ex] \dfrac{\partial F_i^z}{\partial z_j} \end{bmatrix} = -\frac{k_{ij}\Delta t^2}{12 m_i} \begin{bmatrix} \dfrac{\alpha\delta_{ij}^{\alpha-1}(x_{i,t+1}-x_{j,t+1})^2 d_{ij} - \delta_{ij}^{\alpha}[d_{ij}^2 - (x_{i,t+1}-x_{j,t+1})^2]}{d_{ij}^3} \\[2ex] \dfrac{\alpha\delta_{ij}^{\alpha-1}(y_{i,t+1}-y_{j,t+1})^2 d_{ij} - \delta_{ij}^{\alpha}[d_{ij}^2 - (y_{i,t+1}-y_{j,t+1})^2]}{d_{ij}^3} \\[2ex] \dfrac{\alpha\delta_{ij}^{\alpha-1}(z_{i,t+1}-z_{j,t+1})^2 d_{ij} - \delta_{ij}^{\alpha}[d_{ij}^2 - (z_{i,t+1}-z_{j,t+1})^2]}{d_{ij}^3} \end{bmatrix} \qquad i \neq j \tag{3.48}$$

$$\begin{bmatrix} \dfrac{\partial F_i^x}{\partial y_i} \\[2ex] \dfrac{\partial F_i^x}{\partial z_i} \\[2ex] \dfrac{\partial F_i^y}{\partial z_i} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial F_i^y}{\partial x_i} \\[2ex] \dfrac{\partial F_i^z}{\partial x_i} \\[2ex] \dfrac{\partial F_i^z}{\partial y_i} \end{bmatrix} = -\frac{\Delta t^2}{12m_i}\sum_{j=1}^{N} k_{ij} \begin{bmatrix} \dfrac{(x_{i,t+1}-x_{j,t+1})(y_{i,t+1}-y_{j,t+1})(-\alpha\delta_{ij}^{\alpha-1}d_{ij}-\delta_{ij}^{\alpha})}{d_{ij}^3} \\[2ex] \dfrac{(x_{i,t+1}-x_{j,t+1})(z_{i,t+1}-z_{j,t+1})(-\alpha\delta_{ij}^{\alpha-1}d_{ij}-\delta_{ij}^{\alpha})}{d_{ij}^3} \\[2ex] \dfrac{(y_{i,t+1}-y_{j,t+1})(z_{i,t+1}-z_{j,t+1})(-\alpha\delta_{ij}^{\alpha-1}d_{ij}-\delta_{ij}^{\alpha})}{d_{ij}^3} \end{bmatrix} \tag{3.49}$$

$$\begin{bmatrix} \dfrac{\partial F_i^x}{\partial y_j} \\[2ex] \dfrac{\partial F_i^x}{\partial z_j} \\[2ex] \dfrac{\partial F_i^y}{\partial z_j} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial F_i^y}{\partial x_j} \\[2ex] \dfrac{\partial F_i^z}{\partial x_j} \\[2ex] \dfrac{\partial F_i^z}{\partial y_j} \end{bmatrix} = -\frac{k_{ij}\Delta t^2}{12m_i} \begin{bmatrix} \dfrac{(x_{i,t+1}-x_{j,t+1})(y_{i,t+1}-y_{j,t+1})(\alpha\delta_{ij}^{\alpha-1}d_{ij}+\delta_{ij}^{\alpha})}{d_{ij}^3} \\[2ex] \dfrac{(x_{i,t+1}-x_{j,t+1})(z_{i,t+1}-z_{j,t+1})(\alpha\delta_{ij}^{\alpha-1}d_{ij}+\delta_{ij}^{\alpha})}{d_{ij}^3} \\[2ex] \dfrac{(y_{i,t+1}-y_{j,t+1})(z_{i,t+1}-z_{j,t+1})(\alpha\delta_{ij}^{\alpha-1}d_{ij}+\delta_{ij}^{\alpha})}{d_{ij}^3} \end{bmatrix} \quad i \neq j \tag{3.50}$$

d_{ij} and δ_{ij} in eqs. 3.47 to 3.50 represent the distance and the overlap between the two shapes at the new iteration respectively:

$$d_{ij} = \sqrt{(x_{t+1,i}-x_{j,t+1})^2 + (y_{i,t+1}-y_{j,t+1})^2 + (z_{i,t+1}-z_{j,t+1})^2} \tag{3.51}$$

$$\delta_{ij} = R_i + R_j - d_{ij} \tag{3.52}$$

Eqs. 3.47 to 3.50 are enough to set up all the elements in the Jacobian matrix with i having values from 1 to n. The Jacobian needs to be updated by new solved positions at every iteration.

The Jacobian matrix is a sparse matrix in general. This is due to the fact that, except for small test cases, each particle can only have collisions with few other particles in the system at the same time. Therefore and in order to save memory usage, Compressed sparse row format is employed to store the Jacobian [98].

3.8.3 Linear Solver

Once the linear system in eq. 3.43 is set up, a linear solver can be employed to solve it. The Gauss-Seidel iterative method is chosen to solve the system. Noting from eqs. 3.47 to 3.50 that the Jacobian matrix is symmetric and diagonally dominant, the Gauss-Seidel method is a suitable choice [50].

3.8.4 Convergence

Convergence in the Newton-Raphson method is defined as when δ in eq. 3.44 approaches zero which is enforced by a tolerance value. Convergence is defined strictly in the algorithm in the sense that all particles in the system need to converge before the program can proceed to the next time step. In other words, the Newton-Raphson iterations continue until all elements of the δ matrix in eq. 3.43 become smaller than the tolerance.

3.8.5 Rotational Motion

The equation of rotational motion of the particles are not solved implicitly but the torque acting on shapes is calculated as the average of the two torques at the beginning and the end of that time step. In this way, the positions at the next time step are also used to compute the torque acting on particles. This is in contrast to the explicit method where only the force at the beginning of the time step is used to calculate the torque on the particles.

3.8.6 Parallelization

The parallelization principles explained in section 2.8.2 can be utilized in the implicit algorithm too. Each processor would deal with the particles in its specific sub-domain as well as its ghost particles as illustrated in fig. 2.33. In the implicit method, it would imply that eq. 3.43 is set up for each processor where n is the number of particles in the sub-domain plus its ghost particles. In a strict implementation, the new position of the ghost particles need to be communicated to the other processor where this particle is bordering with. However, for an efficient parallel implementation the communication during iteration is excluded in DPM. As a consequence of this assumption, the collision between ghost particles of different sub-domains is not solved fully implicitly in the strict sense of the word. This consequence is arguably justified considering the overhead costs of communication between processors at every iteration.

3.9 Verification

Within modelling and simulation community, verification is defined as "the process of determining that a computer model, simulation, or federation of models and simulations implementations represent the developer's conceptual description and specifications" [70]. This section presents different test cases where the performance of the implicit method is evaluated and compared with the explicit and wherever available the analytical results.

Comparison of the accuracy of the proposed implicit method with the more common methods i.e. explicit schemes is not a straightforward task. This is because finding the analytical solution to take as the reference for comparison is a very complicated task and often impossible for real DEM simulations containing a considerable number of particles. What could be done is to verify the results in simple but representative cases where the analytical solution is available. For larger cases, experimental results or simply physical behaviour can be taken as the reference for comparison.

Furthermore, in order to validate a new simulation approach, it is important to test the method not only in one specific case but in different cases and circumstances. It is also necessary to run the simulations several times. A single run can not be considered a representative solution in general.

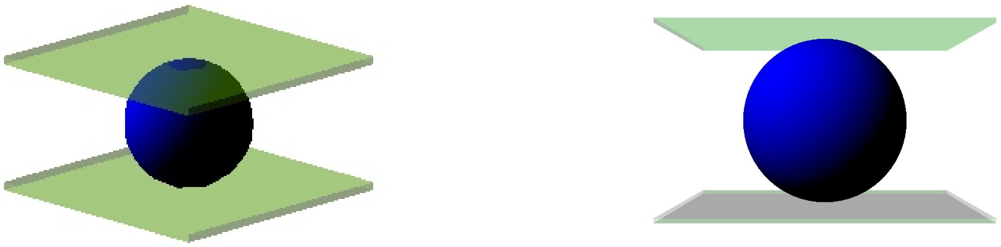
Initially the idea is to verify the implicit method in simple settings where analytical solution is relatively easy to determine. Such a test is performed and described in section 3.9.1. Another simple test is presented in section 3.9.2 which explores the difference between the explicit and implicit methods in detecting an upcoming contact. While these two mentioned sections present simple tests including only one moving object, section 3.9.3 presents a larger case including 2010 moving particles in a hopper. A quite comprehensive comparison between the explicit and implicit methods are presented for the simulation of the hopper discharge. Finally, the results of the implicit method in simulation of the motion of solid fuels on a backward acting grate is presented in section 3.9.4. The results are then compared with the experimental results. The Backward Acting Grate simulation results are not compared with the explicit results. It is merely performed in order to validate the feasibility of the implicit method in modelling the dynamics of granular media. Backward and Forward Acting Grates are the subject of chapter 4 and more advanced simulation results are presented there.

In all cases, the implicit method employed is the Numerov formula presented in eq. 3.25. The explicit method employed for comparison with the implicit results is the Position Verlet scheme presented in eq. 2.67. The Verlet method to predict the position is a very common explicit method [62]. Both Verlet and Numerov methods require two previous values in order to predict the next position which makes them an appropriate pair for conducting a fair comparison.

As the value of the time step is dependent on the specific studied case, it has no general meaning when analysing different cases. Therefore it would be more meaningful to use the contact resolution defined in eq. 2.79 for comparisons. Since a linear force model is used in the following verification tests, the contact duration is not dependent on the impact velocity and can be precisely calculated from the size of the particles and their mechanical properties. Whenever the problem includes poly-disperse particles, the minimum contact duration is considered. A survey of previous DEM work shows that the size of time step is often 50 times smaller than the duration of contact [54, 110, 4, 14, 103].

3.9.1 Particle Sandwich

A simple verification test is presented in this section where the analytical solution is easy to determine. The case chosen is a spherical particle in continuous contact with two parallel walls, as shown in figure 3.5.



(a) Top view

(b) Front view

Figure 3.5: A particle sandwiched between two fixed walls

The distance between the two walls is equal to the sphere's diameter. The sphere is initially positioned in contact with one of the walls which creates an initial overlap. In this type of "sandwich configuration", the particle has no room for free motion and is continuously pushed back by the repulsive force from either of the walls. Gravitational force is excluded and the simple linear force model with no damping and no friction is chosen for this test.

The choice of this setting is justified by the fact that such a continuous exposure to contact forces from different objects resembles the reality of many industrial applications involving dense granular matter. Besides, numerical integration of motion of particles in free motion, i.e under no force or constant gravitational force, is not challenging and the implicit and explicit methodologies do not differ much there. The implicit approach is a distinguished method when the forces are dependant on the positions, which in DEM is true for the contact forces. What is important in free motion is the quick detection of contacts with other objects. The difference between the implicit and explicit methods in detection of contacts is explained in section 3.9.2.

Solution

There are several features in the case described above which makes it easy to solve. These are:

- The motion of the particle is one-dimensional. It reverses its direction at maximum overlap points but remains on the same line of movement.
- There is only one moving object.
- The particle has only one contact at a time. It is either in contact with the lower wall or the upper one and not with both at the same instance.
- The force model is linear.

Considering these simplifications, the equation of motion in eq. 3.32 can be rewritten for this specific case, as shown in eq. 3.53.

$$\frac{\partial^2 r_i}{\partial t^2} + \frac{k_{ij}}{m_i} r_i = \frac{k_{ij}}{m_i}(R_i + r_j) \tag{3.53}$$

where subscript i represents the moving particle and subscript j represents the wall in contact with the particle. The right hand side of the equation is constant.

The general solution to such a second order linear ordinary differential equation, eq. 3.53, is shown in eq. 3.54 [51, 9, 83].

$$r_i = C_1 \sin(\sqrt{\frac{k_{ij}}{m_i}} t) + C_2 \cos(\sqrt{\frac{k_{ij}}{m_i}} t) + R_i + r_j \tag{3.54}$$

The boundary conditions are assumed to be:

$$\begin{cases} r_i(0) = R_i + r_j + 0.15 r_i \\ \dot{r}_i(0) = 0 \end{cases} \tag{3.55}$$

These initial conditions mean the particle is positioned, with zero velocity, experiencing an overlap with the upper wall equal to 15% of its radius. The value of initial overlap being 15% of the particle's radius is chosen arbitrary and does not affect the generality of the case. This is due to the fact that the duration of collision in the linear spring force model is independent of the maximum overlap, as explained in section 2.6.1

Applying the boundary conditions to specify the constants C_1 and C_2, would lead to the particular solution to the differential equation:

$$r_i = 0.15 r_i \cos(\sqrt{\frac{k_{ij}}{m_i}} t) + R_i + r_j \tag{3.56}$$

The material chosen for both the spherical particle and the walls is glass with a density of 2520 kg/m^3 and the stiffness constant is chosen to be 250 kg/s^2. Radius of 4 mm is assumed for the sphere and since no damping is assumed, the coefficient of restitution is 1. Based on these properties and using eq. 2.22, the duration of contact is calculated to be approximately 0.005 seconds. These parameters are summarized in table 3.1.

Table 3.1: physical properties of the spherical glass bead used in the test and the corresponding duration of contact, linear spring force model

parameter	value	unit
Density, ρ	2520	kg/m^3
Stiffness constant, k	250	kg/s^2
Diameter, D	8	mm
Coefficient of restitution, e	1	–
Duration of contact, $T_{collision}$	0.005	seconds

Results

The results are calculated for each time step by exact analytical solution and numerical solutions, both explicit and implicit, for the total duration of 0.45 seconds. Initially, two different time steps are used for the numerical solutions, one relatively small time step and one relatively big time step. The small time step is 0.0001 seconds, which is less than 50 times smaller than the duration of contact given in table 3.1 and the big time step is 0.001 seconds which is more than 5 times smaller than the duration of contact. In other words each contact is solved in approximately 50 and 5 time steps respectively for the small and the big time steps. The comparison of the results are performed in the arbitrary chosen range of 0.4 to 0.45 seconds. The very start of the simulation is not chosen to be analysed because as the time advances, the numerical error of the integration is being added up and the errors become more representative. The results presented are normalized over the maximum value for the position of the particle and time since the absolute values do not make sense in the comparison.

Fig. 3.6 shows the results of the analytical, explicit and implicit solutions for the small time step. It is observed in fig. 3.6 that at such a small time step both explicit and implicit methods provide a good prediction of the particle's position compared with the analytical solution.

As a next step, the comparison is carried out for the large time step and the results are shown in fig. 3.7. It is observed that the results of the explicit prediction are shifted away from the analytical solution at such a large time step while the implicit results are still consistent with the analytical solution.

What is seen in fig. 3.7 is that the prediction of the explicit method of how the particle moves is quite accurate but there is a delay in the response of the explicit method. In other words, the explicit prediction is slightly behind the analytical and implicit solutions in time. Such responses from the explicit and implicit methods are consistent with the theory too recalling that the explicit method remains insensitive to the changes happening during a time step and is in fact one step behind the reality. The implicit method does not "freeze" as such since the position is updated at each iteration during a time step.

While the explicit method's prediction of the maximum overlap is quite accurate with the contact resolution of 5 as shown in fig. 3.7, it will deviate from the analytical maximum overlap as the time step increases. At the relatively very large time step of 0.0035 seconds, which is in fact close to the duration of contact, the explicit method totally failed with the particle moving outside the walls. The implicit method's error with this time step was still bounded. There is a limit, however, for the implicit method as well in the size of the time step as it cannot exceed the

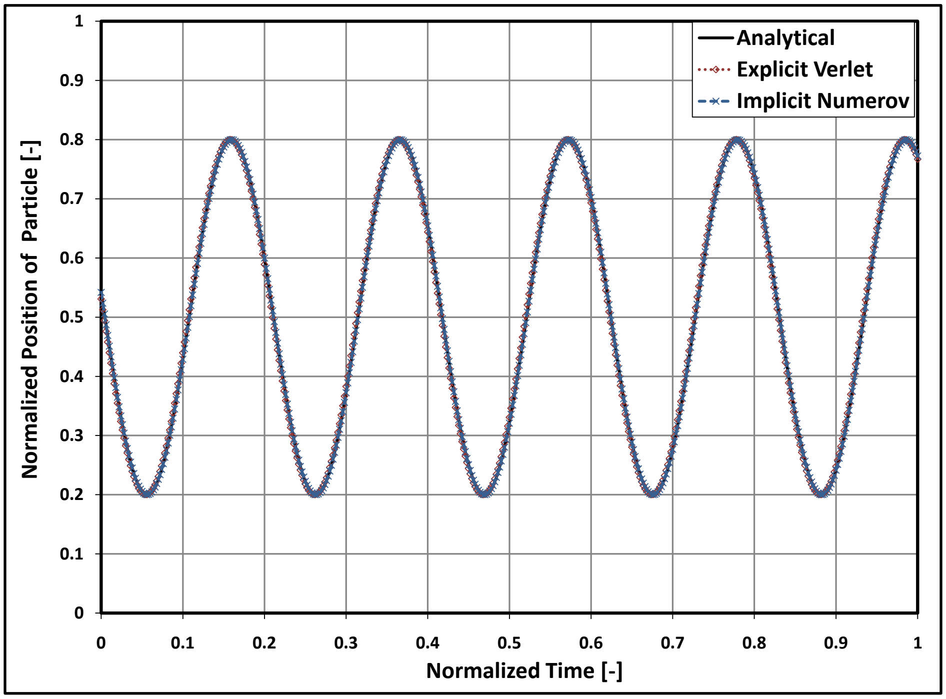


Figure 3.6: Prediction of the particle's position in the sandwich setting by analytical, explicit and implicit methods with a relatively small time step, contact resolution= 50

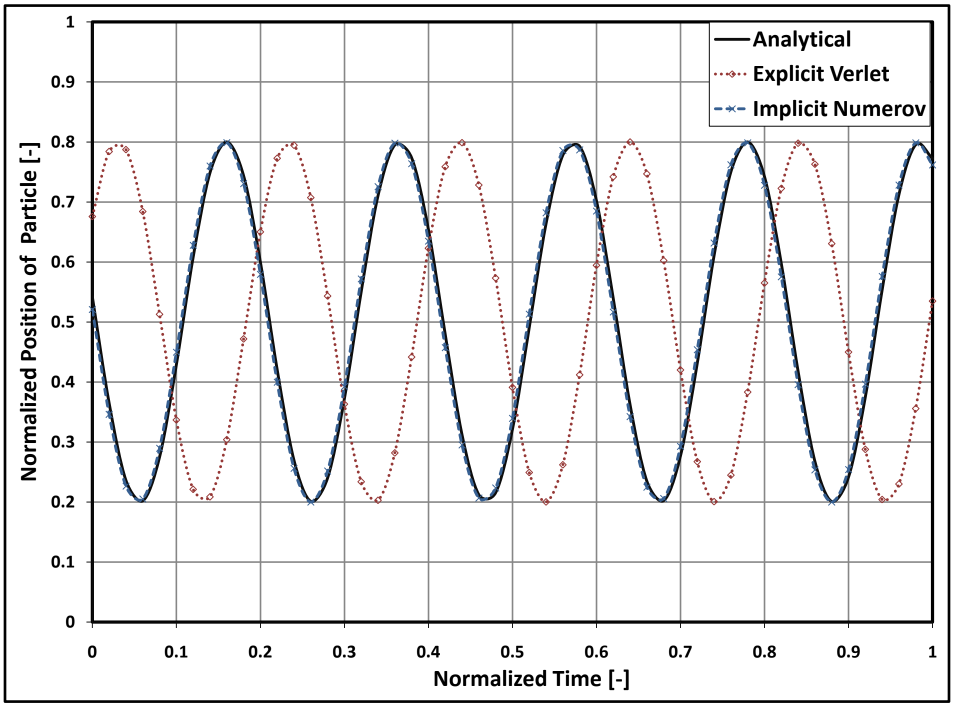


Figure 3.7: Prediction of the particle's position in the sandwich setting by analytical, explicit and implicit methods with a relatively large time step, contact resolution= 5

duration of collision.

The test was performed with other time steps too. Fig. 3.8 shows the biggest relative errors in prediction of the particle's position by explicit and implicit methods for different time steps. Understandably as the time step goes up, the error in both implicit and explicit methods increase but the implicit method predicts the particle's position more accurately using the same or even bigger time steps than the explicit time steps in this case.

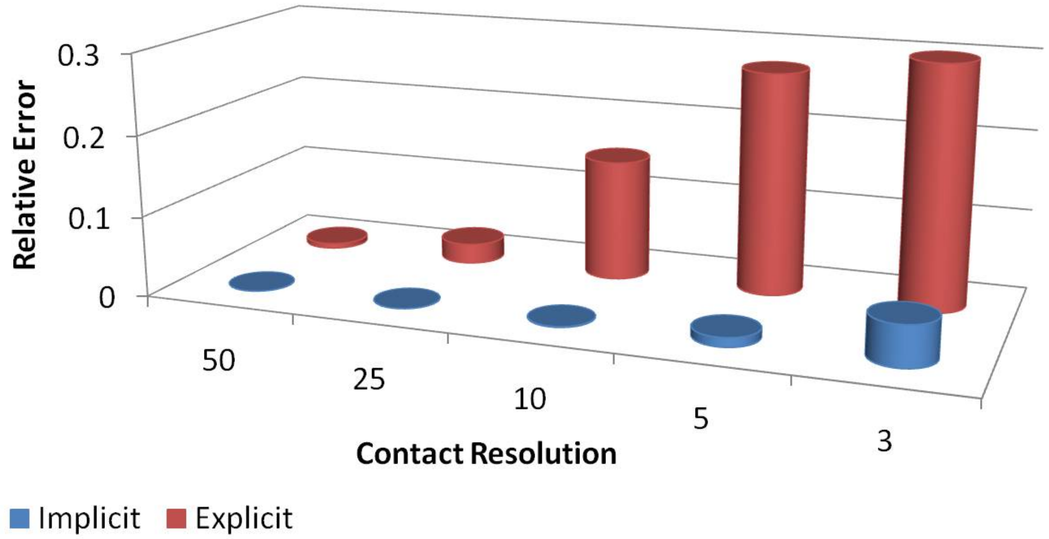


Figure 3.8: Comparison of the relative error of the implicit and explicit methods in prediction of the particle's position in the sandwich setting with different time steps

3.9.2 Contact Detection

In the case presented in section 3.9.1 the distance between the two walls was equal to the particle's diameter. To calculate the overlap with one wall, one could calculate the overlap with the other wall but with a negative sign. The solutions used this fact for simplicity and were dealing only with one wall even though the result was solving the case for two boundary walls. The importance of this is that the solutions did not examine detachment. The particle was always in contact. Therefore this case did not examine how the numerical scheme detects new contacts. This will be examined in the test case presented in this section.

Figures 3.9 and 3.10 show how the implicit and explicit methods, respectively, respond to a particle approaching another object for example a wall. Snapshots 1 to 4 in both figures correspond to consecutive time steps. A relatively large time step is selected in order to illustrate the difference in response more clearly. Snapshot 1 is the initial set up when the particle is moving towards the wall with an initial velocity. At the next time step, snapshot 2, the particle is close to the wall but with no contact yet. Here comes the difference between the implicit and explicit solutions. To calculate the new positions from here, the explicit method uses the current values which indicate no contact and decide the new position. The implicit solution uses the Newton-Raphson iterative method to find the new position and by each iteration it gets closer to the new position and therefore it discovers there will be a contact. At snapshot 2, the implicit solution already knows that there will be a contact in the next time step and it does consider this contact in calculating the new position. In other words, the explicit solution discovers a contact only after it happened while the implicit discovers the contact as it is happening. Due to this early detection of contact, the first overlap calculated by the implicit scheme is smaller than the first overlap in the explicit method as seen in snapshot 3 in figures 3.9 and 3.10. Obviously with a larger overlap, the explicit scheme pushes the particle more forcefully than the implicit scheme which results in a higher rise in snapshot 4.

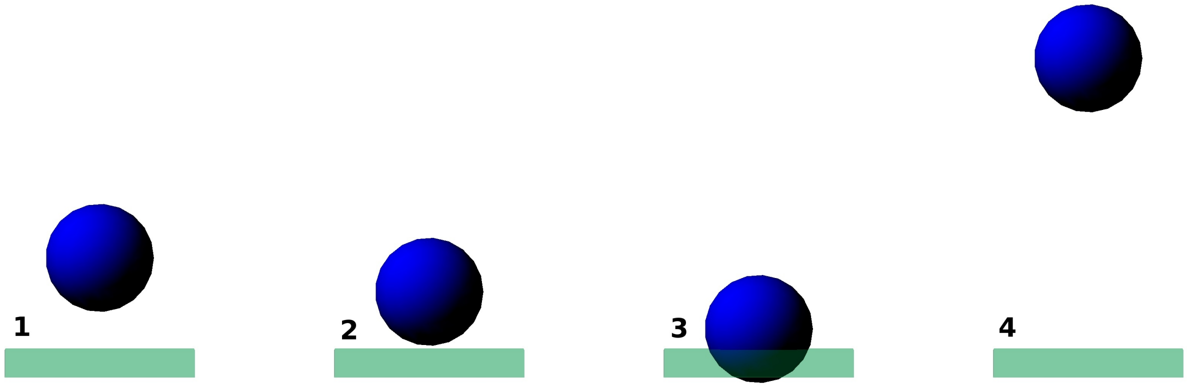


Figure 3.9: Detection of contacts in the explicit solution.

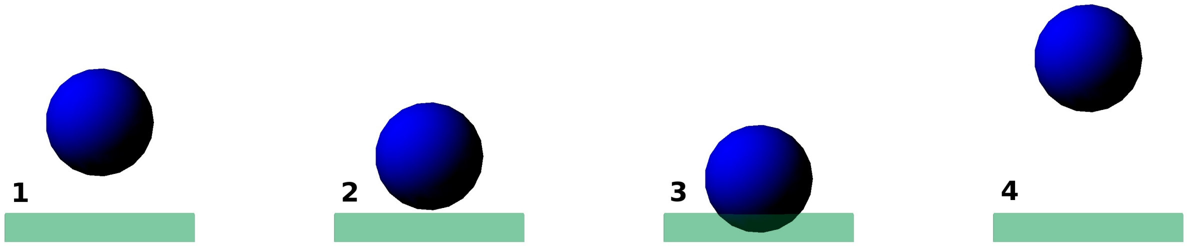


Figure 3.10: Detection of contacts in the implicit solution.

Apart from the size of the time step, the numerical error in this case also depends on how synchronized the time step and the collision start. To clarify this, Fig. 3.11 shows the fraction of a time step before the commencement of the collision. If the collision starts between the time step i and $i+1$, the pre-collision fraction of the time step can be written as:

$$\Delta t_{pre-collision} = \frac{t_{collision-start} - t_i}{\Delta t} \tag{3.57}$$

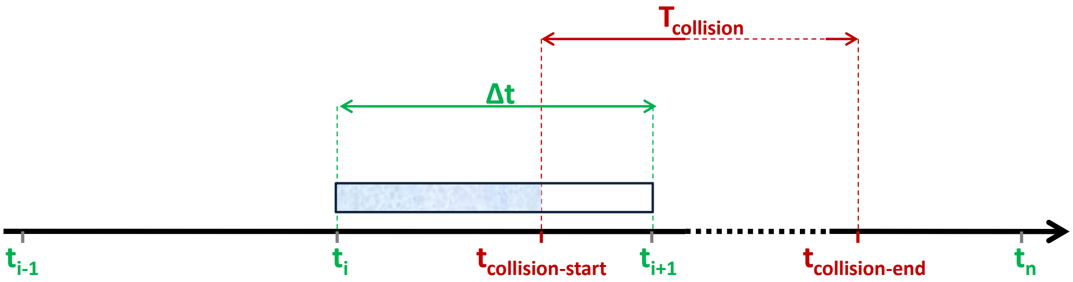


Figure 3.11: Fraction of a time step before the commencement of the collision

In order to discover the influence of $\Delta t_{pre-collision}$ on the prediction results, the simple case of a particle approaching a wall, shown in figs. 3.9 and 3.10, is carried out with different values of $\Delta t_{pre-collision}$. The mechanical and contact properties are the same as shown in table 3.1 and the force model is again the Linear Spring model. Prior to impact, the particle is moving towards the wall with a constant velocity of 0.5 m/s while the gravitational force is excluded. The value of the initial velocity does not have any effect on the duration of contact in the linear spring force model and consequently the resolution of collision and the results are therefore qualitatively independent of the impact velocity. In order to magnify the trend, a large time step corresponding to contact resolution of 1.2 is initially chosen.

Fig. 3.12 shows how the predictions of the explicit and implicit methods depend on the pre-collision fraction of the time step. It shows that the exact time when a collision starts during a time step has less effect on the implicit method than the explicit method. This is explained by the fact that the implicit method can detect a contact during a time step in contrast to the explicit method which only checks for contacts at the start of the time step. Also deduced from fig. 3.12 is that the worst prediction in both explicit and implicit methods is when the collision starts at the same time as the time step starts, i.e. when $\Delta t_{pre-collision}$ approaches zero or one. This is when the integration method is most “shocked” at the next time step because considerable changes in the force are happening during the time step.

Except for small test cases, there is no control, in general, over the start of the collisions in DEM. Therefore, the performance of the integration scheme in detecting the contact should be evaluated for the worst case scenario.

Both explicit and implicit methods’ predictions would improve, of course, with increasing time step. Fig. 3.13 shows the effect of the time step on the maximum overlap predicted by explicit and implicit methods. The test is configured in such a way that the worst case scenario of collision commencement is attained i.e. $\Delta t_{pre-collision} = 0$.

The results in fig. 3.13 show that the differences between the explicit and implicit methods in detecting and solving a contact is mainly in very large time steps. With contact resolutions bigger than 5, the differences become negligible and both methods would give a very good prediction of the maximum overlap compared with the analytical solution.

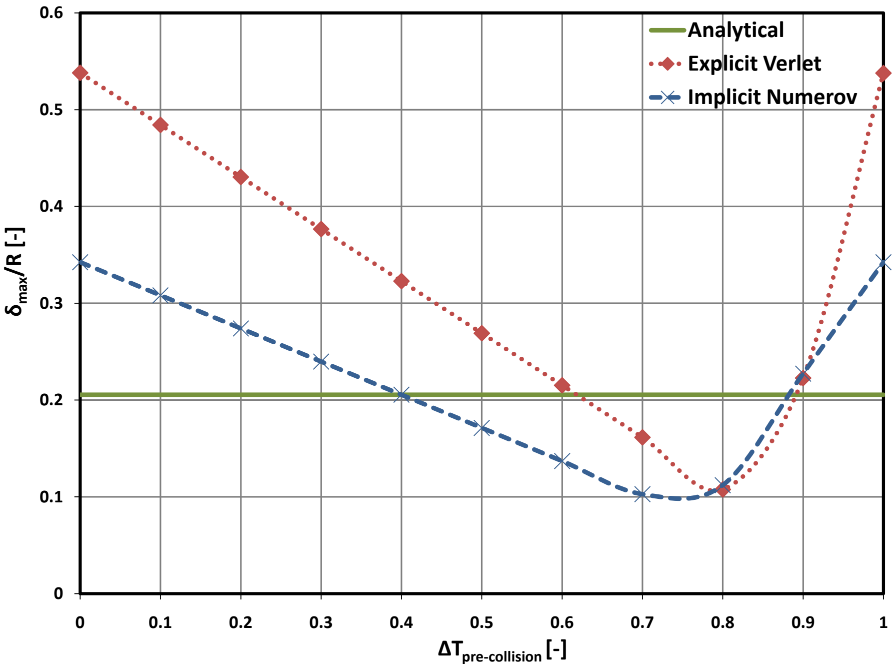


Figure 3.12: Explicit and Implicit methods' prediction of the maximum overlap occurring when a particle approaches a wall in relation to the pre-collision fraction of the time step, contact resolution=1.2

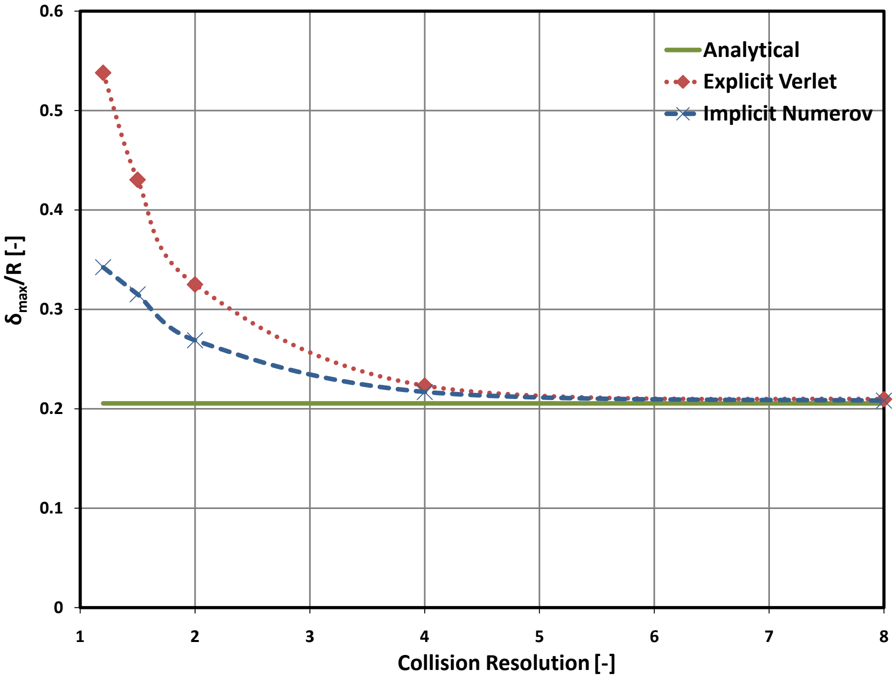


Figure 3.13: Explicit and Implicit methods' prediction of the maximum overlap occurring when a particle approaches a wall in relation to the time step, $\Delta t_{pre-collision} = 0$

3.9.3 Hopper Discharge

A hopper discharge sample case is chosen to validate and compare the results of the implicit method with the explicit method. Hoppers are widely used in the industry for the storage and transport of the granular material. They are not only used as a single storage bin but also often as a linking part between different industrial units. Industries which utilize hoppers range from small scale food or pharmaceutical processing to large scale minerals industry [48]. The engineering design of hoppers, therefore, has been the subject of many studies for many years reported by Jenike, 1967 [47], Enstad, 1975 [28] and Williams, 1977 [111]. More recently the computer simulation tools are utilized to predict the flow of granular particles in a hopper. The case in this section is the sample case introduced by Pöschel and Schwager, 2005 [84]. It is simple enough to enable comprehensive analysis and complex enough to be considered an engineering problem for the implicit method validation. Different tests are performed in order to avoid drawing general conclusions from a single run.

Boundary Conditions

The problem includes 2010 moving spherical particles in a rectangular hopper, as shown in fig. 3.14. The hopper geometry is constructed in such a way that only one layer of particles could be positioned in the depth of the hopper. This means the case is reduced to an almost two dimensional problem. The particles diameter is generated randomly in the range of 8 to 12 mm which leads to a poly-disperse problem. The particles are coloured based on their initial height in the hopper which allows for qualitative evaluation of the motion of the particles by observing the deformation of the stripes during the discharge. The material for the particles and boundary shapes is steel. The gravitational force and the normal repulsive force are considered in this test case while friction and energy dissipation forces are excluded. The linear spring force model is again utilized. The mechanical and contact properties of the particles are summarized in table 3.2.

Table 3.2: physical properties of the spherical steel particles in the hopper discharge simulation and the corresponding duration of contact, linear spring force model

parameter	value	unit
Density, ρ	7850	kg/m^3
Stiffness constant, k	1000	kg/s^2
Diameter, D	8-12	mm
Coefficient of restitution, e	1	–
Duration of contact, $T_{collision}$	0.0032	seconds

Qualitative Analysis

The problem is solved using a relatively small time step i.e. a contact resolution larger than 25 and then is resolved several times each time with a slightly higher time step. With this approach there will be a point, in both implicit and explicit methods, at which the time step is too large that the prediction would fail. What is defined as failure in this qualitative analysis is when the

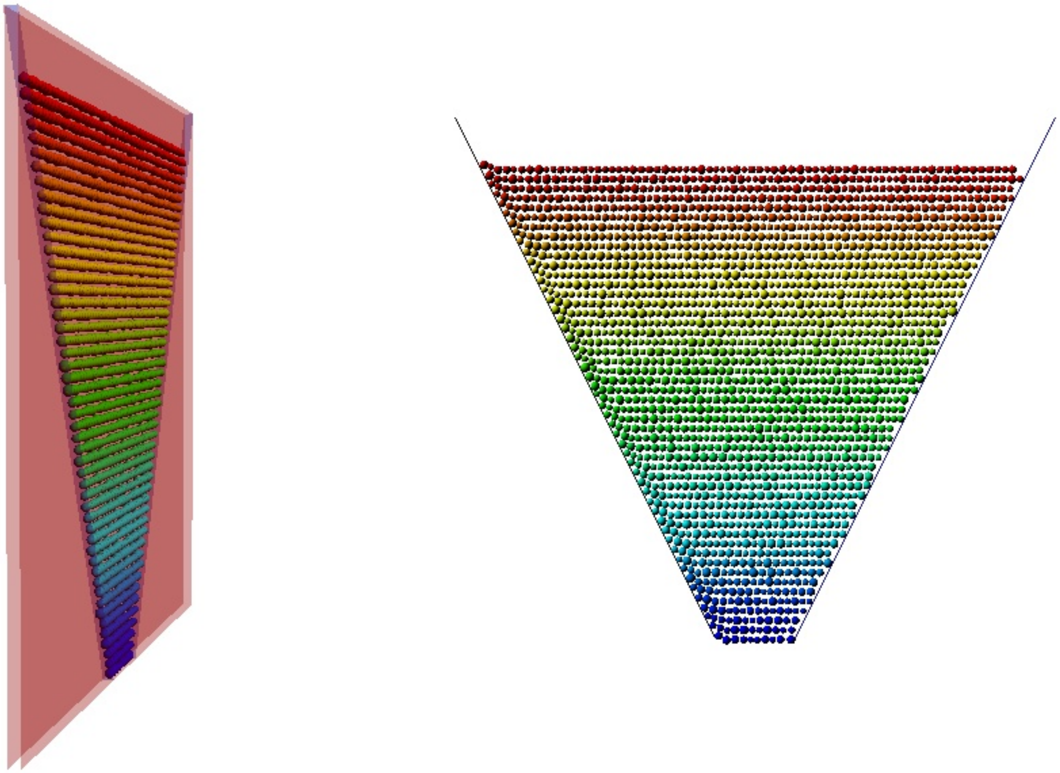

(a) Side view (b) Front view

Figure 3.14: Narrow rectangular hopper resembling an almost 2D problem

animation of results becomes obviously invalid e.g. when one or more particles start to pass the boundary walls instead of bouncing back. In other words, the objective here is to answer the following question: "What is the highest upper limit for the time step where the simulation starts to be visibly invalid?".

It is found out that the explicit method fails at time step of $\Delta t = 0.001$ which corresponds to contact Resolution of 3.2 and the implicit method fails at time step of $\Delta t = 0.0012$ which corresponds to contact Resolution of 2.7. Fig. 3.15 presents Snapshots of the animation of the results with the same contact Resolution of 3.22 for both explicit and implicit methods. The snapshots in fig. 3.15 show the exact same instance of the simulation when few particles in the explicit animation have already passed the hopper wall which is obvious non-physical behaviour while the particles remain inside the boundaries in the implicit snapshot.

Quantitative Analysis

The objective of the quantitative analysis is to investigate differences in the explicit and implicit results in more quantitative and subtle aspects rather than obvious visible non-physical behaviours. A sample of 201 particles, i.e. 10% of total number of particles, with a good distribution of their initial positions throughout the hopper is selected. The parameter chosen as the validation criteria is the mass related mean residence time of the sample particles in the hopper which is defined in

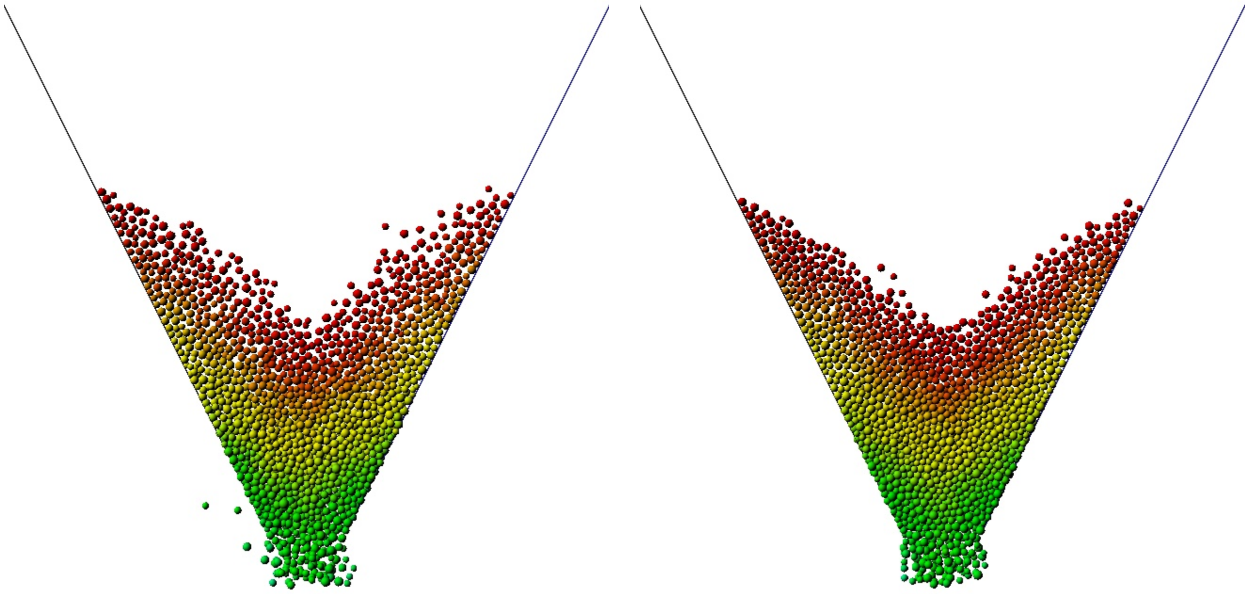

(a) Explicit (b) Implicit

Figure 3.15: Same instance predicted by: (a) explicit method where few particles have obviously passed the hopper wall, (b) implicit method where no particle passes the wall, contact resolution for both cases=3.2

eq. 3.58.

$$t_m = \frac{\sum_{i=1}^{n} m_i.t_i}{\sum_{i=1}^{n} m_i} \tag{3.58}$$

in which t and m stand for the residence time and mass respectively, n is the total number of sample particles and subscript m correspondence to the mean value. In both explicit and implicit methods, the simulation is performed four times. These four cases are essentially the same with one difference i.e. the initial positions of the particles in the hopper are regenerated each time. Since the particles are generated randomly in the specified range, the initial positions are not the same each time even though they are qualitatively similar and contain 2010 particles distributed throughout the hopper. Disturbing the initial positions is done intentionally as an attempt to avoid single analysis and rely on a number of repeated simulations while a small disturbance is introduced each time. This approach is applied to simulations with five different time steps in both explicit and implicit methods. As a result, a total of 40 different simulations are performed for this quantitative analysis.

Fig. 3.16 shows the results of the mean residence time of the particles in both explicit and implicit methods and using different time steps. For each time step, the mean residence time is calculated as the average of the four cases.

The results presented in fig. 3.16 do not indicate any advantages for neither implicit or explicit methods over the other. As the time step is increased the mean residence time deviates from its reference value, which is its value at the small time step here, but this happens quite equally for both explicit and implicit methods. This confirms the hypothesis that the differences between the explicit and implicit methods are mainly recognizable at big time steps. The small gain in accuracy

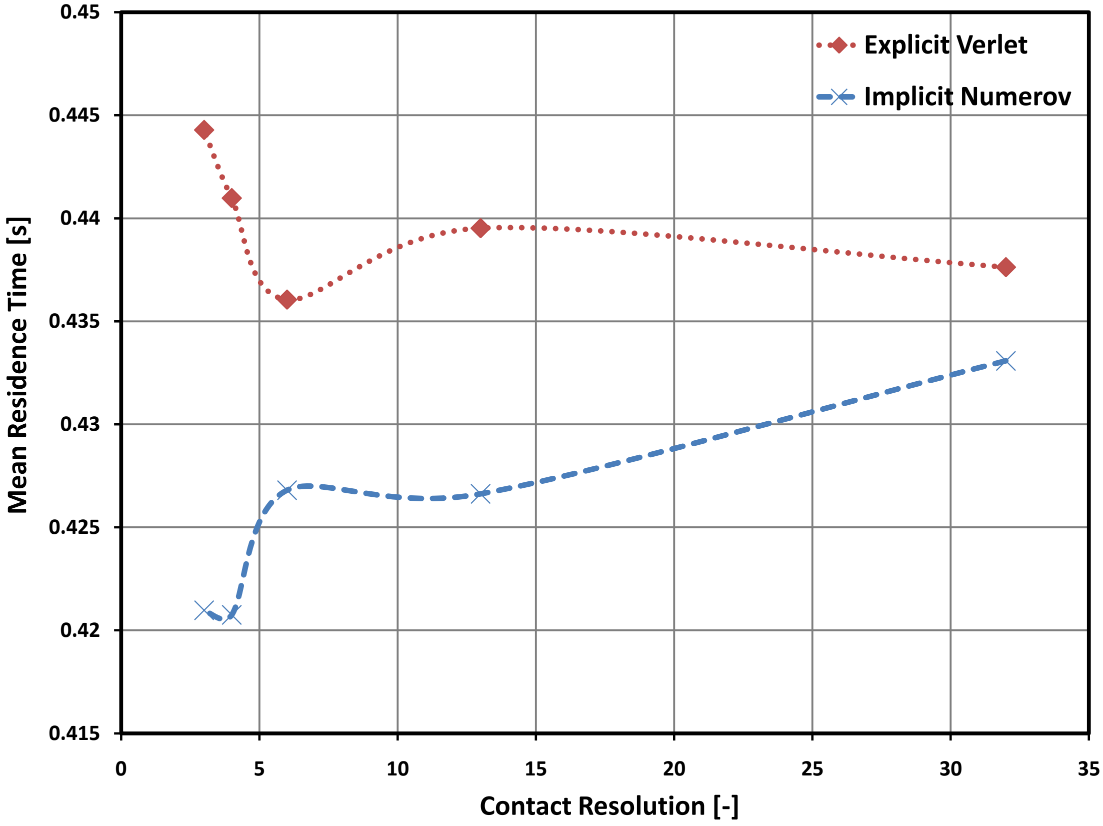


Figure 3.16: Mean residence time of the particles in the hopper

in moderately small time steps do not seem to have a decisive impact on the overall behaviour of the granular particles, at least for the case of hopper discharge.

3.9.4 Qualitative Analysis of Backward Acting Grate

As a final verification test, the implicit method is applied to a backward acting grate device to predict the motion of solid fuel particles on a grate. Backward acting grates are widely used in incineration plants such as waste-to-energy plants. The analysis only involves the dynamics of the solid particles on cold grates which means the chemical conversion due to heating and air flows are not part of this study. Energy dissipation in the normal direction of impact is included in this study based on eq. 3.29. In a similar manner as explained in section 3.4.2, friction forces are also included based on a backward Euler approximation. No comparison with explicit results are performed in this test and verification is carried out in comparison with the experimental results. The experimental results are taken from a study by Beckmann et al., 1995, [6]. The primary objective of this analysis is to test and verify the implicit method. More advanced analysis of backward and forward acting grates are presented in chapter 4.

Experimental Device The backward acting grate simulated in this study is a plexiglass grate device in Clausthal, Germany [5]. The configuration and geometry of the device are summarised in Table 3.3.

Parameter	Value
number of grate bars	11
bar length	190 [mm]
bar height	30 [mm]
bar width	1000 [mm]
grate inclination	25 °
bar inclination	45 °
moving bar amplitude	30 [mm]

Table 3.3: Major geometrical settings of the experimental backward acting grate

A photo of the device is shown in Figure 3.17 with the kind permission of the personnel at CUTEC institute in Clausthal [20]. Every other grate bar is a moving bar.

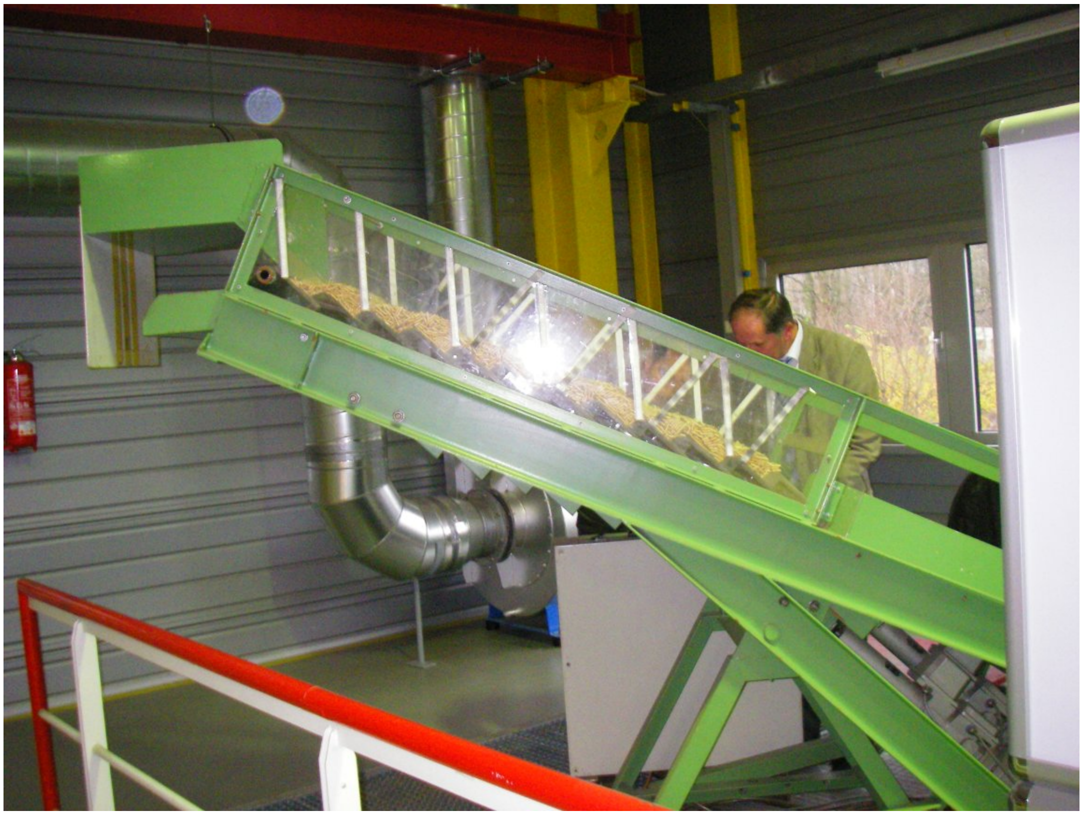


Figure 3.17: Backward acting grate in Clausthal, Germany

Computational Model The same geometrical settings as the parameters in Table 3.3 is applied to the computational model except the width of the grate bars which is reduced to enable faster simulations. The effect of reducing the width of the grate bars on the residence time of particles is discussed in chapter 4. The moving bar velocity is also increased in the numerical study in order of 10 times the experimental study. This is done to enable achieving results quicker in the simulations. This assumption will affect the residence time in such a way that a direct quantitative comparison with the experimental results would not be possible but a qualitative comparison and analysis is still valid because approximately the same bed height is maintained on the grates despite the increased velocity. This is achieved by increasing the mass flow. The moving bar velocity is fixed at 35 mm/s.

The particles are selected to be spherical wooden spheres of 15 mm diameter to be consistent with one of the experiments performed by Beckmann et al. [6]. The mechanical properties of the spherical particles and grate bars are shown in table 3.4.

	particles	grate bars and walls
Density $[kg/m^3]$	550	1190
Normal stiffness $[kg/s^2]$	10^4	10^4
Normal dissipation constant $[kg/s]$	2.7	3.3
Tangential dissipation constant $[kg/s]$	1.2	1.5
Sliding friction coefficient	0.3	0.5
Rolling friction coefficient	0.0015	0.0015

Table 3.4: Mechanical properties of the particles and grate bars with linear spring-dashpot force model

The linear spring-dashpot model, with the coefficients shown in table 3.4, leads to a constant coefficient of restitution of 0.5 for particle-particle contacts and 0.55 for particle-boundary contacts. The implicit Numerov method with the time step of 10^{-4} $[s]$ is employed in parallel computing and domain decomposition techniques are applied. Figure 3.18 shows the decomposition of particles on the backward acting grate into four sub-domains.

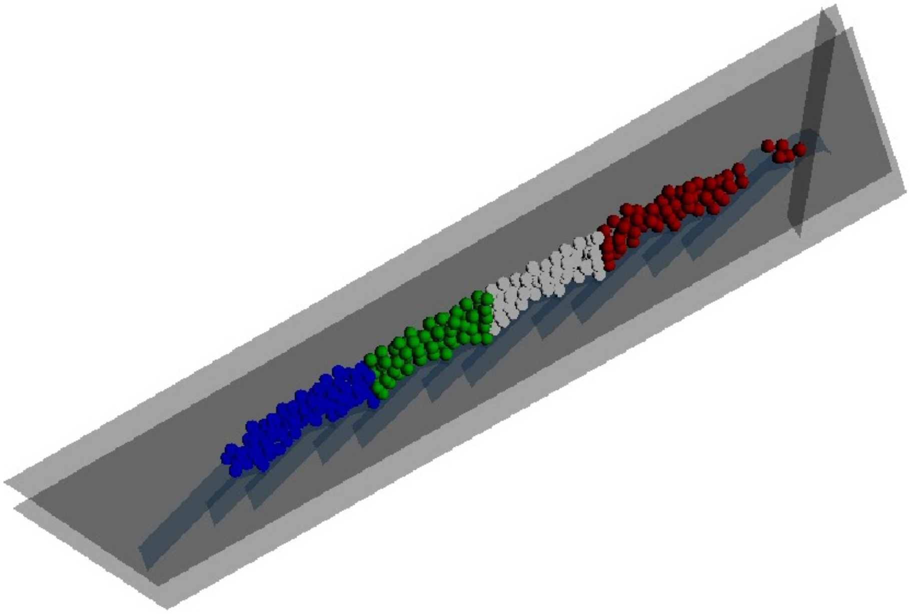
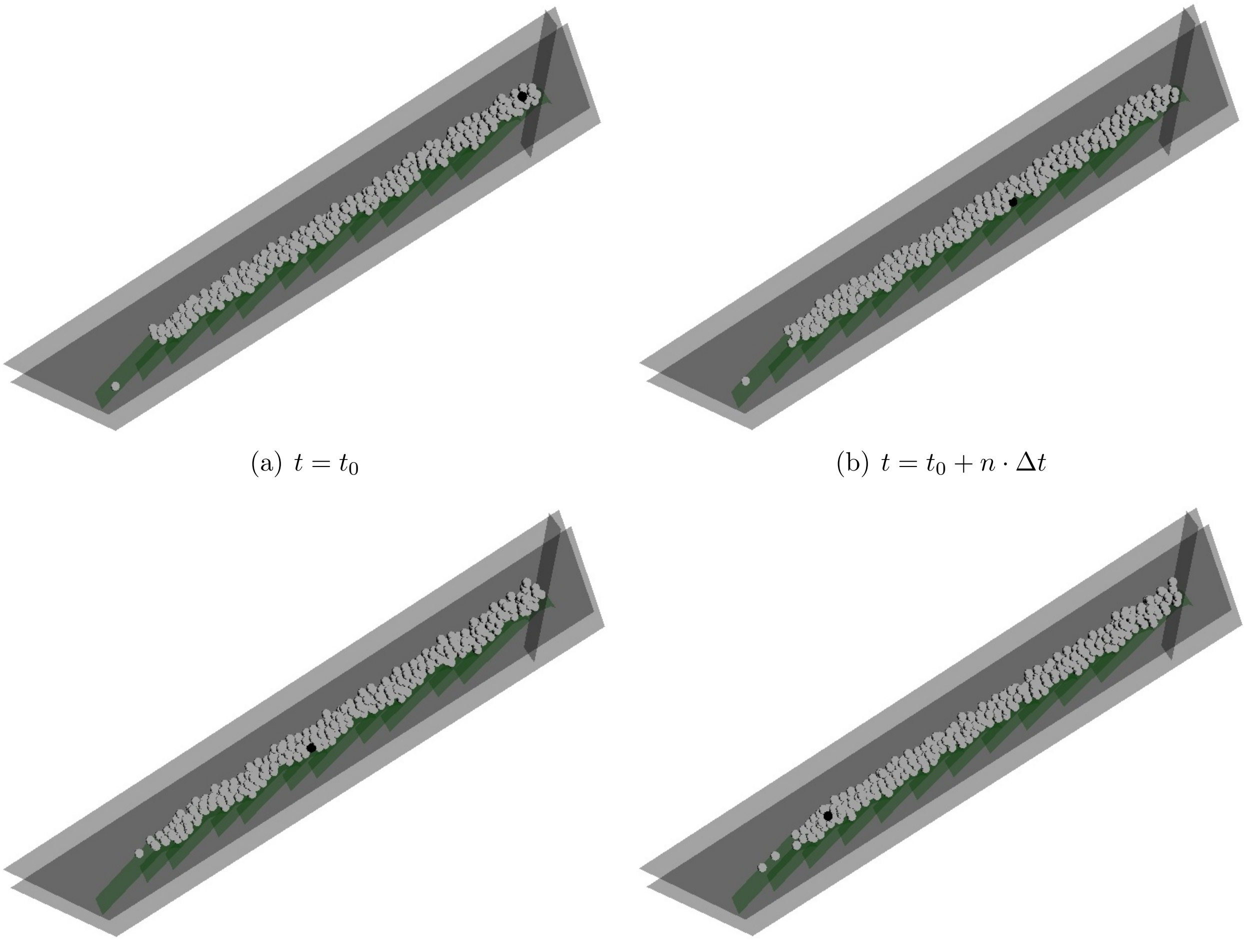


Figure 3.18: Particles on the backward acting grate coloured according to their corresponding sub-domains.

Different simulations are performed in order to evaluate the performance of parallel implementation, the residence time distribution and also to independently analyse the effect of mass flow on the residence time of particles on the grates.

Results

Residence Time Distribution The main simulation performed is a case where the width of the grate bars are reduced to accommodate 2 particles along the width of the bars. In other words, the width of the bars is adjusted equal to the diameter of two particles plus some gap between them. Two side walls confine the particles in this range and prevent them from falling outside the width of the grate. The particles are fed into the device at the first bar by a constant mass flow of 13.6 kg/hr. This will lead to an almost fixed bed height on the grate equal to 33 mm and more or less constant number of particles of 300. The residence time for a sample of 148 particles is recorded. Figure 3.19 shows snapshots of the animation of the results with one particle coloured to demonstrate the flow at fixed time intervals.



(a) $t = t_0$

(b) $t = t_0 + n \cdot \Delta t$

(c) $t = t_0 + 2n \cdot \Delta t$

(d) $t = t_0 + 3n \cdot \Delta t$

Figure 3.19: Snapshots of the animation of the backward acting grate with one particle specified in black to demonstrate its motion in equal time intervals

Figure 3.20 shows the mass increase of the sample particles which complete the flow from the feed point till the end of the grate with time. Both experimental and computational results are illustrated for qualitative comparison. The prediction is based on normalized parameters so that the behaviour of granular particles can be analysed qualitatively. The horizontal and vertical axes in Figure 3.20 are normalised over the mean residence time and the total mass of the sample, respectively. The normalization is applied to each curve, i.e. the experiment and DPM prediction, separately.

The residence time distribution is illustrated in fig. 3.21. The horizontal axis is again time normalized over the mean residence time and the y axis shows the mass fraction of the sample which have that residence time. The vertical axis in Figure 3.21 is not the cumulative mass as it is in fig. 3.20. What fig. 3.21 shows is the mass fraction of the particles which have a certain residence time in relation to the mean residence time.

Figs. 3.20 and 3.21 indicate quite good agreement between the computational and experimental results. The curve in fig. 3.21 shows a small second peak at the normalized time of around 1.45. The second peak might be explained by the fact that the particles can be divided into two categories based on their behaviour on the grate. The first are those who form the top layer of the packed bed and are less affected by the backward motion of the bars. These particles normally have less residence time than the particles at the bottom of the packed bed. The particles at the bottom form the second category which are more affected by the backward motion of the bars especially at higher bar velocities [71]. Since the bar velocity is increased in the numerical part of this study, as explained earlier, this effect is only present in the DPM prediction and not the experimental part. The second peak could be even more prominent in poly-disperse systems where segregation of particles is intensified by differences in size.

Effect of Mass Flow Two additional predictions are preformed to study the effect of decreasing the mass flow on the behaviour of particles on the backward acting grate. Fig. 3.22 shows how the mean residence time decreases as the mass flow increases in both experiments and computations. The agreement between the experimental results and computational model is quite reasonable. In both experiment and DPM prediction curves in fig. 3.22, the mass flow is normalized over the maximum mass flow and the mean residence time is normalized over the maximum value of that particular curve.

Fig. 3.23 shows the relationship between the change in the fixed bed height and the change in the mass flow. Quite good consistency between the computational results and experiments is inferred from fig. 3.23.

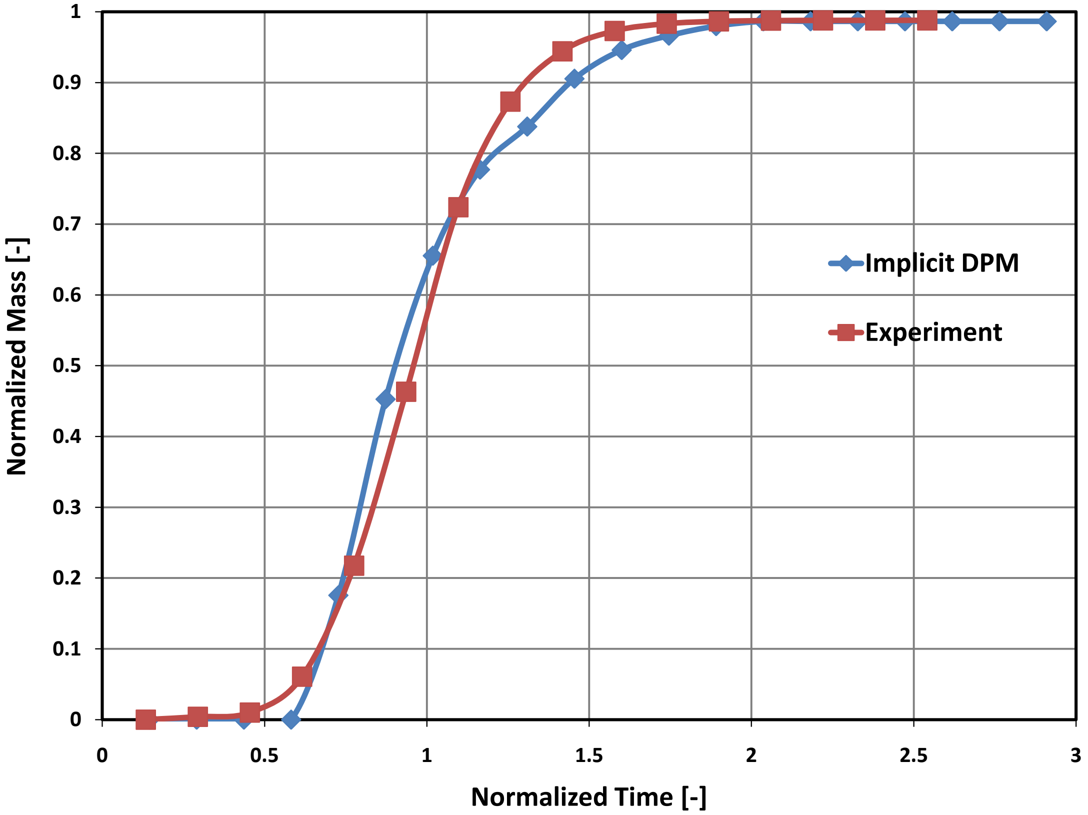


Figure 3.20: Normalised cumulative mass of particles completing the flow on the backward acting grate in relation to the time normalised over mean residence time.

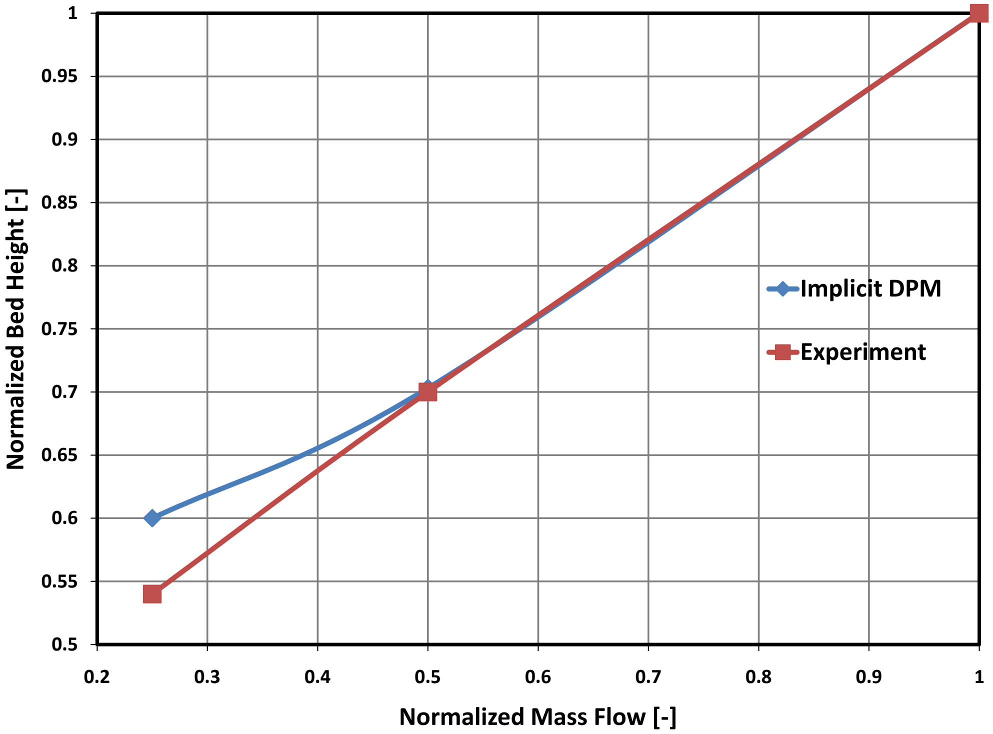


Figure 3.23: Height of the flue bed in dependence on the mass flow.

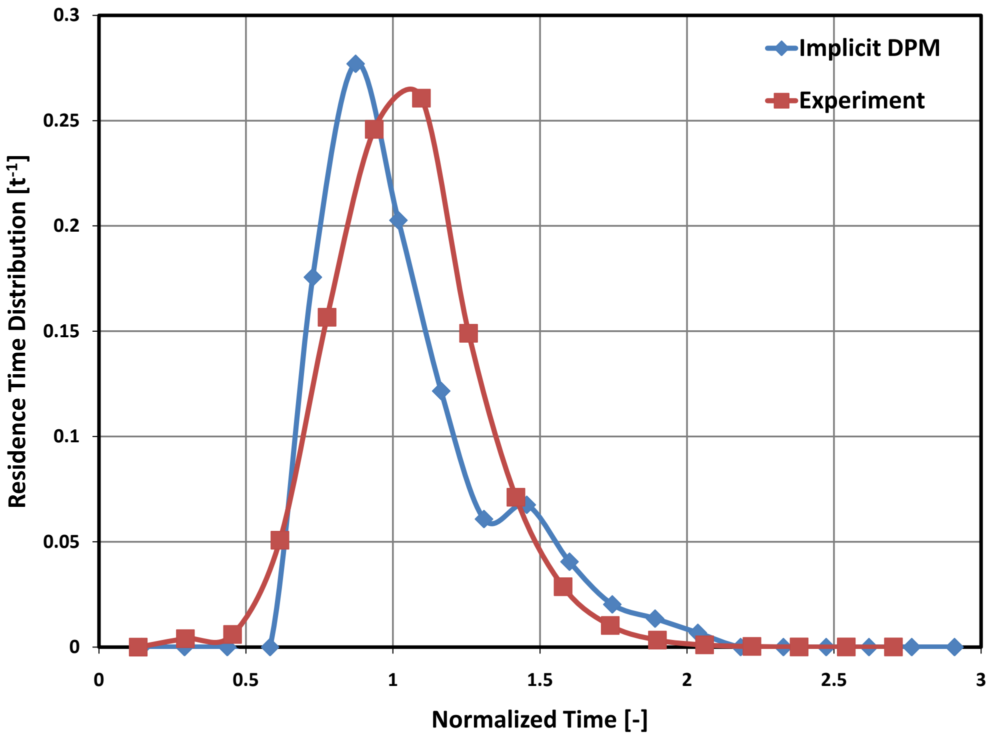


Figure 3.21: Residence time distribution of particles on the backward acting grate with horizontal axis normalised over the mean residence time.

The graphs presented in this section are based on normalized values. The simulation results are normalized over the maximum or mean value of the simulation and the experimental results are normalized in reference to the maximum or mean value of the experiments. Based on such a normalization, it is possible to compare the experimental and simulation results qualitatively. For instance, the plot in fig. 3.23 shows that doubling the mass flow, will increase the bed height 1.43 times (1/0.7) in both experiment and simulation but no information about the absolute values of the mass flow and the bed height can be deduced from the plot. Similarly, the plots in figs. 3.20 and 3.21 can not be consulted to compare the mean residence time in experiment and simulation but does provide information to compare the distribution of the residence time around the mean.

Parallel Performance The performance of the parallel implementation has been tested on the choas.lu cluster consisting of 434 cores with 1 GbE connection. For the purpose of measuring the parallel efficiency, the simulation process was run for 5 seconds both on the sequential and the parallel versions of the program. The flow of particles on the grate is almost constant so running the simulation for long time is not important to measure the efficiency. For running the parallel version, the number of processors used are equal to the number of sub-domains. The execution time and the speed-up of the parallel algorithm are presented in table 3.5. The parallelization efficiency can be calculated based on the method described by Karp et al., 1990 [52] and are presented in fig. 3.24. The efficiency of the parallel run is very good and considerable savings in execution times are achieved based on the information given in table 3.5 and fig. 3.24.

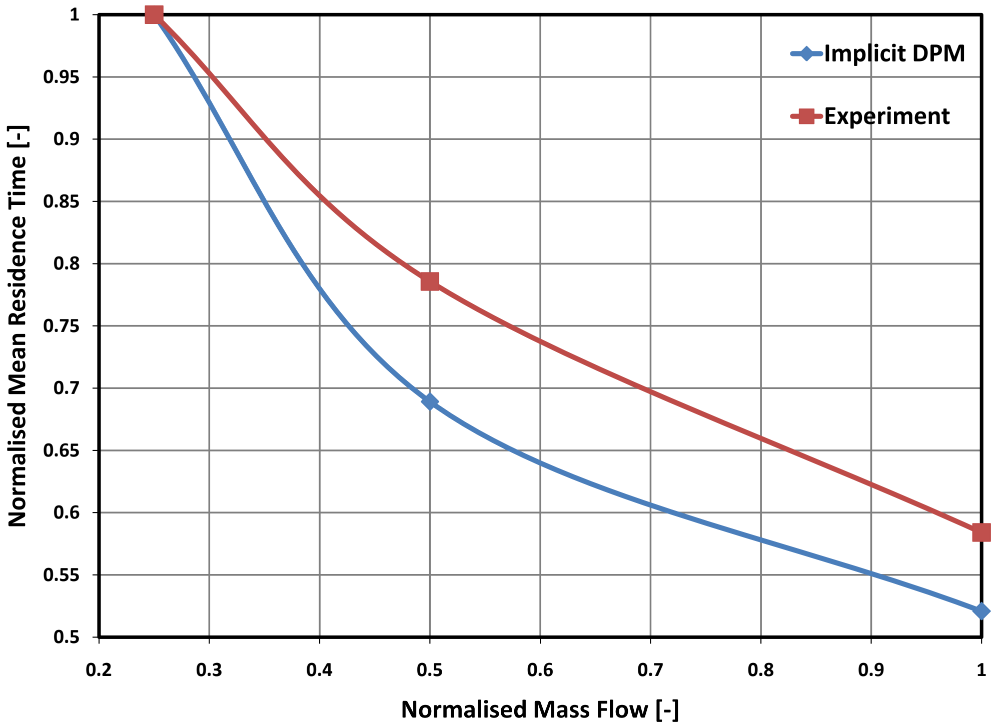


Figure 3.22: Influence of the mass flow on the mean residence time of particles on the backward acting grate

Number of Processors	Execution Time (seconds)	Speed-up
1	71705	-
2	40790	1.76
4	21366	3.36
8	10946	6.55

Table 3.5: Speed-up of the parallel run

3.10 Discussion

Overall, the comparisons and results presented in sections 3.4 and 3.9 show that employing an implicit method enables larger time steps to be used in comparison with an explicit method. However, the differences between the two methods become important only at relatively large time steps, i.e. contact resolutions higher than 5, when explicit methods become unreliable and predict non-physical responses. As long as such failures do not occur, explicit methods are sufficiently accurate in moderately small time steps as the results in section 3.9.3 indicate. The gain in accuracy achieved by implicit integration in smaller time steps do not seem to have a decisive impact on the qualitative behaviour of the granular particles.

In addition, employing a big time step, even in implicit integration, has its own challenges. In all cases studied here, a linear force model is used in which the duration of contact can be predicted

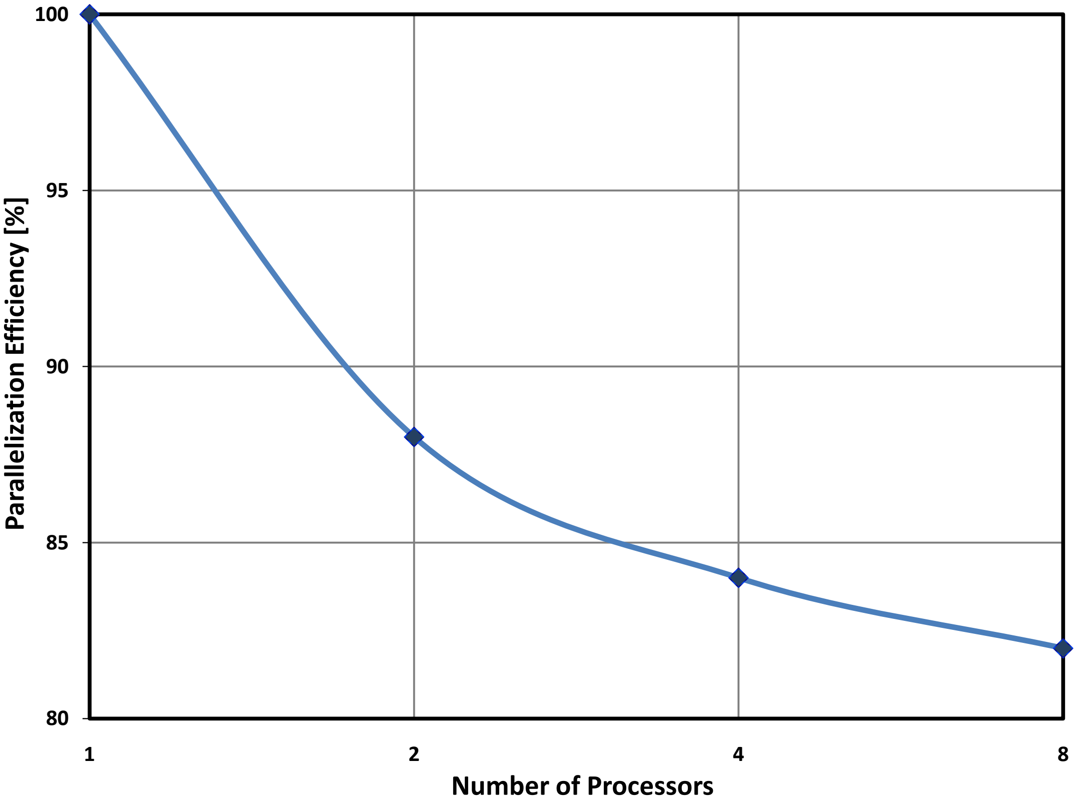


Figure 3.24: Parallelization efficiency in relation to the number of processors

exactly based entirely on the material properties. However, if a non-linear force model is used, the duration of contact can only be vaguely anticipated and therefore the time step needs to be decided considering a safety margin for uncertainties. Confirming the same experience of Schäfer et al., 2010 [93], there is also a risk of non-convergence of the Newton-Raphson method when using big time steps. This problem can be dealt with by adapting the tolerance of the Newton-Raphson method though this solution comes at the cost of losing some accuracy.

Overall, the advantages of using an implicit method seem to be quite small compared to the additional computational costs. With the current state of the art where the computational burden of the force computations are a major obstacle in DEM, any integration method which requires more than one force computation per time step is not recommended. Predictor-corrector schemes instead are recommended which are in fact inexpensive implementations of implicit schemes requiring only one force computation per time step. Other methods to reduce the computational costs such as assembling the Jacobian less frequently, reported by Schäfer et al., 2008 [94], mitigate the costs moderately but also the accuracy. Such methods are not recommended because they still require solving the non-linear system which is quite a costly operation in large systems.

Chapter 4

Analysis of Forward and Backward Acting Grates

4.1 Introduction

On average, each person in the European Union (EU) throws away 6 tons of solid waste every year [30] and the number is on the rise. The Organization for Economic Cooperation and Development (OECD) estimates that by 2020 the amount of waste generated in Europe would be 45% more than in 1995. The primary recommended option by the European Union for treatment of such waste remains incineration [17]. Incineration of solid waste in so called "energy to waste" plants is not only a way to dispose of waste but also to recover its energy content. With increasing awareness towards resource scarcity, incineration and co-incineration of waste with other fuels, are emerging increasingly as an energy production option as well as waste disposal. The main challenge, just like many other alternative fuels, is the technological efficiency. The EU waste incineration directive sets strict guidelines which aims to reduce the negative impacts of waste incineration on the environment [29]. This is to be achieved by improving the operational conditions and technical performance of the incineration plants.

Forward and backward acting grates are widely used in waste-to-energy plants [35, 99]. There are various types of grate firing technologies including stationary, travelling, reciprocating and vibrating grates [114]. All types function with the same main principle i.e. the fuel particles are introduced at one side of the grate and are burned while transported to the other end of the grate. The transport of the particles is either solely due to gravity in case of stationary grate or is supported by grate movements. The characteristics of Municipal Solid Waste (MSW) and biomass as the feedstock to grate firing systems is often not well defined and determined. This makes the process control and design difficult. Numerical simulation techniques are one way of modelling for such systems. Simulation makes it it easier to study the effect of different parameters on the response of the system. The present chapter presents the application of the Discrete Element Method (DEM) to forward and backward reciprocating grates.

An initial attempt in this direction was presented by Peters et al. [80] where a two dimensional DEM study was applied to a forward acting grate. This work was further developed by presenting residence time analysis of the particles [25] and later on by presenting comparisons with experimental results of a forward acting grate test facility [26]. The latest work in this direction also included conversion of a packed bed of fuel particles on a forward acting grate by coupling the Discrete Element Method with a CFD-approach [79].

In the literature, there have been few other authors who have applied numerical simulation tools to forward and backward acting grates. Simsek et al., 2009 [95] simulated the motion and chemical conversion of a packed bed of particles on a forward acting grate. The motion of the particles were simulated by two dimensional Discrete Element Method coupled with three dimensional CFD analysis for the conversion part. This preliminary work was further developed by Sudbrock et al., 2011 [100]. Their work reports the simulation of a simplified grate with no inclination, no feeding and no discharge of the mono-disperse particles and presents comparisons with experimental results. Kruggel-Emden et al., 2007 [58] studied the mixing performance on different grate designs by three dimensional DEM though a parameter study on the influence of different operational parameters is missing.

The work presented in this chapter develops the motion part of the previous works further by introducing three dimensional DEM analysis of both forward and backward acting grates. The residence time distribution of the solid particles on forward and backward acting grates are compared with each other and the effect of different parameters including the mass flow and the particles material are studied. The results of the DEM analysis are directly compared to available experimental results. The analysis here excludes heating and conversion of the particles and is only concerned with the motion of the test particles on an experimental pilot plant as a basis for later combustion.

4.2 Experimental Study

In order to enable direct comparison and validation of the numerical results, the simulation is carried out on a test pilot grate where the residence time behaviour of the particles on the grate was already determined previously by experiments [6]. The device is situated at the CUTEC institute in Clausthal-Zellerfeld, Germany [20, 5]. A photo of the plant is shown in fig. 3.17.

While fig. 3.17 shows the device configured as a reverse acting grate, the same device is convertible to a forward acting grate. The functional difference between the forward and reverse acting grates is that for a forward acting grate, the moving bars strike the particles in the same direction as the grate inclination while in the reverse or also called backward acting grate, the particles are struck in the opposite direction of the grate inclination. Fig. 4.1 illustrates the schematic configuration of both forward and backward acting grates.

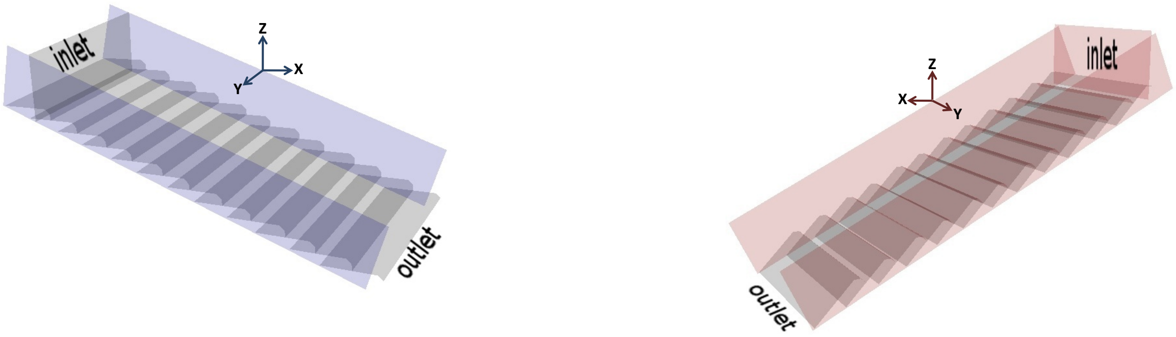


Figure 4.1: Configuration of forward (left) and backward (right) acting grates

The experimental grate consists of 11 bars where every other bar is moving back and forth with a constant velocity. There is a small pause after the moving bar completes a half-cycle and before changing direction to the opposite. The particles are fed into the grate at the inlet at a constant feed rate. The grate's geometric specifications are summarized in Table 4.1 for forward and backward configurations.

	forward acting grate	backward acting grate
number of grate bars	11	11
grate inclination	13 °	25 °
bar inclination	7 °	45 °

Table 4.1: grate specifications for forward and backward configurations

The grate bars are made of acrylic. Different test particles were used in the experiments including swelling clay particles, wooden spheres and ceramic spheres. The information about these particles is not complete and there are few uncertainties concerning the shape, the size and material density of some group of these particles. Therefore few assumptions were undertaken regarding the particles physical characteristics as follows:

- The swelling clay particles are modelled as spheres and their material density is assumed to be 700 kg/m^3 based on the measurements carried out on swelling clay granulates obtained from garden shops. The dependency of the particle density on the particle size is assumed to be negligible.
- The diameter of the swelling clay particles used in the experiments are reported to be between 8 and 25 millimetres but the exact size distribution is not available. Gaussian distribution is assumed for this size range in which the majority of the particles are distributed around the mean [66]. Based on this assumption, the particles in the computational model are generated randomly in the half middle range of the reported size range which is between 12 and 21 millimetres in diameter.
- The ceramic spheres are reported to be 15 millimetres in diameter with single particle mass in the range of 2.25 to 2.44 grams which is not consistent with the reported particle density. The particle diameter is, therefore, assumed to be 12.6 millimetres in order to be consistent with the rest of the data.

Based on these assumptions, the physical characteristics of the particles used in this study are summarized in table 4.2.

Material	Particle diameter $[mm]$	Bulk density $[kg/m^3]$	Material density $[kg/m^3]$
Swelling clay	12 - 21	400	700
Wood	15	350	550
Ceramic	12.6	1300	2200

Table 4.2: Particles' physical data

For further details of the procedures and methods of the experimental study, one may refer to the original article by Beckmann et al. [6].

4.3 Computational Domain

In order to limit the number of particles included in the simulation and therefore the computational costs, the simulations are performed only on a small central portion of the grate width and not the entire width. This means reducing the computational domain along the Y-axis in fig. 4.1. According to Lim et al. [64] and also based on the experiments performed by Hunsinger et al. [42], the motion of the granular material on moving grates is dominated by the movement along the grate inclination. It is also confirmed in this study, section 4.7.1, that the results of the simulation of a narrow grate is almost identical to the results of the same but wider grate. Accordingly the two side walls which are placed to direct the motion of the particles have given the same mechanical and friction properties as moving particles because they are replacing the particles which are now excluded in the reduced domain. The mass flow used in the simulations is also scaled down with the same ratio as down-scaling of the domain. This is based on the assumption that the flow of the new material through the inlet is evenly distributed throughout the entire width of the grate.

A question which arises is how much the computational domain along the width can be reduced without degrading the validity of the method. An important property to preserve when reducing the computational domain is the bulk density of the granular material. Bulk density is a crucial factor determining the bed height and consequently the residence time of the particles on the grate according to eq. 4.1:

$$\rho_{bulk} = \frac{m_{bulk}}{V_{bulk}} = \frac{\dot{m} \cdot t_m}{l \cdot w \cdot h} \tag{4.1}$$

where ρ_{bulk}, V_{bulk}, m_{bulk} represent the bulk density, bulk volume and mass of the packed bed in steady state conditions, l and w being the length and width of the grate respectively, h the bed height, t_m the mean residence time and $\dot{m}$ represent the mass flux. In the computational domain where the grate width and the mass flow are scaled down by the same ratio, the bed height and residence time would be unaffected as long as the bulk density remains unchanged. The bulk density of the granular material when they freely settle does not depend on the size of the container for sufficiently large containers [39]. However if the width of the grate is considerably reduced, the bulk density could be affected. Therefore, the objective here is to investigate theoretically the effect of reducing the grate's width on the bulk density of the particles, at least to some reasonable degree of accuracy. This is important in order to select an appropriate width for the computational domain. To explore this, it is necessary to realize how particles settle on the bars. Fig. 4.2 shows the top view of the arrangement of some mono-disperse particles on a single bar of a relatively wide grate. As a general phenomena, even in the case of particles with high rolling resistance, the next row of particles in relation to the X-axis do not position themselves at the same Y co-ordinate as the preceding row; they fall down into the void space between the two supporting particles.

Since the particles' arrangement pattern is repeated after each two rows of particles along the length of the bar, the bulk density can be compared only for the first two rows. By a similar reasoning, only one layer of particles along the height (Z axis) is considered. Such an alignment of the particles shown in fig. 4.2 represent the maximum possible bulk density for one layer of particles along Z axis. As illustrated in fig. 4.3, the bulk volume of the particles for such a configuration can be written as:

$$V_{bulk} = l \cdot w \cdot h = (R + R + \sqrt{(2R)^2 - R^2}) \cdot (2R \cdot n) \cdot (2R) = 14.93 \cdot n \cdot R^3 \tag{4.2}$$

where R is the radius of the particles and n represents the ratio between the bar width and the particle's diameter. In other words n is the maximum number of particles which could be

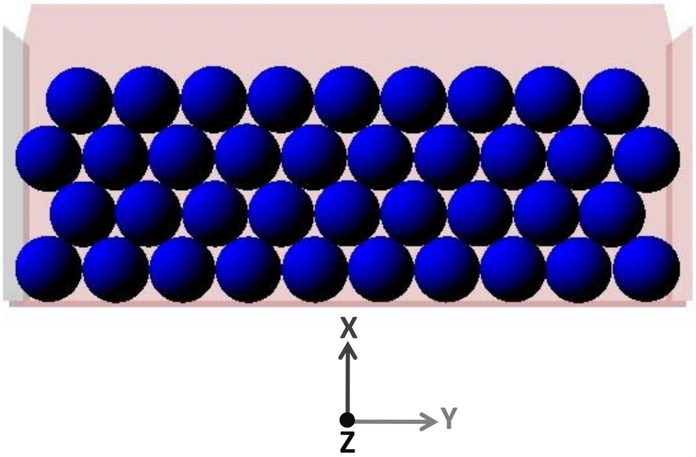


Figure 4.2: Maximal packed arrangement of particles along the grate width, view from top

positioned in one layer along the width of the bar. The width written as $(2R \cdot n)$ in eq. 4.2 is an approximation because the width is not necessarily a multiple of the particles diameter but the remainder can be neglected for a sufficiently wide grate.

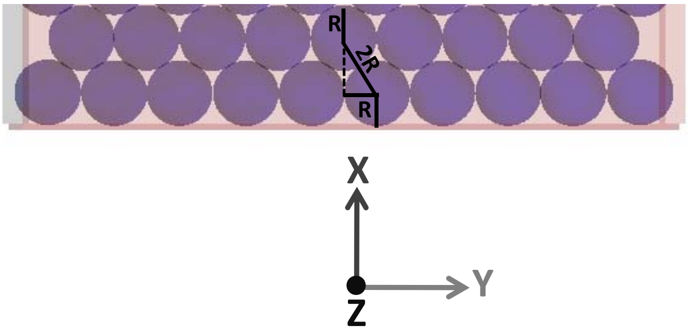


Figure 4.3: Building block of the bulk density of particles representing the bulk density of the whole system

The particles' total mass for this volume can be written as:

$$m_{bulk} = n \cdot m_{sphere} + (n-1) \cdot m_{sphere} + n \cdot m_{cap} \tag{4.3}$$

where m_{sphere} is the mass of a single particle and m_{cap} is the mass of a single particle's doom which falls down into the boundary of the building block from the third row, as shown in fig. 4.3. From the illustration in fig. 4.4, the height of the spherical cap can be written as:

$$h_{cap} = R - (\sqrt{(2R)^2 - R^2} - R) = 0.27R \tag{4.4}$$

The formula for calculation of the volume of the spherical cap is [63]:

$$V_{cap} = \frac{\pi h_{cap}^2}{3}(3R - h_{cap}) \tag{4.5}$$

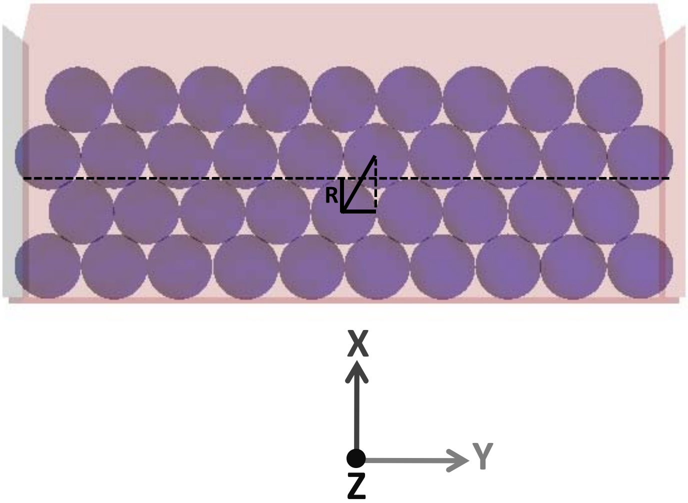


Figure 4.4: A part of the third-row spheres fall into the boundaries of the building block

Substituting h_{cap} from eq. 4.4 into eq. 4.5 gives:

$$V_{cap} = \frac{0.2\pi R^3}{3} = 0.05V_{sphere} \tag{4.6}$$

where $V_{sphere} = \frac{4\pi R^3}{3}$ is the volume of a particle. Obviously, the mass of the spherical cap and the whole sphere are also related with the same ratio as their volume and therefore eq. 4.3 can be rewritten as:

$$m_{bulk} = (2.05n - 1)m_{sphere} \tag{4.7}$$

Finally the particles' bulk density can be concluded from eqs. 4.2 and 4.7:

$$\rho_{bulk} = \frac{0.28(2.05n - 1)}{n} \cdot \rho_{sphere} \tag{4.8}$$

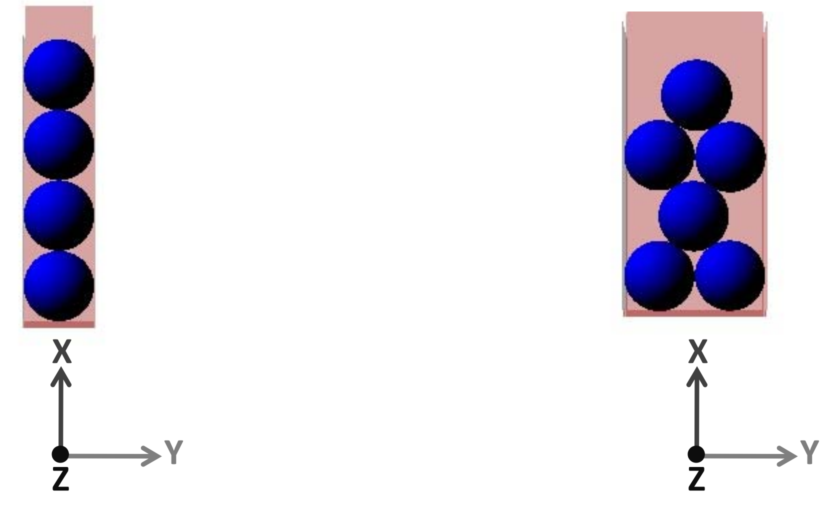
where ρ_{sphere} is the density of a single particle.

It is worth noting once again that the bulk density here is calculated for one layer of particles along the height (Z axis in fig. 4.2). Depending on the bed height, there might be particles on top of this layer which would increase the bulk density slightly because of the spherical caps turning up in the arrangement. However, this effect is the same for both wide and narrow grates and therefore do not affect the objective here which is to compare the bulk densities in narrow and wide grates.

When the grate is very wide, $(2.05n - 1)$ could be approximated as $(2.05n)$ in eq. 4.8 which simplifies the equation to:

$$\rho_{bulk} \simeq 0.57\rho_{sphere} \tag{4.9}$$

The bulk density can be calculated for different widths of the grate according to eq. 4.8 except for the case when the width is reduced to particle diameter $(n = 1)$. This is because when the width is equal to particle diameter, the arrangement of the particles no longer conforms to the same pattern of wider grates. In order to illustrate this, figs. 4.5(a) and 4.5(b) show the top view of the particles arrangement on a single bar when the width is reduced to one particle diameter and two times particle diameter, respectively.



(a) $n = 1$ (b) $n = 2$

Figure 4.5: Arrangement of particles when the grate width is reduced to: (a) one layer of particles along Y-axis and (b) two layers of particles along Y-axis, view from top.

As seen in fig. 4.5, the bulk density in the case of $n = 2$ can be calculated from eq. 4.8 because the pattern of arrangement of particles is the same as the wide grate discussed above. However, the case of $n = 1$ is a special case when the particles bulk density can be written as:

$$\rho_{bulk} = \frac{2 \cdot m_{sphere}}{4R \cdot 2R \cdot 2R} \simeq 0.52\rho_{sphere} \tag{4.10}$$

The chart in fig. 4.6 shows the bulk density of the granular particles depending on the width of the grate. The vertical axis in fig. 4.6 represents the bulk density normalized over the bulk density on a very wide grate (eq. 4.9) and the horizontal axis shows the grate width in units of particle diameter (n).

As seen in the chart in fig. 4.6 and following the discussion above if the objective is to model a very wide bar, a grate as narrow as one particle diameter gives a better approximation of the particles bulk density than a width of two times particle diameter. In fact the approximation only gets better than the former case when the width is constructed wider than 5 times particle diameter as seen in fig. 4.6. In this study the computational domain along the width of the grate is reduced to one particle diameter. The influence of the grate width on the residence time behaviour of the particles is investigated by simulation and the results are presented in sec 4.7.1.

Before closing this section, it is worth pointing out few important remarks:

- When scaling down the domain, the feed rate (mass flow) should be recalculated from the feed rate in the original domain based on the physical realities of the configured system i.e. equal distribution of the mass flow along the entire width in this study. The mass flow should

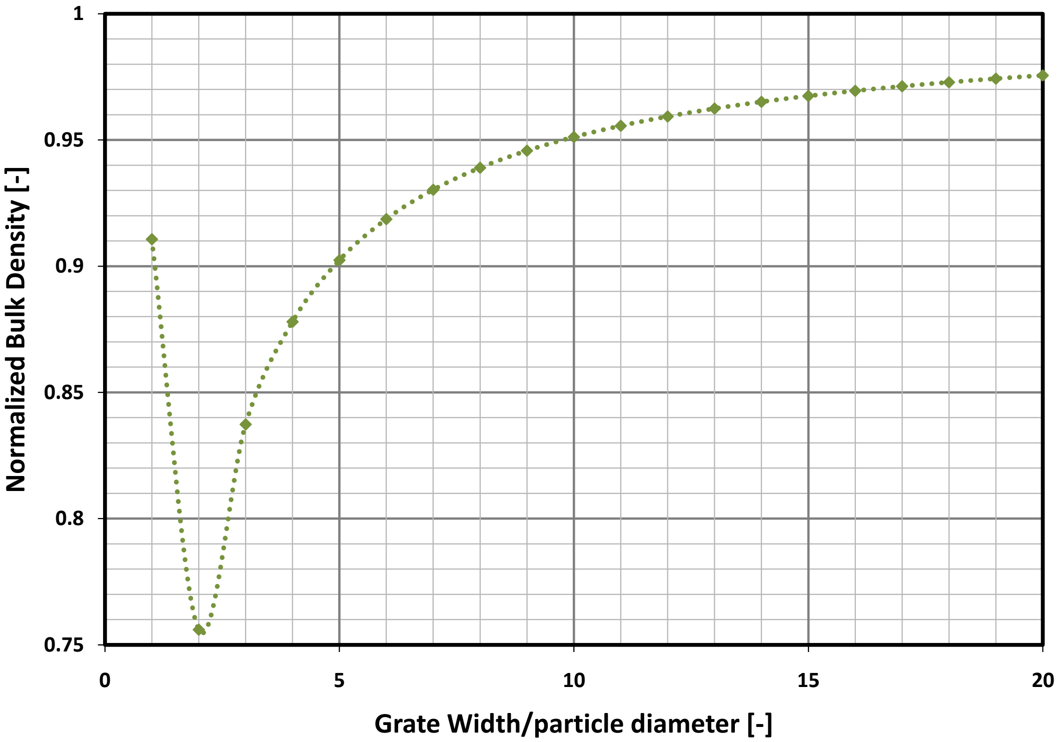


Figure 4.6: The relationship between the particles bulk density and the grate width

not be adjusted in order to produce the desired bulk density and consequently the desired residence time.

- The calculation of the bulk density presented in this chapter for the simple case of mono-disperse particles is carried out to serve as an approximation for proper selection of the computational domain. In practice, the bulk behaviour of granular particles is more complicated to predict analytically due to several inter-related conditional reactions.
- There are of course other ways of handling a large domain. Periodic boundary conditions could be applied along the width of the grate or parallelization could be employed where the domain is decomposed into several sub-domains each taken care of by a different processor. It is also possible to utilize a combination of different strategies to increase the efficiency of the computational method.

4.4 Numerical Method

The Discrete Particle Method (DPM) is used to simulate the motion of solid particles on forward and backward acting grates. The force model used is the Maw-linear-spring force model introduced in chapter 2. The integration scheme is the Position Verlet method and the time step used is 10^{-3} seconds. The parameters used in the DEM study for different materials are summarized in table

4.3. The tangential stiffness and the tangential coefficient of restitution are assumed equal to the normal stiffness and normal coefficient of restitution, respectively.

Parameter	Unit	Particles			Grate bars
Mechanical property		Swelling clay	Wood	Ceramic	Acrylic
Young modulus	N/m^2	10^5	5×10^4	2×10^5	3.2×10^9
Poisson ratio	-	0.4	0.3	0.4	0.4
impact attribute: particle/particle					
Characteristic impact velocity	m/s	2	2	2	-
Coefficient of restitution	[-]	0.7	0.8	0.94	-
Coefficient of sliding friction	[-]	0.5	0.25	0.25	-
Coefficient of rolling friction	[-]	0.002	0.0045	0.00126	-
Collision resolution	[-]	4-7	6	6	-
impact attribute: particle/bar					
Characteristic impact velocity	m/s	2	2	2	-
Coefficient of restitution	[-]	0.55	0.65	0.68	-
Coefficient of sliding friction	[-]	0.8	0.6	0.2	-
Coefficient of rolling friction	[-]	0.002	0.0045	0.00126	-
Collision resolution	[-]	4-6	6	5	-

Table 4.3: Mechanical and contact properties used in the DEM study, Maw linear spring force model

In consistence with the experimental procedure, enough particles are initially placed on the grate bars to create a primary bed of particles. The particles are then generated steadily and with a constant specified rate at the inlet and are let to fall down on the first bar from the same height as the height of the feed hopper in the experimental study. For each particle, the time when the particle enters the system and the time when it exits at the outlet is recorded. Several simulations are performed to be compared with different experiments reported. This is important because the simulation results of a single run or different runs but same boundary conditions can not be considered as a conclusive representative of a general response. On the other hand, different boundary conditions can reveal flaws in the model which might not be recognized otherwise so that a method might lead to acceptable results for mono-disperse granular material but fails when applied to poly-disperse material.

4.5 Residence Time Distribution

The residence time of a particle is defined as the duration of time from the point when a particle is introduced at the inlet until it exits the grate at the outlet. The mass related mean residence time is defined by eq. 3.58. Residence time distribution (RTD) is a concept first introduced by Danckwerts, 1953 [21] which is still widely used among chemical engineers to characterize mixing and flow behaviour in steady state conditions in a system. The Residence Time Distribution (RTD) function E(t) is defined as:

$$E(t) = \frac{C(t)}{\int_0^\infty C(t)\mathrm{d}t} \tag{4.11}$$

where $C(t)$ is the concentration of the tracer material at the outlet. $E(t)\mathrm{d}t$ represents the mass fraction of the particles which have a residence time between $(t - \frac{\mathrm{d}t}{2})$ and $(t + \frac{\mathrm{d}t}{2})$.

The cumulative distribution function F(t) is the mass fraction of particles which have a residence time between 0 and t:

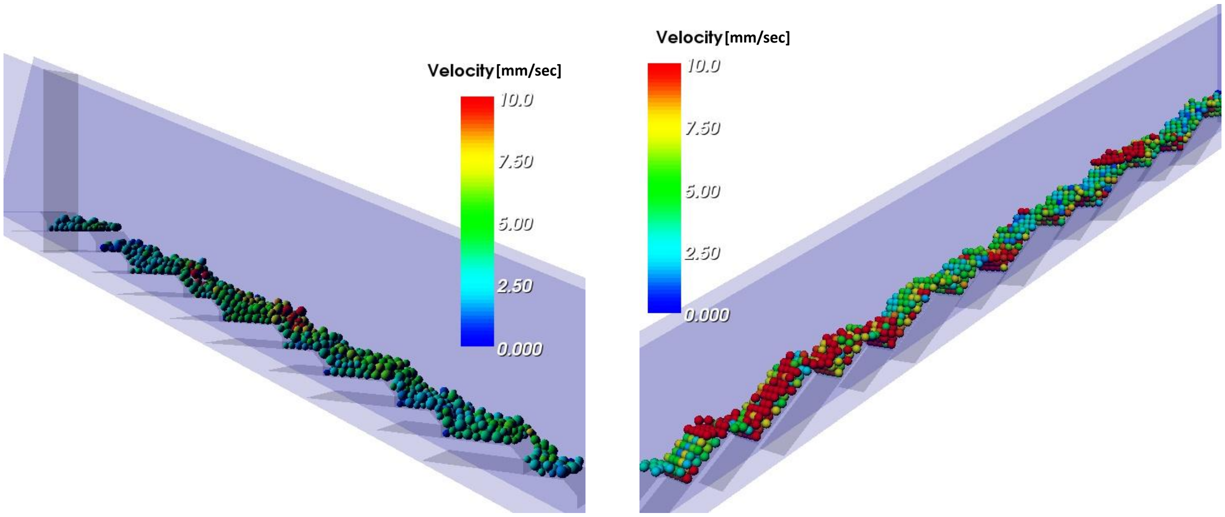
$$F(t) = \int_0^t E(t)\mathrm{d}t \tag{4.12}$$

Residence time distribution curves for both simulation and experiments are presented. The comparison between simulation and experiments are carried out based on absolute values and neither time nor mass fraction are normalized. The simulation results are post-processed employing the same sampling frequency and same sampling points in time as in the experiments which enables direct and one to one comparisons. In the experimental study, the end of the RTD curve is produced by approximations done by extrapolation of data [6]. This choice is explained by the fact that "the tracer material is discharged only incompletely in a reasonable time". However, no approximation or extrapolation is carried out on the results of the numerical predictions and enough time is given for the sample of particles to discharge almost completely.

4.6 Validation Results

4.6.1 Comparison of Forward and Backward Acting Grates

Comparison of the residence time distribution for forward and backward acting grates is carried out with the same parameters as summarized in table 4.4. Snapshots of the animation of the results are presented in fig. 4.7.



(a) forward acting grate (b) backward acting grate

Figure 4.7: Snapshots of the animation of motion of solid particles predicted by DPM

The Residence Time Distribution (RTD) curve is presented in fig. 4.8.

Particles material	Swelling clay
Mass flow $[kg/hr]$	20
Moving bar velocity $[mm/s]$	3.5

Table 4.4: Data for the experiment and simulation of the comparison between forward and backward acting grate

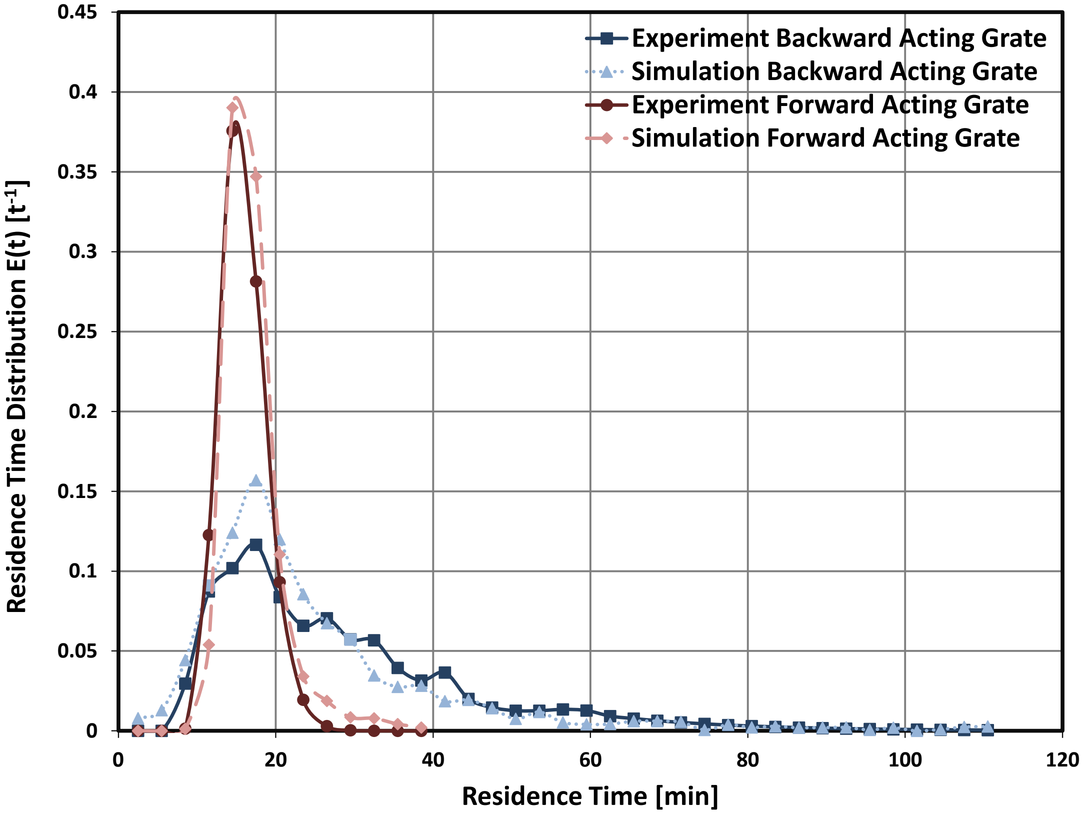


Figure 4.8: Comparison of residence time distribution for forward and backward acting grates by experiments and DPM simulation

As seen in fig. 4.8, the residence time is more distributed in the case of backward acting grate which indicates better mixing for the backward compared to the forward acting grate. The RTD curve of the forward acting grate, on the other hand, resembles the plug flow conditions where the particles exit the system almost in the same order as they entered.

Comparing the simulation results with experimental results shows that the simulation results have slightly higher peaks than the experiments in both forward and backward cases. There are few uncertainties in the input data which might relate to such a difference between simulation and experimental results. These uncertainties include the data regarding the swelling clay particles as already mentioned in section 4.2 and the exact position at which the new material is introduced to the forward acting grate at the inlet. It should also be reminded that the tracer material in the experimental works are not completely discharged and the results presented in this case are representative of approximately 90% of the total tracer material while more than 98% of the samples in the simulations are discharged. This difference might also explain part of the differences observed between the simulation and experimental results in fig. 4.8. Overall, the simulation results show quite good agreement with the experiments.

4.6.2 Wooden Particles on Backward Acting Grate

In consistence with one of the experiments reported, RTD of backward acting grate is also predicted using wooden particles. The specific data for this test are summarized in table 4.5 and the residence time distribution and cumulative distribution are plotted in figs. 4.9 and 4.10, respectively.

Particles material	Wood
Mass flow $[kg/hr]$	40
Moving bar velocity $[mm/s]$	3.5

Table 4.5: Data for the experiment and simulation of wooden particles on backward acting grate

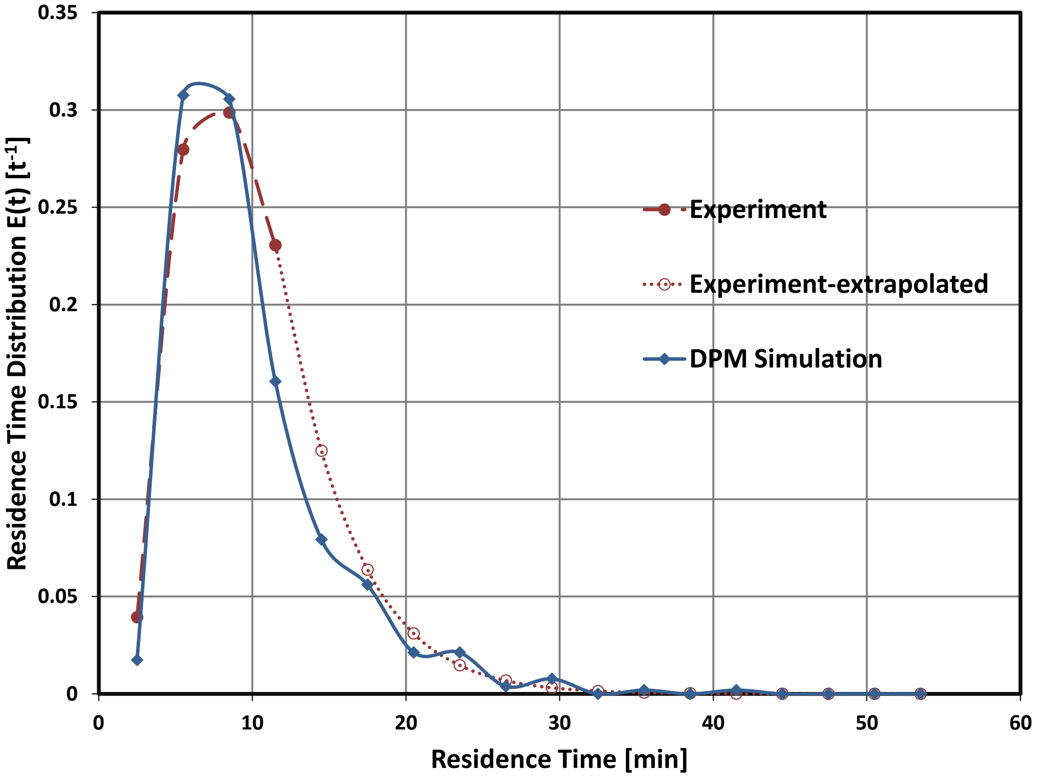


Figure 4.9: Residence time distribution of wooden particles on backward acting grate

Fig. 4.9 shows quite good agreement between the simulation and experimental results. However, as seen in fig. 4.10, the extrapolation of the experimental results has led to the cumulative mass of

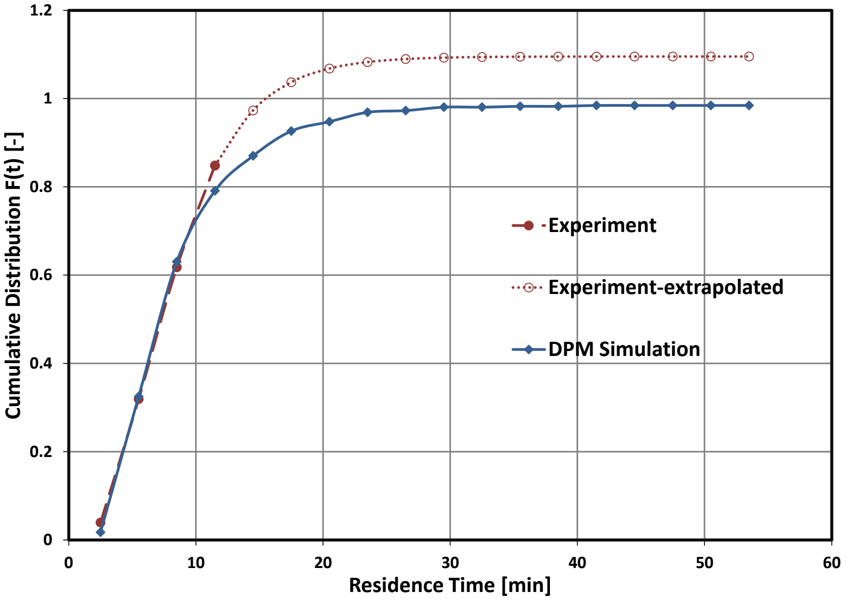


Figure 4.10: Cumulative distribution of wooden particles on backward acting grate

more than 1 which is not physically sensible. The DPM simulation predicts the end of the RTD curve more realistically.

4.6.3 Ceramic Particles on Backward Acting Grate

The residence time distribution of backward acting grate is also predicted using ceramic particles. The simulation data are summarized in table 4.6 while the residence time distribution and cumulative distribution curves are presented in figs. 4.11 and 4.12, respectively.

Particles material	Ceramic
Mass flow $[kg/hr]$	60
Moving bar velocity $[mm/s]$	3.5

Table 4.6: Data for the experiment and simulation of ceramic particles on backward acting grate

Figures 4.11 and 4.12 show that in the prediction the material start to discharge slightly later than the experiments. This might be due to uncertainties in the ceramic particles data explained in section 4.2. Nevertheless, DPM prediction of the distribution of residence time is quite consistent with the distribution curve of the experimental results.

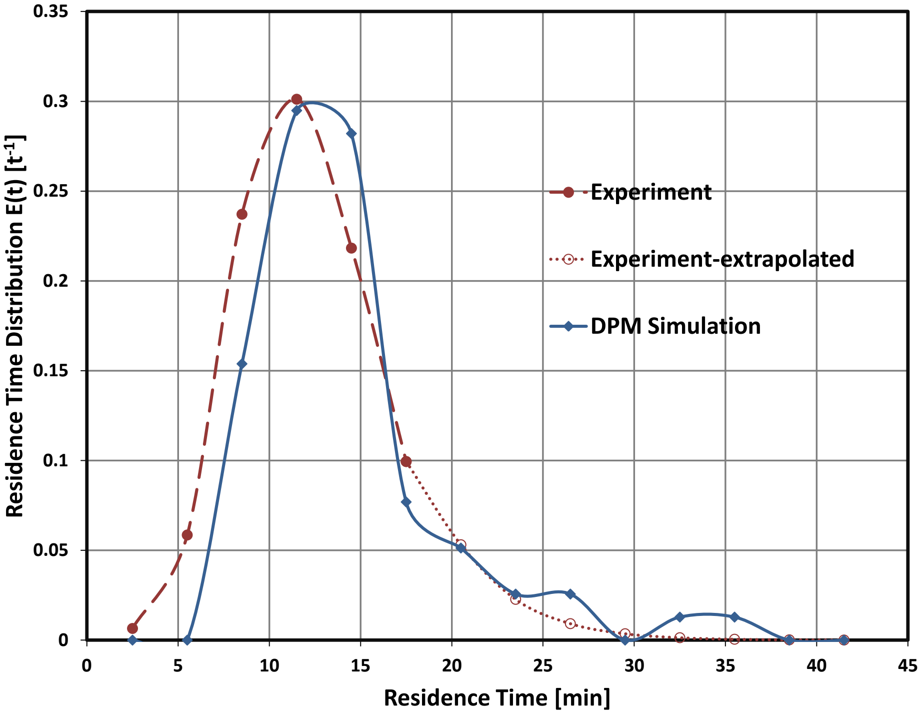


Figure 4.11: Residence time distribution of ceramic particles on a backward acting grate

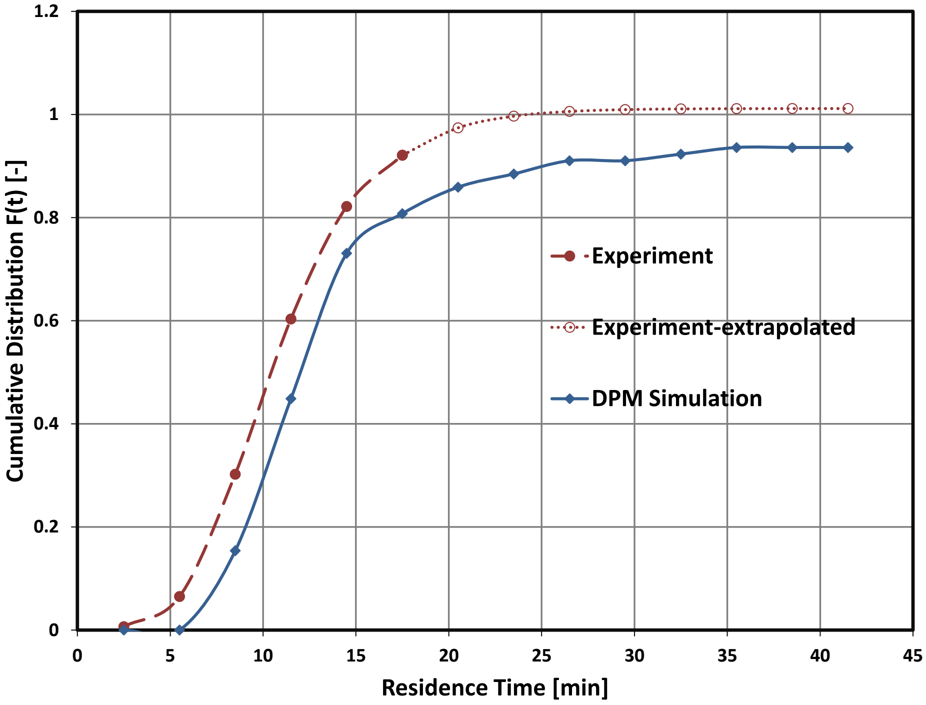


Figure 4.12: Cumulative distribution of ceramic particles on a backward acting grate

4.6.4 Effect of Mass Flow

In order to investigate the effect of mass flow on the mean residence time, the backward acting grate prediction using swelling clay particles was repeated with four more different mass flows. The mass flow or flux is the rate of feeding the new material to the grate at the inlet. The results of this test are shown in fig. 4.13.

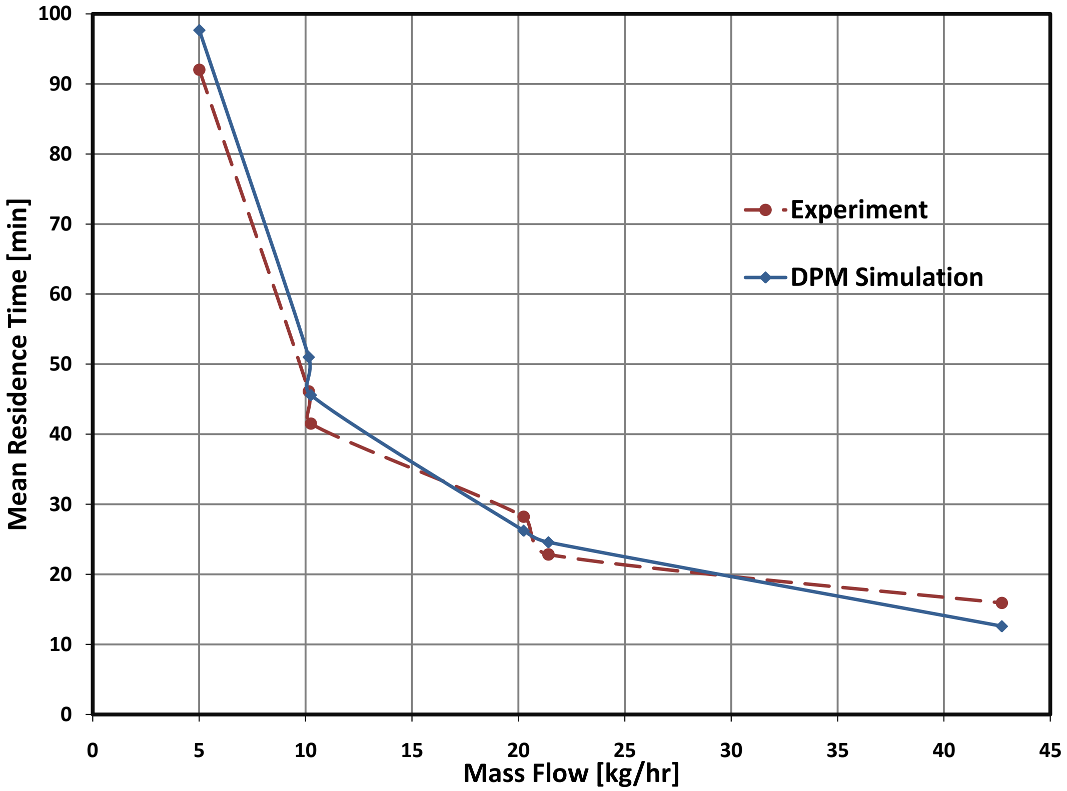


Figure 4.13: Effect of mass flow on the mean residence time

The mean residence time decreases in general by increasing the mass flow as seen in fig. 4.13. This relationship is due to the fact that a higher bed height on the grate is maintained by increasing the mass flow. There is a limit for the increase of the mass flow of course and above this limit, the particles merely slide along the grate inclination towards the outlet. Fig. 4.13 shows a very good prediction of the mean residence time by DPM simulation for different mass flows.

4.7 Sensitivity Analysis

4.7.1 Effect of Grate Width

As discussed in section 4.3, it has been argued that the movement of the granular material on the grate is mainly dominated by the motion along the grate inclination (X axis in fig. 4.1) [64]. Based on this, the computational domain in the simulations above is reduced along the width of the grate to the extent of accommodating only one layer of particles along the width, as already stated in section 4.3. In order to verify the validity of this approach, a simulation is carried out on two forward acting grates which have different widths and otherwise identical. The narrow grate is only wide enough to accommodate one layer of particles along the grate width while the wider bar is wide enough for five layers of particles to be positioned along the width. The objective is merely to investigate the effect of the grate width on the simulation results and therefore imaginary test particles and grate are used in this prediction. The input data for this test are given in table 4.7.

Particles diameter [mm]	8 - 14
Type of grate	Forward acting grate
Number of grate bars	11
Grate inclination	20 °
Bar inclination	0 °
Mass flow [kg/hr]	20
Moving bar velocity [mm/s]	3.5

Table 4.7: The input data for the two simulations performed to investigate the effect of the grate width on the residence time

The force model used in this particular test is a simple linear spring-dashpot, the integration scheme Position Verlet and the time step is 10^{-4} seconds. The mechanical and contact properties are summarized in table 4.8.

Parameter	Unit	Particles	Grate bars
Mechanical property			
Material density	kg/m^3	820	1190
Spring stiffness	N/m	800	1000
impact attribute (particle/particle)			
Coefficient of restitution	[-]	0.4	-
Coefficient of sliding friction	[-]	0.7	-
Coefficient of rolling friction	[-]	0.0009	-
impact attribute (particle/bar)			
Coefficient of restitution	[-]	0.4	-
Coefficient of sliding friction	[-]	0.75	-
Coefficient of rolling friction	[-]	0.0009	-

Table 4.8: Mechanical and contact properties used in the two simulations performed to investigate the effect of the grate width on the residence time

Fig. 4.14 shows the results of the effect of the grate width on the residence time distribution of particles on forward acting grate. It shows that increasing the grate width five times has negligible effects on the residence time distributions of particles on forward acting grate which is in agreement with the earlier findings of Hunsinger et al. [42].

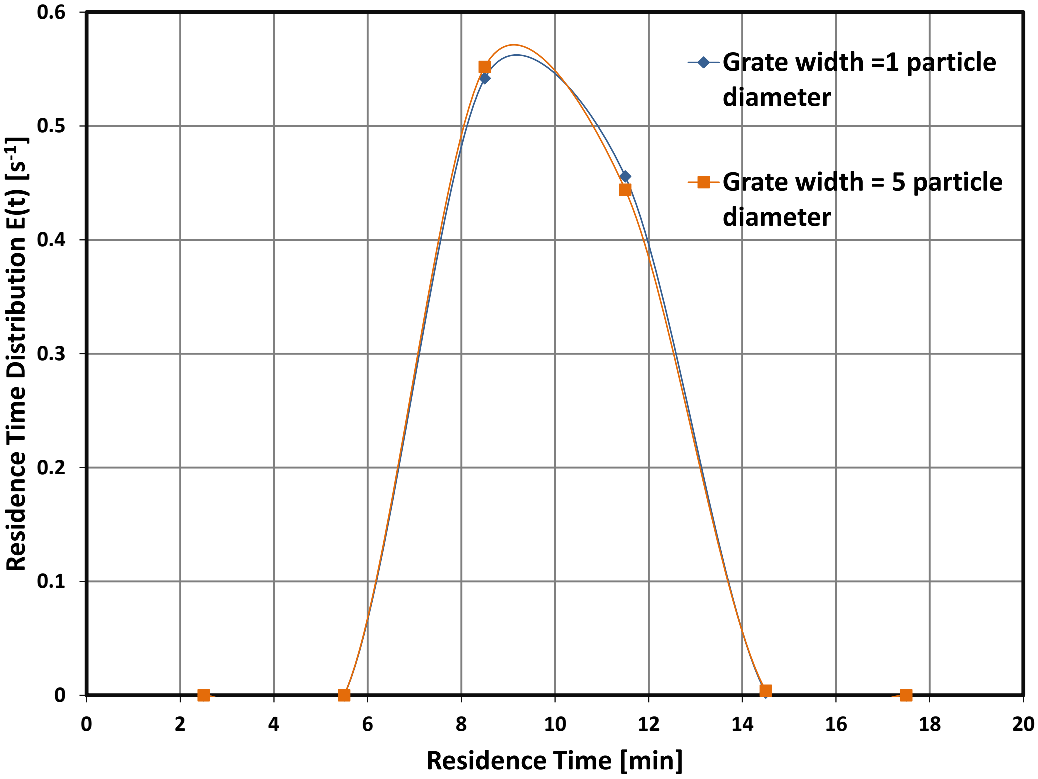


Figure 4.14: Effect of increasing the grate width on the residence time distribution of particles on forward acting grate

4.7.2 Effect of Sampling Frequency

The results presented above are generated with the same sampling frequency as in the experiments. In most of the cases, this frequency was set to three minutes which means the samples are collected and analysed every three minutes. This also means that the residence times in the range of 1.5 minutes above or below the mean of the range are approximated as the mean. For instance in three minutes sampling frequency, the residence times in the range of 15 to 18 minutes will all be approximated by 16.5 minutes. In order to investigate the effect of this approximation, the simulation results of the swelling clay particles on a forward acting grate is post-processed again this time with the sampling frequency of one minute. The results, shown in fig. 4.15, suggest that the approximation of sampling every three minutes is quite acceptable.

Obviously, the sampling frequency should be selected considering also the size of the whole sample. If the sampling is performed too frequently, there will not be enough data in a single sampling range which makes the outcome non-representative. In this analysis, the whole sample consists of 3118 particles. It appears that the sampling frequency of three minutes is quite appropriate for this number of particles.

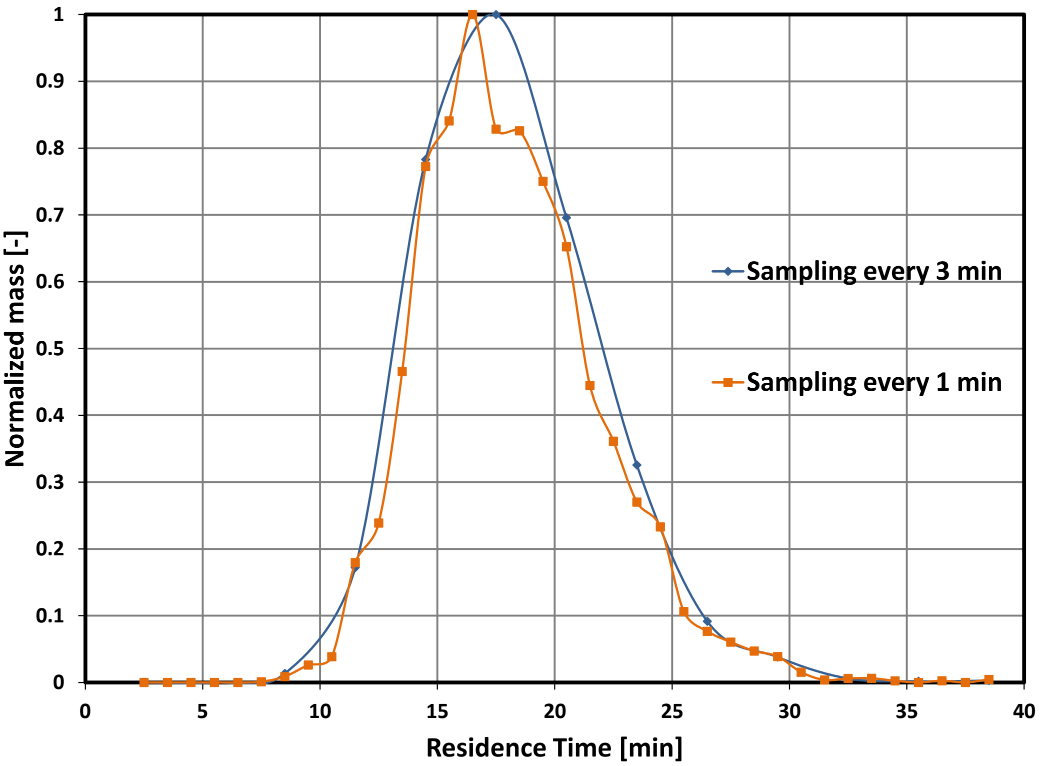


Figure 4.15: Effect of sampling frequency on the simulation results, case of swelling clay particles on forward acting grate

4.7.3 Sensitive Parameters and Recommendations for Future Set-up

As already discussed, the DEM study should be constructed with the same configuration as the experimental set-up in order to achieve a credible validation of the numerical approach. Therefore, it is important to pay thorough attention to the experimental data and particularly sensitive parameters which affect the flow behaviour of the solid particles on the grate. Based on the experience from this study, few sensitive parameters which might be neglected but are influential for numerical reproduction of the system are listed below. It is worth reminding that this is a suggestive list of the more subtle parameters and decent consideration of the whole range of experimental parameters is essential when collecting data.

- It is quite common to report the bulk density of granular material in experimental studies but what is less common to consider is the grain density i.e. the material density of each single particle. This information is very important in DEM which is founded on observing the microscopic behaviour of individual particles. It is also worth noting that the material density is not absolutely constant and varies with particle size. This effect is more pronounced in some materials such as expanding or swelling clay particles. If the variation in particles' density is considerable, the constant density assumption might affect the DEM results. However, in most cases the density can be assumed constant and approximated as the density of the average particle.

- An important parameter which highly affects the residence time of particles especially in backward acting grates is the ratio between particle diameter and the bar height. Particles which have smaller diameter than the height of the grate bar are prone to be trapped at the bottom of the particles bed and exhibit higher residence time than the particles on the top surface of the bed. It is, therefore, important to pay millimetric attention to the bar height when setting up the grate. The exact shape and curve of the grate bars should also be accounted for.

- While it is important to accurately define the grate geometry and the particles size and properties, the question of how the new particles are introduced into the system should not be overlooked. Some specific questions worth noticing are: Where is the exact feed area? Are the feed particles introduced into the system at the inlet at the same point or are scattered into a feed area? From what height are particles let fall on the first bar?

- The velocity and the full travelling distance of the moving bars are essential but not enough information to define the motion of the bars; the period of a double-stroke is also worth noting down. This is because the moving bars do normally have a small pause at the end of a single stroke and before changing direction. This can influence the dynamics of the solid particles on the grate and is worth considering.

4.8 Summary and Conclusions

Numerical analysis of the dynamics of the granular material on moving grates is still a very young research topic. Continuing the few earlier initial attempts in this regard, this chapter presents the application of Discrete Element Method (DEM) to packed bed of particles on forward and backward acting grates. Consistent set of boundary and initial conditions are employed in all DPM predictions in order to allow a sound parametric comparison and analysis. Distributions of the residence times are presented which are used by chemical engineers to characterize the mixing behaviour in reactors [43]. Special emphasis is given to compare the residence time distribution of granular particles on forward and backward acting grates. DPM predictions confirm the conclusion of the experimental results that the backward acting grate configuration induces better mixing of the particles.

The influence of several operating parameters including the mass flux and the particles' material are studied. The results of the simulation and experiments are compared directly and based on absolute values of the residence times which indicates promising competence for DEM in predicting the dynamics of granular material. Such a one-to-one comparison of the results are only possible if all the operating and geometrical conditions are set exactly the same in experiments and simulations. Considering few uncertainties in the experimental data, the correspondence between the simulation and experimental results is good. Occasional deviations of the predictions from experimental results are in the same order of magnitude as the uncertainties in the experimental data.

Future work shall include improving the consistency between the simulation and experiments by clarifying the unknown data in the experiments. The effect of the grate width on the bulk density and the residence time of the particles still need to be further investigated in both mono-disperse and poly-disperse systems. Particle shapes other than spheres need to be studied and in particular systems containing various shapes and sizes of particles need to be analysed.

Chapter 5

Summary and future work

Referring back to the two main objectives of this dissertation introduced in chapter 1, the major conclusions of the dissertation can also be addressed in two separate categories:

5.1 Potentials of the Implicit Method

Several conclusions can be drawn from the results of the comparisons and verification tests presented in sections 3.4 and 3.9 which are summarized below.

- Implicit integration schemes are more robust than the explicit schemes in large time steps i.e. contact resolutions smaller than 5. Earlier detection of contacts and smoother transition between different phases of collision results in errors being bounded in implicit methods. However, the size of the time step is also limited in implicit methods as it can not be bigger than the duration of collision.

- In smaller time steps i.e. contact resolutions bigger than 5, the gain in accuracy in implicit methods is too small to have a decisive impact on the global behaviour of the granular system.

- Though implicit integration allows slightly bigger time steps than explicit methods, it still remains a risky approach to utilize big time steps. This is because the duration of collision in non-linear impact models is not absolutely predictable from the material properties. Therefore, the time step selected should always consider safety margins for uncertainties in impact velocities. Besides, the implicit solution risks non-convergence of the Newton-Raphson method in large time steps. This issue, though, can be managed by adapting the tolerance of the Newton-Raphson method according to the convergence history. This approach, however, does sacrifice some degree of accuracy.

- Effective detection of contacts in both explicit and implicit methods does not only depend on the size of the time step but also on how late a collision starts relative to the start of the time step. Since there is no reasonably efficient way in DEM to synchronize the start of the time step with the start of each collision, the worst case scenario should always be considered when evaluating a method. The implicit method is less sensitive to pre-collision fraction of the time step and is therefore more effective than explicit methods in contact detection.

- Implicit methods are computationally more intensive than explicit methods. A system of non-linear equations needs to be solved at each time step iteratively. This makes implicit methods too expensive for the benefits. There are methods to improve the efficiency of the

Newton-Raphson method such as less frequent Jacobian assembly or less frequent collision detection. However, such methods reduce the costs only moderately while sacrificing some degree of accuracy.

- With the current state of the art where the computational costs of force computation is the main drawback of the DEM, any integration scheme which requires more than one force computation per time step is not recommended. Instead predictor-corrector schemes such as Gear integration scheme are recommended which are in fact inexpensive implementations of implicit schemes.

5.2 Numerical Prediction of Particles' Dynamics in Forward and Backward Acting Grates

The numerical analysis of the residence time behaviour of solid particles on forward and backward acting grates indicates highly promising capabilities of DPM in predicting the motion of granular particles. This claim is supported by direct quantitative comparison of the simulation and experimental results in various cases reported in this dissertation.

Admittedly, this is still one of the initial attempts in predicting the residence time characteristics of forward and backward acting grates numerically. Further expansion and development is necessary to promote the effectiveness of such analyses in serving the industries and accordingly societies. To this end, future work should focus on improving the analysis by:

- Diminishing the uncertainties in experimental data for higher consistency in boundary conditions and thus better validation of the numerical results
- Incorporating more complex shapes to represent solid fuel particles in the numerical approach
- Expanding the parametric analysis to include a whole range of operating conditions which influence the residence time behaviour of fuel particles
- Coupling the dynamics of solid particles with their thermal conversion due to combustion

Bibliography

[1] M.P. Allen and D.J. Tildesley. *Computer Simulations of Liquids.* Clarendon Press, Oxford, 1987.

[2] S.J. Antony, W. Hoyle, and Y. Ding. *Granular Materials, fundamentals and applications.* The Royal Society of Chemistry, Cambridge, 2004.

[3] K.E. Atkinson. *An introduction to numerical analysis.* John Wiley and Sons, New York, 1989.

[4] M. Babic, H.H. Shen., and H.T. Shen. The stress tensor in granular shear flows of uniform, deformable disks at high solids concentrations. *Fluid Mechanics*, 219(81), 1990.

[5] M. Beckmann. *Mathematische Modellierung und Versuche zur Prozeßführung bei der Verbrennung und Vergasung in Rostsystemen zur thermischen Rückstandsbehandlung.* PhD thesis, TU Clausthal, 1995.

[6] M. Beckmann and R. Scholz. Residence time behaviour of solid material in grate systems. In *in: European Conference N05 on Industrial furnaces and boilers*, page 661, Porto, Portugal, 2000.

[7] D. Beeman. Some multistep methods for use in molecular dynamics calculations. *Computational Physics*, 20(2):130–139, 1976.

[8] M.J. Berger and S.H. Bokhari. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers*, 36(5):570–580, 1987.

[9] W.E. Boyce and R.C. DiPrima. *Elementary Differential Equations.* Wiley, New York, 7th edition, 2000.

[10] A. Bridges, A. Hatzes, and D.N.C. Lin. Structure, stability and evolution of saturn's rings. *Nature*, pages 309–333, 1984.

[11] N.V. Brilliantov, F. Spahn, J.M. Hertzsch, and T. Pöschel. The collision of particles in granular systems. *Physica A*, 231:417–424, 1996.

[12] N.V. Brilliantov, F. Spahn, J.M. Hertzsch, and T. Pöschel. Model for collisions in granular gases. *Physical review E*, 53(5):5382–5392, 1996.

[13] C.S. Campbell. *Shear flows of granular materials.* PhD thesis, California Institute of Technology, 1982.

[14] C.S. Campbell. Granular shear flows at the elastic limit. *Fluid Mechanics*, 465:261–291, 2002.

[15] C.S. Campbell. Granular material flows-an overview. *Powder Technology*, 162:208–229, 2006.

[16] P.W. Cleary. Large scale industrial DEM modelling. *Engineering Computations*, 21(2/3/4):169–204, 2004.

[17] European Commission. Waste. http://ec.europa.eu/environment/waste/index.htm.

[18] S.J. Cummins and J.U. Brackbill. An implicit particle-in-cell method for granular materials. *Computational Physics*, 180:506–548, 2002.

[19] P.A. Cundall and O.D.L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29:47–65, 1979.

[20] Cutec institute. www.cutec.de.

[21] P.V. Danckwerts. Continous flow systems. distribution of residence times. *Chemical Engineering Science*, 2:1–13, 1953.

[22] A. Di Renzo and F.P. Di Maio. Comparison of contact-force models for the simulation of collisions in dem-based granular flow codes. *Chemical Engineering Science*, 59:525–541, 2004.

[23] J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors, 2006.

[24] J. Duran and P.G. De Gennes. *Sands, Powders, and Grains: An Introduction to the Physics of Granular Materials*. Springer, 1999.

[25] A. Džiugys, B. Peters, H. Hunsinger, and L. Krebs. Evaluation of the residence time of a moving fuel bed on a forward acting grate. *Granular Matter*, 8:125–135, 2006.

[26] A. Džiugys, B. Peters, H. Hunsinger, and L. Krebs. Experimental and numerical evaluation of the transport behaviour of a moving fuel bed on a forward acting grate. *Granular Matter*, 9:387–399, 2007.

[27] B.J. Ennis, J. Green, and R Davies. The legacy of neglect in the us. *Chemical Engineering Progress*, 90(4):32–43, 1994.

[28] G. Enstad. On the theory of arching in mass low hoppers. *Chemical Engineering Science*, 30:1273–1283, 1975.

[29] The European Parliament and the Council of the European Union. *Directive 2000/76/EC of the European Parliament*, December 2000.

[30] Eurostat. Environmental data centre on waste. http://epp.eurostat.ec.europa.eu.

[31] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer-Verlag, 2002.

[32] F.Y. Fraige and P.A. Langston. Integration schemes and damping algorithms in distinct element models. *Advanced Powder technology*, 15(2):227–245, 2004.

[33] C. W. Gear. *Numerical initial value problems in ordinary differential equations*. Prentice-Hall, Englewood Cliffs, 1971.

[34] C.W. Gear. The numerical integration of ordinary differential equations of various orders. Technical report, Argonne National Laboratory, 1966.

[35] K. Göerner. Waste incineration: European state-of-the-art and new development. *IFRF Combustion Journal*, page 32, 2003.

[36] H. Goldstein. *Classical Mechanics*. Addison-Wesley Publishing Company, 1980.

[37] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the störmer/verlet method. *Acta Numerica*, 12:399–450, 2003.

[38] E. Hairer, S.P. Nø rsett, and G. Wanner. *Solving ordinary differential equations I: Nonstiff problems*. Springer-verlag, Berlin, 1993.

[39] H.H. Hausner. Friction conditions in a mass of metal powder. *Int. J. Powder Metall.*, 3:7–13, 1967.

[40] H. Hertz. Ueber die berührung fester elastischer koerper. *Journal für die Reine und Angewandte Mathematik*, 92:156–171, 1882.

[41] C. Hogue. Shape representation and contact detection for discrete element simulations of arbitrary geometries. *Engineering Computations*, 15(2-3):374–390, 1998.

[42] H. Hunsinger, J. Vehlow, B. Peters, and H.H. Frey. Performance of a pilot waste incinerator under different air/fuel ratios. In *in: IT3 International Conference on Incineration and Thermal Treatment Technologies*, Portland, Oregon, USA, MAY 2000.

[43] International Atomic Energy Agency, Vienna. *Radiotracer Residence Time Distribution Method for Industrial and Environmental Applications*, 2008.

[44] D. Janezic and R. Trobec. Parallelization of an implicit runge–kutta method for molecular dynamics integration. *chemical information and computer sciences*, 34(3):641–646, 1994.

[45] G. Jasion, J. Shrimpton, M. Danby, and K. Takeda. Performance of numerical integrators on tangential motion of dem within implicit flow solvers. *Computers and Chemical Engineering*, 35:2218–2226, 2011.

[46] H. Jeffreys and B.S. Jeffreys. *Methods of mathematical physics*. Cambridge University press, Cambridge, England, 1988.

[47] A.W. Jenike. Quantitative design of mass flow bins. *Powder Technology*, 1(4):237–244, 1967.

[48] L. Jintang, P.A. Langston, C. Webb, and T. Dyakowski. Flow of sphero-disc particles in rectangular hoppers-a dem and experimental comparison in 3d. *Chemical Engineering Science*, 59:5917–5929, 2004.

[49] M.A. Kabir, M.R. Lovell, and C.F. Higgs. Utilizing the explicit finite element method for studying granular flows. *Tribology letters*, 22(2):85–94, February 2008.

[50] W. Kahan. *Gauss-Seidel Methods of Solving Large Systems of Linear Equations*. PhD thesis, University of Toronto, 1998.

[51] E. Kamke. *Differentialgleichungen: Lösungsmethoden und Lösungen, I, Gewöhnliche Differentialgleichungen.* B. G. Teubner, Leipzig, 1977.

[52] A.H. Karp and P.F. Horace. Measuring parallel processors performance. *Communications of the ACM*, 33(5):539–543, 1990.

[53] T.C. Ke and J. Bray. Modeling of particulate media using discontinuous deformation analysis. *Journal of Engineering Mechanics*, 121:1234–1243, 1995.

[54] W.R. Ketterhagen, J.S. Curtis, and C.R. Wassgren. Stress results from two-dimensional granular shear flow simulations using various collision models. *Physical Review. E*, 71, 2005.

[55] J. Kozicki and F.V. Donzé. Yade-open dem: an open-source software using a discrete element method to simulate granular material. *Engineering Computations*, 2008. accepted for publication.

[56] H. Kruggel-Emden, S. Rickelt, S. Wirtz, and V. Scherer. A study on the valdity of the multi-sphere discrete element method. *Powder Technology*, 188:153–165, 2008.

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

[58] H. Kruggel-Emden, E. Simsek, S. Wirtz, and V. Viktor. A comparative numerical study of particle mixing on different grate designs through the discrete element method. *Pressure vessel technology*, 129(4):593–600, 2007.

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

[60] L.D. Landau and E.M. Lifshitz. *Theory of Elasticity. Course of Theoretical Physics.* Pergamon Press, Oxford, 1959.

[61] J.P. Latham and A. Munjiza. The modelling of particle systems with real shapes. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical Physical and Engineering Sciences*, 362(1822):1953–1972, 2004.

[62] A.R. Leach. *Molecular Modelling: Principles and Applications.* Prentice Hall, 2001.

[63] S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

[64] C.N. Lim, Y.R. Goh, V. Nasserzadeh, and J. Swithenbank. The modelling of solid mixing in municipal waste incinerators. *Powder Technology*, 114(1):89–95, 2001.

[65] S. Luding, È. Clèment, A. Blumen, J. Rajchenbach, and J. Duran. Anomalous energy dissiaption in molecular-dynamics simulations of grains: the detachment effect. *Phys. Rev. E*, 50:4113, 1994.

[66] G. Marshall and L. Jonker. An introduction to descriptive statistics: A review and practical guide. *Radiography*, 16(4):e1–e7, 2010.

[67] N. Maw, J.R. Barber, and I.N. Fawcett. The oblique impact of elastic spheres. *Wear*, 38:101–114, 1976.

[68] E.W. merrow. Linking r&d to problems experienced in solids processing. *Chemical Engineering Progress*, 81:14–22, 1985.

[69] R.D. Mindlin. Compliance of elastic bodies in contact. *Applied Mechanics*, 16:259–268, 1949.

[70] *Department of Defence Documentation of Verification, Validation & Accreditation (VV&A) for Models and Simulations*, 2008.

[71] M. Nakamura. *Mathematical and Physical Modeling of Mixing and Flow Phenomena of Municipal Solid Waste Particles on a Reverse Acting Grate.* PhD thesis, Columbia University, 2008.

[72] R.M. Nederman. *Statics and kinematics of granular material.* Cambridge University Press, 1992.

[73] D. Negrut. *On the implicit integration of differential algebraic equations of multibody dynamics.* PhD thesis, University of Iowa, July 1998.

[74] N.M. Newmark. A method of computation for structural dynamics. *Engineering Mechanics*, 85:67–94, 1959.

[75] G.T. Nolan and P.E. Kavanagh. Random packing of nonspherical particles. *Powder Technology*, 84:199–205, 1995.

[76] C.S. Peskin and T. Schlick. Molecular dynamics by the backward-euler method. *Communications on Pure and Applied Mathematics*, 42:1001–1031, 1989.

[77] B. Peters. *Thermal Conversion of Solid Fuels.* WIT Press, 2003.

[78] B. peters and A. Džiugys. Bestimmung des kontaktes für die bewegung von elliptischen, granularen medie. In *in: Annual Scientific Conference of the International Association of Applied Mathematics and Mechanics*, Göttingen, 2000.

[79] B. Peters and A. Džiugys. Prediction of conversion of a packed bed of fuel particles on a forward acting grate by the discrete particle method (dpm). *Computer Aided Chemical Engineering*, 29(6):1894–1898, 2011.

[80] B. Peters, A. Džiugys, H. Hunsinger, and L. Krebs. An approach to qualify the intensity of mixing on a forward acting grate. *Chemical Engineering Science*, 60(6):1649–1659, 2005.

[81] B. Peters and A. Džiugys. Numerical simulation of the motion of granular material using object-oriented techniques. *Computer Methods in Applied Mechanics and Engineering*, 191:1983–2007, 2002.

[82] B.T. Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181(3):1086–1096, 2007.

[83] A.D. Polyanin and V.F. Zaitsev. *Handbook of exact solutions for Ordinary Differential Equations.* Chapman and Hall/CRC, Boca Raton, 2nd edition, 2003.

[84] T. Pöschel and T. Schwager. *Computational Granular Dynamics*. Springer, Berlin, 2005.

[85] W.H. Press, W.T. Vetterling, S.A. Teukolsky, and B.P. Flannery. *Numerical recipes*. Cambridge University Press, Cambridge, 1988.

[86] P. Richard. Slow relaxation and compaction of granular systems. *Nature Materials*, 4:121–128, 2005.

[87] G.H. Ristow. Granular dynamics: a review about recent molecular dynamics simulations of granular materials. *Ann. Rev.Comput. Phys. I*, 1:275–308, 1994.

[88] E. Rougier, A. Munjiza, and N.W.M. John. Numerical comparison of some explicit time integration schemes used in dem, fem/dem and molecular dynamics. *Numerical methods in Engineering*, 61:856–879, 2004.

[89] R.J. Sadus. *Molecular Simulation of Fluids: Theory, Algorithms and Object-Orientation*. Elsevier, Amsterdam, 1999.

[90] A. Satoh. Satibility of computational algorithms used in molecular dynamics simulations. *ASME journals of fluids engineering*, 117:531–534, 1995.

[91] A. Satoh. Satibility of various molecular dynamics algorithms. *ASME journals of fluids engineering*, 119:476–480, 1995.

[92] J. Schäfer, S. Dippel, and D.E. Wolf. Force schemes in simulations of granular materials. *Journal de Physique I*, 6:5–20, January 1996.

[93] N. Schäfer and D. Negrut. A quantitative assessment of the potential of implicit integration methods for molecular dynamics simulation. *Computational and Nonlinear Dynamics*, 5(3), 2010.

[94] N. Schäfer, D. Negrut, and R. Serban. Experiments to compare implicit and explicit methods of integration in molecular dynamics simulation. Technical report, University of Wisconsin, November 2008.

[95] E. Simsek, B. Brosch, S. Wirtz, V. Scherer, and F. Krüll. Numerical simulation of grate firing systems using a coupled cfd/discrete element method (dem). *Powder technology*, 193(3):266–273, 2009.

[96] R.D. Skeel, G. Zhang, and T Schlick. A family of symplectic integrators: Stability, accuracy, and molecular dynamics applications. *SIAM J. SCI. COMPUT.*, 18:203–222, 1997.

[97] A.B. Stevens and C.M. Hrenya. Comparison of soft-sphere models to measurements of collision properties during normal impacts. *Powder Technology*, 154:99–109, 2005.

[98] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, Berlin, New york, 2002.

[99] A.K. Subramanian and Y. Marwaha. Use of bagasse and other biomass fuels in high pressure travelling grate boilers. *Int Sugar J*, 108:6–9, 2006.

[100] F. Sudbrock, E. Simsek, S. Rickelt, S. Wirtz, and V. Scherer. Discrete element analysis of experiments on mixing and stocking of monodisperse spheres on a grate. *Powder technology*, 208(1):111–120, 2011.

[101] D. Tabor. The mechanism of rolling friction ii. the elastic regime. *Proc. R. Soc. Lond. A*, 229:198–220, 1955.

[102] J.B. Tatum. Classical mechanics. http://orca.phys.uvic.ca/ tatum/classmechs.html.

[103] P.A. Thompson and G.S. Grest. Granular flow-friction and the dilatancy transition. *Physical Review letters*, 67:1751–1754, 1991.

[104] J.M. Ting. A robust algorithm for ellipse-based discrete element modelling of granular materials. *Comput. geotech.*, 13:175–186, 1992.

[105] Y. Tsuji, T. Tanaka, and T. Ishida. Lagrangian numerical simulation of plug flow of cohesionless particle in a horizontal pipe. *Powder Technology*, 71:239–250, 1992.

[106] R. Tuley, M. Danby, J. Shrimpton, and M. Palmer. On the optimal numerical time integration for lagrangian dem within implicit flow solvers. *Computers and Chemical Engineering*, 2010. accepted for publication.

[107] W.F. van Gunsteren and H.J.C. Berendsen. Algorithm for macromolecular dynamics and constraint dynamics. *Molecular physics*, 34(5):1311–1327, 1977.

[108] L.E. Walizer and J.F. Peters. A bounding box search algorithm for dem simulation. *Computer Physics Communications*, 182(2):281–288, 2011.

[109] O.R. Walton and R.G. Braun. Simulation of rotary-drum and repose tests for frictional spheres and rigid sphere clusters. In *in: Joint DOE/NSF Workshop on Flow of Particulatesand Fluids*, Ithaka, 1993.

[110] O.R. Walton and R.L. Braun. Viscosity, granular-temperatures, and stress calculations for shearing assemblies of inelastic, frictional disks. *Rheology*, 60(5):949–980, 1986.

[111] J.C. Williams. The rate of discharge of coarse granular materials from conical mass flow hoppers. *Chemical Engineering Science*, 32:247–255, 1977.

[112] C.Y. Wu, L.Y. Li, and C. Thornton. Energy dissiaption during normal impact of elastic and elastic-plastic spheres. *Impact Engineering*, 32:593–604, 2005.

[113] K. Yamane. *Simulating and Generating Motions of Human Figures*. Springer, 2004.

[114] C. Yin, L.A. Rosendahl, and S.K. Kaer. Grate-firing of biomass for heat and power production. *Progress in Energy and Combustion Science*, 34(6):725–754, 2008.

[115] D. Zhang and W.J. Whitten. The calculation of contact forces between particle using spring and damping models. *Powder Technology*, 88:59–64, 1996.

[116] Q.J. Zheng, H.P. Zhu, and A.B. Yu. Finite element analysis of the rolling friction of a viscous particle on a rigid plane. *Powder Technology*, 207:401–406, 2010.

[117] H.P. Zhu, Z.Y. Zhou, R.Y. Yang, and A.B. Yu. Discrete particle simulation of particulate systems: Theoretical developments. *Chemical Engineering Science*, 62:3378–3396, 2007.

[118] A. Džiugys and B. Peters. An approach to simulate the motion of spherical and non-spherical fuel particles in combustion chambers. *Granular matter*, 3:231–265, 2001.