

Nomenclature

Roman Symbols

\mathbf{b}	Body force
e	Void ratio of the sample (or) LBM fluid particle velocity
G	Gradient of the function (or) Shear modulus
K	Bulk modulus
N	Shape function or Number of grains/material points
R	Radius of a grain in DEM
r	Radius or hydrodynamic radius of a grain in LBM
V	Volume of a material point
w	Weight function / test function
p	Position of a material point / node

Greek Symbols

χ	Characteristic function
ε	Strain component (or) Coefficient of restitution
η_h	Packing density, defined as the ratio of volume of solids to the total volume (V_s/V)
γ	Viscous damping of a grain
Ω	Domain under consideration
ω	Angular velocity of a grain (or) Local contact natural frequency

φ Angular orientation of a grain

$\bar{\phi}$ Weighting function

Subscripts

i Property of the computational node (MPM) or grain (DEM)

max Maximum

min Minimum

p Property of the material point

Chapter 3

Numerical modelling of granular flow

3.1 Introduction

Most geotechnical analyses involve failure prediction and the design of structures that can safely withstand applied loads. However, it is very important to study the post-failure behaviour to mitigate risk posed by geophysical and gravity-driven flows, such as landslides, avalanches, slope failures, and debris flows. Granular flows are complex problems in continuum mechanics for which no closed-form solution exists. Hence, it is essential to develop alternative solution schemes which are capable of simulating failure mechanisms and post-failure dynamics of granular media.

The dynamics of a homogeneous granular flow involve at least three distinct scales: the *microscopic scale*, which is characterised by the contact between grains; the *meso-scale* that represents micro-structural effects such as grain rearrangement; and the *macroscopic scale*, where geometric correlations can be observed (figure 3.1). Conventionally, granular flows are modelled as continua because they exhibit many collective phenomena. However, on a grain scale, granular materials exhibit complex solid-like and/or fluid-like behaviours. Recent studies, however, suggest that a continuum law may be unable to capture the effect of inhomogeneities at the grain scale level, such as orientation of force chains. Discrete element methods (DEM) are capable of simulating these micro-structural effects; however, they are computationally expensive.

3.2 Continuum modelling of granular flow

Numerical techniques prove to be the most powerful way for modelling the granular assembly. It is important to argue why it is acceptable to model the granular materials as a continuum.

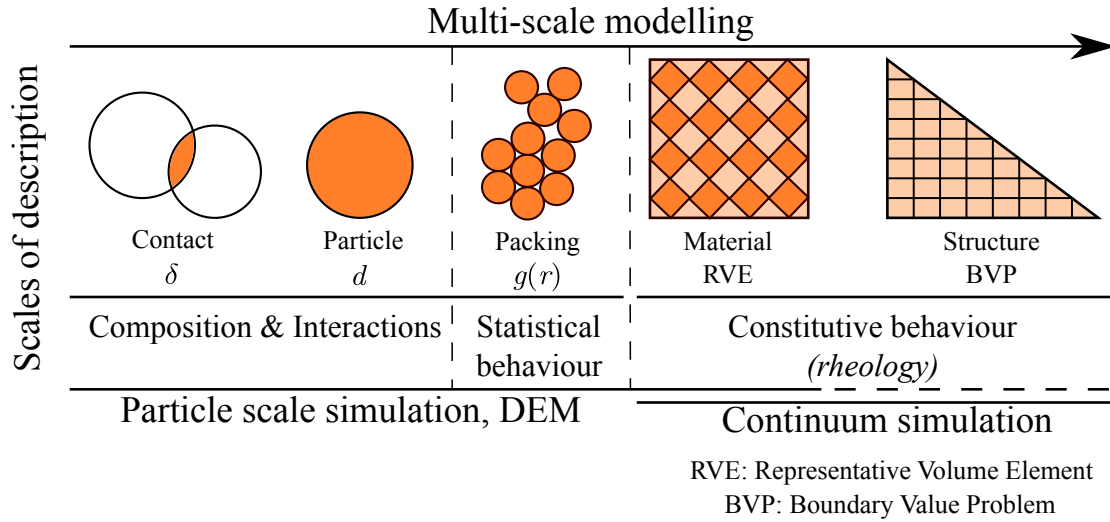


Figure 3.1 Schematic representation of different scales of description involved in the multi-scale modelling of granular materials

Even at the outset, it may appear that such a treatment is objectionable. The most blatant reason is the fact that the micro-constituents of granular matter, i.e. the individual grains, are not small enough to warrant a continuum description (Kamrin et al., 2007). Typical continuum laws are only expected to apply when there is a strong separation of scales, i.e. separation of the micro-scale from the macro-scale, in the flow geometry. Continuum mechanics rely on the fundamental notion of a representative volume element, in which properties averaged over discrete grains exhibit deterministic relationships. Recent work on granular materials suggests that a continuum law may be incapable of revealing inhomogeneities at the grain-scale level (Rycroft et al., 2009). Microscopic features of dense granular materials which seem to defy a simple continuum description include (i) complex, fractal networks of force chains, which are inhomogeneous down to the grain level (Goldhirsch, 2003), (ii) buckling of force chains and instabilities in shear band, (iii) anomalous, non-collisional particle dynamics and proximity to the jamming transition, where geometrical packing constraints suppress any dynamics, and (iv) a wide range of dynamic response, from liquid-like to solid-like behaviours (Aranson and Tsimring, 2001, 2002; Jaeger et al., 1996). The first three points cast serious doubt on the prospects of a continuum law. The third point entails that the full stress tensor must be described, and it may depend on strain, deformation rate, and material variables that can evolve during the process (Rycroft et al., 2009).

Granular materials exhibit many collective phenomena (Jaeger et al., 1996). However, no continuum model is yet capable of describing behaviours such as granular fingering on rough slopes, plug flow and the occurrence of localized shear bands in the granular materials. Most constitutive models, even in the simple case of dry granular flows, cannot describe

the entire range of flow from solid to fluid. In certain cases, granular flow is modelled as a fluid behaviour. Continuum models that are based on averaging techniques applied to representative volume elements are mostly utilised in problems involving quasi-static conditions. The fundamental question is how to meaningfully model granular materials which exhibit complex phenomena.

The oldest approach involves modelling the granular material as a rigid solid, which behaves as an ideal Coulomb material and undergoes failure if the ratio of the shear stress to the normal stress in any plane reaches a critical value of the Coulomb internal friction coefficient μ . The stress is determined based on the mechanical equilibrium of the system along with the hypothesis of *incipient yield*, i.e. the yield criterion is attained everywhere at all times (in a “*limit-state*”) (Rycroft et al., 2009). The fundamental assumption of a limit-state stress field at incipient yield everywhere is questionable. Granular flows can contain regions of stress lying within the yield surface. In fact, discrete-element simulations show that the grains in this region essentially remain static (Staron et al., 2005). Some of the limitations of the Mohr-Coulomb plasticity in modelling granular materials include linearisation of the limit-stress envelope and the inability to account for variable volume change characteristics depending on pre-consolidation pressure during shearing, predicting unrealistic negative pore-pressure upon shearing due to constant rate of dilation, and the inability to predict compressive plastic strains in soft-soils. Mohr-Coulomb plasticity has been used extensively in engineering applications (Nedderman, 1992), but the general solution requires sophisticated numerical techniques to capture shock-like discontinuities in stress and velocity, which arise even in relatively simplistic geometries.

The coaxiality feature of Mohr-Coulomb plasticity is useful in describing debris flows. The principle of coaxiality claims that material should flow by extending along the minor principal stress direction and contracting along the major principal stress direction; the principal planes of stress are aligned with the principal planes of strain-rate. Granular materials deform solely based on the alignment of the principal planes. Under this assumption, the major principal plane is usually vertical due to gravity, and the coaxiality rule requires the material to expand horizontally, as is the case for granular column collapse. However, the coaxiality can be troubling depending on the circumstances. Consider a flat-bottomed quasi-2D silo with smooth side-walls. Under standard filling procedures, the walls provide only enough pressure to keep the grains from sliding farther out. For example, in a slow dense granular flow through a silo, the principal plane remains vertical and coaxiality requires the granular material to expand horizontally, thus making it geometrically impossible for the granular material to converge and exit through the orifice. Depending on the boundary conditions, Mohr-Coulomb plasticity can result in discontinuities or jumps in the velocity and

stress fields (Rycroft et al., 2006). Coaxiality can also violate principles of thermodynamics. Coaxiality only ensures that there is no shear strain-rate in the principal stress reference frame and actually does not directly enforce that the two principal strain-rate axes and the axis of maximal compression (i.e. the major principal strain-rate direction) must align with the major principal stress direction.

Roscoe (1970) showed that the principal axes of strain rate and of stress are generally not coincidental during simple shear tests of sand. Despite the plastic character of the flow, the flow rules obtained by employing the rheological concept of perfect plasticity to granular materials (perfect soil plasticity) are definitely inadequate for a description of fully-developed shear flow. The non-coaxiality of the tensors of stress and strain increment are able to explain the shear flow behaviour. The rotations of the principal stress and the principal plastic strain rate are found to be non-coaxial, particularly at the early stage of loading. However, the axes tend to become coincidental at large shear strains (Yu and Yuan, 2006), which are common in geophysical flows.

Advanced elasto-plastic models based on the *critical state* theory provide a better representation of granular flows in a quasi-static regime, but do not have any rate-dependent effects that are associated with rapid granular flows. At large shear strains, such as geophysical flows, the granular mass is considered to be in a critical state. Another continuum-based model is the partial fluidisation model, which uses a set of equations that describes the flow velocity and the shear stresses along with an auxiliary order parameter to predict granular flow behaviour. The order parameter of the granular media controls the size of the viscous-like contribution to the stress tensor and describes the transition between the flowing and the static components of the granular system (Aranson and Tsimring, 2001). A constitutive model, which considers the solid fraction as the main microscopic parameter for describing dense granular flow, was proposed by Josserand et al. (2004). The stress in the granular material is divided into a rate-dependent part representing the reboundless impact between grains and a rate-independent part associated with longer contacts, i.e. quasi-static regime. Although the model captures shear localization behaviour, it fails to describe granular flow behaviour at rough boundaries. In the basal or frictional layer, grain rotation plays an important role. This rotation causes the grain stress tensor to be non-symmetric and results in a perturbed profile in the solid fraction, phenomena which require specific modelling (Josserand et al., 2004).

In the case of saturated/submerged soil conditions, most continuum techniques do not consider fully coupled behaviours. Rather, they consider the soil-fluid mixture as a single phase material. However, modelling of pore pressure dissipation is important to capture accurate flow behaviour, especially in submarine conditions, where they play a crucial role in the flow dynamics. Presence of ambient fluid may retard the flow due to viscous drag

or accelerate the flow through a lubrication effect. Fully coupled constitutive models are essential to realistically capture the initiation and propagation of rapid granular flows.

Granular materials are composed of distinct grains which interact only at the contact points. It is assumed that the deformations of individual grains are negligible in comparison with the deformation of the granular assembly as a whole. The latter deformation is primarily due to the movement of the grains as rigid bodies. Therefore, it can be argued that precise modelling of grain deformation is not necessary to obtain a good approximation of the overall mechanical behaviour. An Eulerian grain-level continuum model describes the response of individual grains to the applied loads. However, continuum mechanics solves over the whole domain using initial and boundary conditions appropriate for the problem. Hence, continuum models are still widely used to solve engineering problems associated with granular materials and flows.

3.2.1 Mesh-based and mesh-free techniques

In continuum mechanics, there are two different descriptions of the deformation of a continuum, namely Lagrangian and Eulerian descriptions. In the Lagrangian description, the movement of the continuum is specified as a function of the material coordinates and time. This particle description is often applied in solid mechanics. The Eulerian description, on the other hand, focuses on the current configuration focusing on a fixed point in space as time progresses, instead of individual particles as they move through space and time. The Eulerian description is commonly used for describing fluid flows where kinematic properties are of particular interest.

Conventional mesh-based Lagrangian approaches, such as the Finite Element Method or the Finite Difference Method, are capable of modelling history-dependent material behaviour and have well-defined free surfaces. However, they require complex re-meshing and remapping of variables, causing additional errors in simulating large deformation problems (Li and Liu, 2002). Unlike in Lagrangian FEM, the computational mesh in the Eulerian FEM is kept spatially fixed while the material is deforming in time. The Eulerian description produces the capability for handling large deformations without the problem of mesh distortion. As the computational mesh is completely decoupled from the material, convective terms appear in the Eulerian FEM, introducing numerical difficulties because of their non-symmetrical properties (Donea et al., 1982). Additionally, Eulerian FEM is difficult to use with history-dependent constitutive models. The Coupled Eulerian–Lagrangian (CEL) method is an arbitrary Lagrangian-Eulerian method that attempts to capture the advantages of both the Lagrangian and the Eulerian method in modelling large deformation problems in geomechanics (Qiu et al., 2011). This approach involves solving the governing equations in a

Lagrangian step, thus obtaining the material displacement, followed by the Eulerian step where a new mesh is generated and the variables are transferred to the new mesh. This requires greater computation time.

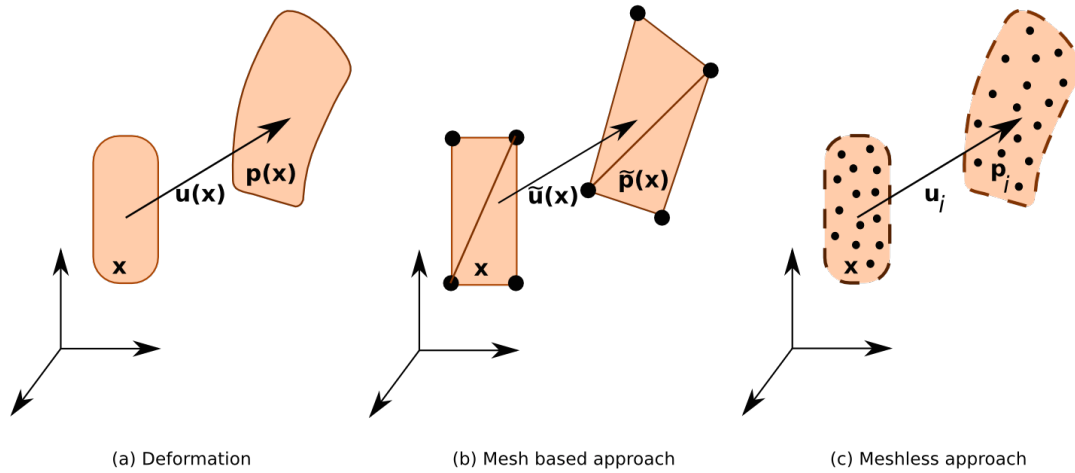


Figure 3.2 Difference between mesh-based and mesh-free techniques in modelling large deformation flow.

An alternative to the mesh-based approach is the use of mesh-less Lagrangian methods (figure 3.2) where the nodes representing the solids transform as the continuum deforms, avoiding the problem of mesh distortion, i.e. the nodes representing the solids can move freely within the domain. Mesh-free methods, such as Smooth Particle Hydrodynamics and Material Point Method, are not constrained by the mesh size and mesh distortion effects, and hence can be effectively used in simulating large deformation problems, such as debris flow and submarine landslides.

The element-free Galerkin (EFG) method is a relatively new mesh-less method, in which the trial functions for the weak form are constructed using moving least squares interpolation (Belytschko et al., 1994). The particle finite element method (PFEM) is another mesh-less method that involves mesh-less finite element interpolation. In PFEM, the nodal points represent the particles and the computational mesh is constructed by connecting these points. The mesh is then used to solve the governing equations in a Lagrangian fashion. In PFEM, large deformation requires frequent re-meshing (Kafaji, 2013).

Smooth Particle Hydrodynamics is the oldest mesh-free technique, in which the domain is discretised into particles that have a spatial distance, called the *smoothing length* over which the material properties are “smoothed” by a kernel function. SPH was developed to solve astrophysical problems (Monaghan, 2005). SPH has been applied in geomechanics for solving large deformation problems (Augarde and Heaney, 2009; Maeda and Sakai, 2010;

Mori, 2008). Although SPH has been successfully used, it has a few drawbacks: SPH exhibits spatial instabilities, as a consequence of the point-wise integration (Bonet and Kulasegaram, 2000), insufficient neighbouring particles causes inconsistencies, and it is computationally expensive as a result of the search for the neighbouring particles (Bandara, 2013).

3.3 Material Point Method (MPM)

The Material Point Method (MPM) (Sulsky et al., 1994, 1995) is a particle based method that represents the material as a collection of *material points*, and their deformations are determined by *Newton's laws of motion*. Sulsky et al. (1994) extended the Particle-in-Cell (PIC) method (Harlow, 1964) to computational solid mechanics by taking advantage of the combined Eulerian-Lagrangian approach. The MPM is a hybrid Eulerian-Lagrangian approach, which uses moving material points and computational nodes on a background mesh. This approach is very effective particularly in the context of large deformations (Andersen and Andersen, 2010; Bandara, 2013; Mackenzie-Helnwein et al., 2010; Mast et al., 2014a; Shin, 2010; Zhang et al., 2009). Although not derived directly from what are classically considered as mesh-free or mesh-less methods, MPM is still considered as a mesh-free approach, primarily because the initial discretisation of the material does not involve a polygonal tessellation, as in the Finite Element Method. However, MPM utilizes a background mesh to perform differentiation, integration, and to solve equations of motion (Steffen et al., 2008). The background mesh can be of any form, though for computational efficiency a Cartesian lattice is adopted.

A typical 2D discretisation of a solid body is shown in figure 3.3. The grey circles in figure 3.3 are the material points x_p , where 'p' represents a material point, and the computational nodes are the points of intersection of the grid (denoted as X_i , where i represents a computational node). MPM involves discretising the domain, Ω , with a set of material points. The material points are assigned an initial value of position, velocity, mass, volume, and stress, denoted as \mathbf{x}_p , v_p , m_p , \mathbf{V}_p and σ_p , respectively. Depending on the material being simulated, additional parameters, like pressure, temperature, pore-water pressure, etc., are specified at the material points. The material points are assumed to be within the computational grid which for ease of computation, is assumed to be a Cartesian lattice (figure 3.3). At every time step t_k , the MPM computation cycle involves projecting the data, such as position, mass, and velocity, from the material points to the computational grid using the standard nodal basis functions, called the *shape functions*, derived from the position of the particle with respect to the grid. Gradient terms are calculated on the computational grid, and the governing equation, i.e. the equation of motion, is solved with the updated

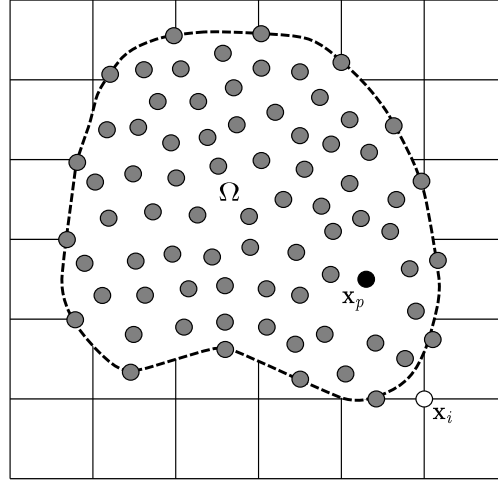


Figure 3.3 Typical discretisation of a domain in MPM. The dotted line represents the boundary of the simulated object, Ω , and each closed point represents a material point used to discretise Ω . The square mesh represents the background grid. Each square in the background grid is a grid cell, and grid nodes are located at the corners of grid cells.

position and velocity values mapping back to the material points. The mesh is reinitialized to its original state and the computational cycle is repeated.

3.3.1 Discrete formulation of the governing equations

The governing differential equation for a continuum is derived from the conservation of mass and momentum:

$$\frac{\partial \rho}{\partial t} \rho \Delta \cdot \mathbf{v} = 0, \quad (3.1)$$

$$\rho \mathbf{a} = \Delta \cdot \boldsymbol{\sigma} + \rho \mathbf{b}, \quad (3.2)$$

where $\rho(\mathbf{x}, t)$ is the mass density, $\mathbf{v}(\mathbf{x}, t)$ is the velocity, $\mathbf{a}(\mathbf{x}, t)$ is the acceleration, $\boldsymbol{\sigma}(\mathbf{x}, t)$ is the Cauchy's stress tensor, and $\mathbf{b}(\mathbf{x}, t)$ is the body force. The vector \mathbf{x} represents the current position of any material point in the continuum at time t . In MPM, the continuum body is discretised into a finite number of material points, N_p . Let \mathbf{x}_p^t ($p = 1, 2, \dots, N_p$) denote the current position of material point p at time t . Each material point, at any given time t , has an associated mass m_p^t , density ρ_p^t , velocity \mathbf{v}_p^t , Cauchy stress tensor $\boldsymbol{\sigma}_p^t$, strain $\boldsymbol{\epsilon}_p^t$, and other necessary internal state variables based on the adopted constitutive model. These material points provide a Lagrangian description of the continuum body. Since material points have a fixed mass at all times, eq. 3.1 is satisfied. The data from the material points are mapped on to the nodes of the computational grid, where the discrete form of eq. 3.2 is described. The

weak form of eq. 3.2 is obtained by multiplying eq. 3.2 with a test function $w(\mathbf{x}, t)$:

$$\int_{\Omega} \rho \mathbf{w} \cdot \mathbf{a} d\Omega = - \int_{\Omega} \rho \boldsymbol{\sigma}^s : \Delta \mathbf{w} d\Omega + \int_{\partial\Omega_{\tau}} w \cdot \boldsymbol{\tau} dS + \int_{\Omega} \rho w \cdot \mathbf{b} d\Omega, \quad (3.3)$$

where $\boldsymbol{\sigma}^s$ is the specific stress (i.e. stress divided by mass density, $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}/\rho$), Ω is the current configuration of the continuum and $\boldsymbol{\tau}$ is the surface traction. Eq 3.3 is obtained by applying the divergence theorem, similar to the standard procedure adopted in Finite Element Methods (Chen and Brannon, 2002; Sulsky et al., 1994, 1995). The differential volume and the surface elements are denoted by $d\Omega$ and dS , respectively.

As the whole continuum is discretised into a finite set of material points, the mass density can be written as

$$\rho(\mathbf{x}, t) = \sum_{p=1}^{N_p} m_p \delta(\mathbf{x} - \mathbf{x}_p^t), \quad (3.4)$$

where δ is the Dirac delta function. Substituting eq. 3.4 in eq. 3.3, the sum of quantities of material points can be evaluated as

$$\begin{aligned} \sum_{p=1}^{N_p} m_p [w(\mathbf{x}_p^t, t) \cdot \mathbf{a}(\mathbf{x}_p^t, t)] &= \sum_{p=1}^{N_p} m_p [-\boldsymbol{\sigma}^s(\mathbf{x}_p^t, t) : \Delta w|_{\mathbf{x}_p^t} \\ &\quad + w(\mathbf{x}_p^t, t) \cdot \boldsymbol{\tau}^s(\mathbf{x}_p^t, t) h^{-1} + w(\mathbf{x}_p^t, t) \cdot \mathbf{b}(\mathbf{x}_p^t, t)], \end{aligned} \quad (3.5)$$

where h is the thickness of the boundary layer upon which the traction boundary conditions are enforced. Since the continuum body is moving in an arbitrary computational mesh, all the boundary conditions are carried by the boundary particles. If only one boundary particle is located in a cell, the cell boundary becomes a part of the continuum boundary, and the cell size represents the thickness of boundary layer. If both boundary and interior particles of the continuum are located in a cell, this cell becomes a mixed one. However, the mixed cell is still treated as a boundary cell. In other words, the interior particles temporarily become boundary ones. To avoid numerical errors, therefore, small cells must be used to contain only boundary particles if possible, and the boundary conditions are enforced in each time step.

It can be noted from eq. 3.5 that the interactions between different material points are reflected only through the gradient terms. In MPM, a background computational mesh is used to calculate the gradient terms. The computational mesh is constructed using 2-node cells for 1-D, 4-node cells for 2-D, and 8-node cells for 3-D problems. These elements are used to define the standard nodal basis functions, $N_i(\mathbf{x})$, associated with the spatial nodes $\mathbf{x}_i(t)$, $i = 1, 2, \dots, N_n$, where N_n represents the total number of mesh nodes. The nodal basis functions are assembled by using the conventional finite-element shape functions (Chen and

Brannon, 2002). The coordinates of any material point in a cell can be represented by

$$x_p^t = \sum_{i=1}^{N_n} x_i^t N_i(\mathbf{x}_p^t). \quad (3.6)$$

Similarly the nodal displacements, velocity and acceleration of any material point in a cell are represented using the basis functions. Thus, the test function must be of the form

$$w_p^t = \sum_{i=1}^{N_n} w_i^t N_i(\mathbf{w}_p^t). \quad (3.7)$$

Equations (3.6) and (3.7) ensure that the associated vectors are continuous across the cell boundary. However, the gradient of these functions is not continuous, due to the use of linear shape functions. Substituting eq. 3.6 and eq. 3.7 into eq. 3.5, the weak form of the equation of motion reduces to

$$\sum_{j=1}^{N_n} m_{ij}^t \mathbf{a}_j^t = \mathbf{f}_i^{int,t} + \mathbf{f}_i^{ext,t}, \quad (3.8)$$

where the nodal mass, m_{ij}^t , is represented as

$$m_{ij}^t = \sum_{p=1}^{N_p} m_p N_i(x^t) N_j(x^t). \quad (3.9)$$

The nodal internal force, $\mathbf{f}_i^{int,t}$ and the nodal external force, $\mathbf{f}_i^{ext,t}$ are defined as

$$\begin{aligned} \mathbf{f}_i^{int,t} &= - \sum_{p=1}^{N_p} m_p \mathbf{G}_{ip}^t \cdot \boldsymbol{\sigma}_p^{s,t}, \\ \mathbf{f}_i^{ext,t} &= - \sum_{p=1}^{N_p} m_p \mathbf{b}_p^t N_i(\mathbf{x}_p^t) + \sum_{p=1}^{N_p} m_p N_i(\mathbf{x}_p^t) \boldsymbol{\tau}_p^{s,t} h^{-1}, \end{aligned} \quad (3.10)$$

where $\mathbf{G}_{ip}^t = \Delta N_i(x)|_{x=X_p^t}$. The nodal accelerations are obtained by explicit time integration of eq. 3.8. To obtain stable solutions, the time step used in the analysis should be less than the critical time step, which is defined as the ratio of the smallest cell size to the wave speed (Chen and Brannon, 2002). The critical time increment is obtained as

$$\Delta t_{crit} = L/c, \quad (3.11)$$

$$c = \frac{K + \frac{4}{3}G}{\rho_s}, \quad (3.12)$$

where L is the background cell size, c is the pressure wave velocity, K and G are the bulk modulus and the shear modulus of the solid and ρ_s is the density of the soil skeleton. The boundary conditions are enforced on the cell nodes, and the nodal velocities are obtained by solving the equation of motion at each node. The strain increment for each material point is determined using the gradients of the nodal basis functions. The corresponding stress increments are computed using the adopted constitutive law. After updating all the material points, the computational mesh is discarded, and a new mesh is defined for the next time step.

3.3.2 Boundary conditions

The Material Point Method uses standard shape functions, similar to those used in the Finite Element Methods. Therefore the essential and the natural boundary conditions can be applied to the background grid nodes in the same way as in the traditional FEM. The free surface boundary conditions are satisfied, as the MPM is formulated in the weak form. Implementation of traction boundary conditions requires a set of material points to represent the boundary layer. Bandara (2013) proposed a friction interaction for the planar boundary condition using Coulomb's friction criterion. The friction boundary algorithm for the solid phase adopted in the present study is shown in algorithm 3.1. The friction boundary conditions are applied on the mesh nodes by controlling the nodal acceleration tangential to the boundary. The nodal accelerations are considered to include the frictional effects instead of the forces, as the forces are proportional to the corresponding accelerations. Both static and kinetic friction are considered, and applied only when the particles are in contact with the boundary. The static and kinematic frictions are applied in the direction tangential to the nodal boundary. Friction forces are applied only if the particles are in contact. The normal velocity and acceleration on the boundary plane is zero. Displacement boundary conditions are applied as velocity constraints on the nodes in the background mesh.

3.3.3 Integration scheme

Love and Sulsky (2006) investigated the energy consistency of MPM and observed that the MPM algorithm is better suited to flow calculations than the Lagrangian finite element method. The energy-consistent MPM formulation is effective in simulating materials that exhibit localised dissipative mechanisms. In dynamic MPM, an explicit time integration scheme is adopted to advance the solution. Bardenhagen (2002) studied the energy consistency of MPM using two different explicit integration schemes. The *update stress first* (USF) scheme involves updating the strain and the stress at the beginning of the time step from the velocities

Algorithm 3.1 Friction boundary algorithm for solid phase in MPM (Bandara, 2013)

```

if  $a_{sl,t}^{k+1} < 0.0$  then
   $v_{sl,t}^{k+1,tmp} = v_{sl}^k + \Delta a_{sl,t}^{k+1}$ 
  if  $v_{sl,t}^{k+1,tmp} = 0$  then ▷ static friction condition
    if  $a_{sl,t}^{k+1} \leq \mu_s |a_{sl,n}^{k+1}|$  then
       $a_{sl,t}^{k+1} = 0.$ 
    else ▷ friction acts in the direction opposite to the tangential force
       $a_{sl,t}^{k+1} = a_{sl,t}^{k+1} - \mu_s |a_{sl,n}^{k+1}| \frac{a_{sl,t}^{k+1}}{|a_{sl,n}^{k+1}|}$ 
    end if
  else ▷ kinetic friction condition
    if  $v_{sl,t}^{k+1} \leq \mu_k |a_{sl,n}^{k+1}| \Delta t$  then
       $a_{sl,t}^{k+1} = -\frac{v_{sl,t}^k}{\Delta t}$ 
    else ▷ friction acts in the direction opposite to movement
       $a_{sl,t}^{k+1} = a_{sl,t}^{k+1} - \mu_k |a_{sl,n}^{k+1}| \frac{v_{sl,t}^{k+1}}{|v_{sl,n}^{k+1}|}$ 
    end if
  end if
end if

```

of the previous time step. In the *update stress last* (USL) approach, the updated particle momentums are used to calculate the nodal velocities, which are then used to update the particle strain and stress. Bardenhagen (2002) observed that the USL approach performed better than the USF. The USL approach dissipates the energy slowly, while the USF approach is found to gain energy (Kafaji, 2013). The USL approach yields almost the same result as using the central difference scheme that is second order in time (Wallstedt and Guilkey, 2008). The *update stress last* approach is used in the present study due to its dissipative nature, which is useful in modelling granular flow problems and numerical stability.

In problems involving a slow rate of loading, i.e. quasi-static problems, the flow of the material is much slower than the speed of wave propagation in the material. Hence, employing an implicit time integration scheme reduces the computational time considerably (Kafaji, 2013). Guilkey and Weiss (2003) proposed an implicit time integration method for MPM using quasi-static governing equations and the Newmark integration scheme. Love and Sulsky (2006) showed that implicit time integration in MPM is unconditionally stable. Although, MPM does not suffer from the limitations of FEM in simulating large deformations, more research is required for applying implicit time integration for large deformation problems. The present study focuses on large deformation problems, hence an explicit time integration scheme with the USL approach is adopted.

3.3.4 Solution scheme

In the present study a template-based three-dimensional C++11 Material Point Method code, developed at the University of Cambridge (Bandara, 2013), is modified and extended to study granular flow problems. The three-dimensional MPM code is parallelised to run on multi-core systems, thus improving the computational efficiency. The algorithm of the MPM code is improved to handle multi-body dynamics and interactions. A step-by-step solution scheme for the Material Point Method implemented in the present study is described below:

- A continuum body is discretised into a finite set of material points corresponding to the original configuration of the body. The number of material points corresponds to the resolution of the mesh size adopted in the Finite Element Method. The material points are followed throughout the deformation of the material, which is a Lagrangian description of the motion.
- An arbitrary computational grid is initialized to describe the natural coordinates of the material points. For the purpose of simplicity, a Cartesian grid is usually adopted.
- The state variables (mass/density, velocity, strain, stress, other material parameters corresponding to the adopted constitutive relation) are initialized at every material point.
- The shape function $N_{ip}^t(x_p)$ and the gradient of the shape function G_{ip}^t for each material point are computed.
- The information and properties carried by each material point are projected onto the background mesh using the shape functions computed from the particle position.
- The nodal mass matrix is obtained as

$$m_i^t = \sum_{p=1}^{N_p} m_p N_{ip}^t(x_p^t), \quad (3.13)$$

where m_i^t is the mass at node i at time t , m_p is the particle mass, N_i is the shape function associated with node i , and x_p^t is the location of the particle at time t .

- The nodal velocity is obtained by mapping the particle velocity onto the nodes using the shape functions. If necessary, the boundary conditions for the nodal velocities are applied.

$$\mathbf{v}_i^t = \sum_{p=1}^{N_p} m_p \mathbf{v}_p^t N_{ip}^t(x_p^t) / m_i^t. \quad (3.14)$$

- The momentum balance equation for the solid phase is solved, and the nodal acceleration is computed as

$$a_i^t = \frac{1}{m_i^t} \left(- \sum_{p=1}^{N_p} G_{ip}^t \sigma_p^t \Omega_p^t + \sum_{p=1}^{N_p} m_p^t \mathbf{b}_p^t N_{ip}^t(x_p^t) \right). \quad (3.15)$$

If necessary, the boundary conditions for the nodal accelerations are applied.

- The nodal velocity at the end of the Lagrangian time step (L) is obtained from the computed nodal acceleration as

$$v_i^L = \mathbf{v}_i^t + a_i^t \Delta t. \quad (3.16)$$

where $\Delta t = (t + 1) - t$.

- The particle position and its velocity are updated according to

$$\begin{aligned} \mathbf{x}_p^{t+1} &= \mathbf{x}_p^t + \Delta t \sum_{i=1}^{N_n} v_i^L N_{ip}^t, \\ \mathbf{v}_p^{t+1} &= \mathbf{v}_p^t + \Delta t \sum_{i=1}^{N_n} a_i^t N_{ip}^t. \end{aligned} \quad (3.17)$$

- The strain increment $\Delta \boldsymbol{\varepsilon}_p^{t+1}$ for the particle is then computed as

$$\Delta \boldsymbol{\varepsilon}_p^{t+1} = \frac{\Delta t}{2} \sum_{i=1}^{N_n} G_{ip}^t \mathbf{v}_i^t + (G_{ip}^t \mathbf{v}_i^t)^T. \quad (3.18)$$

- The stress increment for the particle $\Delta \boldsymbol{\sigma}_p^{t+1}$ is computed from the strain increment using the constitutive model adopted in the simulation

$$\Delta \boldsymbol{\sigma}_p^{t+1} = \mathbf{D} : \Delta \boldsymbol{\varepsilon}_p^{t+1}. \quad (3.19)$$

In large deformation problems, the Jaumann rate is used to update the effective stress of the solid particles

$$\boldsymbol{\sigma}_p^{t+1} = \Delta t \left(\boldsymbol{\sigma}_p^t - \mathbf{W}_p^t - \mathbf{W}_p^t \boldsymbol{\sigma}_p^t \right) + \mathbf{D} : \Delta \boldsymbol{\varepsilon}_p^{t+1}, \quad (3.20)$$

$$\mathbf{W}_p^t = \sum_{i=1}^{N_n} \left[\mathbf{G}_{ip}^t \mathbf{v}_i^t - (\mathbf{G}_{ip}^t \mathbf{v}_i^t)^T \right]. \quad (3.21)$$

- The stress and the strain of the material points are updated based on

$$\begin{aligned}\sigma_p^{t+1} &= \sigma_p^t + \Delta\sigma_p^{t+1}, \\ \epsilon_p^{t+1} &= \epsilon_p^t + \Delta\epsilon_p^{t+1}.\end{aligned}\tag{3.22}$$

- In large deformations, the volume of the solid material points Ω_p is updated using the determinant J of the deformation gradient \mathbf{F}_p^{t+1}

$$\Omega_p^{t+1} = J\Omega_p^{t_0}.\tag{3.23}$$

- The material point density is then updated as

$$\rho_p^{t+1} = \frac{\rho_p^t}{\{1 + tr(\Delta\epsilon_p^{t+1})\}}.\tag{3.24}$$

- At the end of every time step, all the variables on the grid nodes are initialized to zero. The material points carry all the information about the solution, and the computational grid is re-initialised for the next step.

Figure 3.4 illustrates the steps involved in a MPM analysis.

Post-processing

The post-processing stage, as in most analysis, involves visualization and extraction of the data from the analysis. In mesh-less methods, like MPM, structures are generally represented as points which describe a discrete region of the body. MPM facilitates representation of arbitrarily complex geometries and has advantages over strictly grid-based methods, especially in simulations involving large deformations (Bardenhagen et al., 2000). However, MPM poses a whole new set of visualization problems. It is essential to visualize the general configuration of the body as well as observe the finer details like the development of cracks or separation of chunks of material from the body. The body is discretised into conceptual material points, which carry all the relevant information of the corresponding segment. The unique qualities of MPM necessitate the need to visualize the particle data in a way that is informative and appropriate.

In MPM, the particle data represent the finite portion of a larger continuum, and the ability to see and interpret the macroscopic structure created by these particles is vital (Bigler et al., 2006). There are two main aspects in visualizing MPM data: (1) visualization of

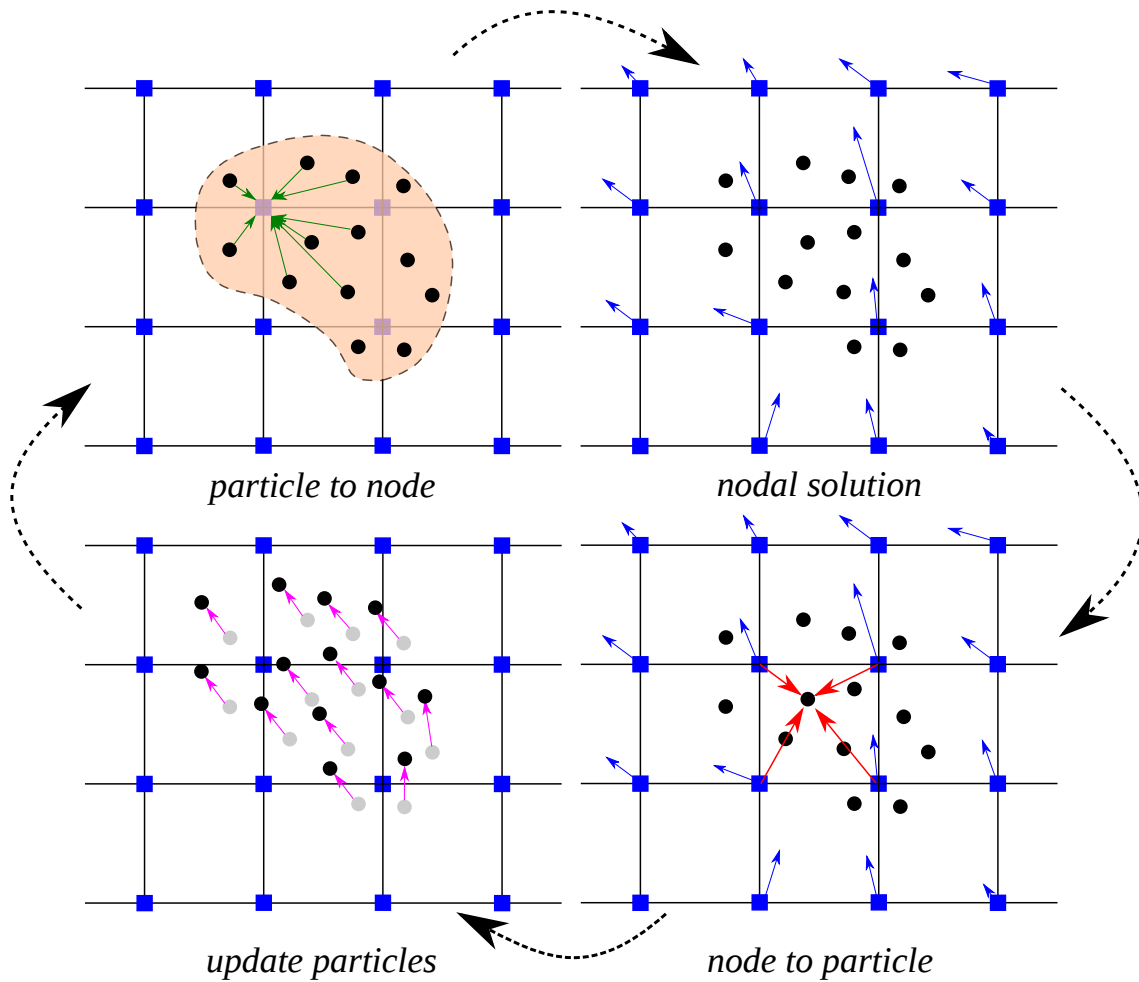


Figure 3.4 Illustration of MPM algorithm (1) A representation of material points overlaid on a computational grid. Arrows represent material point state vectors (mass, volume, velocity, etc.) being projected to the nodes of the computational grid. (2) The equations of motion are solved onto the nodes, resulting in updated nodal velocities and positions. (3) The updated nodal kinematics are interpolated back to the material points. (4) The state of the material points is updated, and the computational grid is reset.

the structure represented by the material points and (2) understanding the qualitative trends associated with the material points like mass, velocity or stress.

The MPM output data contain both the material point and the grid data, and one approach in visualizing MPM data is to render the interpolated particle values on grid nodes using the *iso-surfacing* (Lorensen and Cline, 1987) or *volume rendering* (Levoy, 1988). In regions where the material points are sparse, it is necessary that the grid resolution is sufficiently fine to compensate for the missing features. This results in storing a large amount of unnecessary data in regions where sufficient material points are present. Thus, it is advantageous to visualize MPM data of the material points as particles (Bigler et al., 2006). Particle visualization involves rendering the particles as a spheres or an ellipsoids representing the size and location of the fraction of the continuum (Gumhold, 2003; Krogh et al., 1997; Kuester et al., 2001). In the present study, MPM data points are represented as spheres. Colour mapping of scalar quantities, such as mass, velocity, or stress, of a material point are applied to provide additional qualitative understanding of the data.

3.3.5 GIMP method

The shape functions used in MPM are continuous, and hence penetrations between bodies are handled automatically without the need for any supplemental contact algorithm (Chen and Brannon, 2002). In MPM, the continuum body deforms and moves in an arbitrary computation grid. All boundary conditions are carried by the boundary particles. If a boundary particle is present in a cell, then the cell boundary becomes a part of the continuum body, and the cell size represents the thickness of the boundary. In certain cases, however, both the boundary particle and an interior particle can be found in a cell. In such cases, the cell is still treated as a boundary cell, and the interior particle temporarily acts as a boundary particle. To avoid numerical errors, it is essential to use a smaller cell size along the boundary (Chen and Brannon, 2002).

In MPM simulations, numerical noise is observed when material points cross the cell boundaries as the body deforms. The noise is termed *cell crossing noise*. If a material point is located very close to the cell boundary. Then the results is a discontinuous gradient of the weighing function causing a force imbalance on the grid (Bardenhagen and Kober, 2004). This results in large, non-physical acceleration values causing separation of material points from the continuum (Sulsky et al., 1995). Figure 3.5 illustrates the problem of cell crossing noise. The main reason for the occurrence of cell crossing noise is the use of piecewise linear shape functions. However, this problem, which is predominant when using fine mesh size, can be overcome by changing the order of arithmetic operation, as proposed by Sulsky et al. (1995).

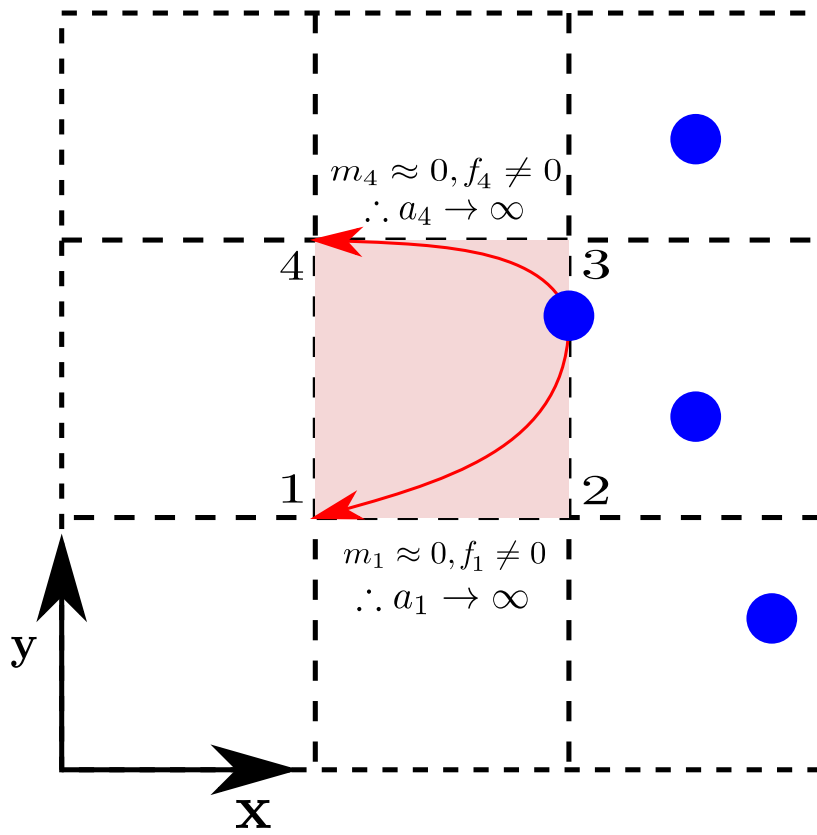


Figure 3.5 Schematic description of the occurrence of cell crossing noise in MPM.

To overcome the problem of cell crossing noise, Bardenhagen and Kober (2004) proposed an alternate method called the Generalized Interpolation Material Point Method that uses smoother shape functions and a larger influence region for each grid node. This approach minimizes the cell crossing noise. The piecewise-linear grid basis functions used are written as

$$\bar{\phi}(x) = \begin{cases} 1 - |x|/h & : & |x| < h \\ 0 & : & \text{otherwise,} \end{cases} \quad (3.25)$$

where h is the grid spacing. The basis function associated with grid node i at position x_i is then $\bar{\phi}_i = \bar{\phi}(x - x_i)$. The basis functions in 3-D are separable functions constructed as $\bar{\phi}_i(x) = \bar{\phi}_i^x(x)\bar{\phi}_i^y(y)\bar{\phi}_i^z(z)$. GIMP is often implemented using the standard piecewise-linear grid basis functions and piecewise-constant particle characteristic functions:

$$\chi_p = \begin{cases} 1 & : & |x| < \frac{1}{2}l_p \\ 0 & : & \text{otherwise,} \end{cases} \quad (3.26)$$

in which case the 1-D MPM and GIMP weighting functions can be grouped together in the general form

$$\bar{\phi} = \begin{cases} 1 - (4x^2 + l_p^2)/(4hl_p) & : & |x| < \frac{l_p}{2} \\ 1 - |x|/h & : & \frac{l_p}{2} \leq |x| < h - \frac{l_p}{2} \\ \left(h + \frac{l_p}{2} - |x|\right)^2/(2hl_p) & : & h - \frac{l_p}{2} \leq |x| < h + \frac{l_p}{2} \\ 0 & : & \text{otherwise,} \end{cases} \quad (3.27)$$

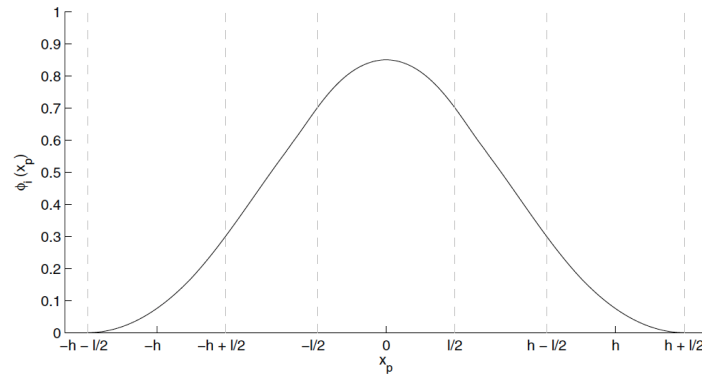
where l_p is the width of the particle characteristic function χ_p . Figure 3.6a shows a 1-D GIMP weighting function $\bar{\phi}_{ip}$ and gradient weighting function $\bar{\nabla}\phi_{ip}$ for a piecewise-constant χ_p with a characteristic length of l . The GIMP weighting function is smooth; however, a discontinuity is observed in the gradient weighting function.

In traditional MPM, boundary conditions need only be applied on those nodes which coincide with the extent of the computational domain. As illustrated in figure 3.6c nodes beyond those boundaries are not influenced by particles within the domain, which can be considered a result of the zero width of the Dirac delta characteristic functions. However, special attention is required to simulate the boundaries in the Generalized Interpolation Material Point Method (GIMP). Namely, because of their increased extents, it is possible for particles to influence, and be influenced by, nodes that lie outside of the simulation domain

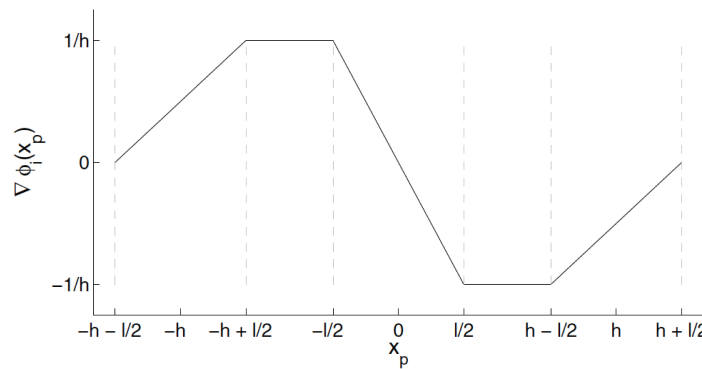
(figure 3.6b). These extra nodes are referred to as the “ghost” nodes. Boundary condition treatment of these nodes for Dirichlet conditions is the same as for the regular boundary nodes, namely, their computed values are replaced by prescribed values (Steffen et al., 2008).

In the present study, the influence of the GIMP method on the run-out behaviour of a granular column collapse experiment is investigated. ?? shows a granular column of height H_0 , and length L_0 , is allowed to collapse and flow on a horizontal plane. The run-out distance observed is proportional to the initial aspect ratio of the column (H_0/L_0). A granular column with an initial aspect ratio of 0.4 is considered for the comparison. The granular column is represented by 32,000 material points arranged uniformly on a regular lattice with a particle spacing of 0.25 mm. Since the scale of the problem being modelled is small and it is important to precisely define the flow surface, a larger number of material points are used to represent the geometry. When modelling geophysical problems, where the tolerance in defining the flow surface is in the order of a few millimetres, however, each material point shall represent individual grains or orders of magnitude larger than the grain size. A grid size of 1 mm is adopted with 16 material points per cell. In order to understand the influence of the number of material points on the accuracy of the solution, a simulation using 4 material points per cell with GIMP is also performed. The granular collapse experiment is performed for a column with an initial aspect ratio of 0.4 using both GIMP method and MPM with 16 material points per cell.

The evolution of run-out at time $t = \tau_c$ and $t = 6\tau_c$, where τ_c is the critical time when the potential energy is fully mobilised, is presented in figure 3.7. At the initial stage of collapse $t = \tau_c$, both MPM and GIMP give almost the same behaviour. The run-out observed in the case of 4 material points per cell is similar to the run-out behaviour for 16 material points per cell. At the end of the flow, the GIMP simulation with 4 material points shows oscillations at the flow front due to fewer material points in the cell. However, both MPM and GIMP show a smoother response at the flow front. The evolution of run-out and height with time for both MPM and GIMP are presented in figure 3.8. For a time up to $t = 1.5\tau_c$ all approaches yield the exact same behaviour. However, as the flow progresses, the number of material points per cell at the flow front decreases, resulting in oscillations. The oscillations decrease with increase in the number of material points. Hence, it is essential to have a larger number of material points, especially at the flow front. The difference in the normalised run-out between the MPM and GIMP methods is about 2.5%. This is due to the difference in the interpolation scheme adopted in both approaches. GIMP method offers a better approximation of the run-out behaviour as a result of the continuous basis function. This also results in a smoother stress and velocity distribution in GIMP than MPM. With increase in the number of material points and the use of finer mesh size decreases the difference between both approaches. The

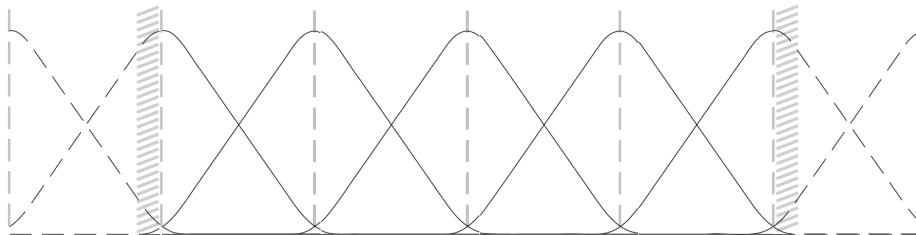


GIMP Weighting Function

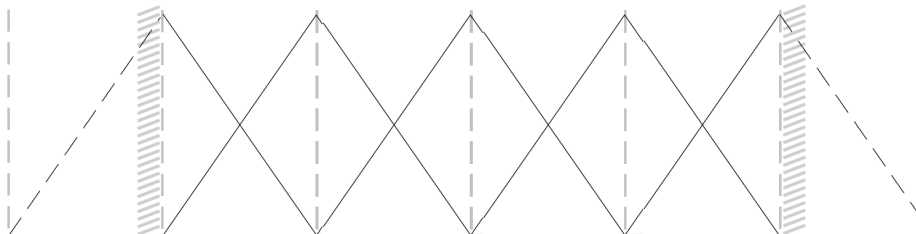


GIMP Gradient Weighting Function

(a) Example GIMP weighting function $\bar{\phi}_{ip}$, and gradient weighting function $\bar{\nabla} \psi_{ip}$ centered at 0 using piecewise linear grid basis functions and piecewise constant particle characteristic functions χ_p . Dotted lines denote breaks in the continuity of the functions.

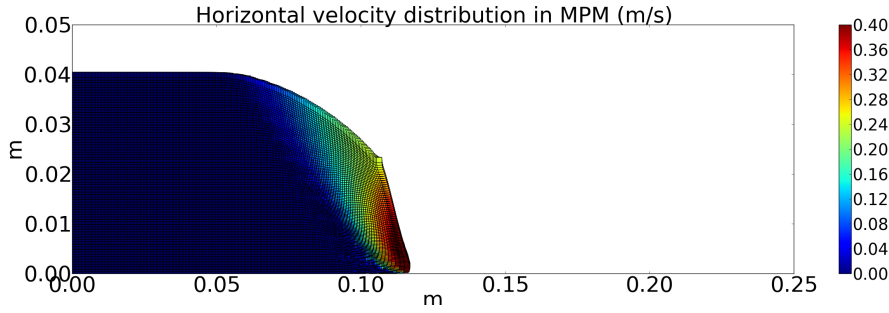


(b) GIMP

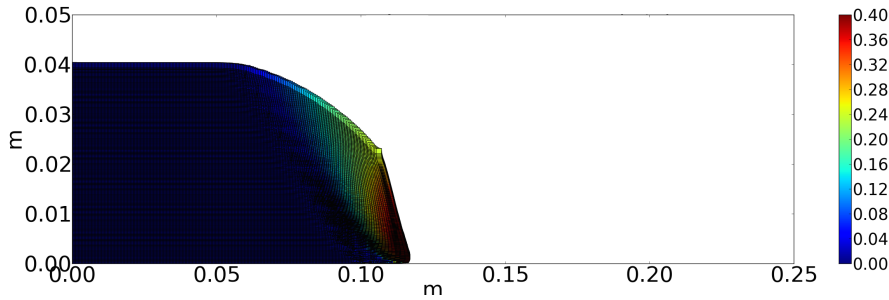


(c) Piecewise-Linear

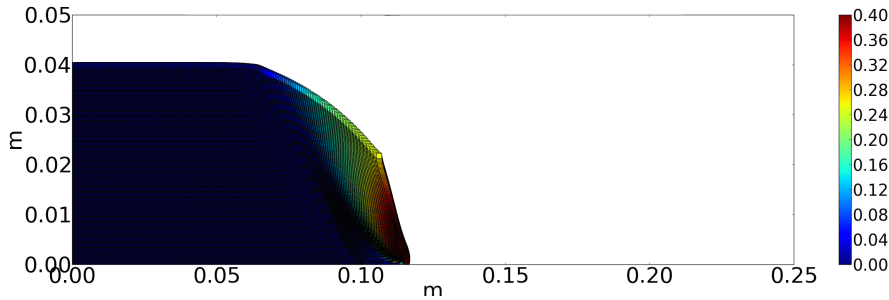
Figure 3.6 Schematic view of the 1-D basis functions used in MPM (Steffen et al., 2008).



(a) GIMP method (4 material points per cell)



(b) GIMP method (16 material points per cell)



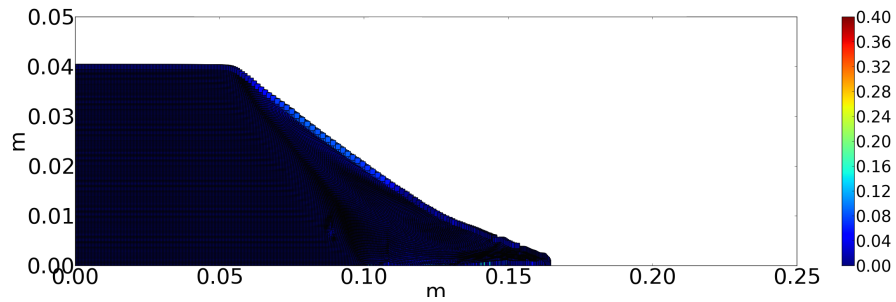
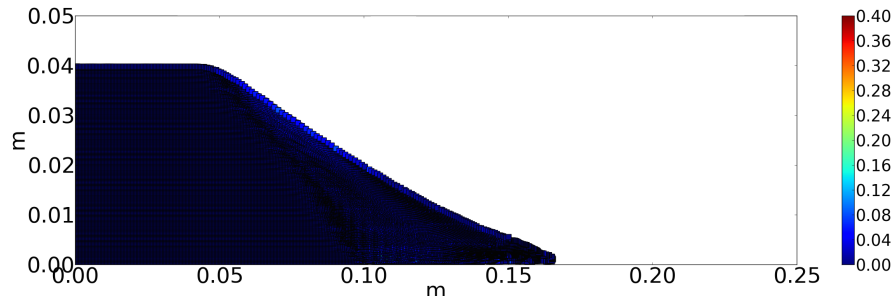
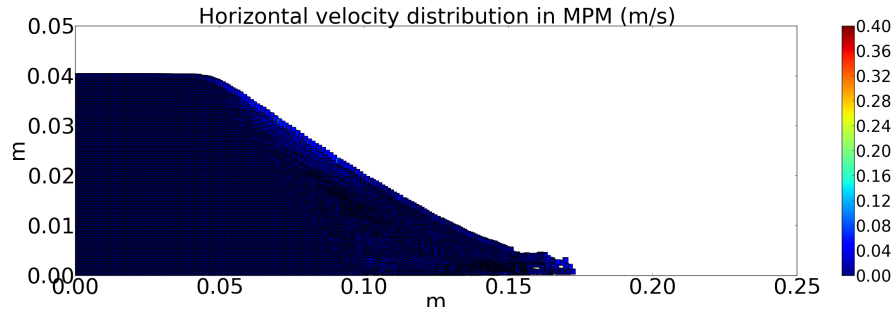
(c) Conventional MPM (16 material points per cell)

Evolution of granular column collapse at $t = \tau_c$.

computational effort using GIMP method is almost twice that of MPM, since GIMP method considers the particles at neighbouring cells due to the larger spread of the basis functions. In the present study, a very fine mesh and 16 material points per cell are used to simulate large deformation problems.

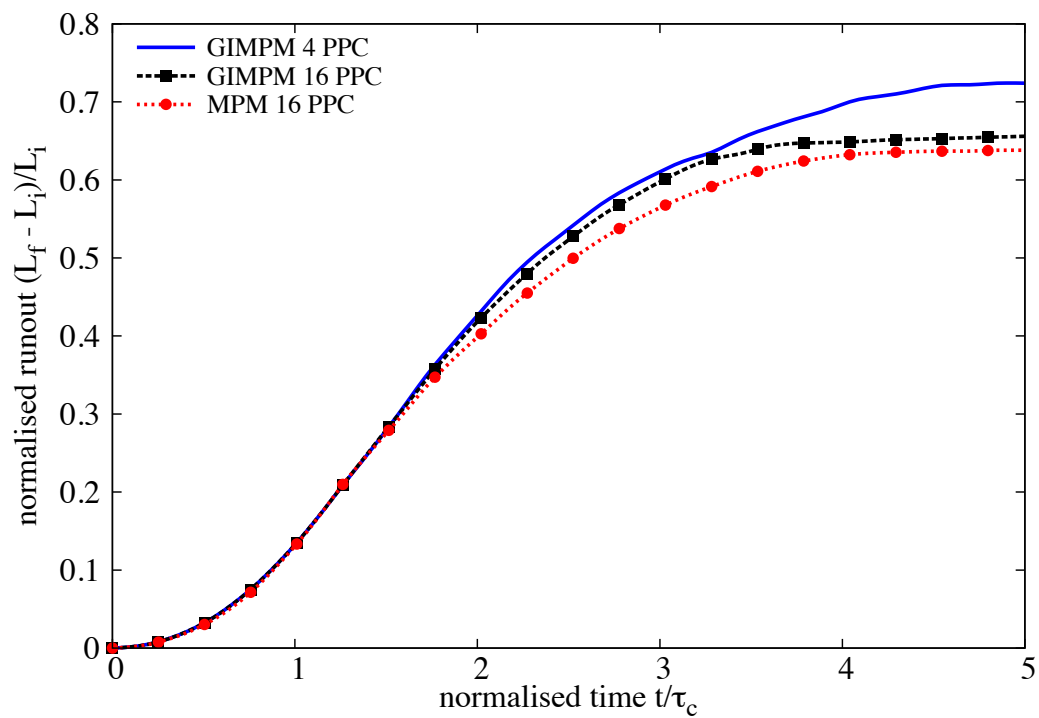
3.3.6 Application of MPM in geomechanics

The studies on using MPM in modelling geotechnical problems are limited. The potential of the Material Point Method in modelling granular flows, due to the discharge of silos, was first recognised by Wieckowski et al. (1999). Bardenhagen et al. (2001) developed a

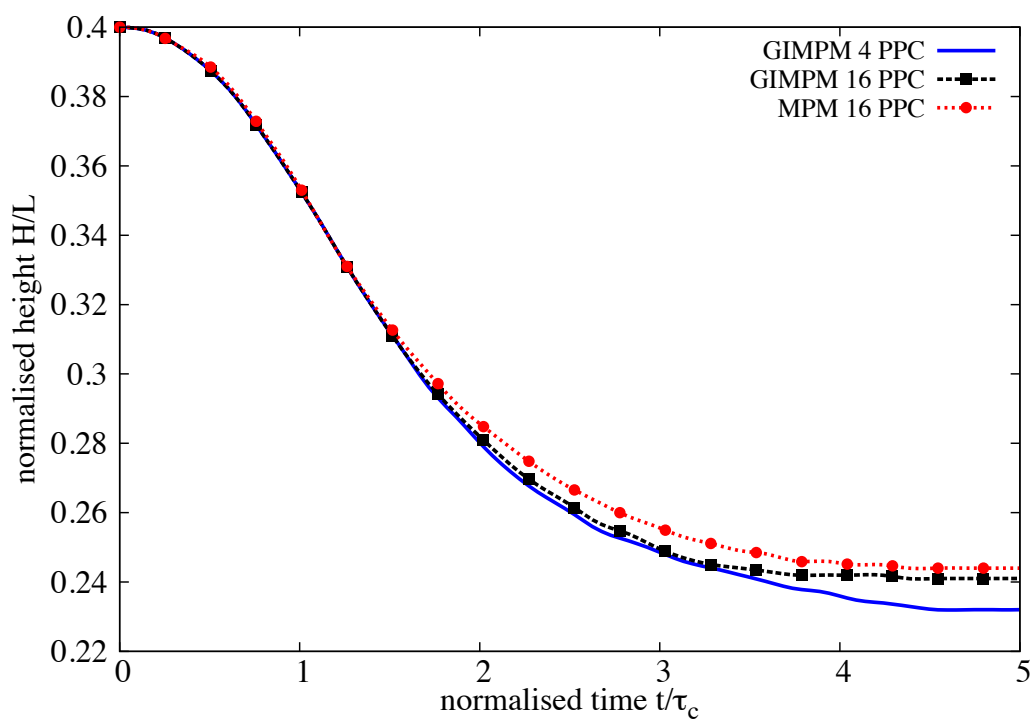


$$t = 6\tau_c$$

Figure 3.7 Comparison between the GIMP method and the conventional MPM on the flow morphology of a granular column collapse ($a = 0.4$).



(a) Evolution of run-out (GIMP method vs. MPM)



(b) Evolution of height (GIMP method vs. MPM)

Figure 3.8 Comparison between the GIMP method and the conventional MPM on the evolution of run-out and height with time for a granular column collapse ($a = 0.4$).

frictional contact algorithm to model granular materials. The Mohr-Coloumb model criterion is used to describe the kinematics of the grains. In this model, a contact is defined when the nodal velocity interpolated from all material points in the cell differs from the nodal velocity interpolated from a single material point. Coetzee et al. (2005) applied this contact algorithm to understand the pull-out behaviour of anchors. Ma et al. (2014) developed a new contact algorithm with a penalty contact function and a limited maximum shear stress for soil-structure interaction, i.e. interaction of pipe-line with debris flow or submarine landslide. The penalty contact function behaves similarly to numerical damping and thus reduces the oscillations during the impact.

Beuth et al. (2010) applied Gaussian integration in quasi-static MPM to model large strain problems. In this approach, however, the conservation of mass is not valid as a larger filled volume of material is considered. Andersen and Andersen (2010) used the GIMP method with an elasto-plastic model to simulate slope failures. The solution is found to be dependant on the number of material points used to describe the slope geometry. Mast et al. (2014a) studied the behaviour of granular column collapse using a non-associative flow rule. They observed a significant increase in the run-out distance for large aspect ratio columns. Mast et al. (2014b) investigated the suitability of the MPM in prediction of large deformation problems, such as snow avalanches.

Guilkey et al. (2007) developed a coupled numerical scheme for fluid-structure interactions. The solid field is modelled in a Lagrangian frame, while an Eulerian frame, compressible CFD, is used for the fluid. As the MPM computation grid is reset at every stage, the same Eulerian grid is adopted for both the solid and the fluid. A similar approach was adopted by Zhang et al. (2008) to model multiphase flows, where the interactions between the solid and gas are modelled using the MPM background mesh as an Eulerian grid. Mackenzie-Helnwein et al. (2010) investigated various techniques to model multiphase drag forces. Both solid and fluid are modelled as Lagrangian particles using the mixture theory. Abe et al. (2013) developed a soil-pore fluid coupled MPM algorithm based on Biot's mixture theory for solving hydro-mechanical interaction problems. Bandara (2013) developed a unique approach in modelling solid-fluid interactions. Two sets of Lagrangian particles are used to represent soil skeleton and pore water, separately. Bandara (2013) applied the coupled MPM to solve large deformation problems such as slope failure due to seepage.

In the present study, the Material Point Method is used to model large deformation problems, such as collapse of dry columns and soil slopes subjected to horizontal excitation. The suitability of the continuum approach (MPM) in modelling large deformation problems is also investigated.

3.4 Particulate modelling of granular flows

Granular materials often exhibit different behaviours under different circumstances. Fluidized granular material often resembles a liquid and reveals surface waves. In certain situations, granular materials behave more like solids exhibiting plastic deformations. Despite the wide variation in the physical and the chemical properties of the grains, the discrete granular structure has a rich generic phenomenology, motivating researchers to understand the fundamental behaviour of these materials.

A granular material can be considered as a continuous material if it is viewed at a macroscopic scale, ignoring the fact that it is composed of grains. On a macroscopic scale, the behaviour of the granular material can be approximately defined using continuum mechanics. However, on a grain level, granular materials exhibit complex solid-like and/or fluid-like behaviours depending on the way the grains interact with each other. Analytical and finite element models, which consider granular materials as a continuum, cannot take into account the local geometrical processes that govern the mechanical behaviour of a non-homogeneous soil. The application of continuum models to describe granular flow poses subtle problems for statistical analysis (Mehta and Barker, 1994). The grain-level description of the granular material enriches the macro-scale variables, which poorly account for the local rheology of the materials.

Numerical models, such as the Discrete Element approach proposed by Cundall and Strack (1979), are capable of simulating the granular material as a discontinuous system. Although modern measurement techniques can probe local granular variables, like grain position, velocities, contact forces, they have inherent limitations in acquiring those variables. The *discrete-element* approach is a powerful and reliable research tool to study the behaviour of granular materials at the grain-scale. This approach involves applying Newton's equation of motion simultaneously to all grains described as rigid solid bodies by considering the contact forces and the external forces acting on the grains. For a given boundary condition, the collective mechanical response of grains to the external force leads to relative motion between grains constrained in a dense state, and by inelastic collisions in the loose state. Cundall and Strack (1979) applied this method to granular geomaterials, and terming it the *Distinct Element Method*, to differentiate it from the existing *Finite Element Method* used in geomechanics. The attribute "distinct" refers to the degrees of freedom of individual grains, but it was later replaced by "discrete" to underline the discrete nature of the system.

The interactions between the individual grains are governed by unilateral contact laws, and the mechanism of energy dissipation is through friction and inelastic collisions. Moreover, granular materials have a wide variation in their grain shape and size distribution that require appropriate numerical treatments. In DEM, the normal reaction force, which prevents the

interpenetration of two grains, is proportional to the depth of penetration. Thus, frictional contact between grains can be expressed as a function of the configuration variables, which describe the positions and velocities of the grains (Radjai and Dubois, 2011).

Discrete-Element methods, which describe interactions between grains based on the explicit overlap between the grains, are termed as *smooth methods*. Another approach is the *non-smooth approach* (Jean, 1999), which describes the behaviour of discrete elements using the main features of uni-laterality and Coulomb friction, and neglect the finer details such as interpenetration and overlap between grains. The fundamental difference between the non-smooth method and the common discrete element method lies in the treatment of small length and time scales involved in the dynamics of granular media. In DEM, the grains are treated as rigid bodies, but the contacts between grains are assumed to obey the visco-elastic constitutive law. The time-stepping schemes used for the numerical integration of the equations of motion in DEM imply that the contact interactions involve smaller time and length scales. In the non-smooth Contact Dynamics (CD) method, these small scales are neglected and their effects are absorbed into the contact laws. In non-smooth formulation, the grain dynamics are described at a larger scale than the elastic response time and displacement scales (Jean, 1999; Radjai and Richefeu, 2009).

DEM simulations can more easily capture the complex flow mechanics of large-deformation problems than the continuum approach. Tang et al. (2009) used 2D discrete element modelling to understand the mechanism of the Tsaoling landslide triggered by the Chi-Chi earthquake. The researchers were able to establish the landslide has a low-friction coefficient (about 0.15) and a medium strength. They were also able to back-calculate a maximum velocity of sliding reached 50 m/s. Similarly, Tang et al. (2013) performed 3D discrete-element simulations to understand the transportation and deposition of the 2009 Hsiaolin landslide. The authors estimated the friction coefficient of landslide-mass to have reached a critical value of 0.1, at which the mass begins to slide and reach a maximum velocity of 40 - 50 m/s. Two-dimensional DEM simulations have been used to analyse the temporal and spatial evolution of slope failure and landslides from the intact, pre-failure slope to the restabilised, post-failure slope composed of bonded material (Katz et al., 2014). The pre-failure slope material disintegration is found to be the fundamental element in determining the size and geometry of the resultant landslides. Liu and Koyi (2013) studied the kinematics and internal deformation of granular slopes which experience flow-like behaviour. They observed that dilatant grain-shearing flow is the dominating mechanism in the movement of granular slopes. DEM is capable of probing the material response in a detailed scale, where conventional experiments or field tests are not feasible. Hence, in the present study, 2D discrete element simulations are performed to understand the behaviour of granular flows. Rickenmann et al.

(2006) applied 2D debris-flow simulation models to two well-documented field events. Two-dimensional simulations provided reasonable results for the extent of the deposits and the relative spatial distribution of deposit thicknesses. Nevertheless, some general characteristics of debris flow deposits, which are necessary for hazard assessment, may be reasonably well-simulated with these simple modelling approaches if rheologic or friction parameters can be measured or calibrated.

3.5 Discrete Element Method

The Discrete Element Method (DEM) computes the equilibrium and the trajectories of a classical multi-body system. DEM is a simple and flexible discrete-element approach, which involves applying Newton's second law of motion to each grain to describe the deformation of the granular assembly

$$m_i \frac{d^2 x_i}{dt^2} = \mathbf{F}_i, (i = 1, \dots, N), \quad (3.28)$$

where N is the number of grains in the simulation, m_i is the mass of a grain i , x_i is its position, and \mathbf{F}_i is the force exerted on a grain. The method consists of calculating the forces \mathbf{F}_i and then solving the ordinary differential in eq. 3.28. In general, the system of coupled non-linear differential equations cannot be solved analytically. The approximate numerical solution of these equations, which describes the trajectories of all the grains of the system is called the Discrete Element Method.

DEM simulations are similar to the real experiments, involving the generation of samples (initial conditions) with N grains and solving the Newton's equation of motion for the system until the properties of the system no longer change with time (equilibration of the system). The computation of the forces and torques is the central part of the Discrete Element Method simulation. The dynamics of the granular material is governed by Newton's equation of motion which depends on the centre-of-mass coordinates and the Euler angles of the grains i ($i = 1, 2 \dots, N$):

$$\frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{1}{m_i} \vec{\mathbf{F}}_i(\vec{r}_j, \vec{v}_j, \vec{\phi}_j, \vec{\omega}_j), \quad (3.29)$$

$$\frac{\partial^2 \vec{\phi}_i}{\partial t^2} = \frac{1}{\hat{J}_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\phi}_j, \vec{\omega}_j), (j = 1, \dots, N). \quad (3.30)$$

The force $\vec{\mathbf{F}}_i$ and the torque $\vec{\mathbf{M}}_i$, which act on a grain i of mass m_i and the tensorial moment of inertia \hat{J}_i are functions of the grain positions \vec{r}_j , their angular orientations $\vec{\phi}_j$, and their corresponding velocities \vec{v}_j and $\vec{\omega}_j$. In a two-dimensional system, the angular orientation of

a grain is described by a single (scalar) quantity ϕ_i and the moment of inertia reduces to a scalar value J_i .

For grains in the absence of long range fields, the force $\vec{\mathbf{F}}_i$ and the torque $\vec{\mathbf{M}}_i$ acting upon the grain i are given as a sum of the pairwise interaction of a grain i with all other grains in the system:

$$\vec{\mathbf{F}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{F}}_{ij}, \quad \vec{\mathbf{M}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{M}}_{ij}. \quad (3.31)$$

The limitation to pairwise interaction is an abstraction, which is justified if the grain deformation at the contact is trivial. To describe the deformation of granular assemblies one has to take into account the effect of multi-grain interactions. This generalised method can be applied to a wide range of systems. The Discrete Element Method can be used to study the behaviour of grains in rapid flows and in static assemblies. The method treats both conditions in exactly the same way; it is not necessary to divide the system and then treat each condition differently. The simplest model for a grain is a sphere. In a two-dimensional case, the sphere is reduced to a circular disk. Simulations using spherical grains are numerically very effective, since grain collisions can be easily identified and described in a simplistic way (Poschel and Schwager, 2005).

In the present study, a two-dimensional DEM code is developed in C++ to study the micro-scale rheology of dry granular flows. Existing DEM codes do not yet fully support micro-scale fluid solid coupling using the Lattice Boltzmann approach, which will be used to study the mechanism of submarine landslides. A new C++ based high performance DEM code is developed in the present study. The features implemented in the present study, and the general overview of DEM are discussed below.

3.5.1 The Forces

The force \mathbf{F}_i in eq. 3.28 represents both the grain to grain interaction force and other external forces acting on the system. Therefore, the force \mathbf{F}_i is expressed as

$$\mathbf{F}_i = \sum_{j \neq i} \mathbf{F}_{ij} + \mathbf{F}_{ext,i}, \quad (3.32)$$

where \mathbf{F}_i is the force exerted by a grain j on i . The external force $\mathbf{F}_{ext,i}$ is most often the force of gravity, $\mathbf{F}_{ext,i} = m_i \mathbf{g}_i$. The methodology to incorporate any other external forces in the simulation is the same. However, the computation of the interaction forces depends on

the numerical method adopted in the study. The methodology used in the present study is described below.

Consider two grains i and j in contact (figure 3.9). The contact force can be decomposed into two components, as the normal (F_n) and the tangential (F_t) components

$$\mathbf{F}_{ij} = F_n \mathbf{n} + F_t \mathbf{t}. \quad (3.33)$$

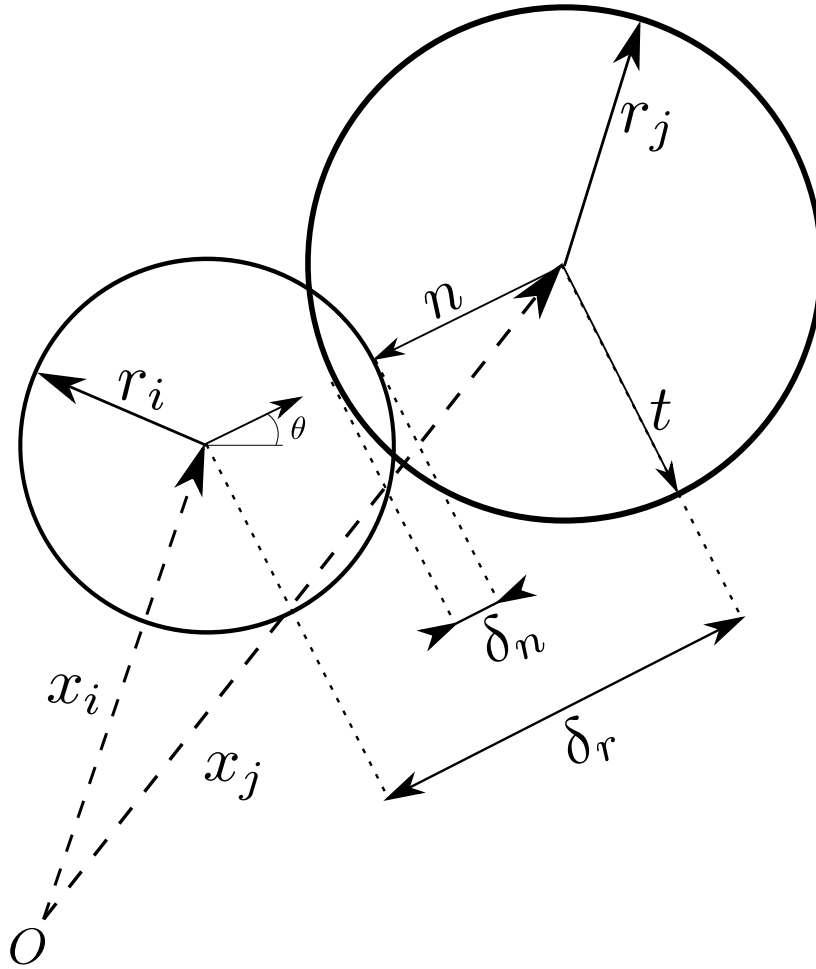


Figure 3.9 Grains i and j in contact, and the separation δ_n is used to calculate the normal force

where \mathbf{n} and \mathbf{t} are unit vectors, pointing in the normal and the tangential directions. The procedure adopted to calculate the normal and tangential forces is discussed.

Normal force

When grains collide, part of the kinetic energy is dissipated as heat and the other part causes deformation of the grain. These deformations generate interaction forces. In DEM, the grains

are considered to be rigid while their contact is assumed to be soft. Thus, the grains do not change their shape; instead they overlap. The shapes of the grains are conserved on average, after many collisions. The overlap at the contact is limited to very small deformations, which are achieved by defining a repulsive normal force that opposes the overlap. The mutual compression (δ_n) of the grains i and j is defined as

$$\delta_n = |x_i - x_j| - r_i - r_j, \quad (3.34)$$

where x_i and x_j are vectors fixing the centres of the grains, and r_i and r_j are their radii (figure 3.9). When $\delta_n > 0$, the two grains are not in contact, and there is no interaction. When $\delta_n < 0$, the two grains overlap, and there is a repulsive normal force that pushes the two grains apart. As a simple model, consider the contact as a linear spring with damping. The repulsive force depends linearly on δ_n and is controlled by the stiffness of the grain. The energy dissipation due to the interaction between grains is an intrinsic characteristic of the granular material and is incorporated by adding a damping force that opposes the relative velocity for the duration of the contact. The interaction force at the contact is idealized as a simple spring-dashpot system, with elastic and dissipative constants (Luding et al., 1994).

$$F_n = \begin{cases} 0, & \delta_n > 0 \\ -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, & \delta_n < 0 \end{cases} \quad (3.35)$$

The constant k_n characterises the stiffness of the grain and must be chosen to be sufficiently large so that the overlap between the grains remains small. Nevertheless, the solution has an undesirable property of generating an attractive force (Poschel and Schwager, 2005). It arises just before the two grains separate. In this case, we have $d\delta_n/dt > 0$ while δ_n approaches zero. To avoid the attractive force, the force is computed in two stages: a candidate force \hat{F}_n is calculated and verified as to whether it is non-negative

$$\hat{F}_n = -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, \quad F_n = \begin{cases} 0, & \hat{F}_n \leq 0 \\ \hat{F}_n, & \hat{F}_n > 0 \end{cases} \quad (3.36)$$

For pairwise collisions, the normal force (F_n), represented as, $k_n \delta_n + \gamma_n$ causes a decrease in the relative normal velocity of the grains by a factor ε . This factor is the *coefficient of restitution* and is defined as $\varepsilon \approx u'/u$, where u is the absolute normal relative velocity before the collision and u' corresponds to the post-collision value. The relative velocity, $d\delta_n/dt > 0$,

can be obtained by differentiating eq. 3.34. Thus, we obtain

$$\frac{d\delta_n}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n}, \quad (3.37)$$

where $\mathbf{v}_i = dx_i/dt$ is the velocity of the grain i and $\mathbf{v}_j = dx_j/dt$ is the velocity of the grain j . The numerical integration of eq. 3.37 yields the separation δ_n and permits us to generalise the model so as to incorporate the tangential forces. By integrating Newton's equation of motion, it is found that the linear force corresponds to the co-efficient of restitution, which is defined as

$$\varepsilon = \exp\left(-\frac{\pi\gamma_n}{2m^{eff}} / \sqrt{\frac{Y}{m^{eff}} - \frac{\gamma_n^2}{2m^{eff}}}\right). \quad (3.38)$$

Tangential force

Grains are not perfect spheres, they have a complicated surface texture. Therefore, at oblique collisions, there is a tangential force in addition to a normal force. Even perfectly smooth spheres exert a tangential force due to their bulk viscosity (Poschel and Schwager, 2005). To build a heap of spheres on a flat surface, the grains as well as the surface have to be sufficiently rough, indicating the dependency of the tangential force on the surface properties of the granular materials. For realistic simulations of granular materials, it is important to consider the tangential force in DEM. The tangential force is considered in a similar fashion to the normal force, arising from a spring stretched by the relative motion of the grain. Tangential forces are modelled by considering the relevant relative tangential velocity of the grain surfaces at the point of contact. The point of contact is an approximation, as the description of the normal force assumes a compression δ_n , which implies a contact surface in 3-D or a contact line in 2-D. Assuming a tangential spring of length δ_t that exerts an opposing force to the relative tangential displacements (ignoring the effect of relative rolling between the grains), the tangential force can be postulated similar to the normal force (eq. 3.37) as

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \times \mathbf{t}. \quad (3.39)$$

As with eq. 3.28, this equation must also be numerically integrated. The grains are in contact when $\delta_t < 0$, whereas when $\delta_t = 0$, the grains no longer exert a force on each other. With these assumptions, δ_t can be calculated similar to the normal force. The tangential force is assumed to be governed by Coulomb's friction law:

$$|F_t| \leq \mu F_n, \quad (3.40)$$

where F_t is the tangential force and μ is the friction coefficient. It is therefore necessary to constrain the tangential force to remain less than or equal to μF_n . To impose the condition in eq. 3.40, two stages similar to the normal force computation are adopted. The first step is to evaluate the candidate force, and the force is then accepted if it obeys the condition in eq. 3.40.

$$\hat{F}_t = -k_t \delta_t - \gamma_t \frac{d\delta_t}{dt}, \quad F_t = \begin{cases} \text{sgn}(\hat{F}_t), & |\hat{F}_t| \geq \mu F_n \\ \hat{F}_t, & |\hat{F}_t| < \mu F_n \end{cases} \quad (3.41)$$

where k_t is the stiffness of the tangential spring and γ_t is the damping constant. If $|F_t| = \mu F_n$, the contact is sliding; otherwise, it is non-sliding. It can be noted that the normal force (eq. 3.36) and the tangential force (eq. 3.41) are handled in the same way in DEM. When the grains slide against each other, they do not retain any memory of their initial position, and hence do not return to their original position. In order to model this behaviour, a limiting value of δ_t is imposed. When the contact slides, $\delta_t = \pm \mu F_n / k_t$ is imposed.

In addition to sliding, the grains can roll relative to one another around their centre of mass due to the tangential force acting at their contact surfaces. In which case, $d\delta_t/dt = 0$. It is important to assume that the grains touch at a single point instead of overlapping, i.e. $\delta_n = 0$. This point is located at $x_i - r_i \mathbf{n} = x_j + r_j \mathbf{n}$. If we consider that this point belongs to grain i , its velocity is $\mathbf{v}_i + r_i (\boldsymbol{\omega} \times \mathbf{n})$. If it belongs to grain j , its velocity is $\mathbf{v}_j + r_j (\boldsymbol{\omega} \times \mathbf{n})$. The relative velocity is the difference between these two velocities.

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{t} - (r_i \boldsymbol{\omega}_i + r_j \boldsymbol{\omega}_j) \times \mathbf{n}. \quad (3.42)$$

It should be noted that eq. 3.42 is only an approximation, as the grains in DEM do not touch at points, but overlap. It is therefore an approximation that produces an error of order $O(\delta_n/r)$ (Radjai and Dubois, 2011). It is assumed that the contact forces are exerted at the point of contact, which implies that the tangential force is accompanied by torque acting on two grains. If the overlap is zero, these torques are

$$\boldsymbol{\tau}_{ij} = -(a_i \mathbf{n}) \times (F_t \mathbf{t}), \quad (3.43)$$

$$\boldsymbol{\tau}_{ji} = -(a_j \mathbf{n}) \times (F_t \mathbf{t}). \quad (3.44)$$

The torques modify the angular velocities of the grains. It is therefore necessary to incorporate the equation for the angular coordinates of the grains in eq. 3.28

$$I_j \frac{d\omega_i}{dt} = \sum_{j \neq i} \tau_{ij}, \quad (3.45)$$

where I_j is the moment of inertia of grain j . Eq 3.44 is only valid when $\delta_n = 0$. The torque is a vector product of the force and its lever arm. It is assumed that the lever arms have lengths equal to r_i and r_j , which is true only when the grains do not overlap. Hence in this case they produce an error of order $O(\delta_n/r)$. It is nevertheless desirable to damp this type of motion (Radjai and Dubois, 2011). A rolling resistance can be adopted in order to model the shape effect of non-spherical grains, which accounts for moments arising from the fact that the line of action of the normal contact force in the case of non-spherical particles no longer passes through the centre of mass of the particles and hence generates rotational moments.

The interaction between two solid bodies is much more complex than that described by the simple linear model. Nevertheless, the linear force law has several advantages. It is simple to implement, and its harmonic behaviour is well-understood, which makes it easier to interpret the results. The most common non-linear interaction law is the Hertz law (Hertz, 1882). In certain situations, such as a quasi-static packing, a non-linear law can have significant influence on the acoustic properties and on the global stiffness (Agnolin and Roux, 2007). However, in the case of rapid granular flows, the interaction force between the grains has almost no effect on the phenomenon, and a linear law can be used to describe this kind of behaviour (Radjai and Dubois, 2011).

3.5.2 Numerical algorithm and integration scheme

The efficiency of a DEM algorithm is mainly determined by its efficiency for computing the interaction forces between grains. If we consider a model system with pairwise interactions, then we have to consider the contribution of the force on grain i due to all of its neighbours. If we consider only the interaction between a grain and the nearest image of another grain, then for a system of N grains, we must evaluate $N \times (N - 1)/2$ pair distances. Consider a system of 1000 grains. At every time step, all possible pairs of grains have to be considered to compute the interaction forces, and hence, $N(N - 1)/2 \approx 500,000$ force computations are required. For short-range grain interactions, the majority of these force evaluations is unnecessary as the corresponding grains are located far apart and do not necessarily touch each other. For a dense system of equally sized grains, the grains can have contacts with not more than 6 grains, thereby reducing the number of force computations required to $3N \approx 3000$. In

the preliminary force computation scheme, at least 166 times more pair interactions are considered than necessary. Therefore, the numerical methods employed in the DEM should try to minimize the computation of interaction forces (Poschel and Schwager, 2005). There are three different methods for the efficient computation of the forces the *Verlet* algorithm, the *link-cell* algorithm, and a *lattice* algorithm. The *Verlet* algorithm described in Grubmuller et al. (1991) is implemented in the present study.

Verlet list algorithm

The Verlet list algorithm assumes a cut-off value, so that only neighbouring grains that contribute to the energy of grain i are considered. It is advantageous to exclude the grains that do not interact in the memory-expensive energy computation. Verlet (1967) developed a book-keeping technique, commonly referred to as the Verlet list or neighbour list, which is illustrated in figure 3.10. In this method, a second cut-off radius $r_v > r_c$ is introduced, and before the interactions are calculated, a list is made (the Verlet list) of all grains within a radius r_v of the grain i . In the subsequent calculations of the interactions, only those grains in this list will be considered. The idea of the Verlet algorithm is based on a simple property of grain dynamics: the neighbourhood relation between grains can only change slowly, i.e. two grains which are close to each other at a given time step will remain as neighbours, at least in the following few time steps. During initialization, the neighbourhood relations between the grains, i.e. the distance of all close pairs of grains, are computed. Two grains are considered to be neighbours if the distance of their surfaces is smaller than a predefined distance, termed the *Verlet distance*

$$(|\vec{r}_i - \vec{r}_j| - R_i - R_j) < \text{Verlet distance}. \quad (3.46)$$

For each grain, there is a *Verlet list* in which the close neighbours are saved. To initialise the Verlet lists efficiently, a grid that covers the simulation area is defined. Its mesh size is larger than the largest grain. For construction of the lists, only pairs whose grains reside in the same or adjacent grids are considered. This procedure guarantees the detection of all close pairs of grains (Poschel and Schwager, 2005). Redundancy in Verlet lists, i.e. if grain i is a neighbour of j , then grain j is a neighbour of i , is avoided by imposing a restriction on the list of grain i , such that it contains only neighbours with index $j < i$. For the computation of interaction forces, the Verlet list of grain i is scanned, and only pairs which are recorded in one of the Verlet lists are considered. Hence, the Verlet list of each grain i is scanned, and the interaction force of i with each entry j in its list is computed.

Initially, to build the Verlet list, the grains are sorted into a grid of mesh size $dx \times dy$. For each grid, there is a list of grains residing in the cell. During the simulation, the

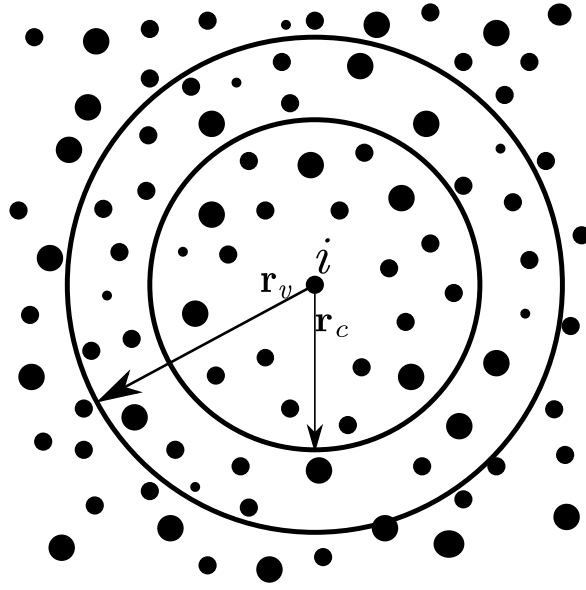


Figure 3.10 The Verlet list: a grain i interacts with those grains with the cut-off radius r_c . The Verlet list contains all the grains within a sphere with radius $r_v > r_c$

neighbourhood relation among the grains change; therefore, the Verlet lists have to be updated. The decision to update a Verlet list depends on how far the grains have travelled since the time when the present list was built. The Verlet list of a grain i must contain at any time all neighbours j with $j < i$. This assures that two grains i and j never touch and are not considered as neighbours, i.e. j is not in the list of i , and i is not in the list of j . Hence,

$$|\vec{r}_i - \vec{r}_j| - R_i - R_j > 0. \quad (3.47)$$

The above condition is required for all pairs (i, j) of grains which are *not* known as neighbours. This condition is a criterion to update the Verlet lists (Poschel and Schwager, 2005). Assume at the instant when the Verlet lists are constructed that the surfaces of the grains have the distance $|\vec{r}_i - \vec{r}_j| - R_i - R_j > \text{Verlet distance}$, i.e. they are not classified as neighbours. If the Verlet lists are updated before one of these grains has travelled the distance $\text{verletdistance}/2$ since the lists were constructed then they can never collide without being recognized as neighbours first. This is explained in figure 3.11. The impact of optimisation of the Verlet list algorithm has negligible effect on the computation time, as the algorithm is quite efficient already and only consumes a small percentage of the total computation time in construction of the Verlet lists. The implementation of the Verlet list algorithm in force computation drastically reduces the computation time in comparison to the linear algorithm. The performance of the Verlet list algorithm is controlled by two crucial parameters: the

number of cells N_c , for the construction of the Verlet lists and the Verlet distance r_v (Poschel and Schwager, 2005).

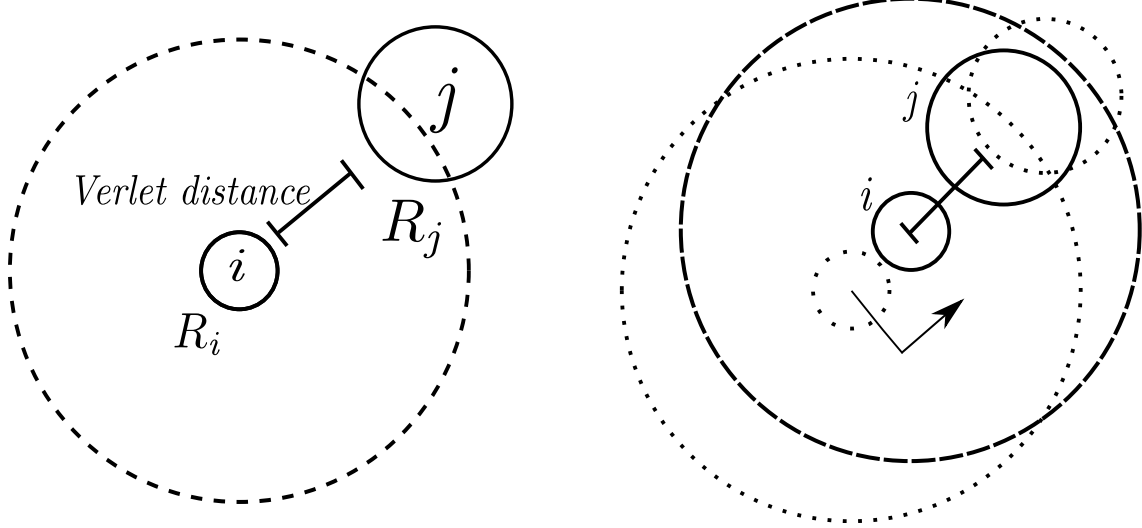


Figure 3.11 Checking the validity of Verlet lists. *Left:* the grains i and j are not recognized as neighbours since the distance of their surfaces is larger than the Verlet distance. The radius of the dashed circle is $R_i + R_{max} + \text{Verlet distance}$. *Right:* in the most critical case the grains approach each other directly, travelling at the same velocity. As soon as one of the grains has travelled the distance $\text{Verlet distance}/2$ (arrows), the Verlet lists have to be rebuilt. The grains i and j are now recognized as neighbours. Redrawn from Poschel and Schwager (2005).

Leap frog, or Verlet integration algorithm

The Discrete Element Method involves numerically solving Newton's equation of motion eq. 3.28, which is an ordinary differential equation. Choosing an integration algorithm is important, as the forces are not always differentiable in time, and the temporary derivative of the force is discontinuous when the contact splits. It is also essential to numerically integrate eq. 3.42 with the same precision as eq. 3.28. At first, computational speed seems important. However, it is usually not very relevant because the fraction of time spent on integrating the equation of motions (as opposed to computing the interactions) is negligible. Accuracy for large time steps is more important because the larger the time step used, the fewer evaluations of the forces needed per unit of simulation time. Hence, it is advantageous to use a sophisticated algorithm that allows use of larger time step.

Algorithms that allow the use of large time steps achieve efficiency by storing information on increasingly higher-order derivatives of the grain coordinates. Consequently, they tend to require more memory storage. However, the most important aspect to consider is the energy conservation. It is important to distinguish between two kinds of energy conservation: the

short-term and the long-term. The sophisticated higher-order algorithms tend to have very good energy conservation for short times. However, they often have undesirable features that result in drifting of the overall energy for longer times. In contrast, the Verlet-style algorithms tend to have only moderate short-term energy conservation, but little long-term drift (Frenkel and Smit, 1996). In this case, such algorithms are not useful. They are more complicated to program, and they do not yield a more precise solution (Radjai and Dubois, 2011). It might be important to have an algorithm that accurately predicts the trajectories of all grains for both short and long durations, however no such algorithm exists.

In certain cases, two trajectories that are initially very close may diverge exponentially as time progresses. Any integration error, however small it may be, will always diverge the predicted trajectory exponentially from the true trajectory. This phenomenon is called the Lyapunov instability, and it poses a serious threat to the fundamental basis of DEM simulations. However, this problem might not be serious (Frenkel and Smit, 1996). The aim of DEM is not to predict precisely what will happen to a system, but to predict the average behaviour of the system that is prepared in an initial state about which we know something (initial position, velocity and energy), but not everything. Hence, DEM differs from other methods, which are used to predict the trajectories. However, considerable numerical evidence suggests that the shadow orbits exist, which is a true trajectory of a multi-body system that closely follows the numerical trajectory for a time that is longer in comparison with the time that is required for the Lyapunov instability to develop (Frenkel and Smit, 1996).

Because Newton's equations of motions are time-reversible, so too should the integration algorithms be time-reversible. The "leapfrog" algorithm, or the Verlet integration algorithm, is a numerical scheme used to integrate the Newton's equation of motion to calculate the trajectories of grains and is implemented in DEM by Verlet (1967). The Verlet algorithm is fast and requires less storage memory. It is not particularly accurate for long time steps, and hence, we should expect to compute the forces on all grains rather frequently. Its short-term energy conservation is satisfied (in versions that use more accurate expression for velocity), but most importantly, it exhibits little long-term energy drifts. This is related to the fact that the Verlet algorithm is time-reversible and area preserving. However, it does not conserve the total energy of the system exactly (Frenkel and Smit, 1996). The Verlet algorithm is simply based on a truncated Taylor expansion of grain coordinates.

$$t(t + \Delta t) = rt + \mathbf{v}(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots \quad (3.48)$$

If we truncate this expansion beyond the term Δt^2 , we obtain Euler's algorithm, which looks similar to the Verlet Algorithm, but it does not preserve energy and it has significant energy drifts. The simplest among the Verlet schemes is the *Leap frog algorithm*, which evaluates the velocities at half-integer time steps and uses these velocities to compute the new positions. The position of each grain is calculated at time $t = 0, \Delta t, 2\Delta t, \dots$, where Δt is the time step. On the other hand, their velocities are calculated at intermediate times, that is, at $t = \Delta t/2, 3\Delta t/2, \dots$. Let the position of a grain at time $t = k\Delta t$ be written as x_k , and its velocity at time $t = \Delta t(k + 1/2)$ be written $\mathbf{v}_{k+1/2}$, and its acceleration at $t = k\Delta t$ be \mathbf{a}_k . Then the following equation is used to advance systematically

$$\mathbf{v}_{k+1/2} = \mathbf{v}_{k-1/2} + \mathbf{a}_k \Delta t, \quad (3.49)$$

$$x_{k+1} = x_k + \mathbf{v}_{k+1/2} \Delta t \quad (3.50)$$

This algorithm determines the new grain position with an error of order $O(\Delta t^4)$. But eq. 3.49 hides a difficulty in the application of this algorithm to granular materials (Radjai and Dubois, 2011). This difficulty is the calculation of acceleration at time $t = k\Delta t$. The velocities are known at $t = (k - 1/2)\Delta t$, but not at $t = k\Delta t$. One option for resolving this problem is to write

$$\mathbf{v}_k = \mathbf{v}_{k-1/2} + \mathbf{a}_{k-1} \Delta t / 2. \quad (3.51)$$

The equation uses the acceleration of the preceding time step to estimate the velocity. This approximation does not diminish the order of the algorithm. Eq 3.51 estimates \mathbf{v}_k with an error of order $O(\Delta t^2)$, producing an error of the same order in the calculation of the force in eq. 3.49. But this causes only an error of order $O(\Delta t^3)$ in the velocity and an error of order $O(\Delta t^4)$ in the position. However, this problem does not exist in energy conservation systems, because the computed forces do not depend on the velocities of the grains. The heaviest computational task is the evaluation of forces, not the integration of equations. The Verlet integration scheme is summarized in eq. 3.52 and figure 3.12. To calculate the forces and acceleration, it requires the positions and velocities at time t :

$$\begin{aligned} \mathbf{v}(t + \Delta t/2) &= \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t) \Delta t, \\ x(t + \Delta t) &= x(t) + \mathbf{v}(t + \Delta t/2) \Delta t, \\ \mathbf{v}(t) &= \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t - \Delta t) \Delta t / 2. \end{aligned} \quad (3.52)$$

The analysis of the DEM formulation reveals that the linear force law gives the model a harmonic character, showing that it is very closely related to simple models widely applied in physics and mechanics. The shortest time scales often arise from the oscillations of one or

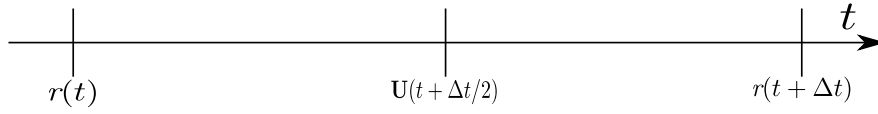


Figure 3.12 Verlet integration scheme

two grains. The integration algorithm must resolve these movements with sufficient precision. Thus, the time steps used must be smaller than these time scales; the most rapid frequency is usually ω_N , the characteristic oscillation frequency of very short waves. This frequency is proportional to ω_o , which is easier to estimate. Therefore, it is essential to choose a time step $\Delta t \approx \varepsilon/\omega_o$, where ε is a constant that depends on the integration algorithm. Values such as $\varepsilon \approx 0.01$ are often a reasonable choice (Radjai and Dubois, 2011). In the case of rapid granular flows, the time step must be small enough so that the fastest grains move only by a small fraction of their size during one time step. The grains must be stiff enough so that violent collisions do not lead to large overlaps between grains.

3.5.3 Boundary conditions

In many cases, the dynamic and static properties of a granular system are substantially affected by the interaction of the granular material with the system boundaries, i.e. by the properties of the container or the surface on which the material is present. The effect of boundary conditions on the response of the granular assembly can be noticed in the convective motion of granular material in vibrating containers, the formation of density waves in pipes, the motion of granular material on conveyors, and the clogging of hoppers. In these and many other cases, careful definition of the interaction between the granular material and the contact surface is essential. Of particular importance is the realistic modelling of the wall surface roughness. Unfortunately the mechanical interaction of granular materials with a rough wall is poorly understood (Poschel and Schwager, 2005). A simple way to define the wall property is to build up the wall from grains, which obey the same rules of interaction as the grains of granular material. By varying the size and position of the wall grains, system boundaries of adjustable roughness can be described. However, the surface roughness that characterizes the frictional properties of the wall has to be derived iteratively, and may not represent the real conditions. In the present case, a solid wall with corresponding stiffness, damping and frictional characteristics is introduced to model the interaction between grains and the wall. The interaction force is computed in a similar fashion to that of a pair of grains in contact and is divided into the normal and tangential components. The compression of the grain upon collision with the wall is calculated along the normal direction to the wall and the grain contact.

Periodic boundary

The effect of a wall on the response of grains is very critical, especially in numerical simulation where the number of grains is relatively fewer in comparison to the experiments. The undesired effect of a wall can be eliminated using periodic boundary conditions, i.e. a periodic extension of the simulation area in one or more dimensions. Any grain leaving the system at one side is reintroduced at the opposite side, and correspondingly the interaction forces between grains at opposite sides of the simulation area are taken into account. In this framework, the simulation domain becomes a unit area containing grains with periodic copies paving the whole system. The periodic boundary conditions extend the system boundaries to infinity, so that the simulation cell simply plays the role of a coordinate system to locate grain positions (see figure 3.13).

The external stresses or displacements are applied on the simulation box by constraining the degrees of freedom of the wall, which are alternatively kept free or fixed depending on whether a stress or a displacement is monitored in a system. With periodic boundary conditions, this role is played by the collective degrees of freedom carried by the coordinate system, whose basis vectors become dynamic variables, and their conjugate stresses are expressed as a state function of the granular configuration (Parrinello and Rahman, 1980). In the case of granular systems, there is dissipation of energy during grain interactions. The kinematics, equation of dynamics, and the time-stepping schemes for DEM are discussed in detail in Radjai et al. (2011). The periodicity in position implemented in the present study is discussed below.

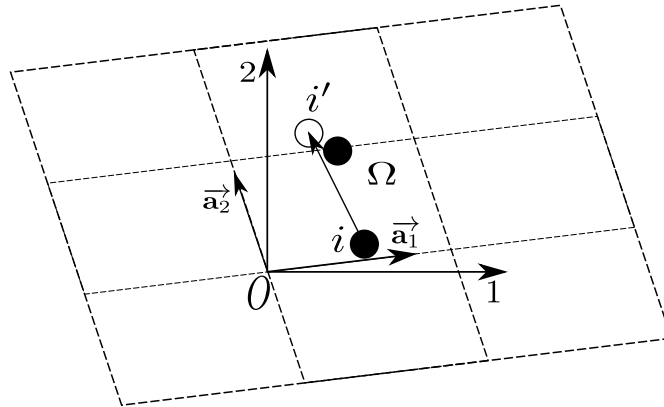


Figure 3.13 A 2D simulation cell ω with its basis vectors in an absolute frame. A grain located at the right boundary interacts with the image of another grain located at the left boundary.

Let us consider a collection of N_p grains with their centres contained in a cell of volume V . The cell can have any shape allowing for a periodic tessellation of space. The simplest shape is a parallelepiped i.e. parallelogram in 2D. The cell and its replicas define a regular

lattice characterized by its basis vectors (\vec{a}_1, \vec{a}_2) . In the case of a parallelogram, the basis vectors may simply be the two sides of the parallelogram; figure 3.13. The origin O of the simulation cell is a vertex of the cell of coordinates $(0, 0)$ and its replicas are defined by two indices (i_1, i_2) corresponding to a translation of the origin by the vector $i_1 \vec{a}_1 + i_2 \vec{a}_2$. Then, the coordinates $\vec{r}(i)$ of the image i' of a grain $i \in \Omega$ of coordinates $\vec{r}(i)$ are given by:

$$\vec{r}(i') = \vec{r}(i) + \sum_{k=1}^2 i_k \vec{a}_k. \quad (3.53)$$

The grains belonging to the cell Ω , characterized by $i_1 = i_2 = 0$, can interact with the grains of the same cell but also with image grains in the neighbouring cells characterized by $i_k \in 1, 1$. There are $3^D - 1$ cells surrounding the simulation cell and they are involved in the search of contact partners for each grain. The distance between two grains i and $j \in \Omega$ is the shortest distance separating i from j or from one of its images j' . As the system evolves in time, a grain i may leave but one of its images i' enters at the same moment. In order to keep all original grains in the cell, the status “original” should be reserved to the grains whose centres belongs to Ω . Hence, whenever a grain i leaves the simulation cell, it becomes an image of i' , which then becomes the original. This means that a grain crossing a border of the simulation cell, returns to the cell by crossing another border.

3.5.4 Validation of DEM

It is essential to validate the developed DEM code. In the present study, a preliminary validation of the DEM code is performed by studying a ball rolling down an inclined plane. O’Sullivan et al. (2003) showed that the validation of DEM code using the above problem confirms the appropriate implementation of shear contact model in DEM. The theoretical solution for a ball rolling down an inclined plane is derived by Ke and Bray (1995). A disk is resting on a rigid inclined plane oriented at an angle $\beta = 45^\circ$ to the horizontal plane (figure 3.14a). The disk has a radius r of 1m and a density of 2650 kg/m^3 . The friction angle ϕ between the disk and the rigid plane is varied. The disk rolls/slides down the plane due to gravity. The DEM solution for the sliding case ($\phi = 0^\circ$) is compared with the theoretical solution by Ke and Bray (1995). The normal force, shear force, angular rotation, accelerations \ddot{a} and angular acceleration $\ddot{\theta}$ are compared with the theoretical solution (table 3.1). The accumulation of disk rotation at time $t = 0.2 \text{ s}$ versus the friction angle is presented in figure 3.14b. The angular rotation observed in the DEM matches the theoretical solution, which validates the shear contact model used in DEM. The DEM model implemented in

the present study is further validated by comparing it with the granular column collapse simulations performed by Zenit (2005) and using PFC2D (Itasca PFC2D, 1998).

Table 3.1 Comparison of theoretical and DEM result for a disk on an inclined plane of 45°

	$\phi = 0^\circ$ (sliding)			
	$N = N/(mg \cos \beta)$	$S = S/(mg)$	\ddot{a}/g	$\ddot{\theta}/g$
Theoretical (Ke and Bray, 1995)	1.000	0.000	0.7071	0.000
DEM	1.000	0.000	0.7071	0.000

3.5.5 Cumulative β distribution

For a DEM analysis, the sample must be representative of the grain properties including the particle size distribution (PSD) both in volume (mainly for the smallest particles) and in number (for the largest particles). This representativeness of PSD is a condition for a sample to be a representative volume element. However, this condition cannot be satisfied in DEM for a poly-disperse sample, due to the upper limit on the computation capacity. It is important to extract a discrete ensemble of grain sizes such that a statistical representation of the size classes is obtained (Radjai and Dubois, 2011).

The β distribution in its cumulative form is able to satisfactorily model the size span (ratio of the largest to the smallest grain) and the shape descriptor (relative weights of the maximum and the minimum grain sizes). The cumulative β distribution (Voivret et al., 2007) is defined as

$$\beta(x, a, b) = \frac{1}{B(a, b)} \int_0^x t^{a-1} (1-t)^{b-1} dt, \quad (3.54)$$

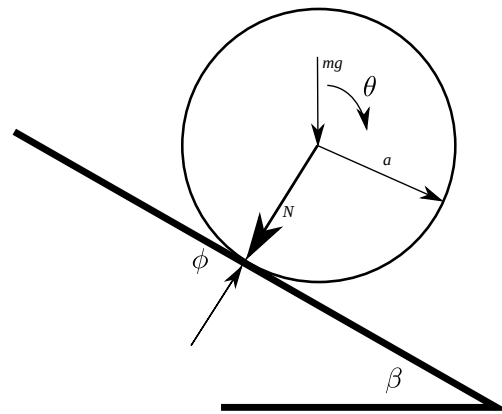
where $a > 0$ and $b > 0$ are the two parameters of the distribution. The $B(a, b)$ function is given by

$$B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}, \quad (3.55)$$

where Γ is the gamma function defined by

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt. \quad (3.56)$$

The cumulative β (CB) distribution is defined and normalised in the interval $[0, 1]$ with $\beta(0) = 0$ and $\beta(1) = 1$. To represent the grading curve of grain size range $[d_{min}, d_{max}]$ with



(a) Layout

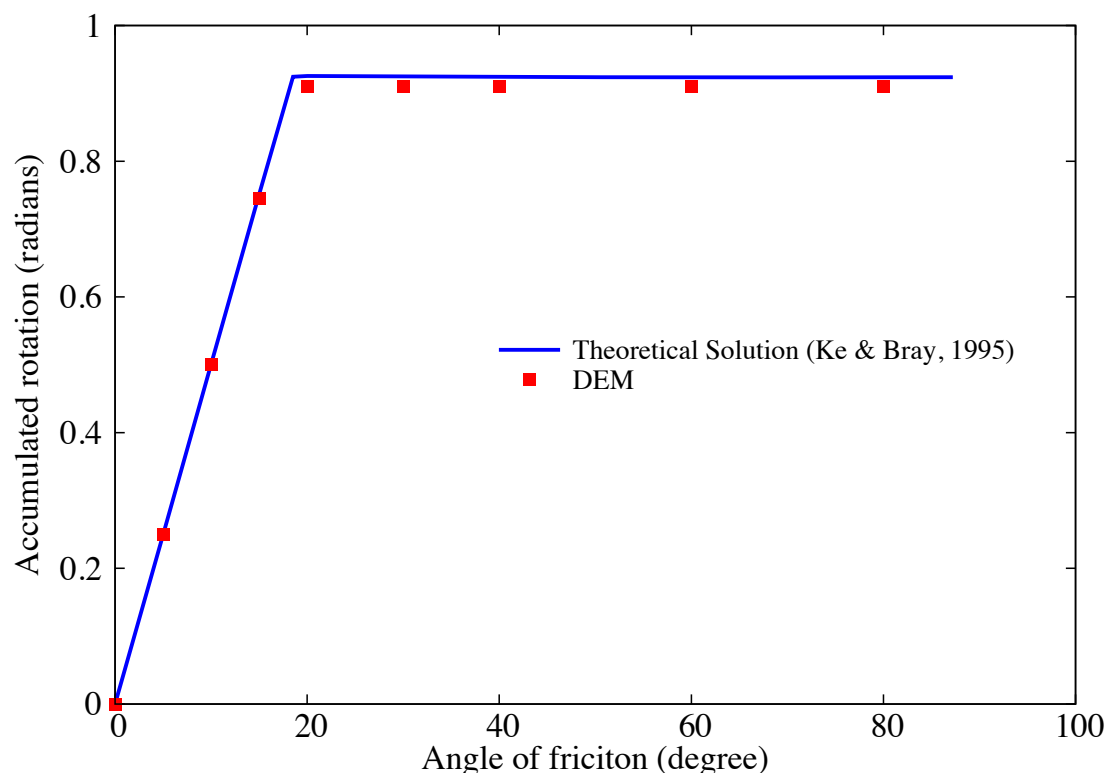
(b) Accumulated disk rotation at $t = 0.2$ s versus friction angle ($\beta=45^\circ$).

Figure 3.14 Validation of DEM using a disk rolling down an inclined plane

CB, the argument x is replaced by the reduced diameter d_r

$$d_r(d) = \frac{d - d_{min}}{d_{max} - d_{min}}, \quad (3.57)$$

which varies in the range of $[0,1]$. The model grading curve $h(d)$ is given by the CD distribution in terms of the reduced diameters as

$$h(d, a, b) = \beta(d_r(d); a, b). \quad (3.58)$$

The parameters a and b of the CB model allows the shape descriptor $h(d)$ to vary easily. The size span is characterised by the ratio $r = d_{max}/d_{min}$. The size span can also be defined as

$$s = \frac{d_{max} - d_{min}}{d_{max} + d_{min}}. \quad (3.59)$$

A value of $s = 0$ indicates a strictly mono-disperse grain size and a value of $s = 1$ indicates an infinitely poly-disperse grain size distribution. Voivret et al. (2007) observed a well graded particle size distribution for $a > 1$ and $b > 1$.

In the present study, a value of $a = 4$ and $b = 4$ is adopted to get a well graded curve. In the present study, samples with different poly-dispersity $r = 1.5, 1.8, 2$ and 6 are used. ?? shows a poly-disperse sample $r = 1.5$ generated using the CB method, for $a = 4$ and $b = 4$. The PSD curve of the generated sample is also presented.

3.5.6 Particle assembling methods

In order to simulate a granular assembly, it is essential to assign an initial position and velocity to all the grains in the system. Particle positions should be chosen to be compatible to the structure (granular fabric) we are trying to simulate. In any event, the grains should not be positioned such that there is an appreciable overlap between grains. In order to achieve the initial position of the grains, various grain-assembling methods can be adopted. The grain assembling methods can be classified into two broad categories: dynamic methods and geometrical approaches. The dynamic approach involves packing of grains using laws of mechanics and contacts, while in the geometrical method the grains are packed considering their geometry, i.e. grain size, shape and its position. In general, the packing of grains can be categorized into two types: crystal/lattice packing, like hexagonal or square pattern of mono-disperse grains, and random packing with varying density employing mono-disperse or poly-disperse grains. The crystalline packing arrangements, such as hexagon and square lattices, are easier to generate, however they have non-trivial effects on the response of the

granular system (Staron et al., 2005). Hexagonal packing is the densest possible arrangement for mono-dispersed spherical grains. In 2D, the packing of mono-dispersed circles on a hexagonal lattice yields a packing density, defined as the ratio of volume of solids to the total volume (V_s/V), of $\eta_h = \frac{1}{6}\pi\sqrt{3} \approx 0.9068$

The rheology of a granular material is controlled by the geometry of the assembly, which includes the grain shape, size distribution, and their arrangement. This prevailing role of geometry sometimes permits to simplify the dynamics in favour of a better description of the geometry and/or higher numerical efficiency (Radjai and Dubois, 2011). For example, a dense granular packing may be efficiently constructed by replacing the equations of dynamics by simple displacement rules satisfying the geometrical constraints. Purely geometrical procedures can be much simpler and numerically faster than dynamic or quasi-static methods. Contrary to dynamic simulation methods, the geometrical methods allow for quick assembling of a large number of grains. Such a packing may then be used as the initial state for dynamic simulations. The issue of the assembling methods is to construct configurations of grains as close as possible to a state of mechanical equilibrium with built-in packing properties. This can be a target packing density for a given grain size distribution. In the same way, the average connectivity of the grains (coordination number) and the anisotropy of the contact network are basic geometrical properties. The coordination number represents the mechanical response of packing. The homogeneity of the grain assembly in terms of packing fraction and connectivity is another important property, which depends on the assembling rules. In the present study, the initial grain packing is obtained using the ballistic deposition technique.

Ballistic deposition

Initially a random arrangement of grains which do not touch each other is generated. The radii of the grains are chosen from the interval of (R_{min}, R_{max}) in such a way that the total mass of all grains from a certain size interval is the same for all sizes, thus ensuring that neither larger nor smaller grains dominate the system. The grains are arranged randomly on a regular lattice. In the second step, the grains arranged in a regular lattice are allowed to fall down maintaining a constant potential head between layers of grains (figure 3.15).

The construction of the packing proceeds layer by layer from the substrate, hence this deposition model is also known as bottom-to-top restructuring model. In 2D, two contacts are sufficient to balance a grain if its centre of gravity lies between the two contacts. This corresponds to a position of local stable equilibrium. In this method, the order of deposited grains is generally random and independent of their sizes. The mechanically stable sample obtained from this method is presented in figure 3.16. Dense granular samples are prepared

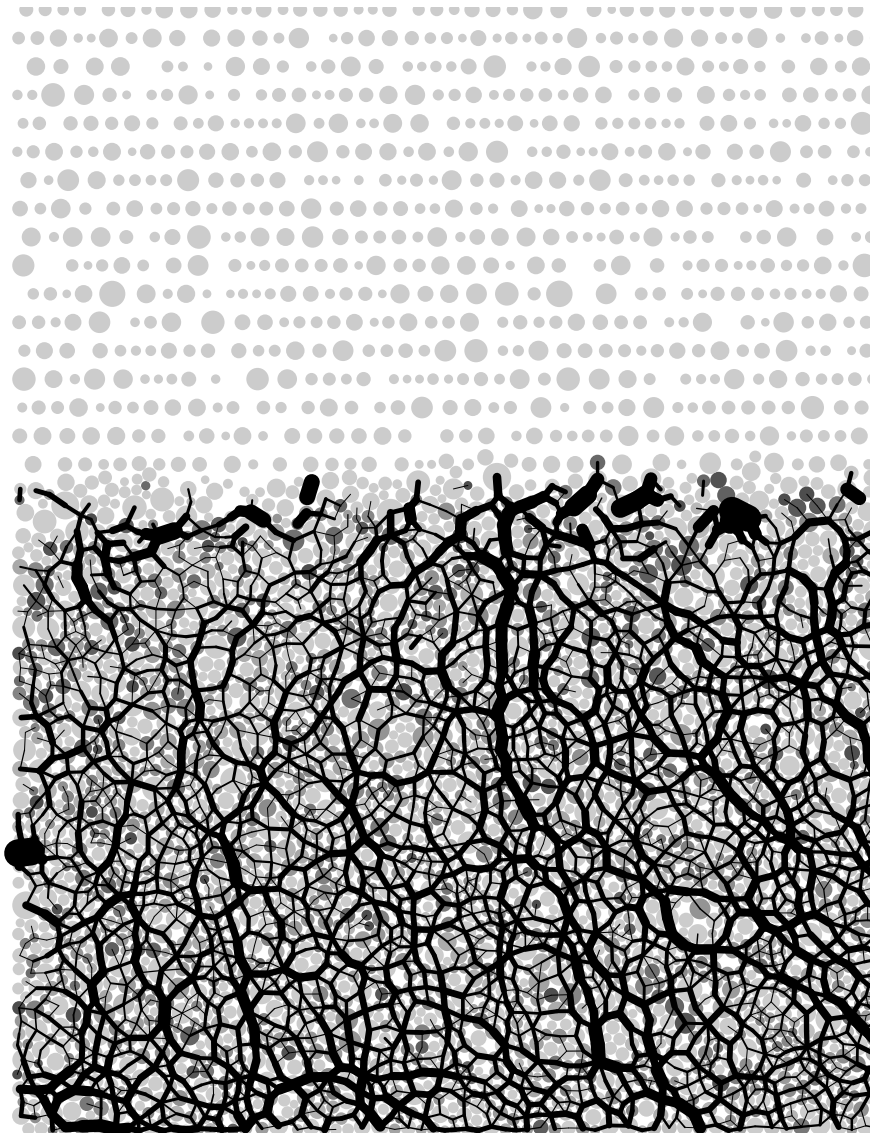


Figure 3.15 Generation of a poly-disperse sample using ballistic deposition technique

with zero initial friction angle. Zero frictional resistance ensures the densest possible packing. The friction angle is turned on during the analysis. To generate a dense sample, the sample is subjected to vibrations at varying frequencies.

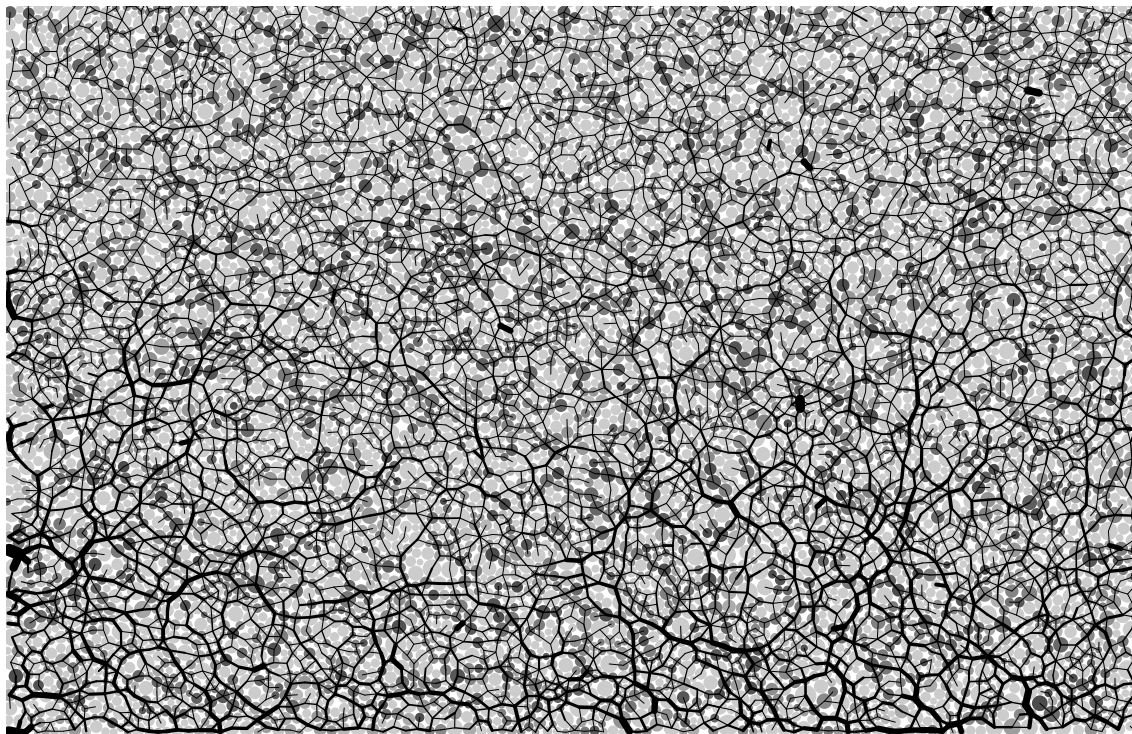


Figure 3.16 A poly-disperse DEM sample generated using ballistic deposition technique

Although there is no optimum specimen generation technique (O’Sullivan, 2011), there is a need to assess the homogeneity of the packing density generated (Jiang et al., 2003). The homogeneity of the sample is assessed by measuring the void-ratio within sub-volumes. The generated specimen is discretised into horizontal bands of $2.5d_{50}$. The homogeneity of the packing is calculated by the variance S in the void ratio.

$$S = \frac{1}{N_{layer} - 1} \sum_{i=1}^{N_{layer}} (e - e_i)^2, \quad (3.60)$$

where N_{layer} is the number of layers and e is the overall void ratio. A variance S of 2.74% is observed, which is less than 5% Jiang et al. (2003), indicating a homogeneous sample.

3.5.7 Voronoi tessellation

In order to extract bulk properties, such as packing density, stresses and strains, from a DEM simulation, it is important to quantify the granular texture. A useful geometrical

representation of granular texture consists of dividing the space occupied by the particles into contiguous cells. This procedure is called ‘tessellation’. Voronoi tessellation is one of the mostly commonly used technique. For a finite set of points p_1, \dots, p_n in the Euclidean space, the domain/plane is discretised into convex polygons such that each polygon contains exactly one point p_i and every point in a given polygon is closer to its generating point p_i than to any other. The inverse of the Voronoi tessellation is the Delaunay triangulation.

In the present study, the Fortune (1992) sweep line algorithm is implemented to tessellate the region surrounding each grain in the geometry. Once each grain has a corresponding area, the macroscopic properties such as the bulk density and stresses can be extracted from micro-mechanical properties such as the local packing density and force chains.

The Fortune sweep line algorithm involves a sweep line and a beach line, both of which move through the plane from left to right as the algorithm progresses. The sweep line is a straight line which moves from left to right across the plane. At any time during the algorithm, the points (grains) to the left of the sweep line will be incorporated into a Voronoi cell. While the points on the right of the sweep line are yet to be considered. The beach line is a complex curve, composed of pieces of parabolas that divides the plane within which the Voronoi diagram is known. As the sweep line crosses a point, a parabola evolves from the generating point. As the sweep line progresses, the vertices of the beach line, at which two parabolas cross, trace out the edges of the Voronoi diagram. The beach line progresses by keeping each parabola base exactly half way between the points initially swept over with the sweep line, and the new position of the sweep line. Figure 3.17 shows the Fortune sweep line algorithm in progress. In the present study, a modified version of the sweep-line algorithm is used to construct an additively weighted Voronoi diagram, in which the distance to each site (grain) is offset by the weight of the site, i.e., the radius of the grain.

The Voronoi tessellation is used to study the evolution of packing fraction during the collapse of a granular column. Figure 3.18a shows the Voronoi tessellation of the run-out for a granular column with an initial aspect ratio of 6 at time $t = 3\tau_c$. The distribution of local packing density for the run-out is shown in figure 3.18b. Dark regions represent dense packing, while loose regions are shown in white. The run-out at this stage has a bulk packing density of 81.23%. It is difficult to tessellate the free surface and the Voronoi cells on the free surface do not represent the actual packing density. Hence, during the evaluation of the macroscopic density, the packing density of surface grains that are larger than a threshold value are ignored. Voronoi Tessellation is a useful tool to extract continuum properties from DEM simulations. In the present study, the Voronoi tessellation is used to understand the evolution of packing density and entrainment of water in the flow front (due to hydroplaning) in granular flows.

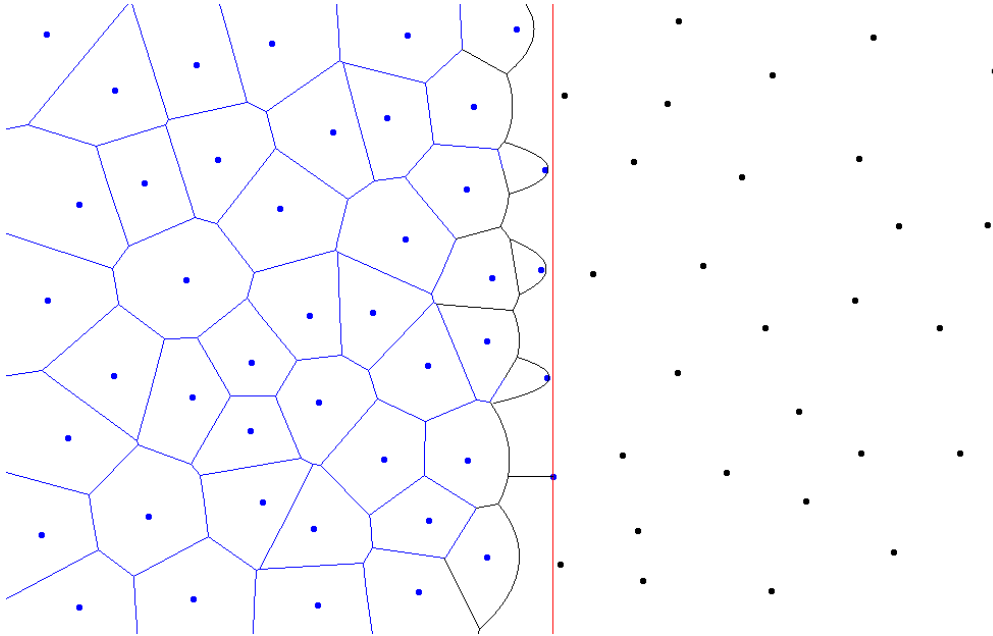
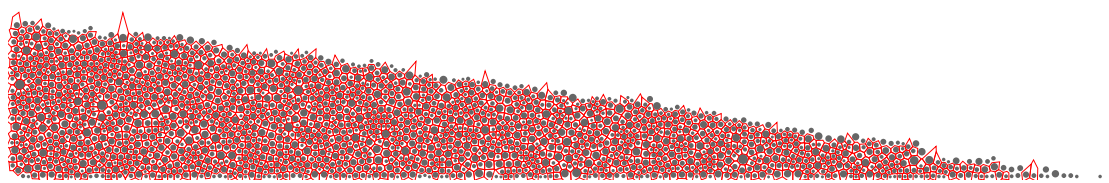
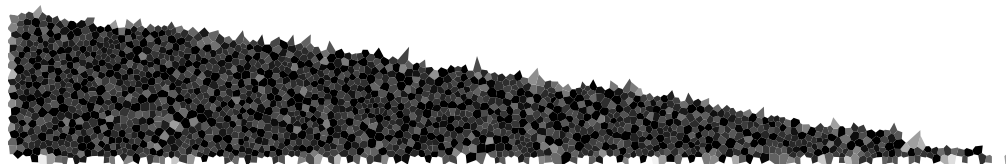


Figure 3.17 Fortune sweep line algorithm for generating Voronoi Tessellation. Generated from <http://www.diku.dk/hjemmesider/studerende/duff/Fortune/>.



(a) Voronoi Tessellation



(b) Local packing density (dark - dense and light - loose)

Figure 3.18 Voronoi tessellation of a run-out profile showing the local packing density

3.6 Summary

A plane-strain Material Point Method is implemented in the present study to describe the continuum response of granular flows. When the material points move from one cell to another, it results in numerical oscillations called as the cell-crossing noise. The Generalised Interpolation Material Point GIMP method is adopted to minimize these oscillations in large-deformation problems. The grain-scale response is captured using a two-dimensional DEM code. The discrete grains are generated using a cumulative β distribution, which mimics the particle size distribution curves. The sample is generated using the ballistic deposition technique and the homogeneity of the generated sample is verified by investigating the variance in the void-ratio at different layers. A sweep-line Voronoi tessellation approach is adopted to extract macroscopic parameters such as stresses and packing density from micro-scale properties such as the local packing fraction. These multi-scale tools are used to understand the rheophysics of dry granular flows and to evaluate the suitability of MPM as a continuum approach in modelling granular flow behaviour.

References

- Abe, K., Soga, K., and Bandara, S. (2013). Material Point Method for Coupled Hydromechanical Problems. *Journal of Geotechnical and Geoenvironmental Engineering*, 140(3).
- Agnolin, I. and Roux, J.-N. (2007). Internal states of model isotropic granular packings. I. Assembling process, geometry, and contact networks. *Phys. Rev. E*, 76(6):061302.
- Andersen, S. and Andersen, L. (2010). Modelling of landslides with the material-point method. *Computational Geosciences*, 14(1):137–147.
- Aranson, I. S. and Tsimring, L. S. (2001). Continuum description of avalanches in granular media. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 64(2 I):203011–203014.
- Aranson, I. S. and Tsimring, L. S. (2002). Continuum theory of partially fluidized granular flows. *Physical Review E*, 65(6):061303.
- Augarde, C. and Heaney, C. (2009). The use of meshless methods in geotechnics. In *Proceedings of the 1st International Symposium on Computational Geomechanics*, France.
- Bandara, S. (2013). *Material Point Method to simulate Large Deformation Problems in Fluid-saturated Granular Medium*. PhD thesis, University of Cambridge.
- Bardenhagen, S. (2002). Energy Conservation Error in the Material Point Method for Solid Mechanics. *Journal of Computational Physics*, 180:383–403.
- Bardenhagen, S. and Kober, E. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496.
- Bardenhagen, S. G., Brackbill, J. U., and Sulsky, D. (2000). The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):529–541.
- Bardenhagen, S. G., Guilkey, J. E., Roessig, K. M., Brackbill, J. U., Witzel, W. M., and Foster, J. C. (2001). An improved contact algorithm for the material point method and application to stress propagation in granular material. *Computer Modeling in Engineering and Sciences*, 2(4):509–522.
- Belytschko, T., Lu, Y. Y., and Gu, L. (1994). Element-free Galerkin Method. *International Journal for Numerical Methods in Engineering*, 37(2):229–256.

- Beuth, L., Wieckowski, Z., and Vermeer, P. A. (2010). Solution of quasi-static large-strain problems by the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 35(13):1451–1465.
- Bigler, J., Guilkey, J., Gribble, C., Hansen, C., and Parker, S. (2006). A case study: Visualizing material point method data. *Proceedings of Euro Vis 2006*, pages 299–306.
- Bonet, J. and Kulasegaram, S. (2000). Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47(6):1189–1214.
- Chen, Z. and Brannon, R. (2002). An evaluation of the material point method. *Sandia National Laboratories (SAND2002-0482)*.
- Coetzee, C. J., Vermeer, P. A., and Basson, A. H. (2005). The modelling of anchors using the material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 29(9):879–895.
- Cundall, P. A. and Strack, O. D. L. (1979). A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65.
- Donea, J., Giuliani, S., and Halleux, J. (1982). An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723.
- Fortune, S. (1992). Voronoi diagrams and Delaunay triangulations. In Du, D. and Hwang, F., editors, *Computing in Euclidean geometry*, pages 193–233. World Scientific Publishing Company.
- Frenkel, D. and Smit, B. (1996). *Understanding molecular simulation: From algorithms to applications*. Academic Press.
- Goldhirsch, I. (2003). Rapid granular flows. *Annual Review of Fluid Mechanics*, 35:267–293.
- Grubmuller, H., Heller, H., Windemuth, A., and Schulten, K. (1991). Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular simulations*, 6:121–142.
- Guilkey, J., Harman, T., and Banerjee, B. (2007). An Eulerian–Lagrangian approach for simulating explosions of energetic devices. *Computers & Structures*, 85(11-14):660–674.
- Guilkey, J. and Weiss, J. (2003). Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method. *International Journal for Numerical Methods in Engineering*, 57(9):1323–1338.
- Gumhold, S. (2003). Splatting illuminated ellipsoids with depth correction. In *VMV*, volume 2003, pages 245–252.
- Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Computer Methods in Physics*, 3:319–343.
- Hertz, H. (1882). Über die Berührung fester elastischer Körper. *Journal für die Reinen und Angewandte Mathematik*, 92:156–171.

- Itasca PFC2D (1998). Particle Flow Code in Two Dimensions.
- Jaeger, H., Nagel, S., and Behringer, R. (1996). Granular solids, liquids, and gases. *Reviews of Modern Physics*, 68(4):1259–1273.
- Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):235–257.
- Jiang, M., Konrad, J., and Leroueil, S. (2003). An efficient technique for generating homogeneous specimens for DEM studies. *Computers and Geotechnics*, 30(7):579–597.
- Josserand, C., Lagree, P. Y., and Lhuillier, D. (2004). Stationary shear flows of dense granular materials: A tentative continuum modelling. *European Physical Journal E*, 14(2):127–135.
- Kafaji, I. (2013). *Formulation of a dynamic material point method (MPM) for geomechanical problems*. PhD thesis, University of Stuttgart.
- Kamrin, K., Rycroft, C. H., and Bazant, M. Z. (2007). The stochastic flow rule: a multi-scale model for granular plasticity. *Modelling and Simulation in Materials Science and Engineering*, 15(4):S449–S464.
- Katz, O., Morgan, J., Aharonov, E., and Dugan, B. (2014). Controls on the size and geometry of landslides: Insights from discrete element numerical simulations. *Geomorphology*, 220:104–113.
- Ke, T.-C. and Bray, J. (1995). Modeling of Particulate Media Using Discontinuous Deformation Analysis. *Journal of Engineering Mechanics*, 121(11):1234–1243.
- Krogh, M., Painter, J., and Hansen, C. (1997). Parallel sphere rendering. *Parallel Computing*, 23(7):961–974.
- Kuester, F., Bruckschen, R., Hamann, B., and Joy, K. (2001). Visualization of particle traces in virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 151–157.
- Levoy, M. (1988). Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37.
- Li, S. and Liu, W. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55:1.
- Liu, Z. and Koyi, H. (2013). Kinematics and internal deformation of granular slopes: insights from discrete element modeling. *Landslides*, 10(2):139–160.
- Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169.
- Love, E. and Sulsky, D. L. (2006). An unconditionally stable, energy-momentum consistent implementation of the material-point method. *Computer Methods in Applied Mechanics and Engineering*, 195(33-36):3903–3925.
- Luding, S., Clément, E., and Blumen, A. (1994). Anomalous energy dissipation in molecular-dynamics simulations of grains: The "detachment" effect. *Physical Review*, 50:4113.

- Ma, J., Wang, D., and Randolph, M. (2014). A new contact algorithm in the material point method for geotechnical simulations. *International Journal for Numerical and Analytical Methods in Geomechanics*, 38(11):1197–1210.
- Mackenzie-Helnwein, P., Arduino, P., Shin, W., Moore, J. A., and Miller, G. R. (2010). Modeling strategies for multiphase drag interactions using the material point method. *International Journal for Numerical Methods in Engineering*, 83(3):295–322.
- Maeda, K. and Sakai, H. (2010). Seepage failure and erosion of ground with air bubble dynamics. In *Geoenvironmental Engineering and Geotechnics: Progress in Modeling and Applications*, pages 261–266. ASCE.
- Mast, C. M., Arduino, P., Mackenzie-Helnwein, P., and Miller, G. R. (2014a). Simulating granular column collapse using the Material Point Method. *Acta Geotechnica*, page In print.
- Mast, C. M., Arduino, P., Miller, G. R., and Mackenzie-Helnwein, P. (2014b). Avalanche and landslide simulation using the material point method: flow dynamics and force interaction with structures. *Computational Geosciences*, 18(5):817–830.
- Mehta, A. and Barker, G. (1994). The dynamics of sand. *Reports on Progress in Physics*, 57:383.
- Monaghan, J. (2005). Smoothed particle hydrodynamics. *Reports on progress in physics*.
- Mori, H. (2008). *The SPH method to simulate river levee failures*. Msc thesis, University of Cambridge.
- Nedderman, R. M. (1992). *Statics and Kinematics of Granular Materials*. Cambridge University Press.
- O’Sullivan, C. (2011). *Particulate discrete element modelling*. Taylor and Francis.
- O’Sullivan, C., Bray, J., and Li, S. (2003). A new approach for calculating strain for particulate media. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(10):859–877.
- Parrinello, M. and Rahman, A. (1980). Crystal structure and pair potentials: a molecular-dynamics study. *Physics Review Letters*, 45:1196.
- Poschel, T. and Schwager, T. (2005). *Computational Granular Dynamics*. Springer-Berlin.
- Qiu, G., Henke, S., and Grabe, J. (2011). Application of a Coupled Eulerian–Lagrangian approach on geomechanical problems involving large deformations. *Computers and Geotechnics*, 38(1):30–39.
- Radjai, F. and Dubois, F. (2011). *Discrete-element modeling of granular materials*. ISTE Wiley, London; Hoboken, N.J.
- Radjai, F. and Richefeu, V. (2009). Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41(6):715–728.

- Radjai, F., Voivret, C., and McNamara, S. (2011). Discrete-element modelling of granular materials. In Radjai, F. and Dubois, F., editors, *Discrete-element modelling of granular materials*, chapter Periodic B, pages 181–198. ISTE Wiley.
- Rickenmann, D., Laigle, D., McArdell, B. W., and Hübl, J. (2006). Comparison of 2D debris-flow simulation models with field events. *Computational Geosciences*, 10(2):241–264.
- Roscoe, K. (1970). The influence of strains in soil mechanics. *Geotechnique*, 20(2):129–170.
- Rycroft, C. H., Bazant, M. Z., Grest, G. S., and Landry, J. W. (2006). Dynamics of random packings in granular flow. *Physical Review E*, 73(5):051306.
- Rycroft, C. H., Kamrin, K., and Bazant, M. Z. (2009). Assessing continuum postulates in simulations of granular flow. *Journal of the Mechanics and Physics of Solids*, 57(5):828–839.
- Shin, W. (2010). *Numerical simulation of landslides and debris flows using an enhanced material point method*. Phd thesis, University of Washington.
- Staron, L., Radjai, F., and Vilotte, J. P. (2005). Multi-scale analysis of the stress state in a granular slope in transition to failure. *The European physical journal. E, Soft matter*, 18(3):311–20.
- Steffen, M., Wallstedt, P., Guilkey, J., Kirby, R., and Berzins, M. (2008). Examination and analysis of implementation choices within the material point method. *Computer Modeling in Engineering and Sciences*, 32:107–127.
- Sulsky, D., Chen, Z., and Schreyer, H. (1994). A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):179–196.
- Sulsky, D., Zhou, S.-J., and Schreyer, H. L. (1995). Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1-2):236–252.
- Tang, C., Hu, J., Lin, M., and Angelier, J. (2009). The Tsaoling landslide triggered by the Chi-Chi earthquake, Taiwan: insights from a discrete element simulation. *Engineering Geology*, 106(1-2):1–19.
- Tang, C., Hu, J., and Yuan, R. (2013). The transportation and deposition of the 2009 Hsialin landslide in Taiwan revealed from 3D granular discrete element simulation. In *Geophysical Research Abstracts*. EGU General Assembly.
- Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159(1):98.
- Voivret, C., Radjai, F., Delenne, J. Y., and El Youssoufi, M. S. (2007). Space-filling properties of polydisperse granular media. *Physical Review E*, 76(2):021301.
- Wallstedt, P. and Guilkey, J. (2008). An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics*, 227:9628–9642.

- Wieckowski, Z., Youn, S.-K., and Yeon, J.-H. (1999). A particle-in-cell solution to the silo discharging problem. *International Journal for Numerical Methods in Engineering*, 45(9):1203–1225.
- Yu, H. and Yuan, X. (2006). On a class of non-coaxial plasticity models for granular soils. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 462(2067):725–748.
- Zenit, R. (2005). Computer simulations of the collapse of a granular column. *Physics of Fluids*, 17(Compendex):031703–1–031703–4.
- Zhang, D. Z., Zou, Q., Vanderheyden, W. B., and Ma, X. (2008). Material point method applied to multiphase flows. *Journal of Computational Physics*, 227(6):3159–3173.
- Zhang, H. W., Wang, K. P., and Chen, Z. (2009). Material point method for dynamic analysis of saturated porous media under external contact/impact of solid bodies. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1456–1472.