

# Chapter 3

## Numerical modelling of granular flows

### 3.1 Introduction

Most of the geotechnical analysis involves failure prediction and 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 the 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 a continuum because they exhibit many collective phenomena. However, on a grain scale, the granular materials exhibit complex solid-like and/or fluid-like behaviour. 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, which are micro-structural effects. Discrete element methods (DEM) are capable of simulating these micro-structural effects, however they are computationally expensive.

### 3.2 Continuum modelling of granular flows

The most powerful way of modelling the granular assembly is through numerical techniques. It is important to argue, why it is acceptable to model the granular materials as a continuum. At



Figure 3.1 Multi-scale modelling of granular materials

1 the outset, it may even appear for some reasons why such a treatment is objectionable. Most  
 2 obvious is the fact that the micro-constituents of granular matter, i.e. the individual grains are  
 3 not small enough to warrant a continuum description (Kamrin et al., 2007). Typical continuum  
 4 laws are only expected to apply when there is a strong separation of scales, i.e. separation of  
 5 the micro-scale from the macro-scale, in the flow geometry. Continuum mechanics relies on  
 6 the fundamental notion of a representative volume element, in which, properties averaged over  
 7 discrete grains exhibit deterministic relationships. Recent works on granular materials suggest  
 8 that a continuum law may be incapable of revealing in-homogeneities at the grain-scale level,  
 9 such as force chains, fabric tensor, etc (Rycroft et al., 2009).

10 Granular materials exhibit many collective phenomenon (Jaeger et al., 1996). However, no  
 11 continuum model is still capable of describing the parabolic flow, the plug flow (Rycroft et al.,  
 12 2006) and the occurrence of localized shear bands in the granular materials. Most constitutive  
 13 models, even in the simple case of dry granular flows, cannot describe the entire range of  
 14 flow from solid to fluid. In certain cases, the granular flow is modelled as a fluid behaviour.  
 15 Continuum models that are based on averaging techniques applied to a representative volume  
 16 elements are not successful even in quas-static conditions. The fundamental question is how to  
 17 model granular materials which exhibit complex phenomenon, meaningfully.

18 The oldest approach involves modelling the granular material as a rigid solid, which behaves  
 19 as an ideal Coulomb material and undergoes failure if the ratio of the shear stress to the normal  
 20 stress in any plane reaches a critical value of the Coulomb internal friction coefficient  $\mu$ .  
 21 The stress is determined based on the mechanical equilibrium of the system along with the  
 22 hypothesis of *incipient yield*, i.e. the yield criterion is attained everywhere at all times. In limit-

---

3.2 Continuum modelling of granular flows3

---

state Mohr-Coulomb plasticity, these conditions are assumed to hold even if the wall allows for a plastic yielding, due to the assumption of coaxiality ([Rycroft et al., 2009](#)). The fundamental assumption of a limit-state stress field at incipient yield everywhere is questionable.

Granular flows can contain regions lying within the yield surface. For example, in the case of a granular column collapse the central cone remains stagnant, and thus cannot be considered as yielded. In fact, discrete-element simulations show that the grains in this region essentially remain static ([Staron et al., 2005](#)). The coaxiality feature of Mohr-Coulomb plasticity is useful in describing the debris flow. Granular material deforms solely based on the alignment of the principle plane. In general, the major principle plane is usually vertical due to gravity, and the coaxiality rule requires the material to expand horizontally, which is the case for granular column collapse. However, the coaxiality can be troubling depending on the circumstances, for example, in a slow dense granular flow through a silo, the principle plane remains vertical and the 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 discontinuity or jumps in the velocity and stress fields ([Rycroft et al., 2006](#)).

Advanced elasto-plastic models based on the *critical state* theory provide a better representation of granular flows in quasi-static regime, but they fail to capture the mechanism of rapid granular flows which involves rate dependent behaviour. Another continuum based model is the partial fluidization model, which uses a set of equations that describes the flow velocity and the shear stresses along with a auxiliary order parameter to predict the 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 rate-dependent part representing the rebound-less 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 the granular flow behaviour at rough boundaries.

In the case of saturated/submerged soil conditions, most of these techniques do not consider the fully coupled behaviour. Instead, 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 lubrication effect. Fully coupled constitutive models are essential to realistically capture the initiation and propagation of rapid granular flows.

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

### 11 **3.2.1 Mesh-based and mesh-free techniques**

12 In continuum mechanics, there are two different view points describing the deformation of a  
13 continuum, namely Lagrangian and Eulerian description. In the Lagrangian description the  
14 movement of the continuum is specified as a function of the material coordinates and time.  
15 This is a particle description that is often applied in solid mechanics. On the other hand, the  
16 Eulerian description focuses on the current configuration, giving attention to what is occurring  
17 at a fixed point in space as time progresses, instead of giving attention to individual particles as  
18 they move through space and time. The Eulerian description is commonly used for describing  
19 fluid flows where kinematic properties are of great interest.

20 Conventional mesh based Lagragian approaches, such as Finite Element Method or Finite  
21 Difference Method are capable of modelling history dependant material behaviour and have  
22 well-defined free surface. However, they require complex re-meshing and remapping of  
23 variables, which cause additional errors in simulating large deformation problems ([Li and Liu, 2002](#)). Unlike Lagrangian FEM, in Eulerian FEM the computational mesh is kept spatially fixed  
24 while the material is deforming in time. This allows the capability of handling large deformation  
25 without the problem of mesh distortion. As the computational mesh is completely decoupled  
26 from the material, convective terms appear in Eulerian FEM, introducing numerical difficulties  
27 because of their non-symmetrical properties ([Donea et al., 1982](#)). Additionally, Eulerian FEM  
28 is difficult to use with history dependent constitutive models. Coupled Eulerian–Lagrangian  
29 (CEL) method is an arbitrary Lagrangian-Eulerian that attempts to capture the advantages of  
30 both the Lagrangian and the Eulerian method in modelling large deformation problems in  
31 geomechanics ([Qiu et al., 2011](#)). This approach involves solving the governing equation in the  
32 Lagrangian step and thus obtaining the material displacement followed by the Eulerian step  
33 where a new mesh is generated and the variables are transferred to the the new mesh. Thus  
34 requiring higher computational time.

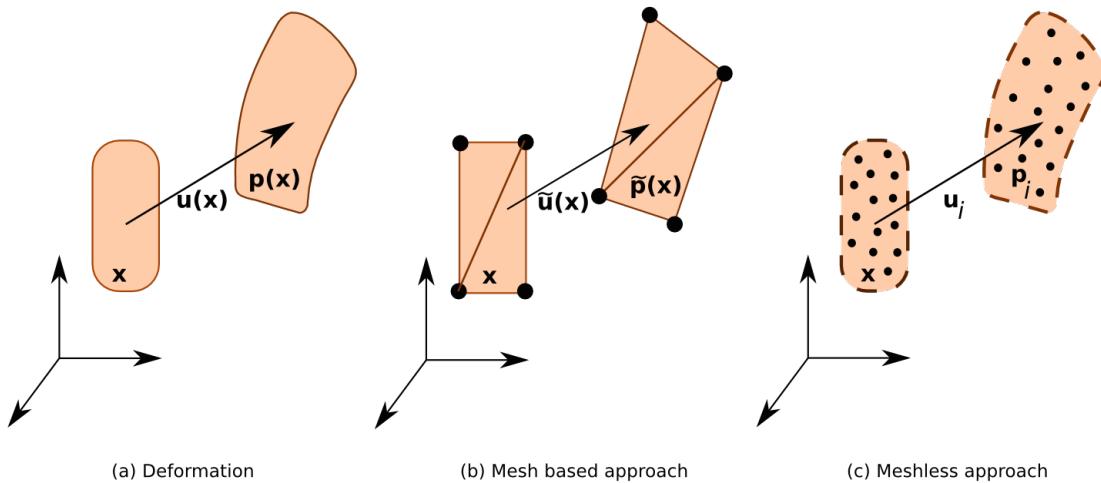


Figure 3.2 Difference between mesh-based and mesh-free techniques in modelling large deformation flow.

An alternative to the mesh-based approach is using meshless Lagrangian methods (figure 3.2) where the nodes representing the solid transforms as the continuum deforms, avoiding the problem of mesh distortion, i.e., the nodes representing the solid 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 meshless 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 meshless method that involves the meshless 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 Lagrangian fashion. In PFEM, large deformation requires frequent re-meshing (Kafaji, 2013).

Smooth Particle Hydrodynamics is the oldest meshfree technique, in 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 problem ([Monaghan, 2005](#)). SPH has been widely applied in geomechanics for solving large deformation problem ([Augarde and Heaney, 2009](#); [Maeda and Sakai, 2010](#); [Mori, 2008](#)). Although, SPH has been widely adopted it has a few drawbacks: SPH exhibits spatial instabilities, as a consequence of the point-wise integration ([Bonet and Kulasegaram, 2000](#)),

- <sup>1</sup> insufficient neighbouring particles causes inconsistencies and is computationally expensive as  
<sup>2</sup> a result of search for the neighbouring particles ([Bandara, 2013](#)).

### <sup>3</sup> 3.3 Material Point Method (MPM)

<sup>4</sup> Material Point Method (MPM) ([Sulsky et al., 1994, 1995](#)) is a particle based method that  
<sup>5</sup> represents the material as a collection of *material points*, and their deformations are deter-  
<sup>6</sup> mined by *Newton's laws of motion*. [Sulsky et al. \(1994\)](#) extended the Particle-in-Cell (PIC)  
<sup>7</sup> method ([Harlow, 1964](#)) to computational solid mechanics by taking advantage of the combined  
<sup>8</sup> Eulerian-Lagrangian approach. Material Point Method is a hybrid Eulerian-Lagrangian ap-  
<sup>9</sup> proach, which uses moving material points, and computational nodes on a background mesh.  
<sup>10</sup> This approach is very effective particularly in the context of large deformations ([Andersen and](#)  
<sup>11</sup> [Andersen, 2010; Bandara, 2013; Mackenzie-Helnwein et al., 2010; Mast et al., 2014a; Shin,](#)  
<sup>12</sup> [2010; Zhang et al., 2009](#)). Although, not derived directly from what is classically considered as  
<sup>13</sup> mesh-free or mesh-less methods, MPM is still considered as a mesh-free approach, primarily  
<sup>14</sup> because the initial discretisation of the material does not involve a polygonal tessellation, as in  
<sup>15</sup> Finite Element Method. However, MPM utilizes a background mesh to perform differentiation,  
<sup>16</sup> integration, and to solve equations of motions ([Steffen et al., 2008](#)). The background mesh can  
<sup>17</sup> be of any form, however for computational efficiency a Cartesian lattice is adopted. A typical  
<sup>18</sup> 2D discretisation of a solid body is shown in figure 3.3.

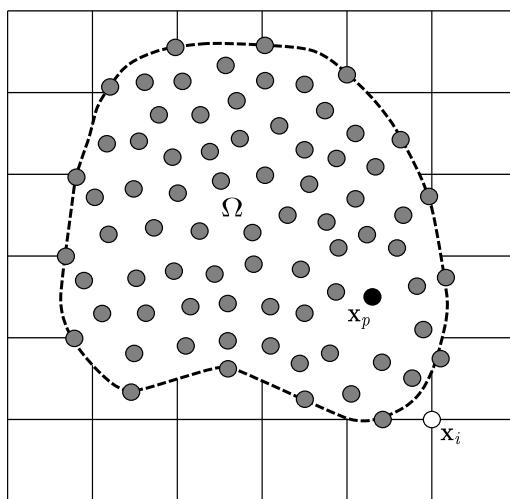


Figure 3.3 Typical discretisation of a domain in MPM. The dotted line represents the boundary of the simulated object  $\Omega$  and each closed point represents a material point used to discretise  $\Omega$ . The square mesh represents the background grid. Each square in the background grid is a grid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

The grey circles 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  $\Omega$  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  $\sigma_p$ . 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 the ease of computation, is assumed to be a Cartesian lattice (see figure 3.3). At every time step  $t_k$ , 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 based on the position of particle with respect to the grid. Gradient terms are calculated in the computational grid and the governing equation, i.e. the equation of motion, is solved and the updated position and velocity values are mapped 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:

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

$$\rho a = \Delta \cdot \sigma + \rho b, \quad (3.2)$$

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,  $\sigma(\mathbf{x}, t)$  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 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  $\rho_p^t$ , velocity  $v_p^t$ , Cauchy stress tensor  $\sigma_p^t$ , strain  $\epsilon_p^t$ , 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  $w(\mathbf{x}, t)$ .

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

where  $\sigma^s$  is the specific stress (i.e. stress divided by mass density,  $\sigma^s = \sigma/\rho$ ),  $\Omega$  is the current configuration of the continuum,  $\tau$  is the 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:

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

where  $\delta$  is the Dirac delta function. Substituting [eq. 3.4](#) in [eq. 3.3](#), the sum of quantities of material points can be evaluated as:

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

where  $h$  is the thickness of the boundary layer. 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  $x_i(t), i = 1, 2, \dots, N_n$ , 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 co-ordinates 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 has to be of the form

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

The equations (3.6) and (3.7) ensures that the associated vectors are continuous across the cell boundary. However, the gradient of these functions are 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

## 3.3 Material Point Method (MPM)

9

motion reduces to

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

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

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

where  $\mathbf{G}_{ip}^k = \Delta N_i(x)|_{x=X_p^t}$ . 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  $\rho_s$  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, hence 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 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 conditions are applied on the mesh nodes by controlling the nodal acceleration tangential to the boundary. The nodal accelerations were 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 are 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 constrains 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 MPM algorithm is more suited to flow calculations than Lagrangian finite element methods. In dynamic MPM, the 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 involves updating the strain and the stress at the beginning of the time step from the velocities 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 in certain conditions simulate artificial damping, and the numerical stability.

In problems involving 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, using the Newmark integration scheme. [Love and Sulsky \(2006\)](#) showed that the implicit time integration in MPM to be unconditionally stable. Although, MPM did not suffer from the limitations of FEM in simulating large deformations, more research is required for applying implicit time integration for large deformation problems.

### 3.3.4 Solution scheme

A step-by-step solution scheme for Material Point Method 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 Finite Element Method. The material points are followed throughout the deformation of the material, which gives 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 parameter 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 is computed.
- The information and properties carried by each material point is projected on to 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  $t$ ,  $m_p$  the particle mass,  $N_i$  the shape function associated with node  $i$ , and  $x_p^t$  the location of the particle at time  $t$ .

- The nodal velocity is obtained by mapping the particle velocity to 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:

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

- Strain increment  $\Delta \boldsymbol{\varepsilon}_p^{t+1}$  for 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

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

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

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

$$\begin{aligned} \boldsymbol{\sigma}_p^{t+1} &= \boldsymbol{\sigma}_p^t + \Delta \boldsymbol{\sigma}_p^{t+1}, \\ \boldsymbol{\varepsilon}_p^{t+1} &= \boldsymbol{\varepsilon}_p^t + \Delta \boldsymbol{\varepsilon}_p^{t+1}. \end{aligned} \quad (3.22)$$

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

$$\Omega_p^{t+1} = J\Omega_p^{t_0}. \quad (3.23)$$

- The material point density is then updated as

$$\rho_p^{t+1} = \frac{\rho_p^t}{\{1 + \text{tr}(\Delta\boldsymbol{\epsilon}_p^{t+1})\}}. \quad (3.24)$$

- 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 any analysis, involves visualization and extraction of the data from the analysis. In mesh-less methods, like the material point method, 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). MPM poses a whole new set of visualization problem, it is essential to visualize the general configuration of the body as well as observe the finer details like 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 represents 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 the structure represented by the material points, and (2) understanding the qualitative trend associated with the material points like mass, velocity or stress.

The MPM output data contains both the material point and the grid data, one approach in visualizing MPM data is by rendering the interpolated particle values on grid nodes using *isosurfacing* (Lorensen and Cline, 1987) or *volume rendering* (Levoy, 1988) technique. In regions where the material points are sparse, it is necessary that the grid resolution to be sufficiently

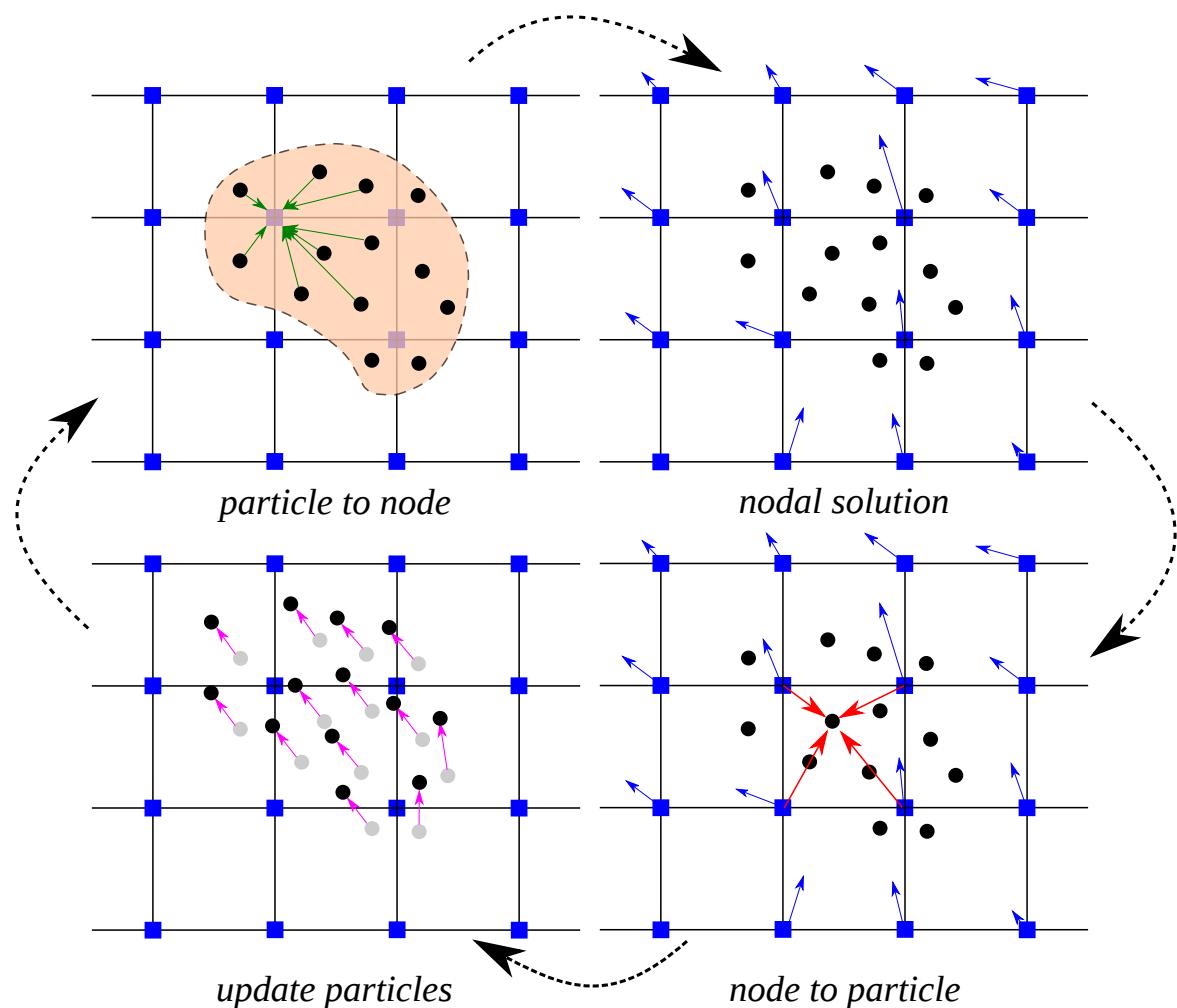


Figure 3.4 Illustration of MPM algorithm (1) A representation of material points overlaid on a computational grid. Arrows represent material point state vector (mass, volume, velocity, etc.) being projected to the nodes of the computational grid. (2) The equations of motion is solved on the nodes, resulting 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 are updated and the computational grid is reset.

fine to compensate for the missing features. This results in storing 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 sphere or an ellipsoid 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 Effect of mesh size and number of material points per cell

The accuracy of MPM simulations largely depends on the number of material points representing the continuum. MPM utilizes a grid to compute the deformation of a continuum, hence the size of the cells affects the accuracy of the results. Generally in MPM, the number of particles per cell controls the accuracy of the simulation. Guilkey et al. (2003) recommends higher particle density, such as 4 particles per cell, for large deformation problems. Very low particle density will result in non-physical opening of cracks in large deformation simulations. However, a higher value of particle density affects the computational time.

Abe et al. (2013) observed that for a coarse mesh, the numerical error decreases with increase in the number of material points per cell. In contrast, they observed an opposite trend for the fine meshes. The influence of numerical noise due to particles crossing the background mesh was not observed in coarse meshes. Coetzee et al. (2005) also found that the numerical error decreases with increase in mesh refinement.

In the present study, the effect of mesh size and the number of material points per cell on the run-out behaviour is investigated. The run-out behaviour of a granular slope of length 0.8 m and height 0.2 m is subjected to a gradient impact velocity is studied. The initial static slope is set into motion by applying a horizontal gradient velocity  $v_{0x}(y) = k(y_{max} - y)$  with  $k > 0$ . ?? shows the initial configuration and distribution of initial horizontal impact velocity of 50J in the granular mass. Detailed description of this problem and the material properties used in the simulations are found in ???. A mesh size of 0.0125 m is adopted. The number of material points per cell (PPC) is varied as 4, 16, 25, 36, 64, 81 and 100.

The effect of number of material points on the run-out behaviour is presented in figure 3.5. At a low input energy of 50 J, 4 and 16 material points per cell result in a longer run-out distance, where as the run-out distance converges when the number of PPC is more than 25. While at a high input energy of 500 J, both 4 and 16 PPC predict almost the same run-out distance, but is higher than the run-out predicted with more than 25 material points per cell.

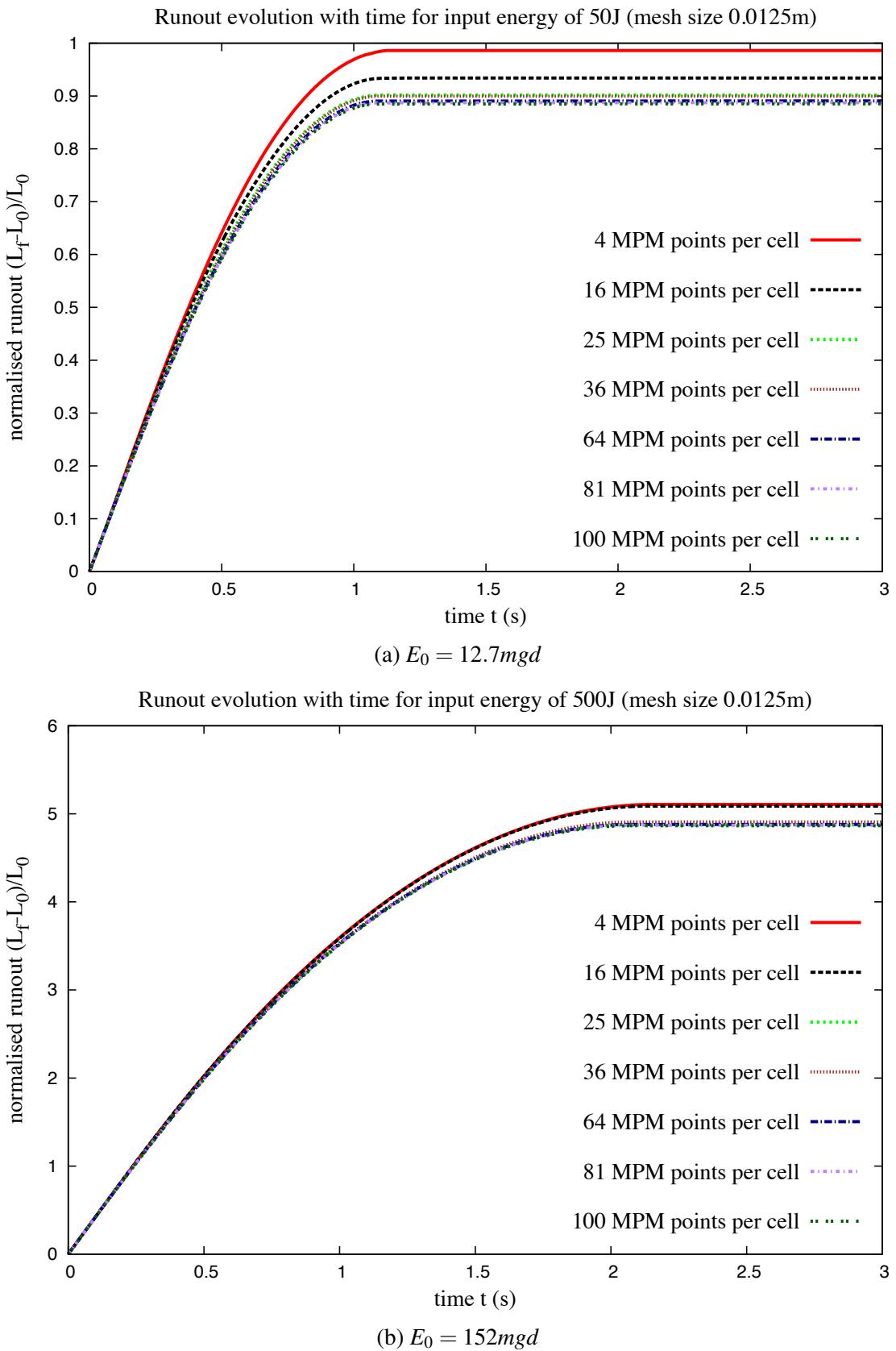


Figure 3.5 Evolution of run-out with time for varying material points per cell.

The evolution of the granular pile during the initial stage of flow is show in figure 3.6 for different number of material points per cell. At low input energy, fewer material points per cell results in larger separation of the spreading mass from the left wall. Distinct shear bands can be observed for more than 16 PPC. The flow structure remains unchanged with increase in PPC of more than 25. At a higher input energy (figure 3.7), almost all cases predict similar flow structure, except in the case of 4 PPC.

Figure 3.8 shows the evolution of kinetic energy with time for varying number of material points per cell. At low input energy, the horizontal kinetic energy evolution is identical for all cases. A slightly quicker run-out evolution during the spreading phase can be observed for the case of 4 PPC. However, increase in the number of material points per cell significantly affects the evolution of the vertical kinetic energy  $E_{ky}$ . At low energy, a large proportion of the input energy is dissipated in the destabilisation process. This results in material points falling behind the spreading mass to the fill the cavity. Fewer material points per cell results in cell crossing noise as the material points filling the cavity experience free-fall due to gravity. The effect of cell-crossing noise can be seen in the osciallation of vertical kinetic energy for fewer material points per cell. However, at high input energy, most of the input velocity is dissipated during the spreading process. This means that only a small fraction of energy is available in the vertical component resulting in almost similar behaviour for all cases. Four material points per cell predicts a higher peak vertical kinetic energy in comparison with other cases, unlike the low energy case, no oscillations were observed for the high input energy.

The effect of mesh size on the flow kinematics is studied by comparing two mesh sizes: 0.01 m and 0.0125 m (figure 3.9). It can be observed that the run-out distance converges with an increase in the number of material points per cell in both cases. Less than 1% difference in the run-out distance was observed between a mesh size of 0.0125 m and 0.01 m. The final run-out duration is almost unaffected by the increase in the number of material points per cell.

This shows that the run-out distance is affected by the number of material points per cell. However, the duration of the run-out is independent of the number of material points per cell. The computation time increases with increase in the number of material points per cell and decrease in the mesh size. However, the run-out distance converges with increase in number of material points per cell. Hence, an optimum number of 25 material points per cell was adopted in this case. In summary, for conducting a successful MPM analysis, a careful selection of the mesh size and the number of particles is necessary.

### 3.3.6 GIMPM

The shape functions used in MPM are continuous, and hence penetration between bodies are handled automatically without the need for any supplemental contact algorithm (Chen and

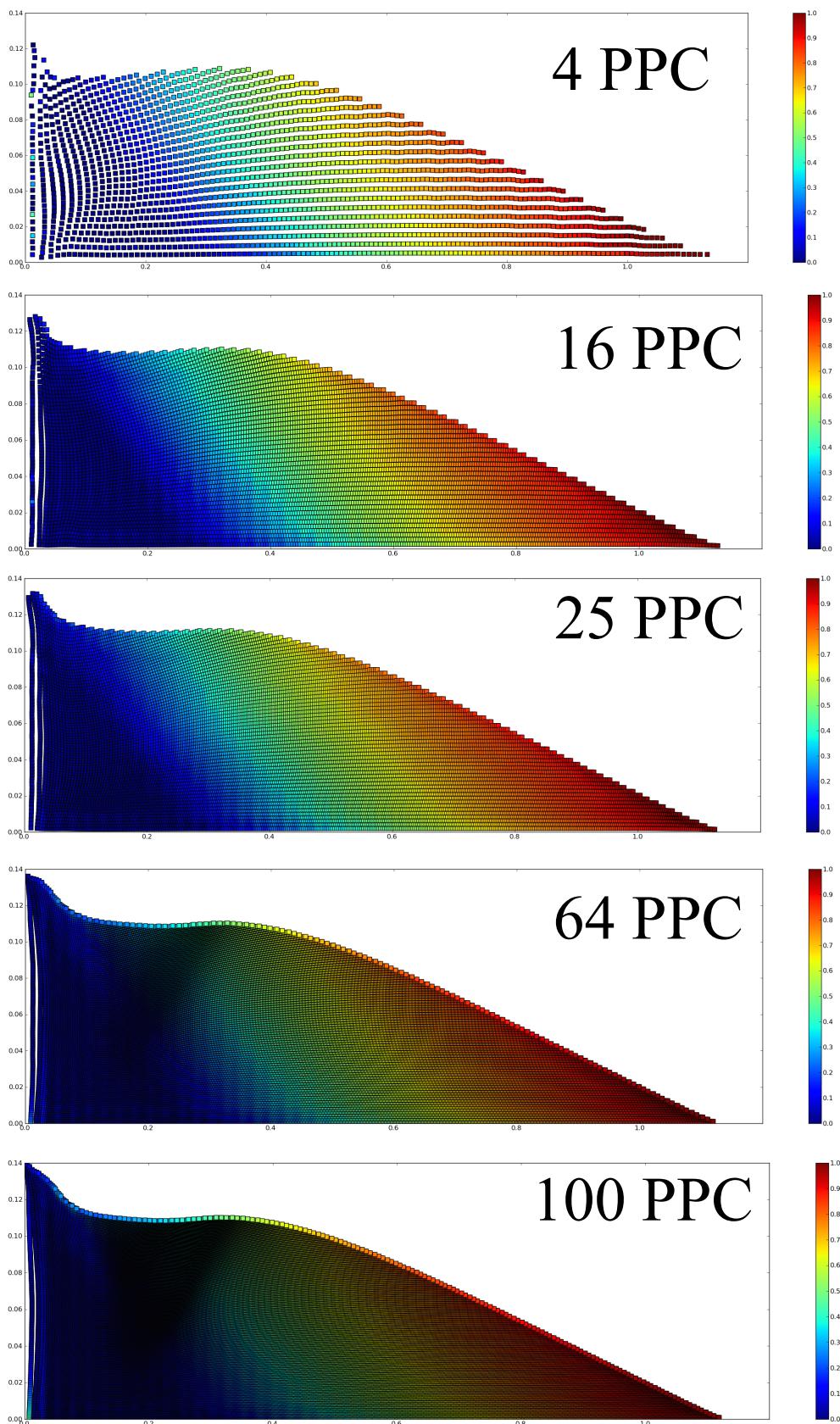


Figure 3.6 Effect of number of material points on cell on the run-out behaviour  $E_0 = 12.7mgd$ . Velocity profile (m/s) of granular pile subjected to gradient impact loading.

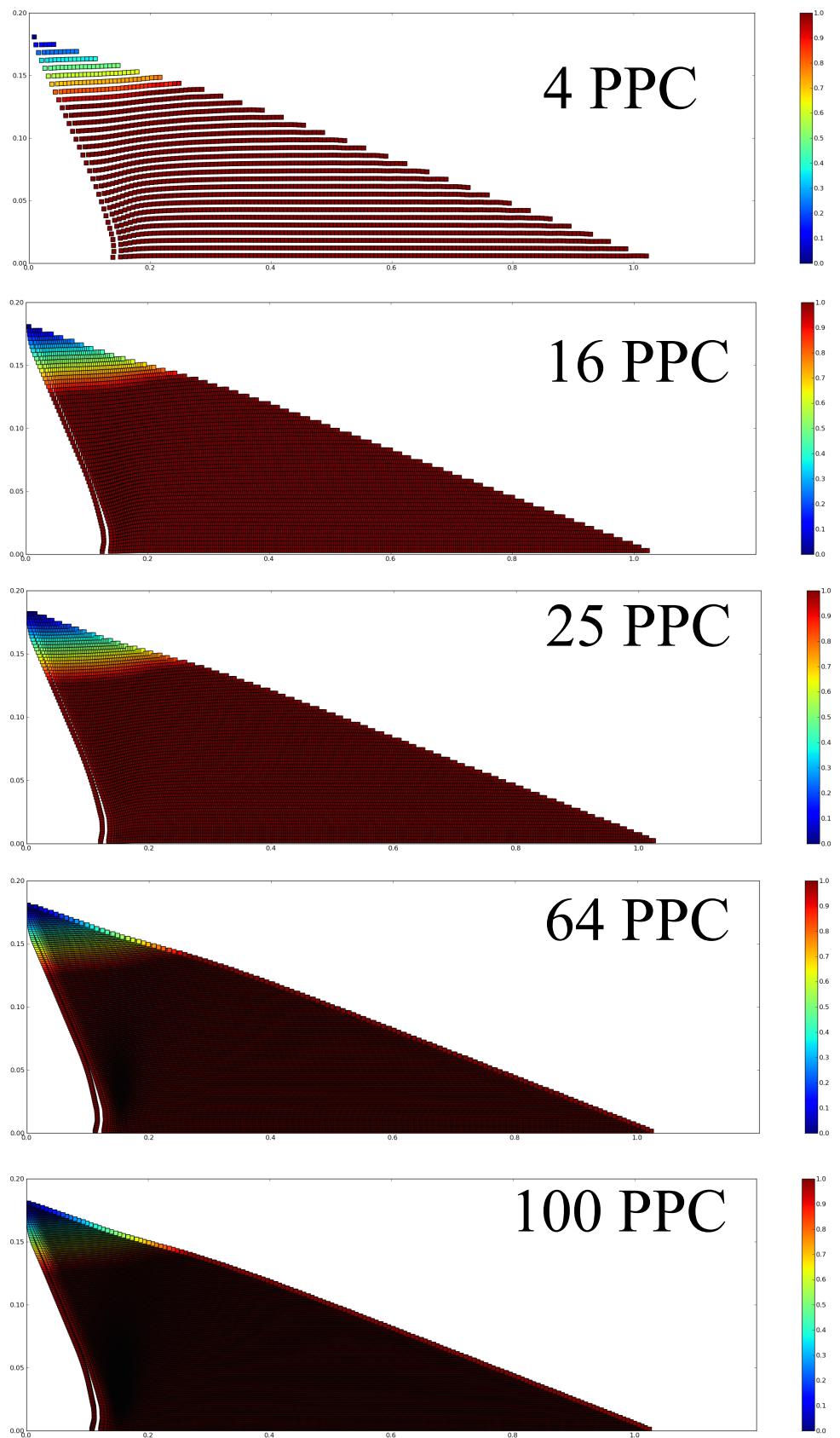


Figure 3.7 Effect of number of material points on cell on the run-out behaviour  $E_0 = 152mgd$ . Velocity profile (m/s) of granular pile subjected to gradient impact loading.

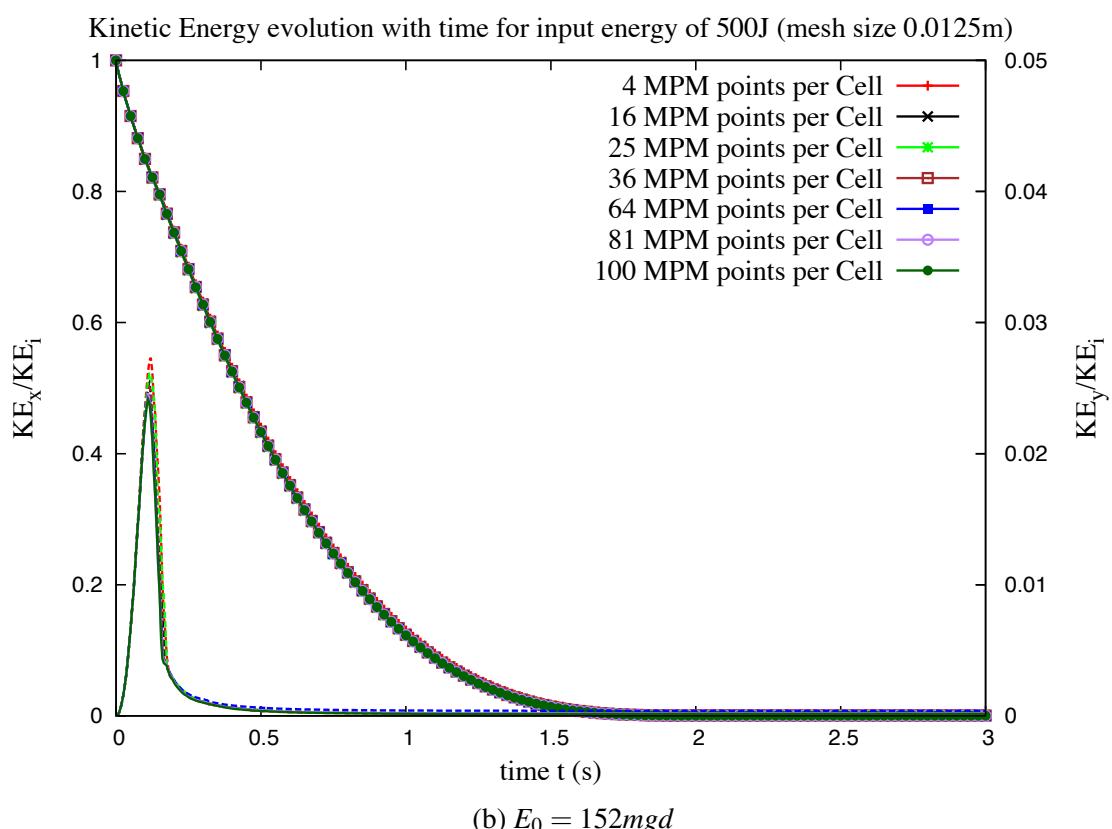
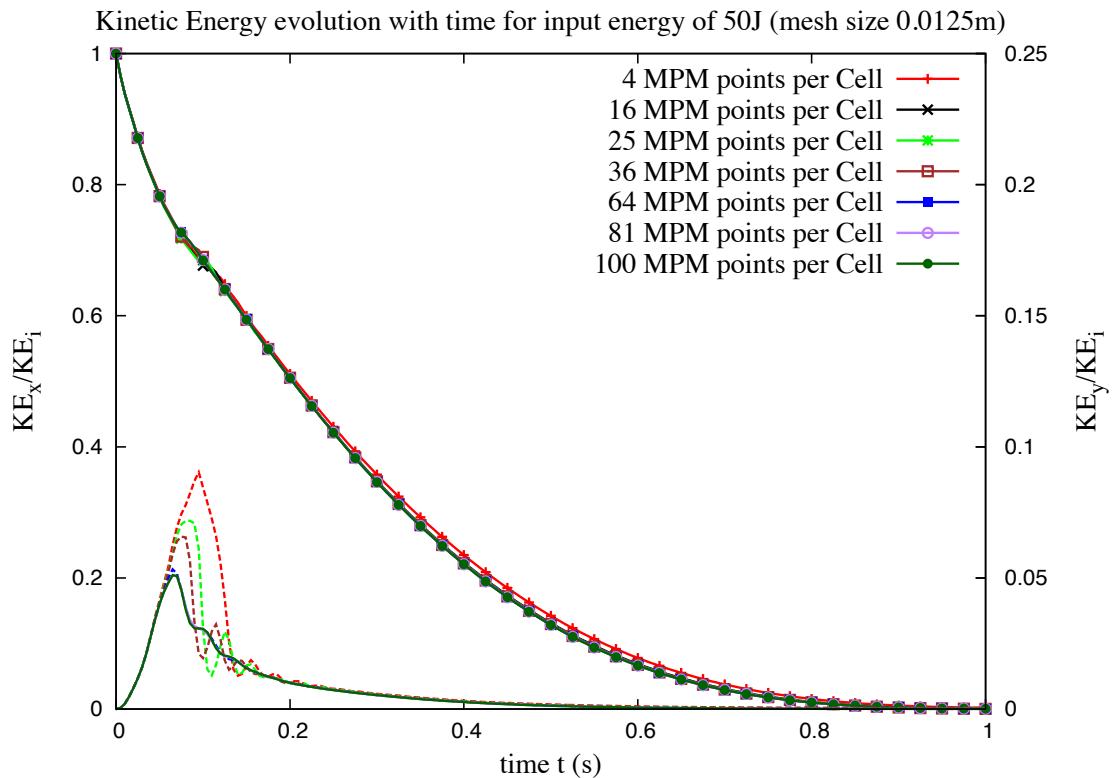


Figure 3.8 Evolution of kinetic with time for varying material points per cell

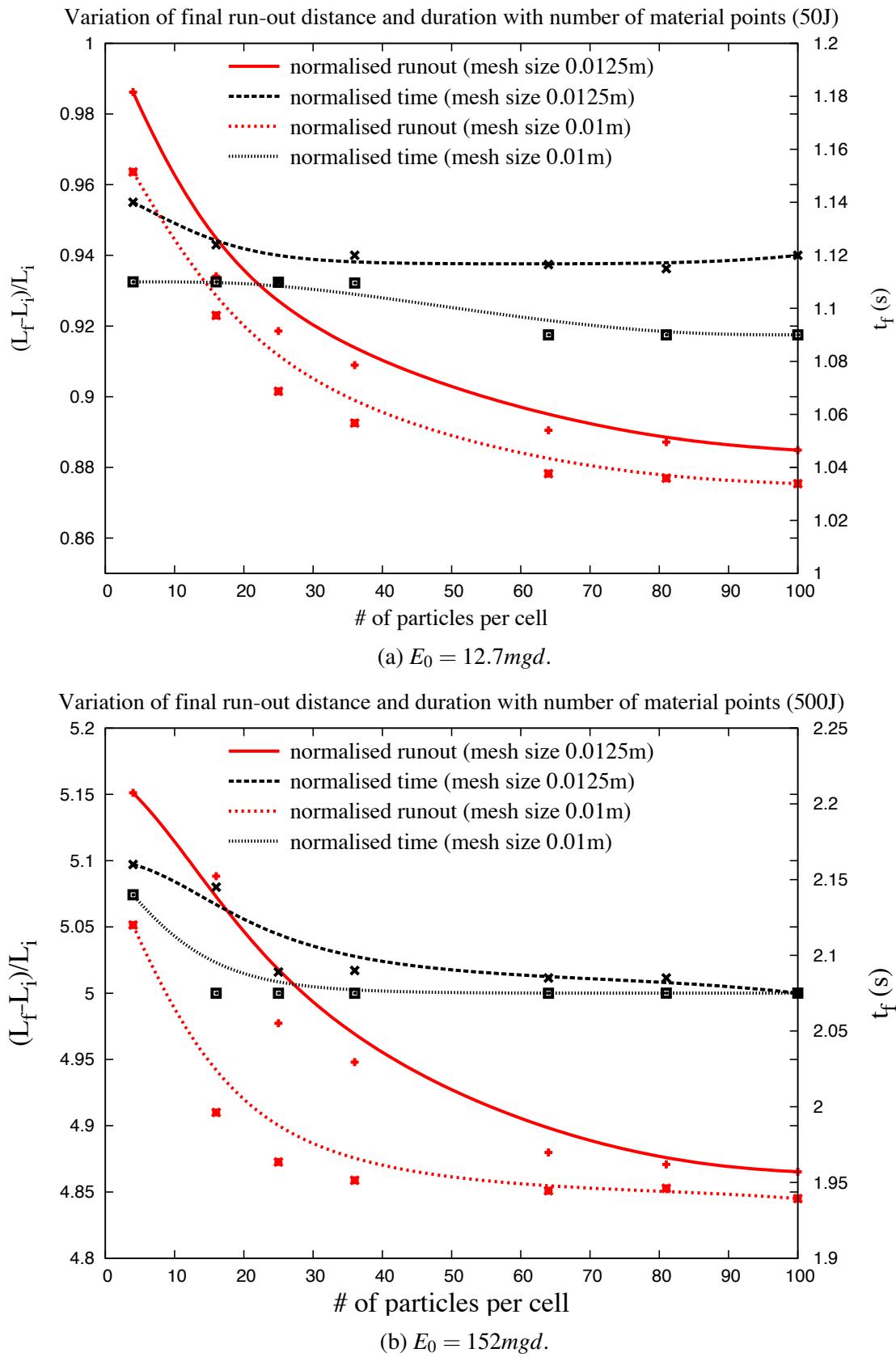


Figure 3.9 Evolution of run-out and duration of flow for varying material points per cell.

<sup>1</sup> Brannon, 2002). In MPM, the continuum body deforms and moves in an arbitrary computation  
<sup>2</sup> grid and all the boundary conditions are carried by the boundary particles. If a boundary particle  
<sup>3</sup> is present in a cell, then the cell boundary becomes a part of the continuum body, and the cell  
<sup>4</sup> size represents the thickness of the boundary. However, in certain cases both the boundary  
<sup>5</sup> particle and an interior particle can be found in a cell, in which case the cell is still treated  
<sup>6</sup> as a boundary cell, and the interior particle temporarily acts as a boundary particle. To avoid  
<sup>7</sup> numerical errors, it is essential to consider smaller cell size along the boundary (Chen and  
<sup>8</sup> Brannon, 2002).

<sup>9</sup> In MPM simulations, numerical noises are observed when the material points crosses the  
<sup>10</sup> cell boundaries as the body deforms, this is termed as *cell crossing noise*. If a material point  
<sup>11</sup> is located very close to the cell boundary, it results in discontinuous gradient of the weighing  
<sup>12</sup> function causing a force imbalance on the grid (Bardenhagen and Kober, 2004). This results  
<sup>13</sup> in large non-physical acceleration values resulting in separation of material points from the  
<sup>14</sup> continuum (Sulsky et al., 1995). Figure 3.10 illustrates the problem of cell crossing noise.  
<sup>15</sup> The main reason for the occurrence of cell crossing noise is the use of piecewise linear shape  
<sup>16</sup> functions. However, this problem, which is predominant when using fine mesh size, can be  
<sup>17</sup> overcome by changing the order of arithmetic operation as proposed by Sulsky et al. (1995).

<sup>18</sup> To overcome the problem of cell crossing noise, Bardenhagen and Kober (2004) proposed  
<sup>19</sup> an alternate method called the Generalized Interpolation Material Point Method that uses  
<sup>20</sup> smoother shape functions and a larger influence region for each grid node. This approach  
<sup>21</sup> minimizes the cell crossing noise. A piecewise-linear grid basis functions are written as

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

<sup>24</sup> where  $h$  is the grid spacing. The basis function associated with grid node  $i$  at position  $x_i$   
<sup>25</sup> is then  $\phi_i = \phi(x - x_i)$ . The basis functions in 3-D are separable functions constructed as  
<sup>26</sup>  $\phi_i(x) = \phi_i^x(x)\phi_i^y(y)\phi_i^z(z)$ . GIMP is often implemented using the standard piecewise-linear grid  
<sup>27</sup> basis functions and piecewise-constant particle characteristic functions:

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

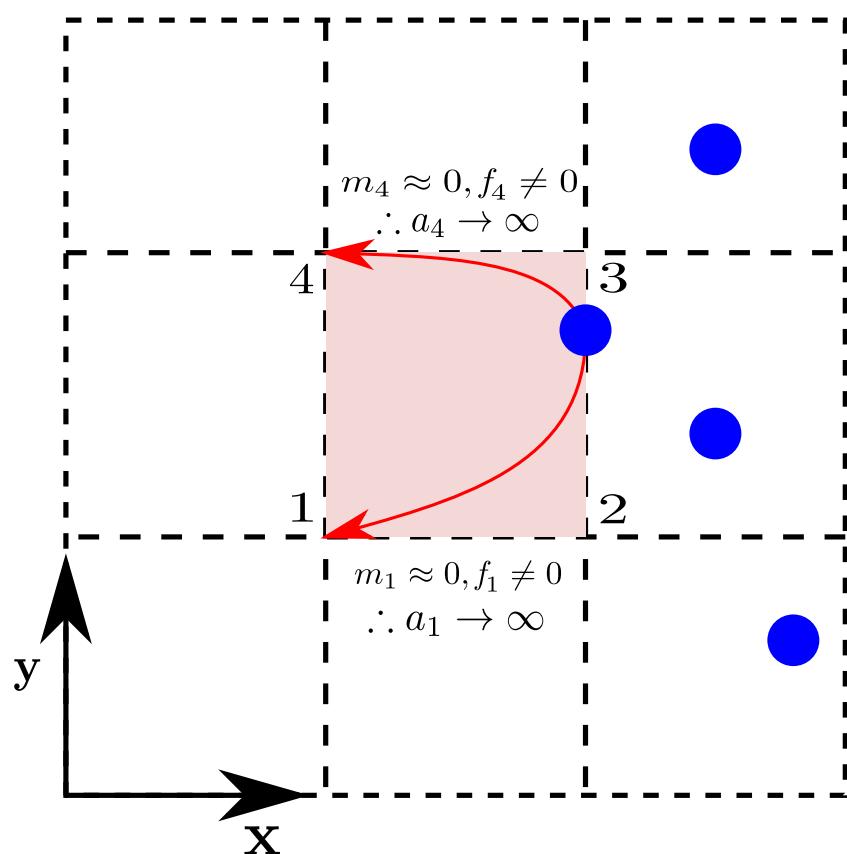


Figure 3.10 Cell crossing noise

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

$$3 \quad \phi = \begin{cases} 1 - (4x^2 + l_p^2)/(4hl_p) & : |x| < \frac{l_p}{2} \\ 1 - |x|/h & : \frac{l_p}{2} \leq |x| < h - \frac{l_p}{2} \\ \left(h + \frac{l_p}{2} - |x|\right)^2/(2hl_p) & : h - \frac{l_p}{2} \leq |x| < h + \frac{l_p}{2} \\ 0 & : otherwise, \end{cases} \quad (3.27)$$

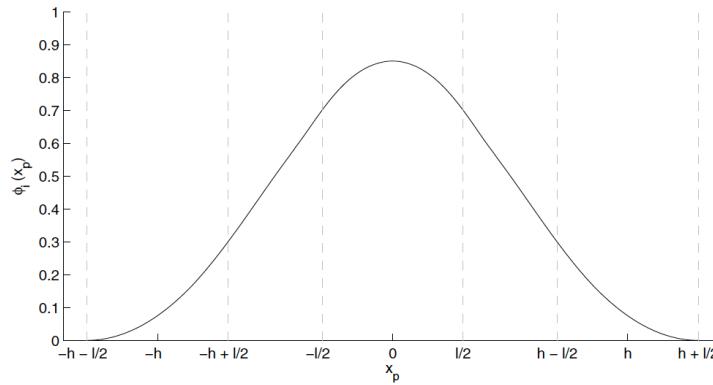
4

- 5 where  $l_p$  is the width of the particle characteristic function  $\chi_p$ . figure 3.11a shows a 1-D GIMP  
 6 weighting function  $\bar{\phi}_{ip}$  and gradient weighting function  $\nabla\bar{\phi}_{ip}$  for a piecewise-constant  $\chi_p$  with  
 7 a characteristic length of  $l$ . The GIMP weighting function is smooth, however a discontinuity is  
 8 observed in the gradient weighting function.

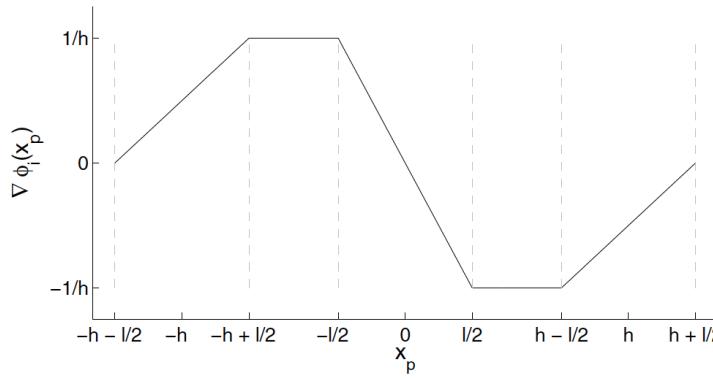
9 In traditional MPM, boundary conditions only need be applied on those nodes which  
 10 coincide with the extents of the computational domain. As illustrated in figure 3.11c nodes  
 11 beyond those boundaries are not influenced by particles within the domain. This can be  
 12 considered a result of the zero width of the Dirac delta characteristic functions. However,  
 13 special attention is required to simulate the boundaries in the Generalized Interpolation Material  
 14 Point Method (GIMP). Namely, because of their increased extents, it is possible for particles to  
 15 influence, and be influenced by, nodes that lie outside of the simulation domain (figure 3.11b).  
 16 These extra nodes are referred to as the “ghost” nodes. Boundary condition treatment of these  
 17 nodes for Dirichlet conditions is the same as for the regular boundary nodes, namely, their  
 18 computed values are replaced by prescribed values (Steffen et al., 2008).

19 In the present study, the influence of GIMP method on the run-out behaviour of a granular  
 20 column collapse experiment is investigated. A granular column of height  $H_0$  and length  $L_0$   
 21 is allowed to collapse and flow on a horizontal plane (??). The run-out distance observed is  
 22 proportional to the initial aspect ratio of the column ( $H_0/L_0$ ). Further details about the granular  
 23 collapse experiment and the material properties used in the simulations are presented in ??.  
 24 A granular column with an initial aspect ratio of 0.4 is considered for the comparison. The  
 25 granular column is represented by 32,000 material points arranged uniformly on a regular lattice  
 26 with a particle spacing of 0.25 mm. A grid size of 1 mm is adopted with 16 material points  
 27 per cell. In order to understand the influence of the number of material points on the accuracy  
 28 of the solution, simulation using 4 material points per cell with GIMP is also performed. The  
 29 granular collapse experiment is performed for a column with an initial aspect ration of 0.4  
 30 using both GIMP method and MPM with 16 material points per cell.

31 The evolution of run-out at time  $t = \tau_c$  and  $t = 6\tau_c$ , where  $\tau_c$  is the critical time when the  
 32 potential energy is fully mobilised, are presented in figure 3.11. At the initial stage of collapse

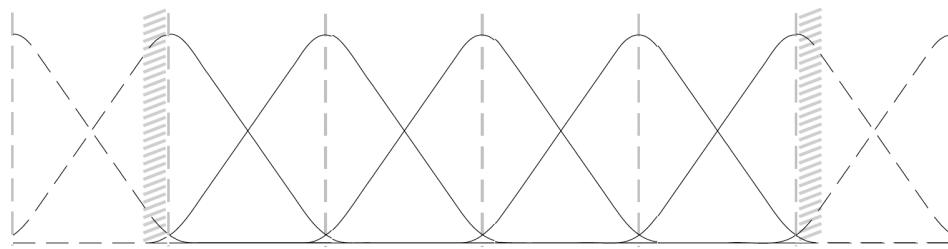


(a) GIMP Weighting Function

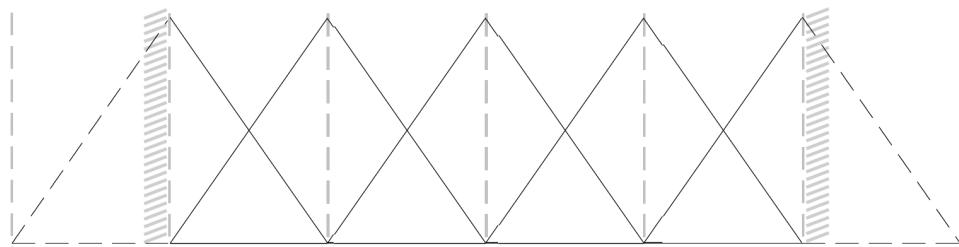


(b) GIMP Gradient Weighting Function

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



(b) GIMP



(c) Piecewise-Linear

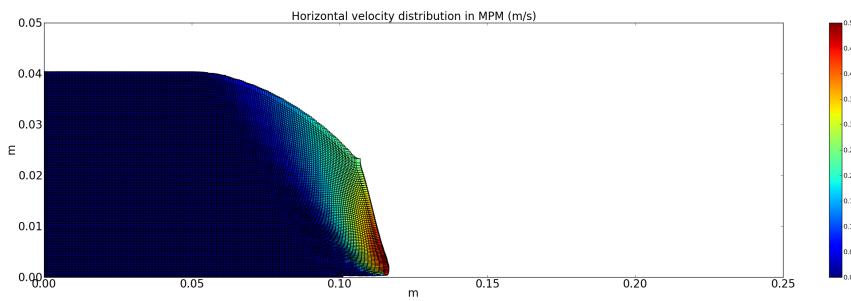
Figure 3.11 1-D basis functions used in MPM ([Steffen et al., 2008](#)).

<sup>1</sup>  $t = \tau_c$ , both MPM and GIMP give almost the same behaviour. The run-out observed in the  
<sup>2</sup> case of 4 material points per cell is similar to the run-out behaviour for 16 material points  
<sup>3</sup> per cell. At the end of the flow, GIMP with 4 material points show oscillations at the flow  
<sup>4</sup> front due to fewer material points in a cell. However, both MPM and GIMP show a more  
<sup>5</sup> smoother response at the flow front. The evolution of run-out and height with time for both  
<sup>6</sup> MPM and GIMP are presented in figure 3.12. For a time up to  $t = 1.5\tau_c$  all approaches yield  
<sup>7</sup> the exact same behaviour. However, as the flow progresses, the number of material points per  
<sup>8</sup> cell at the flow front decreases and this results in oscillations. The oscillations with decrease  
<sup>9</sup> in the number of material points can be seen in the 4 material points per cell. Hence, it is  
<sup>10</sup> essential to have a larger number of material points, especially at the flow front. The difference  
<sup>11</sup> in the normalised run-out between MPM and GIMP method is about 2.5%. This is due to  
<sup>12</sup> the difference in the interpolation scheme adopted in both approaches. GIMP method offers  
<sup>13</sup> a better approximation of the run-out behaviour as a result of the continuous basis function.  
<sup>14</sup> This also results in a more smooth stress and velocity distribution in GIMP than MPM. With  
<sup>15</sup> increase in the number of material points and the use of finer mesh size decrease the difference  
<sup>16</sup> between both approaches. The computational effort using GIMP method is almost twice that  
<sup>17</sup> of MPM, since GIMP method considers the particles at neighbouring cells due to the larger  
<sup>18</sup> spread of the basis functions. In the present study, a very fine mesh and 16 material points per  
<sup>19</sup> cell are used to simulate large deformation problems.

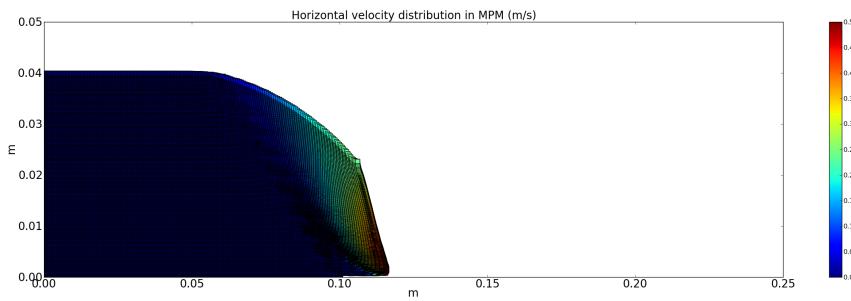
### <sup>20</sup> 3.3.7 Application of MPM in geomechanics

<sup>21</sup> The studies on using the material point method in modelling geotechnical problems are limited.  
<sup>22</sup> The potential of Material Point Method in modelling granular flows, discharge of silos, was  
<sup>23</sup> first recognised by Wieckowski et al. (1999). Bardenhagen et al. (2001) developed a frictional  
<sup>24</sup> contact algorithm to model granular materials. The Mohr-Coloumb is used to describe the  
<sup>25</sup> kinematics of the grains. In this model, a contact is defined when the nodal velocity interpolated  
<sup>26</sup> from all material points in the cell differs from the nodal velocity interpolated from a single  
<sup>27</sup> material point. Coetzee et al. (2005) applied this contact algorithm to understand the pull-out  
<sup>28</sup> behaviour of anchors. Ma et al. (2014) developed a new contact algorithm with a penalty contact  
<sup>29</sup> function and a limited maximum shear stress for soil-structure interaction, i.e, interaction of  
<sup>30</sup> pipe-line with debris flow or submarine landslide. The penalty contact function behaves similar  
<sup>31</sup> to numerical damping and thus reduces the oscillations during the impact.

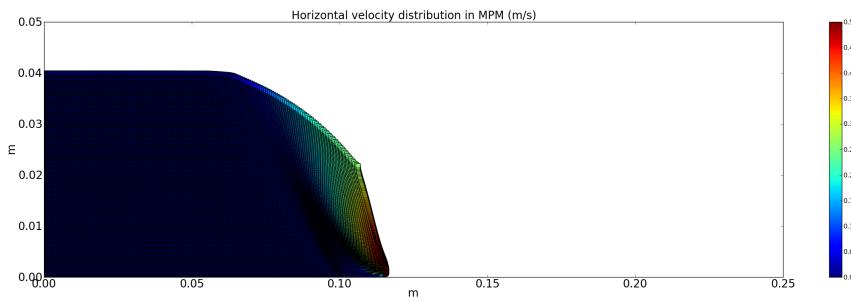
<sup>32</sup> Beuth et al. (2010) applied Gaussian integration in quasi-static MPM to model large strain  
<sup>33</sup> problems. However in this approach, the conservation mass is not valid as a larger filled  
<sup>34</sup> volume of material is considered. Andersen and Andersen (2010) used GIMP method with  
<sup>35</sup> an elasto-plastic model to simulate slope failures. The solution is found to be dependant on



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

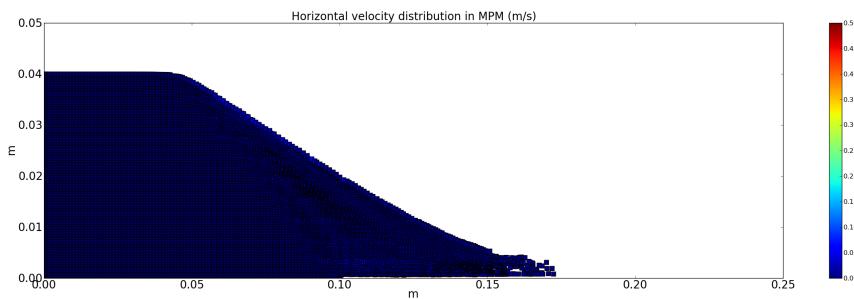


(b) GIMP method (16 material points per cell)

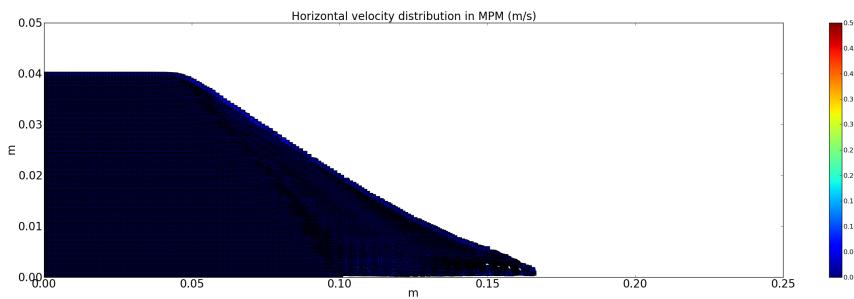


(c) Conventional MPM (16 material points per cell)

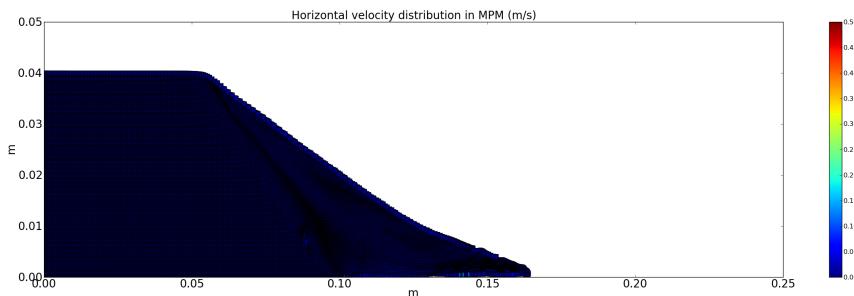
Evolution of granular column collapse at  $t = \tau_c$ .



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



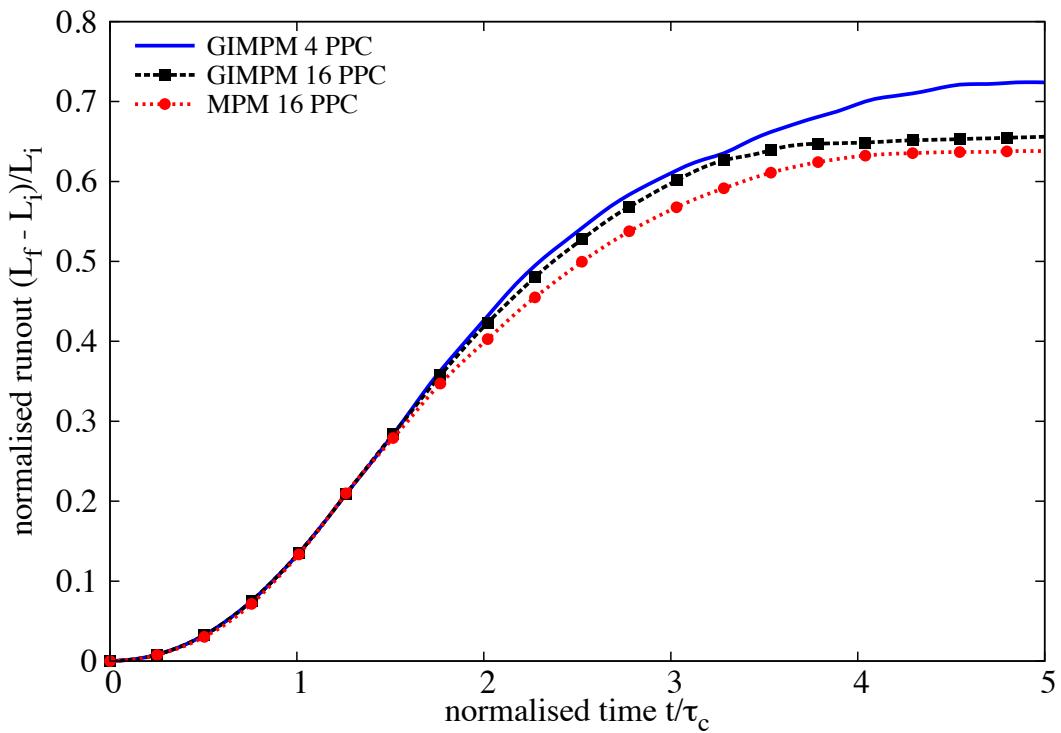
(b) GIMP method (16 material points per cell)



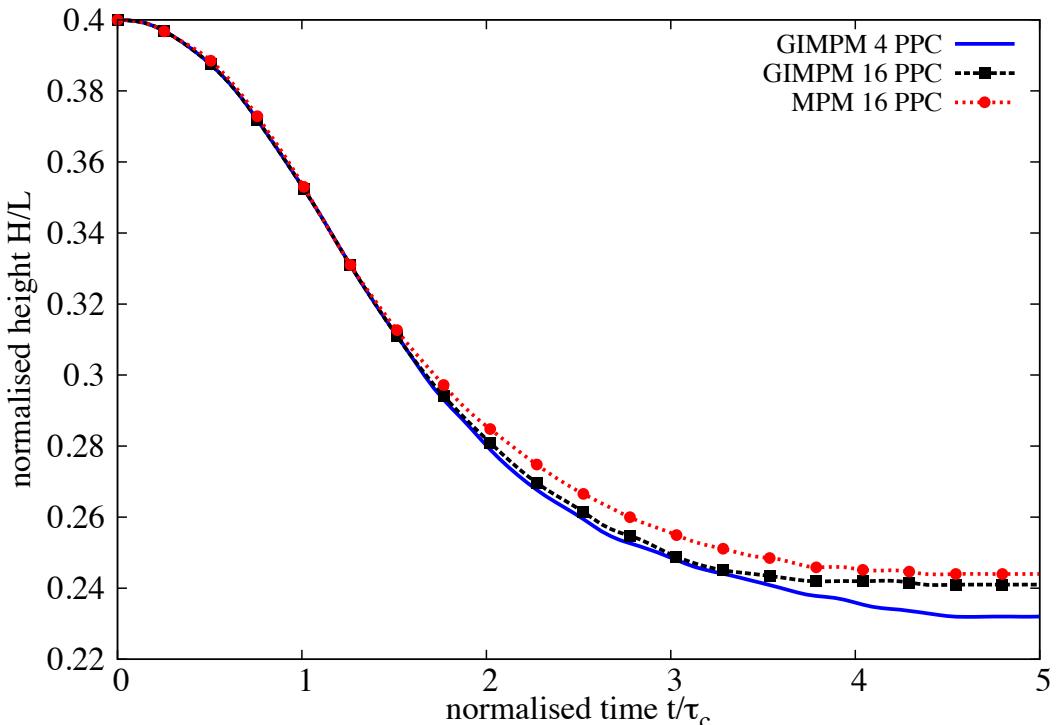
(c) Conventional MPM (16 material points per cell)

$$t = 6\tau_c$$

Figure 3.11 Evolution of granular column collapse.



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



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

Figure 3.12 Evolution of run-out and height with time. Difference between conventional MPM and GIMP method.

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 Material Point Method is realistic prediction of large deformation problems such as snow avalanches.

[Guilkey et al. \(2007\)](#) developed a coupled numerical scheme for fluid-structure interaction. The solid field is modelled in a Lagrangian frame, while an Eulerian frame, compressible CFD, is used for the fluid. As MPM computation grid is reset at every stage, the same Eulerian grid is adopted for both the solid and the fluid. Similar approach was adopted by [Zhang et al. \(2008\)](#) to model multiphase flows, where the interactions between the solid and gas are modelled using MPM background mesh as an Eulerian grid. [Mackenzie-Helnwein et al. \(2010\)](#) investigated various techniques to model multiphase drag forces. Both solid and the 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 impact loading. The suitability of continuum approach (MPM) in modelling large deformation problems is also investigated.

### 3.4 Particulate modelling of granular flows

Granular materials often exhibit different behaviour under different circumstances. Fluidized granular material often resembles a liquid, and reveals surface waves. In certain situations, granular materials behave more like a solid exhibiting plastic deformations. Despite the wide variations in the physical and chemical properties of the grains, the discrete granular structure has a rich generic phenomenology, which motivates us 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 could be approximately defined using continuum mechanics. However, On a grain level, the granular materials exhibit complex solid-like and/or fluid like behaviour depending on the way the grains interact with each other. The Analytical and Finite Element models, which consider granular materials as 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 based on the Discrete Element approach proposed by ? are capable of simulating the granular material as a discontinuous system. Although, modern measurement techniques can probe into the local granular variables, like grain position, velocities, contact forces, etc., 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 the relative motion between grains constrained in a dense state and/or by in-elastic collisions in the loose state. ? applied this method to granular geomaterials, and called it the *Distinct Element Method*, to differentiate 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. A similar method called *Discrete Element Method* (DEM) was used at the same time for the simulation of molecular systems with classical schemes that could be directly applied to granular materials. In-spite of the formal analogy (grains and force laws) between granular and molecular systems, the physics is fundamentally different. The interactions between 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 the Molecular Dynamics approach, 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 by neglecting the finer details such as interpenetration and overlap between grains. The fundamental difference between the non-smooth method and the common DEM or Discrete Element Method (DEM) approach lies in the treatment of small length and time scales involved in the dynamics of granular media. In DEM-type 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 Discrete Element Method, imply that the contact interactions involve smaller time and length scales. In the CD method, these small scales are neglected and their effects are absorbed into the contact laws. In non-smooth formulation, the grain dynamics is described at a larger scale than the elastic response time and displacement scales ([Jean, 1999](#); [Radjai and Richefeu, 2009](#)).

### 3.5 Discrete Element Method

Discrete Element Method computes the equilibrium and the trajectories of a classical multi-body system. The Discrete Element Method 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 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 as Discrete Element Method. The idea of Discrete Element Method goes back to Alder and Wainwright who in 1957 investigated the physics of hard sphere gases. The name implies that it was first applied to molecular scale problems before being applied to granular materials by ?. Discrete Element Method simulations are identical to the real experiments, this involve 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). After equilibration, the actual analysis is performed.

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  $i$  ( $i=1, \dots, N$ ):

$$\begin{aligned} \frac{\partial^2 \vec{r}_i}{\partial t^2} &= \frac{1}{m_i} \vec{\mathbf{F}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j) \\ \frac{\partial^2 \vec{\varphi}_i}{\partial t^2} &= \frac{1}{J_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j), (j = 1, \dots, N) \end{aligned}$$

The force  $\vec{\mathbf{F}}_i$  and the torque  $\vec{\mathbf{M}}_i$ , which act on grain  $i$  of mass  $m_i$  and the tensorial moment of inertia  $\hat{J}_i$  are (sometimes complicated) 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  $\varphi_i$  and the moment of inertia reduces to a scalar value  $J_i$ . The Newton's equation of motion can be written as

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

$$\frac{\partial^2 \vec{\varphi}_i}{\partial t^2} = \frac{1}{J_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\phi}_j, \vec{\omega}_j), (j = 1, \dots, N). \quad 3 \\ 4 \\ 5$$

For granular 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 sum of the pairwise interaction of grain  $i$  with all other grains of the system

$$\vec{\mathbf{F}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{F}}_{ij}, \quad \vec{\mathbf{M}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{M}}_{ij} \quad (3.29) \quad 6 \\ 7 \\ 8$$

The limitation to pairwise interaction is an abstraction, which is justified if the grains 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 method is general and 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 to static assemblies. The method treats both the conditions in exactly the same way, it is not necessary to divide the system and then treat each condition differently. The simplest model for granular grain is a sphere. In a two-dimensions 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 simple way (?).

### 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

$$\mathbf{F}_i = \sum_{j \neq i} \mathbf{F}_{ij} + \mathbf{F}_{ext,i}, \quad (3.30) \quad 23 \\ 24 \\ 25$$

where  $\mathbf{F}_i$  is the force exerted by 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.

Let us consider two grains  $i$  and  $j$ , in contact (see figure 3.13). The contact force can be decomposed into two components, as the normal ( $F_n$ ) and the tangential ( $F_t$ ) components

$$\mathbf{F}_{ij} = F_n \mathbf{n} + F_t \mathbf{t}. \quad (3.31)$$

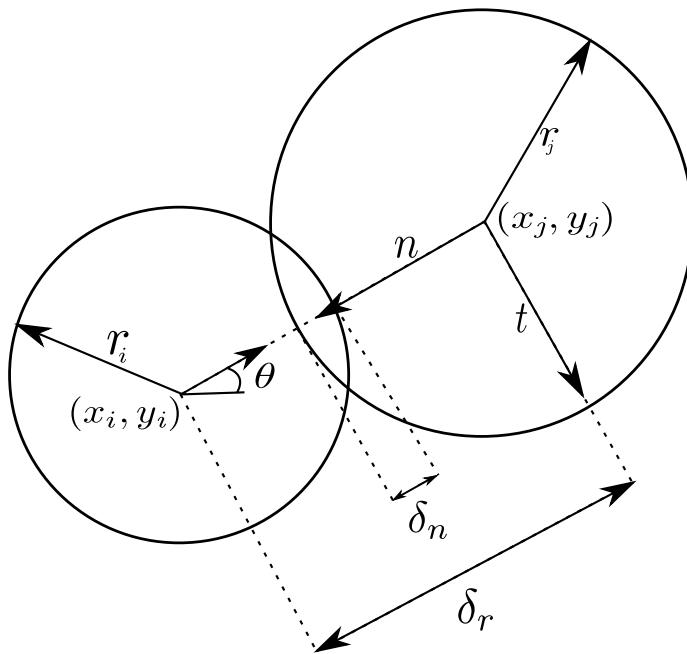


Figure 3.13 Grains  $i$  and  $j$  in contact, and the separation  $\delta_n$  is used to calculate the normal force

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 are discussed.

### Normal force

When granular 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. 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 an 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 ( $\delta_n$ ) of the grains  $i$  and  $j$  is defined as

$$\delta_n = |x_i - x_j| - r_i - r_j \quad (3.32)$$

where  $x_i$  and  $x_j$  are the centres of the grains and  $r_i$  and  $r_j$  are their radii (see Figure 3.13). 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. The simplest model is to consider the contact as a linear spring with damping. The repulsive force depends linearly on  $\delta_n$ , 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).

$$F_n = \begin{cases} 0, & \delta_n > 0 \\ -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, & \delta_n < 0 \end{cases} \quad (3.33)$$

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

$$\hat{F}_n = -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, \quad F_n = \begin{cases} 0, & \hat{F}_n \leq 0 \\ \hat{F}_n, & \hat{F}_n > 0 \end{cases} \quad (3.34)$$

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  $\epsilon$ . This factor is the *coefficient of restitution*, and is defined as  $\epsilon \approx g'/g$ , where  $g$  is the absolute normal relative velocity before the collision and  $g'$  corresponds to the post-collision value. The relative velocity,  $d\delta_n/dt > 0$ , can be obtained by differentiating eq. 3.32. Thus we obtain

$$\frac{d\delta_n}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n}, \quad (3.35)$$

where  $\mathbf{v}_i = dx_i/dt$  is the velocity of grain  $i$  and  $\mathbf{v}_j = dx_j/dt$  is the velocity of grain  $j$ . The numerical integration of eq. 3.35 yields the separation  $\delta_n$  and permits us to generalise the model and treat the tangential forces as well, as explained in the next section. By integrating Newton's equation of motion it is found that the linear force corresponds to the co-efficient of restitution,

<sup>1</sup> which is defined as

$$\varepsilon = \exp\left(-\frac{\pi\gamma_n}{2m^{eff}}/\sqrt{\frac{Y}{m^{eff}} - \frac{\gamma^n}{2m^{eff}}^2}\right). \quad (3.36)$$

### <sup>3</sup> Tangential force

<sup>4</sup> Granular grains are not perfect spheres, but have a complicated surface texture, therefore at  
<sup>5</sup> oblique collisions, besides the normal force there is a tangential force too. Even perfectly  
<sup>6</sup> smooth spheres exert a tangential force due to their bulk viscosity (?). To build a heap of  
<sup>7</sup> spheres on a flat surface, the grains as well as the surface has to be sufficiently rough, indicating  
<sup>8</sup> the dependence of the tangential force on the surface properties of the granular materials. For  
<sup>9</sup> realistic simulation of granular materials, it is important to consider the tangential force in  
<sup>10</sup> Discrete Element Method. The tangential force is considered in a similar fashion as the normal  
<sup>11</sup> force, arising from a spring stretched by the relative motion of the grain. Tangential forces are  
<sup>12</sup> modelled by considering the relevant relative tangential velocity of the grain surfaces at the  
<sup>13</sup> point of contact. The point of contact is an approximation, as the description of the normal  
<sup>14</sup> force assumes a compression  $\delta_n$ , which implies a contact surface in 3-D or a contact line in 2-D.  
<sup>15</sup> Assuming a tangential spring of length  $\delta_t$  exerts an opposing force to the relative tangential  
<sup>16</sup> displacements (ignoring the effect of relative rolling between the grains), the tangential force  
<sup>17</sup> can be postulated similar to the normal force ([eq. 3.35](#)) as:

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_i) \cdot \mathbf{t} \quad (3.37)$$

<sup>19</sup> This equation must also be numerically integrated, just like [eq. 3.28](#). The grains are in contact  
<sup>20</sup> when  $\delta_t < 0$ , and when  $\delta_t = 0$ , the grains no longer exert a force on each other. With these  
<sup>21</sup> assumptions,  $\delta_t$  can be calculated similar to the normal force. The tangential force is assumed  
<sup>22</sup> to be governed by Coulomb's friction law.

$$|F_t| \leq \mu F_n \quad (3.38)$$

<sup>24</sup> where  $F_t$  is the tangential force and  $\mu$  is the friction coefficient. It is therefore necessary to  
<sup>25</sup> constrain the tangential force to remain less than or equal to  $\mu F_N$ . To impose the condition  
<sup>26</sup> in [eq. 3.38](#), two-stages similar to the normal force computation is adopted. The first step is to  
<sup>27</sup> evaluate the candidate force, and is then accepted if it obeys the condition in [eq. 3.38](#).

$$\hat{F}_t = -k_t \delta_t - \gamma_t \frac{d\delta_t}{dt}, \quad F_t = \begin{cases} sgn(F_t), & |\hat{F}| \geq \mu F_n \\ \hat{F}_t, & |\hat{F}| < \mu F_n \end{cases} \quad (3.39)$$

where  $k_t$  is the stiffness of the tangential spring and  $\gamma$  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. 3.34) and the tangential force ( eq. 3.39) are handled in the same way in Discrete Element Method. When the grains slide against each other, they do not retain any memory of their initial position, and hence do not return to its original position. In order to model this behaviour, a limiting value of  $\delta_t$  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 about their centre of mass due to the tangential force acting at their contact surface. In this 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  $v_i + r_i(\boldsymbol{\omega} \times \mathbf{n})$ . If it belongs to grain  $j$ , its velocity is  $v_j + r_j(\boldsymbol{\omega} \times \mathbf{n})$ . The relative velocity is the difference between these two velocities.

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{t} - (r_i \boldsymbol{\omega}_i + r_j \boldsymbol{\omega}_j) \times \mathbf{n} \quad (3.40)$$

It should be noted that the eq. 3.40 is only an approximation, as the grains in Molecular Dynamics 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. It implies that the tangential force is accompanied by torque acting on two grains. If the overlap is zero, these torques are:

$$\tau_{ij} = -(a_i \mathbf{n}) \times (F_t \mathbf{t}), \tau_{ji} = -(a_j \mathbf{n}) \times (F_t \mathbf{t}), \quad (3.41)$$

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:

$$I_j \frac{d\boldsymbol{\omega}_i}{dt} = \sum_{j \neq i} \tau_{ij} \quad (3.42)$$

where  $I_j$  is the moment of inertia of grain  $j$ . The eq. 3.41 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). The interaction between two solid bodies is much more complex than that is 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

1 packing, a non-linear law can have significant influence on the acoustic properties (?), and on  
2 the global stiffness (?). However, in case of rapid granular flows, the interaction force between  
3 the grains has almost no effect on the phenomenon, and a linear law can be used to describe  
4 this kind of behaviour ([Radjai and Dubois, 2011](#)).

5 **3.5.2 Numerical algorithm and integration scheme**

6 The efficiency of a Discrete Element Method program is mainly determined by its efficiency to  
7 compute the interaction forces between grains. If we consider a model system with pairwise  
8 interactions, we have to consider the contribution of the force on grain  $i$  due to all its neighbours.  
9 If we consider only the interaction between a grain and the nearest image of another grain,  
10 then for a system of  $N$  grains, we must evaluate  $N \times (N - 1)/2$  pair distances. Consider a  
11 system of 1000 grains, at every time step all possible pairs of grains have to be considered to  
12 compute the interaction forces, hence,  $N(N - 1)/2 \approx 500,000$  force computations are required.  
13 For short-range grain interactions, the majority of these force evaluations is unnecessary as  
14 the corresponding grains are located far apart and do not necessarily touch each other. For a  
15 dense system of equally sized grains, the grains can have contacts with not more than 6 grains,  
16 this reduces the number of force computation required to  $3N \approx 3000$ . In the preliminary force  
17 computation scheme, at least 166 times more pair interactions are considered than necessary.  
18 Therefore, the numerical methods employed in the Discrete Element Method program should  
19 try to minimize the computation of interaction forces (?). There are three different methods  
20 for the efficient computation of the forces, the *Verlet* algorithm, the *link-cell* algorithm, and a  
21 *lattice* algorithm. The *Verlet* algorithm described in [Grubmuller et al. \(1991\)](#) is implemented in  
22 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 a 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 3.14. In this method a second cut-off radius  $r_v > r_c$  is introduced, and before we calculate the interactions, 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: 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 as neighbours if the distance of their surface is smaller than a predefined distance *Verlet distance*:

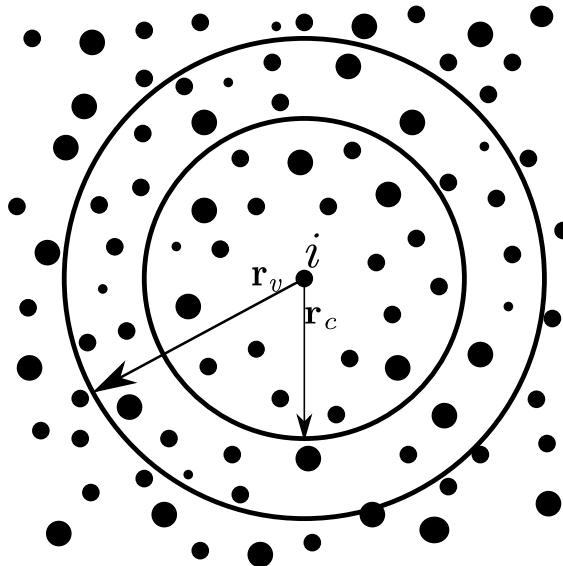


Figure 3.14 The Verlet list: a grain  $i$  interacts with those grains with the cut-off radius  $r_c$ , the Verlet list contains all the grains within a sphere with radius  $r_v > r_c$

$$(|\vec{r}_i - \vec{r}_j| - R_i - R_j) < \text{Verlet distance} \quad (3.43)$$

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

<sup>1</sup> pairs of grains (?). Redundancy in Verlet lists, i.e. if grain  $i$  is a neighbour of  $j$ , then grain  $j$   
<sup>2</sup> is a neighbour of  $i$ , are avoided by imposing a restriction on the list of grain  $i$  contains, such  
<sup>3</sup> that it contains only neighbours with index  $j < i$ . For the computation of interaction forces, the  
<sup>4</sup> Verlet list of grain  $i$  is scanned and only pairs which are recorded in one of the Verlet lists are  
<sup>5</sup> considered. Hence, the Verlet list of each grain  $i$  is scanned and the interaction force of  $i$  with  
<sup>6</sup> each entry  $j$  in its list is computed. Initially to build the Verlet list, the grains are sorted into  
<sup>7</sup> a grid of mesh size  $dx \times dy$ . For each grid there is a list of grains residing in the cell. During  
<sup>8</sup> the simulation, the neighbourhood relation among the grains change, therefore, the Verlet lists  
<sup>9</sup> have to be updated. The decision to update a Verlet list depends on how far the grains have  
<sup>10</sup> travelled since the time when the present list was built. The Verlet list of a grain  $i$  must contain  
<sup>11</sup> at any time all neighbours  $j$  with  $j < i$ . This assures that two grains  $i$  and  $j$  never touch and are  
<sup>12</sup> not considered as neighbours, i.e.  $j$  is not in the list of  $i$  and  $i$  is not in the list of  $j$ . Hence,

$$\left| \vec{r}_i - \vec{r}_j \right| - R_i - R_j > 0 \quad (3.44)$$

<sup>14</sup> The above condition is required for all pairs  $(i,j)$  of grains which are *not* known as neighbours.  
<sup>15</sup> This condition is a criterion to update the Verlet lists (?). Assume at the instant when the  
<sup>16</sup> Verlet lists are constructed, the surfaces of the grains have the distance  $\left| \vec{r}_i - \vec{r}_j \right| - R_i - R_j >$   
<sup>17</sup> Verlet distance, i.e. they are not classified as neighbours. If the Verlet lists are updated  
<sup>18</sup> before one of these grains has travelled the distance *verletdistance*/2 since the lists were  
<sup>19</sup> constructed, they can never collide without being recognized as neighbours first. This is  
<sup>20</sup> explained in figure 3.15. The impact of optimisation of the Verlet list algorithm has negligible  
<sup>21</sup> effect on the computation time, as the algorithm is quite efficient already and only consumes a  
<sup>22</sup> few percent of the total computation time in construction of the Verlet lists. The implementation  
<sup>23</sup> of the Verlet list algorithm in force computation drastically reduces the computation time in  
<sup>24</sup> comparison to the linear algorithm. The performance of the Verlet list algorithm is controlled  
<sup>25</sup> by two crucial parameters: the number of cells  $N_c$ , for the construction of the Verlet lists and  
<sup>26</sup> the Verlet distance  $r_v$  (?).

## <sup>27</sup> Leap frog or Verlet integration algorithm

<sup>28</sup> Discrete Element Method involves numerically solving Newton's equation of motion eq. 3.28,  
<sup>29</sup> which is an ordinary differential equation. Choosing an integration algorithm is important, as  
<sup>30</sup> the forces are not always differentiable in time, and the temporary derivative of the force is  
<sup>31</sup> discontinuous when the contact splits. It is also very essential to numerically integrate eq. 3.40  
<sup>32</sup> with the same precision as eq. 3.28. At first, computational speed seems important. It is usually  
<sup>33</sup> not very relevant because the fraction of time spent on integrating the equation of motions

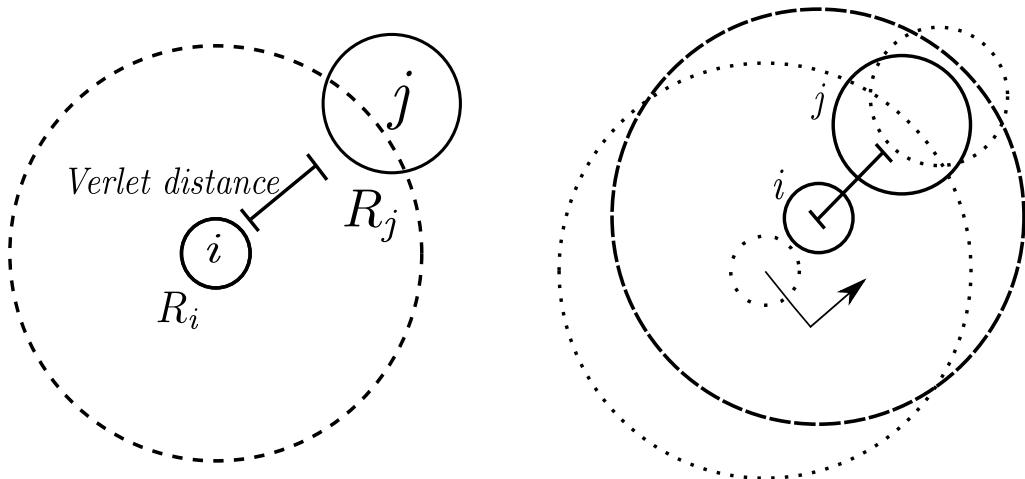


Figure 3.15 Checking the validity of Verlet lists. *Left:* the grains  $i$  and  $j$  are not recognized as neighbours since the distance of their surface is larger than the Verlet distance. The radius of the dashed circle is  $R_i + R_{max} + \text{Verlet distance}$ . *Right:* in the most unfortunate 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. Figure redrawn after ?.

(as opposed to computing the interactions) is small. Accuracy for large time steps is more important, because the larger the time step that we can use, the fewer evaluation of the forces are needed per unit of simulation time. Hence, this would suggest that 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 this 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 time and long time. The sophisticated higher-order algorithms tend to have very good energy conservation at short times. However, they often have undesirable feature that results 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 (?). In this case, such algorithms are not useful. They are more complicated to program, and do not yield a more precise solution (?). It might seem 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, would always diverge the predicted trajectory exponentially from the true trajectory. This phenomenon is called the Lyapunov instability, and it is devastating to the whole idea of Discrete Element Method simulation, but we have good reasons to assume that even this problem need not be serious (?). First of all, one should realize that the aim

of the Discrete Element Method simulation is not to predict precisely what will happen to a system, but to predict the average behaviour of the system that was prepared in an initial state about which we know something (initial position, velocity and energy), but not everything. Hence, Molecular Dynamics technique differs from other methods, which are used to predict the trajectories. However, considerable numerical evidence suggest that the shadow orbits exists, 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 (?).

Newton's equations of motion are time reversible, and so should be the integration algorithm. 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 was implemented in Discrete Element Method simulation by [Verlet \(1967\)](#). 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 fair (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 (?). Verlet algorithm is simply based on a truncated Taylor expansion of grain co-ordinates.

$$t(t + \Delta t) = rt + \mathbf{v}(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots \quad (3.45)$$

If we truncate this expansion beyond the term  $\Delta t^2$ , we obtain the Euler's algorithm, which looks similar to the Verlet Algorithm, but it does not preserve energy and have 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  $\Delta t$  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:

$$\mathbf{v}_{k+1/2} = \mathbf{v}_{k-1/2} + \mathbf{a}_k \Delta t, \quad x_{k+1} = x_k + \mathbf{v}_{k+1/2} \Delta t \quad (3.46)$$

This algorithm determines the new grain position with an error of order  $O(\Delta t^4)$ . But [eq. 3.46](#) hides a difficulty in the application of this algorithm to granular materials ([Radjai and Dubois, 2011](#)). The problem is that the acceleration must be calculated at time  $t = k\Delta t$ . But the velocities are known at  $t = (k - 1/2)\Delta t$ , and not at  $t = k\Delta t$ . One way to get around this problem is to

write:

$$\mathbf{v}_k = \mathbf{v}_{k-1/2} + \mathbf{a}_{k-1}\Delta t / 2 \quad (3.47)$$

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. 3.47 estimates  $\mathbf{v}_k$  with an error of order  $\mathbf{O}(\Delta t^2)$ , which produces an error of the same order in the calculation of the force in eq. 3.46. But this causes only an error of order  $\mathbf{O}(\Delta t^3)$  in the velocity and an error of order  $\mathbf{O}(\Delta t^4)$  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 and not the integration of equations. The Verlet integration scheme is summarized in eq. 3.48 and figure 3.16. To calculate the forces and acceleration, it requires the positions and velocities at time t:

$$\mathbf{v}(t + \Delta t / 2) = \mathbf{v}(t - \Delta t / 2) + \mathbf{a}(t)\Delta t \quad (12)$$

$$x(t + \Delta t) = x(t) + \mathbf{v}(t + \Delta t / 2)\Delta t \quad (13)$$

$$\mathbf{v}(t) = \mathbf{v}(t - \Delta t / 2) + \mathbf{a}(t - \Delta t)\Delta t / 2 \quad (3.48) \quad (14)$$

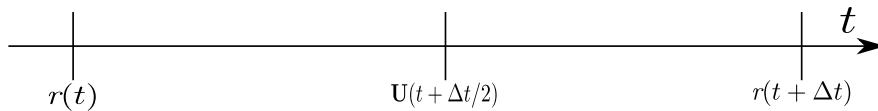


Figure 3.16 Verlet integration scheme

The analysis of the Discrete Element Method formulation reveals that the linear force law gives the model a harmonic character, showing that it is very closely related to simple models widely used in physics and mechanics. The shortest time scales often arise from the oscillations of one or 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  $\omega_N$ , the characteristic oscillation frequency of very short waves. This frequency is proportional to  $\omega_0$ , which is easier to estimate. Therefore, it is essential to choose a time step  $\Delta t \approx \epsilon / \omega_0$ , where  $\epsilon$  is a constant that depends on the integration algorithm. Values such as  $\epsilon \approx 0.01$  are often a reasonable choice (?). 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.

### **1 3.5.3 Boundary conditions**

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

21 **Periodic boundary**

22 The effect of a wall on the response of grains is very critical, especially in numerical simulation  
23 where the number of grains is relatively fewer in comparison to the experiments. The undesired  
24 effect of a wall can be eliminated using periodic boundary conditions, i.e. a periodic extension  
25 of the simulation area in one or more dimensions. Any grain leaving the system at one side is  
26 reintroduced at the opposite side, and correspondingly the interaction forces between grains at  
27 opposite sides of the simulation area are taken into account. In this framework, the simulation  
28 domain becomes a unit area containing grains with periodic copies paving the whole system.  
29 The periodic boundary conditions extend the system boundaries to infinity, so that the simulation  
30 cell simply plays the role of a coordinate system to locate grain positions [3.17](#).

31 The external stresses or displacements are applied on the simulation box by constraining  
32 the degrees of freedom of the wall, which are alternatively kept free or fixed depending on  
33 whether a stress or a displacement is monitored in a system. With periodic boundary conditions,  
34 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 (?). 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 Discrete Element Method are discussed in detail in [Radjai et al. \(2011\)](#). The periodicity in position implemented in the present study is discussed below.

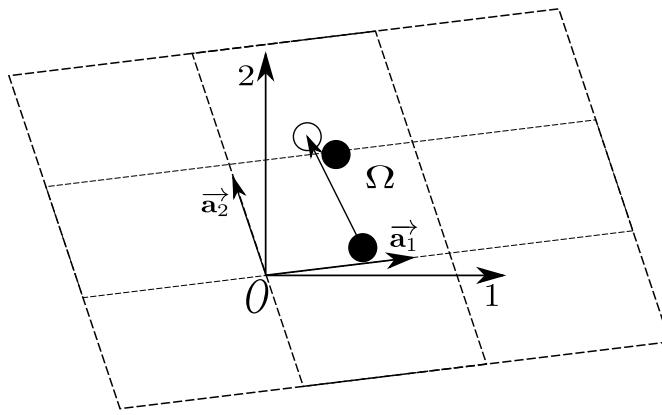


Figure 3.17 A 2D simulation cell  $\omega$  with its basis vectors in an absolute frame. A grain located at the right boundary interacts with the image of another grain located at the left boundary.

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 3.17. 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:

$$\vec{r}(i') = \vec{r}(i) + \sum_{k=1}^2 i_k \vec{a}_k \quad (3.49)$$

The grains belonging to the cell  $\Omega$ , 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  $\Omega$ . Hence, whenever a grain  $i$  leaves the simulation cell, it becomes an image of  $i'$ , which

<sup>1</sup> then becomes the original. This means that a grain crossing a border of the simulation cell,  
<sup>2</sup> returns to the cell by crossing another border.

### <sup>3</sup> 3.5.4 Particle Assembling Methods

<sup>4</sup> In order to simulate a granular assembly, it is essential to assign an initial position and velocity to  
<sup>5</sup> all the grains in the system. Particle positions should be chosen to be compatible to the structure  
<sup>6</sup> (granular fabric) we are trying to simulate. In any event, the grains should not be positioned  
<sup>7</sup> such that there is an appreciable overlap between grains. In order to achieve the initial position  
<sup>8</sup> of the grains, various grain-assembling methods can be adopted. The grain assembling methods  
<sup>9</sup> can be classified into two broad categories: dynamic methods and geometrical approaches. The  
<sup>10</sup> dynamic approach involves packing of grains using laws of mechanics and contacts, while  
<sup>11</sup> in the geometrical method the grains are packed considering their geometry, i.e. grain size,  
<sup>12</sup> shape and its position. In general, the packing of grains can be categorized into two types:  
<sup>13</sup> crystal/lattice packing, like hexagonal or square pattern of mono-disperse grains, and random  
<sup>14</sup> packing with varying density employing mono-disperse or poly-disperse grains. The crystalline  
<sup>15</sup> packing arrangements, such as hexagon and square lattices, are easier to generate, however  
<sup>16</sup> they have non-trivial effects on the response of the granular system ([Staron et al., 2005](#)).  
<sup>17</sup> Hexagonal packing is the densest possible arrangement for mono-dispersed spherical grains. In  
<sup>18</sup> 2D, the packing of mono-dispersed circles on a hexagonal lattice yields a packing density of  
<sup>19</sup>  $\eta_h = \frac{1}{6}\pi\sqrt{3} \approx 0.9068$

<sup>20</sup> The rheology of a granular material is controlled by the geometry of the assembly, which  
<sup>21</sup> includes the grain shape, size distribution, and their arrangement. This prevailing role of  
<sup>22</sup> geometry sometimes permits to simplify the dynamics in favour of a better description of the  
<sup>23</sup> geometry and/or higher numerical efficiency ([Radjai and Dubois, 2011](#)). For example, dense  
<sup>24</sup> granular packing may be efficiently constructed by replacing the equations of dynamics by  
<sup>25</sup> simple displacement rules satisfying the geometrical constraints. Purely geometrical procedures  
<sup>26</sup> can be much simpler and numerically faster than dynamic or quasi-static methods. Contrary  
<sup>27</sup> to dynamic simulation methods, the geometrical methods allow for quick assembling of a  
<sup>28</sup> large number of grains. Such packing may then be used as the initial state for dynamic  
<sup>29</sup> simulations. The issue of the assembling methods is to construct configurations of grains as  
<sup>30</sup> close as possible to a state of mechanical equilibrium with built-in packing properties. This can  
<sup>31</sup> be a target packing density for a given grain size distribution. In the same way, the average  
<sup>32</sup> connectivity of the grains (coordination number) and the anisotropy of the contact network are  
<sup>33</sup> basic geometrical properties. The coordination number represents the mechanical response of  
<sup>34</sup> 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. This distribution can be obtained, if the radii are chosen according to the probability distribution:

$$p(R) = \frac{R_{min}R_{max}}{R_{max} - R_{min}} \frac{1}{R^2} \quad (3.50)$$

Random numbers according to the above distribution can be generated from equi-distributed random numbers  $z \in [0, 1]$  via the transformation

$$R = \frac{R_{min}R_{max}}{R_{max} - z(R_{max} - R_{min})} \quad (3.51)$$

This transformation is applied to initialise the grain radii and the grains are arranged randomly on a regular lattice. The configuration of grains obtained after this step is presented in ??.  
In the second step, the grains arranged in a regular lattice are allowed to fall down and are packed using the *random deposition with relaxation method*, a ballistic deposition technique. The geometrical methods help in this way to improve numerical efficiency in the preparation phase. For example, gravitational deposition of grains located initially on a regular grid can require hours of computation whereas a nearly similar result may be obtained by means of a geometrical method in only a few minutes. The drawback is that the resulting sample will not be in mechanical equilibrium and no information is available on the contact forces. Nevertheless, depending on the relaxation rule, the sample may still be sufficiently close to equilibrium to be considered as a good starting point for mechanical simulations. Hence, a combined approach of ballistic deposition and Discrete Element Method is adopted in the present study to generate mechanically stable samples. The random deposition and relaxation method, first proposed by [Vold \(1959\)](#); ? and developed by [Jullien et al. \(1992\)](#) and [Meakin and Jullien \(1985\)](#), is adopted in the present study. The general principle of this method is quite simple (see figure 3.18); it consists of placing the grains consecutively on a substrate or a layer of already deposited grains. Each grain first touches the substrate or a deposited grain, then undergoes a relaxation process (single-grain restructuring) along the steepest descent (steepest-descent model) until a more stable position according to a stability criterion is reached. The

1 construction of the packing proceeds layer by layer from the substrate, hence this deposition  
 2 model is also known as bottom-to-top restructuring model. The first step is to release a grain  
 3 from a random position above the substrate. Upon contact with the first deposited grain, the  
 4 grain rolls following the steepest descent until a new contact is formed with a second grain. In  
 5 2D, two contacts are sufficient to balance a grain if its centre of gravity lies between the two  
 6 contacts. This corresponds to a position of local stable equilibrium. If this criterion is not met,  
 7 the grain continues to roll and the procedure is iterated until a local stable position is reached.  
 8 The wall effects are eliminated by adopting periodic boundaries (Sec 3.5.3) in the horizontal  
 9 direction (perpendicular to that of deposition). figure 3.18 shows a small sample prepared by  
 10 this method (the grey grains are periodic images of the black ones). In this method, the order of  
 11 deposited grains is generally random and independent of their sizes. The mechanically stable  
 12 sample obtained from equilibrating the random deposited and relaxed sample is presented in ???

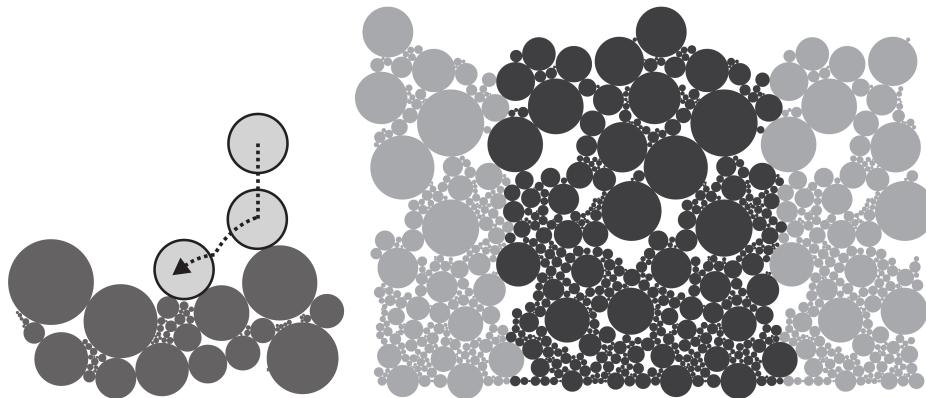


Figure 3.18 (a)*Ballistic deposition*: first contact followed by steepest descent; (b) small-scale periodic sample ([Radjai and Dubois, 2011](#))

13

14 **3.5.5 Voronoi Tessellation**

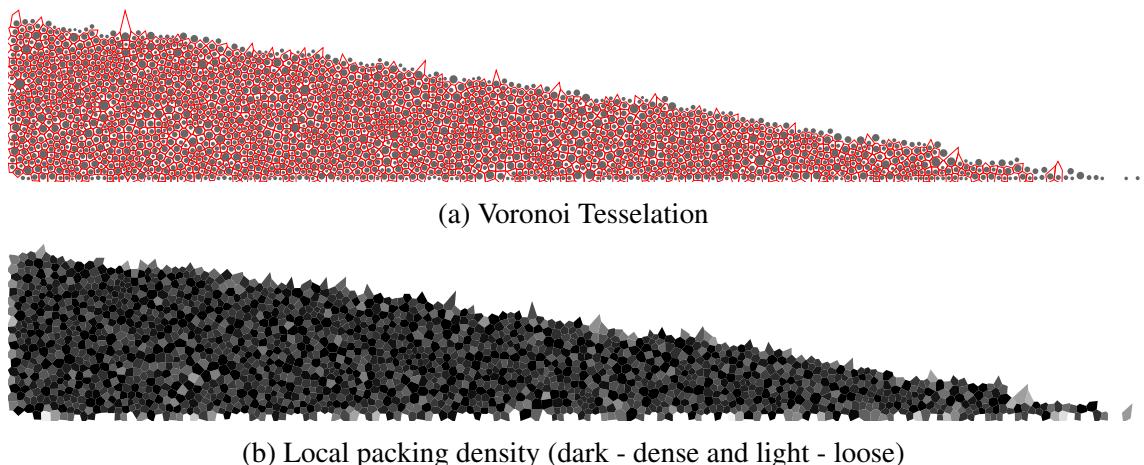


Figure 3.19 Voronoi tessellation to average bulk properties

Draft - v1.0

Saturday 27<sup>th</sup> September, 2014 – 21:06

# References

- Abe, K., Soga, K., and Bandara, S. (2013). Material Point Method for Coupled Hydromechanical Problems. *Journal of Geotechnical and Geoenvironmental Engineering*, page 04013033. 1  
2  
3  
4
- Andersen, S. and Andersen, L. (2010). Modelling of landslides with the material-point method. *Computational Geosciences*, 14(1):137–147. 5  
6
- 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. 7  
8  
9
- Augarde, C. and Heaney, C. (2009). The use of meshless methods in geotechnics. In *Proceedings of the 1st International Symposium on Computational Geomechanics*, France. 10  
11
- Bandara, S. (2013). *Material Point Method to simulate Large Deformation Problems in Fluid-saturated Granular Medium*. PhD thesis, University of Cambridge. 12  
13
- Bardenhagen, S. (2002). Energy Conservation Error in the Material Point Method for Solid Mechanics. *Journal of Computational Physics*, 180:383–403. 14  
15
- Bardenhagen, S. and Kober, E. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496. 16  
17
- 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. 18  
19  
20
- 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*, volume 2. 21  
22  
23
- Belytschko, T., Lu, Y. Y., and Gu, L. (1994). Element-free Galerkin Method. *International Journal for Numerical Methods in Engineering*. 24  
25
- 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*, pages n/a–n/a. 26  
27  
28
- 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. 29  
30

- <sup>1</sup> Bonet, J. and Kulasegaram, S. (2000). Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International journal for numerical ...*
- <sup>4</sup> Chen, Z. and Brannon, R. (2002). An evaluation of the material point method. *Sandia National Laboratories (SAND2002-0482)*.
- <sup>6</sup> 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.
- <sup>9</sup> 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 ...*
- <sup>12</sup> 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.
- <sup>15</sup> 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.
- <sup>17</sup> Guilkey, J., Harman, T., Xia, A., Kashiwa, B., and McMurtry, P. (2003). An Eulerian–Lagrangian approach for large deformation fluid structure interaction problems, Part 1: algorithm development. *Advances in Fluid Mechanics*, 36:143–156.
- <sup>20</sup> 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*.
- <sup>23</sup> Gumhold, S. (2003). Splatting illuminated ellipsoids with depth correction. volume 2003, pages 245–252.
- <sup>25</sup> Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Computer Methods in Physics*, 3:319–343.
- <sup>27</sup> Hertz, H. (1882). Über die Beruhrung fester elastischer Körper. *ournal fur die Reinen und Angewandte Mathematik*, 92:156–171.
- <sup>29</sup> Jaeger, H., Nagel, S., and Behringer, R. (1996). Granular solids, liquids, and gases. *Reviews of Modern Physics*, 68(4):1259–1273.
- <sup>31</sup> Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):235–257.
- <sup>33</sup> 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.
- <sup>35</sup> Jullien, R., Meakin, P., and Pavlovitch, A. (1992). Random packings of spheres built with sequential models. *Journal of Applied Physics*, 25:4103.
- <sup>37</sup> 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. 1  
2  
3
- Krogh, M., Painter, J., and Hansen, C. (1997). Parallel sphere rendering. *Parallel Computing*, 23(7):961–974. 4  
5
- Kuester, F., Bruckschen, R., Hamann, B., and Joy, K. (2001). Visualization of particle traces in virtual environments. pages 151–157. 6  
7
- Levoy, M. (1988). Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37. 8  
9
- Li, S. and Liu, W. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55:1. 10  
11
- Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169. 12  
13
- 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. 14  
15  
16
- Luding, S., Clément, E., and Blumen, A. (1994). Anomalous energy dissipation in molecular-dynamics simulations of grains: The "detachment" effect. *Physical Review*, 50:4113. 17  
18
- 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. 19  
20  
21
- 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. 22  
23  
24
- Maeda, K. and Sakai, H. (2010). Seepage failure and erosion of ground with air bubble dynamics. *Geoenvironmental Engineering and ...*. 25  
26
- Mast, C. M., Arduino, P., Mackenzie-Helnwein, P., and Miller, G. R. (2014a). Simulating granular column collapse using the Material Point Method. *Acta Geotechnica*. 27  
28
- 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. 29  
30  
31
- Meakin, P. and Jullien, R. (1985). Structural readjustment effects in cluster-cluster aggregation. *Journal de Physique*, 46(9):1543–1552. 32  
33
- Mehta, A. and Barker, G. (1994). The dynamics of sand. *Reports on Progress in Physics*, 57:383. 34  
35
- Monaghan, J. (2005). Smoothed particle hydrodynamics. *Reports on progress in physics*. 36

- <sup>1</sup> Mori, H. (2008). *The SPH method to simulate river levee failures*. Msc thesis, University of Cambridge.
- <sup>3</sup> Qiu, G., Henke, S., and Grabe, J. (2011). Application of a Coupled Eulerian–Lagrangian approach on geomechanical problems involving large deformations. *Computers and Geotechnics*.
- <sup>6</sup> Radjai, F. and Dubois, F. (2011). *Discrete-element modeling of granular materials*. ISTE Wiley, London; Hoboken, N.J.
- <sup>8</sup> Radjai, F. and Richefeu, V. (2009). Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41(6):715–728.
- <sup>10</sup> Radjai, F., Voivret, C., and McNamara, S. (2011). Discrete-element modelling of granular materials. chapter Periodic b, pages 181–198. ISTE Wiley.
- <sup>12</sup> 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.
- <sup>14</sup> 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.
- <sup>16</sup> Shin, W. (2010). *Numerical simulation of landslides and debris flows using an enhanced material point method*. PhD thesis.
- <sup>18</sup> Staron, L., Radjai, F., and Villette, 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.
- <sup>21</sup> 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.
- <sup>24</sup> 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.
- <sup>26</sup> Sulsky, D., Zhou, S.-J., and Schreyer, H. L. (1995). *Application of a particle-in-cell method to solid mechanics*, volume 87.
- <sup>28</sup> Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159(1):98.
- <sup>30</sup> Vold, M. J. (1959). A numerical approach to the problem of sediment volume. *Journal of Colloid Science*, 14(2):168–174.
- <sup>32</sup> 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.
- <sup>35</sup> 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.

- 
- 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. 1  
2
- 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. 3  
4  
5