

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 μ .
 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.

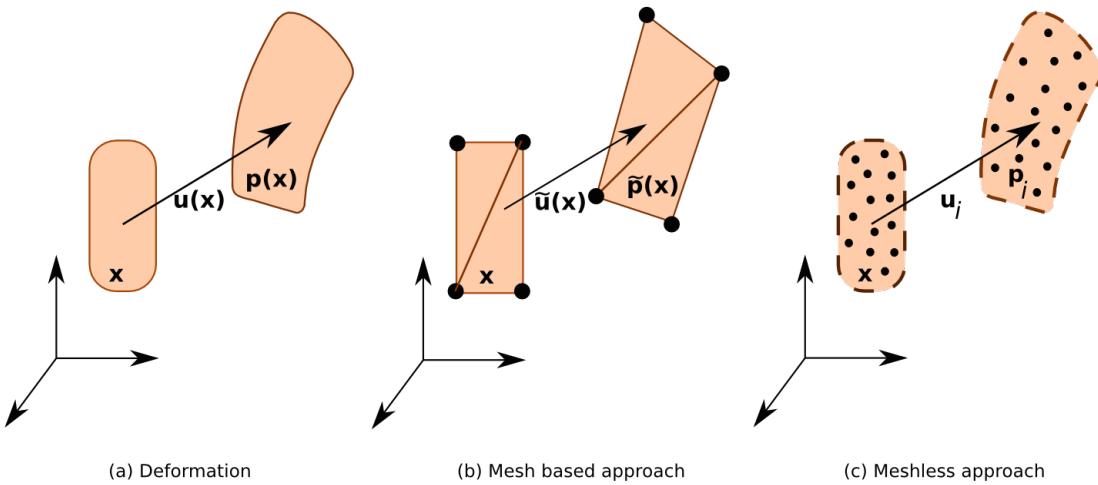


Figure 3.2 Difference between mesh-based and mesh-free techniques in modelling large deformation flow.

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),

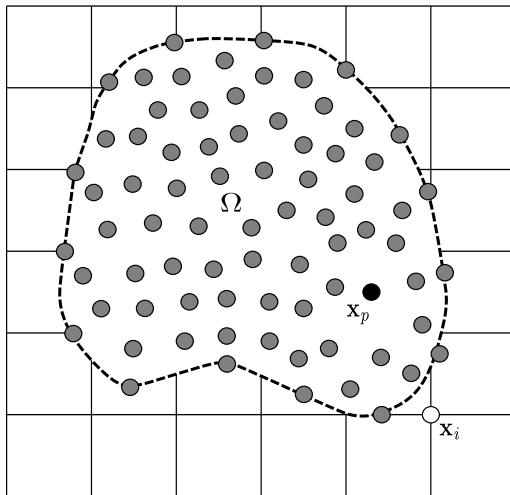


Figure 3.3 Typical discretisation of a domain in MPM. The dotted line represents the boundary of the simulated object Ω and each closed point represents a material point used to discretise Ω . The square mesh represents the background grid. Each square in the background grid is a grid cell, and grid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

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

³ 3.3 Material Point Method (MPM)

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

The grey-shaded 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). The Material Point Method involves discretising the domain Ω with a set of material points. The material points are assigned an initial value of position, velocity, mass, volume, and stress denoted as \mathbf{x}_p , v_p , m_p , \mathbf{V}_p and σ_p . Depending on the material being simulated, additional parameters, like pressure, temperature, etc., should be specified at the material points. The material points are assumed to be within the computational grid, which is assumed to be a Cartesian lattice for convenience (see figure 3.3). At every time step t_k , the MPM computation cycle involves projecting the data, such as position, mass, and velocity from the material points to the computational grid using standard nodal basis functions, called *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.

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

and

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

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 . MPM discretises the continuum body 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 ρ_p^t , velocity v_p^t , Cauchy stress tensor σ_p^t , strain ϵ_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

¹ multiplying eq. 3.2 with a test function $w(\mathbf{x}, t)$.

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

⁴ where $\boldsymbol{\sigma}^s$ is the specific stress (i.e. stress divided by mass density, $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}/\rho$), Ω is the current
⁵ configuration of the continuum, $\boldsymbol{\tau}$ is the traction. eq. 3.3 is obtained by applying the divergence
⁶ theorem, similar to the standard procedure adopted in Finite Element Methods (Chen and
⁷ Brannon, 2002; Sulsky et al., 1994, 1995). The differential volume and the surface elements
⁸ are denoted by $d\Omega$ and dS , respectively.

⁹ As the whole continuum is discretised into a finite set of material points, the mass density can
¹⁰ be written as:

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

¹¹ where δ is the Dirac delta function. Substituting eq. 3.4 in eq. 3.3, the sum of quantities of
¹² material points can be evaluated as:

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

¹³ where h is the thickness of the boundary layer. It can be noted from eq. 3.5 that the interactions
¹⁴ between different material points are reflected only through the gradient terms. In MPM, a
¹⁵ background computational mesh is used to calculate the gradient terms. The computational
¹⁶ mesh is constructed using 2-node cells for 1-D, 4-node cells for 2-D, and 8-node cells for
¹⁷ 3-D problems. These elements are used to define the standard nodal basis functions, $N_i(\mathbf{x})$,
¹⁸ associated with the spatial nodes $x_i(t), i = 1, 2, \dots, N_n$, where N_n represents the total number of
¹⁹ mesh nodes. The nodal basis functions are assembled by using the conventional finite-element
²⁰ shape functions (Chen and Brannon, 2002). The co-ordinates of any material point in a cell can
²¹ 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:

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

[eq. 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 [3.7](#) into [eq. 3.5](#), the weak form of the equation of motion reduces to:

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

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

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

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

$$\mathbf{f}_i^{int,t} = - \sum_{p=1}^{N_p} M_p \mathbf{G}_{ip}^t \cdot \boldsymbol{\sigma}_p^{s,t} \quad (3.10)$$

$$\mathbf{f}_i^{ext,t} = - \sum_{p=1}^{N_p} M_p \mathbf{b}_p^t N_i(\mathbf{x}_p^t) + \sum_{p=1}^{N_p} M_p N_i(\mathbf{x}_p^t) \tau_p^{s,t} h^{-1} \quad (3.10)$$

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 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 all the material points have been completely updated, the computational mesh is discarded, and a new mesh is defined for the next time step.

¹ 3.3.2 Boundary conditions

² The Material Point Method uses standard shape functions, similar to those used in the Finite
³ Element Methods, hence the essential and the natural boundary conditions can be applied to the
⁴ background grid nodes in the same way as in the traditional FEM. The free surface boundary
⁵ conditions are satisfied, as MPM is formulated in the weak form. Implementation of traction
⁶ boundary conditions requires a set of material points to represent the boundary layer. [Bandara](#)
⁷ ([2013](#)) proposed a friction interaction for the planar boundary condition, using Coulomb's
⁸ friction criterion. The nodal accelerations were considered to include the frictional effects
⁹ instead of the forces, as the forces are proportional to the corresponding accelerations. The
¹⁰ static and kinematic frictions are applied tangential to the nodal boundary. Friction forces
¹¹ are applied, only if the particles are in contact. The normal velocity and acceleration on the
¹² boundary plane is zero.

¹³ 3.3.3 Solution scheme

¹⁴ A step-by-step solution scheme for Material Point Method ([Bandara, 2013](#)) is described below.
¹⁵ It involves:

- ¹⁶ • A continuum body is discretised into a finite set of material points corresponding to
¹⁷ the original configuration of the body. The number of material points corresponds to
¹⁸ the resolution of the mesh size adopted in Finite Element Method. The material points
¹⁹ are followed throughout the deformation of the material, which gives a Lagrangian
²⁰ description of the motion.
- ²¹ • An arbitrary computational grid is initialized to describe the natural coordinates of the
²² material points. For the purpose of simplicity, a Cartesian grid is usually adopted.
- ²³ • The state variables (mass/density, velocity, strain, stress, other material parameter corre-
²⁴ sponding to the adopted constitutive relation) are initialized at every material point.
- ²⁵ • For each material point, the mapping of the properties from the particles to the cell nodes
²⁶ is accomplished using the shape functions computed from the particle position. The
²⁷ nodal mass matrix is obtained as:

$$\text{28} \quad m_i^k = \sum_{p=1}^{N_p} M_p N_{ip}^k \quad (3.11)$$

³⁰ where m_i^t is the mass at node i at time t , M_p the particle mass, N_i the shape function
³¹ associated with node i , and x_p^t the location of the particle at t .

- The nodal velocity is obtained by mapping the particle velocity to the nodes using the shape functions:

$$v_i^k = \sum_{p=1}^{N_p} m_p v_p^t N_{ip}^t / m_i^t \quad (3.12)$$

1
2
3
4

- Strain increment $\Delta\epsilon_p^{t+1}$ for particle is then computed as:

$$\Delta\epsilon_p^{t+1} = \frac{\Delta t}{2} \sum_{i=1}^{N_n} \mathbf{G}_{ip}^t \mathbf{v}_i^t + (\mathbf{G}_{ip}^t \mathbf{v}_i^t)^T \quad (3.13)$$

5
6
7

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

$$\Delta\sigma_p^{t+1} = \mathbf{D} : \Delta\epsilon_p^{t+1} \quad (3.14)$$

8
9
10

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

$$\begin{aligned} \sigma_p^{t+1} &= \sigma_p^t + \Delta\sigma_p^{t+1} \\ \epsilon_p^{t+1} &= \epsilon_p^t + \Delta\epsilon_p^{t+1} \end{aligned} \quad (3.15)$$

13
14
15

- The material point density is then updated as:

$$\rho_p^{t+1} = \frac{\rho_p^t}{\{1 + tr(\Delta\epsilon_p^{t+1})\}} \quad (3.16)$$

16
17
18

- Then the nodal acceleration is computed using the equation of motion as:

$$\mathbf{a}_i^t = (\mathbf{f}_i^{int,t} + \mathbf{f}_i^{ext,t}) / m_i^t \quad (3.17)$$

19

- The nodal velocity is obtained from the computed nodal acceleration as:

$$v_i^L = \mathbf{v}_i^k + a_i^k \Delta t \quad (3.18)$$

22
23
24

- Finally, the particle position and its velocity are updated according to:

$$\begin{aligned} \mathbf{x}_p^{t+1} &= \mathbf{x}_p^t + \Delta t \sum_{i=1}^{N_n} v_i^L N_{ip}^t \\ \mathbf{v}_p^{t+1} &= \mathbf{v}_p^t + \Delta t \sum_{i=1}^{N_n} a_i^t N_{ip}^t \end{aligned} \quad (3.19)$$

- At the end of every time step, all the variables on the grid nodes are initialized to zero.

⁶ figure 3.4 illustrates the steps involved in a MPM analysis.

7 Post processing

8 The final step in any analysis is post-processing. It involves visualization and extraction of
9 the data from the analysis. In mesh-less methods, like the Material Point Method, structures
10 are generally represented as points which represent a discrete region of the body. The MPM
11 facilitates representation of arbitrarily complex geometries and is robust in computing large
12 deformation problems. It has advantages over strictly grid based methods in simulations
13 involving contact between multiple objects (Bardenhagen et al., 2000). The MPM poses a
14 whole new set of visualization problem, it is essential to visualize the general configuration of
15 the body as well as to observe the finer details like development of cracks or separation of chunk
16 of material from the body. The body is discretised into conceptual material points, which carry
17 all the relevant information of the corresponding segment. The unique qualities of the MPM
18 necessitate the need to visualize the particle data in a way that is informative and appropriate.
19 In the MPM, the particle data represents the finite portion of a larger continuum, and the
20 ability to see and interpret the macroscopic structure created by these particles is vital (Bigler
21 et al., 2006). There are two vital aspects in visualizing the MPM data: (1) visualization of the
22 structure represented by material points, and (2) understanding the qualitative trend associated
23 with the material points like mass, velocity or stress. The MPM output data contains both the
24 material point and the grid data, one approach in visualizing the MPM data is by rendering the
25 interpolated particle values on grid nodes using *iso-surfacing* (Lorensen and Cline, 1987) or
26 *volume rendering* (Levoy, 1988) technique. In regions where the material points are sparse, it is
27 necessary that the grid resolution to be sufficiently fine to compensate for the missing features.
28 This results in storing large amount of unnecessary data in regions where sufficient material
29 points are present. Thus, it is advantageous to visualize the MPM data of the material points as
30 particles (Bigler et al., 2006). Particle visualization involves rendering the particles as a sphere
31 or an ellipsoid representing the size and location of the fraction of the continuum (Gumhold,

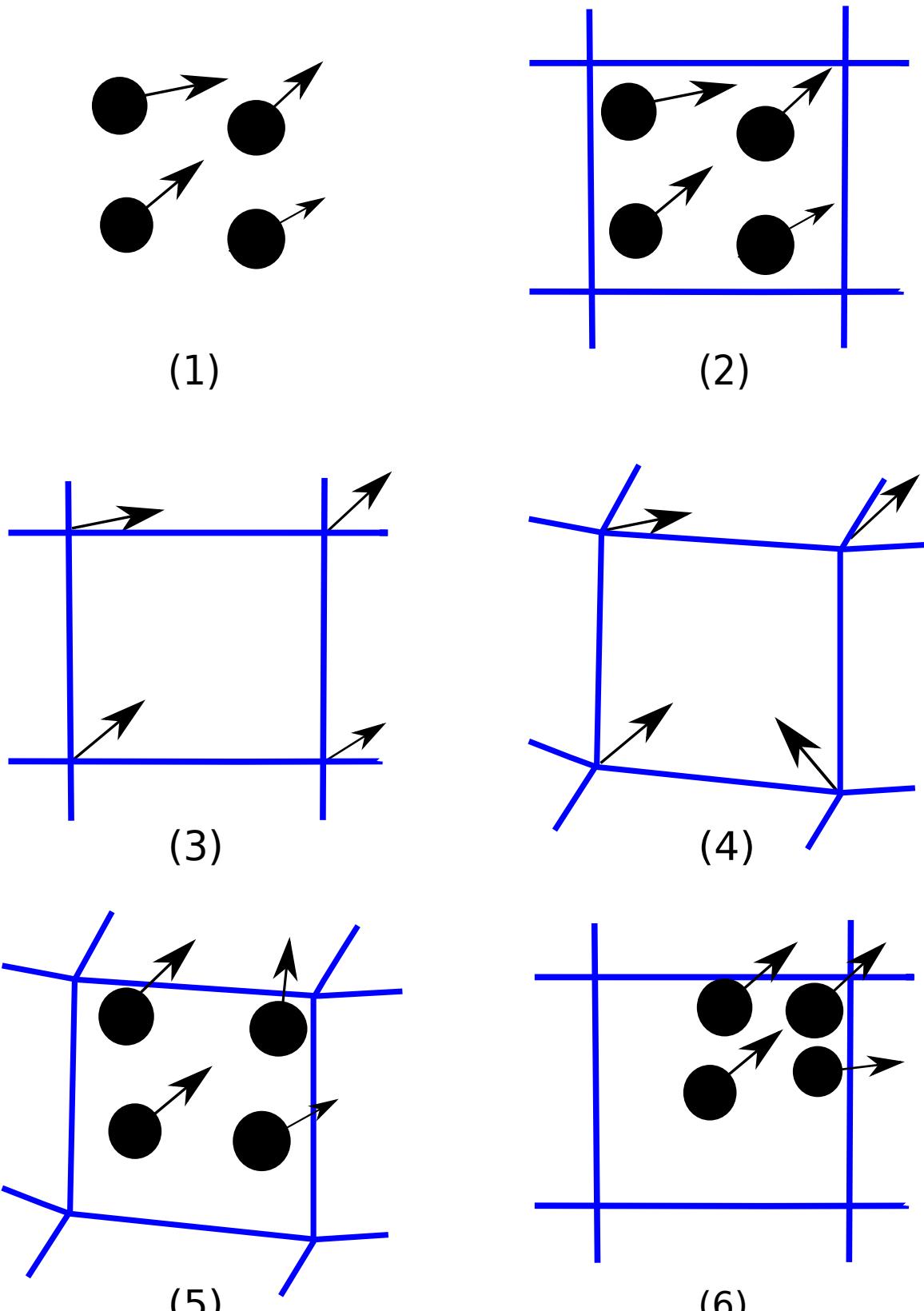


Figure 3.4 Illustration of the steps in the MPM algorithm for particles occupying a single cell in the background grid. (1). A representation of four material points (filled circles), overlaid with the computational grid (solid lines). Arrows represent displacement vectors. (2). The material point state vector (mass, volume, velocity, etc.) is projected to the nodes of the computational grid. (3). The discrete form of the equations of motion is solved on the computational grid, resulting in updated nodal velocities and positions. (4). The updated nodal kinematics are interpolated back to the material points and their state updated. (5). The computational grid is reset to its original configuration and the process repeated. Reproduced after [Bigler et al. \(2006\)](#)

¹ 2003; [Krogh et al., 1997](#); [Kuester et al., 2001](#)). Colour mapping of scalar quantities such as
² mass, velocity, or stress of a material point provide additional qualitative understanding of data.

³ The accuracy of the MPM simulations largely depends on the number of material points
⁴ representing the continuum. The MPM utilizes a grid to compute the deformation of the
⁵ continuum, hence the size of the cells affects the accuracy of the results. Generally in the MPM,
⁶ the number of particles per cell controls the accuracy of the simulation. [Guilkey et al. \(2003\)](#)
⁷ recommends higher particle density, such as 4 particles per cell, for large deformation problems.
⁸ Very low particle density will result in non-physical opening of cracks in large deformation
⁹ simulations and can be a source for the cell crossing noise. However, higher value of particle
¹⁰ density affects the computation time.

¹¹ 3.3.4 Effect of mesh size and number of material points per cell

¹² [Abe et al. \(2013\)](#) observed that for a coarse mesh, the numerical error decreases with increase in
¹³ the number of material points per cell. In contrast, they observed an opposite trend for the fine
¹⁴ meshes (0.01 m). The influence of numerical noise due to particles crossing the background
¹⁵ mesh was not observed when the mesh size is greater than 0.05 m. [Coetzee et al. \(2005\)](#) also
¹⁶ found that the numerical error decreases with increase in mesh refinement.

¹⁷ In the present study, the effect of mesh size and the number of material points per cell on
¹⁸ the run-out behaviour is investigated. For a mesh size of 0.0125 m, the number of material
¹⁹ points per cell (PPC) is varied as 4, 16, 25, 36, 64, 81 and 100. The effect of number of material
²⁰ points on the run-out behaviour is presented in figure 3.5. At a low input energy of 50 J, 4
²¹ and 16 material points per cell result in longer run-out distance, where as the run-out distance
²² converges when the number of PPC is more than 25. While at a high input energy of 500 J,
²³ both 4 and 16 PPC predict almost the same run-out distance, but is higher than the run-out
²⁴ predicted with more than 25 material points per cell.

²⁵ The evolution of the granular pile during the initial stage of flow is show in figure 3.6 for
²⁶ different number of material points per cell. At low input energy, fewer material points per cell
²⁷ results in larger separation of the spreading mass from the left wall. Distinct shear bands can
²⁸ be observed for more than 16 PPC. The flow structure remains unchanged with increase in PPC
²⁹ of more than 25. At a higher input energy (figure 3.7), almost all cases predict similar flow
³⁰ structure, except in the case of 4 PPC.

³¹ Figure 3.8 shows the evolution of kinetic energy with time for varying number of material
³² points per cell. At low input energy, the horizontal kinetic energy evolution is identical for all
³³ cases. A slightly quicker run-out evolution during the spreading phase can be observed for
³⁴ the case of 4 PPC. However, increase in the number of material points per cell significantly
³⁵ affects the evolution of the vertical kinetic energy E_{ky} . At low energy, a large proportion of the

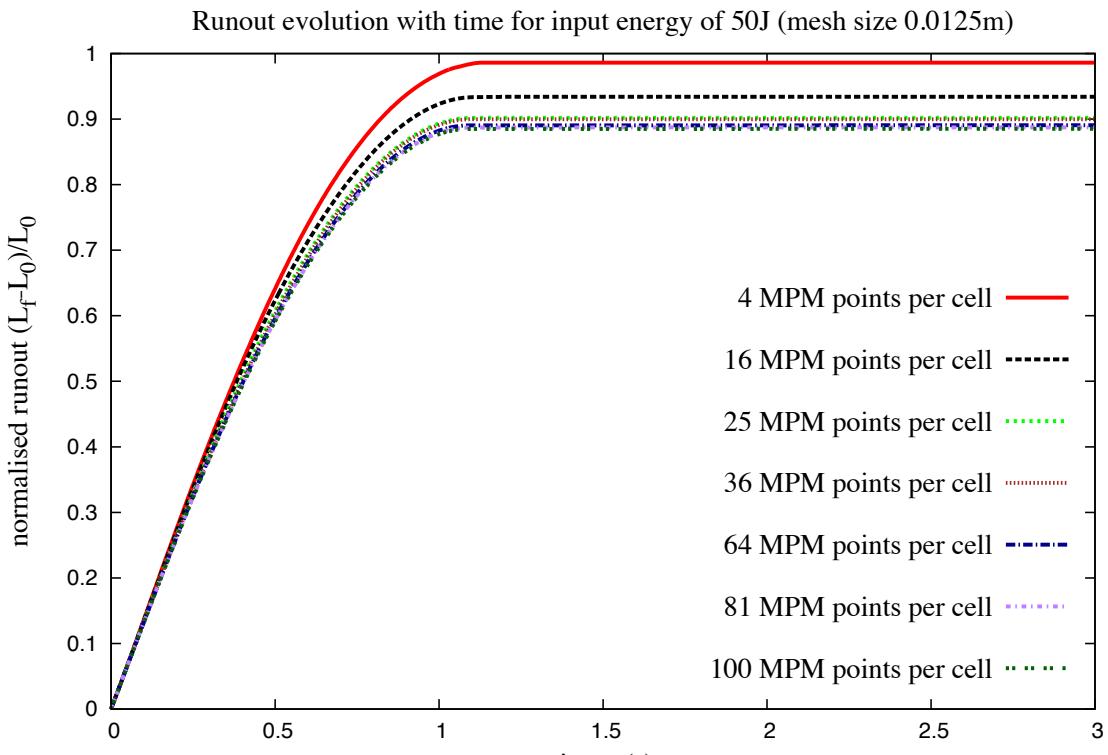
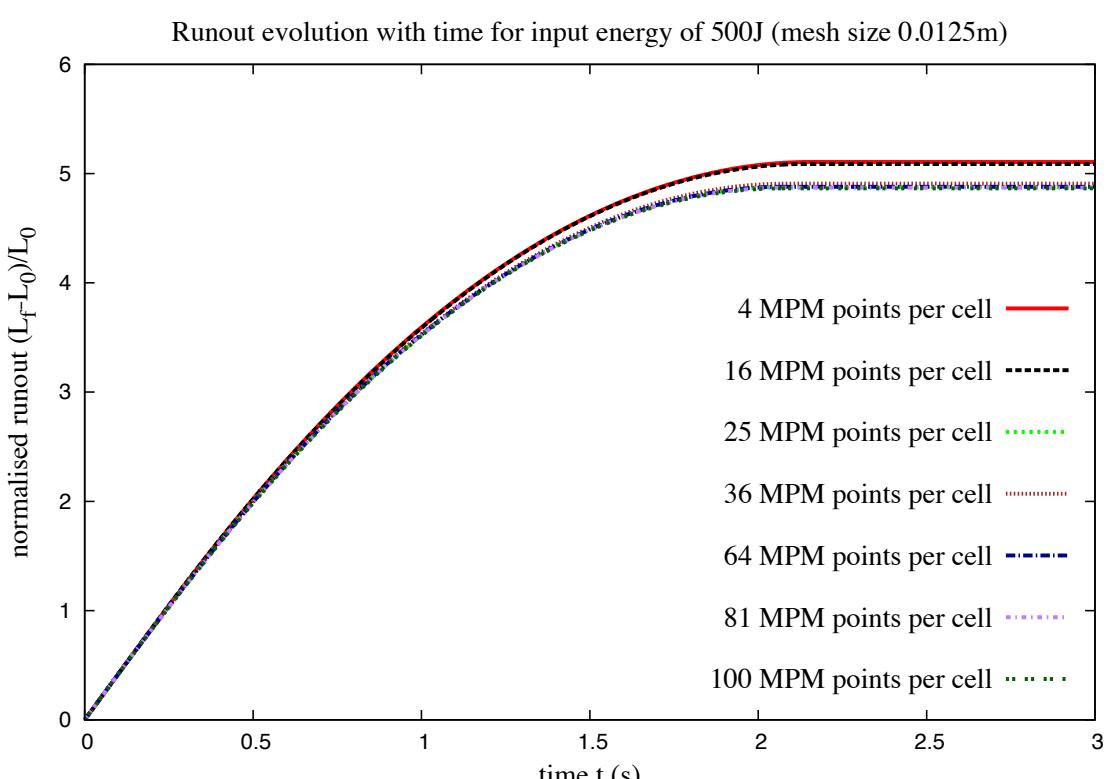
(a) $E_0 = 12.7mgd$ (b) $E_0 = 152mgd$

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

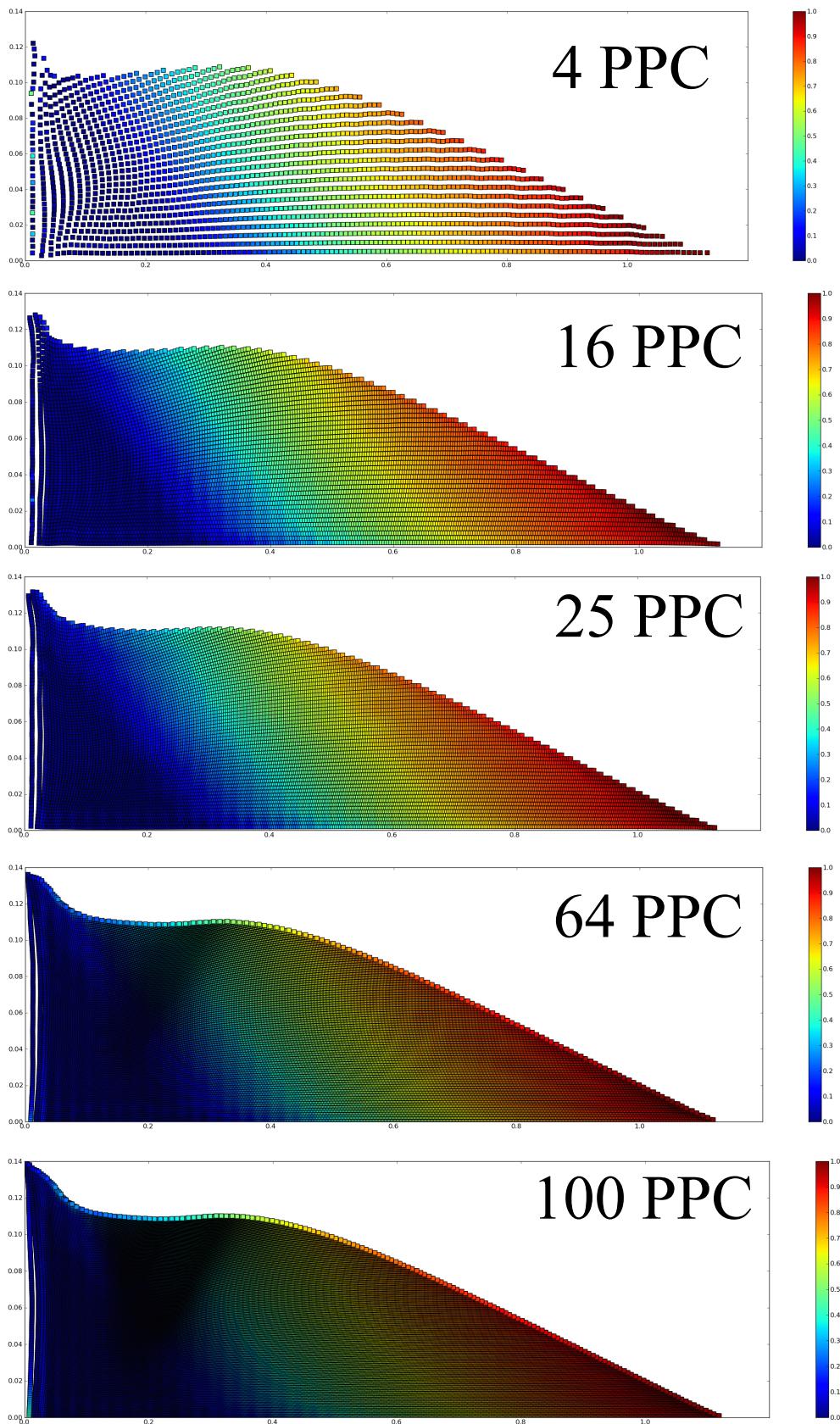


Figure 3.6 Effect of number of material points on cell on the run-out behaviour $E_0 = 12.7mgd$. Velocity profile (m/s) of granular pile subjected to gradient impact loading.

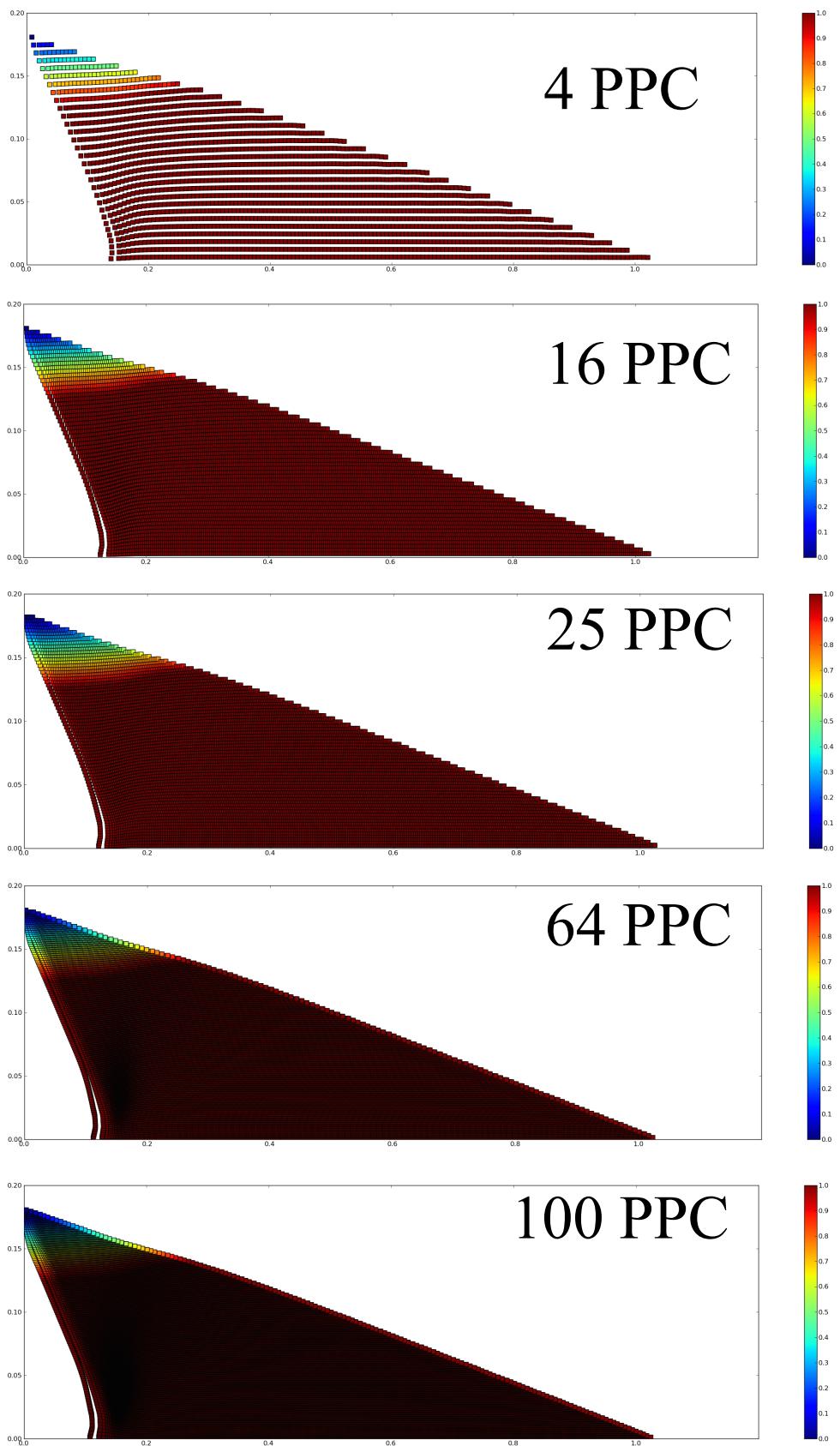


Figure 3.7 Effect of number of material points on cell on the run-out behaviour $E_0 = 152mgd$. Velocity profile (m/s) of granular pile subjected to gradient impact loading.

1 input energy is dissipated in the destabilisation process. This results in material points falling
 2 behind the spreading mass to the fill the cavity. Fewer material points per cell results in cell
 3 crossing noise as the material points filling the cavity experience free-fall due to gravity. The
 4 effect of cell-crossing noise can be seen in the oscillation of vertical kinetic energy for fewer
 5 material points per cell. However, at high input energy, most of the input velocity is dissipated
 6 during the spreading process. This means that only a small fraction of energy is available in the
 7 vertical component resulting in almost similar behaviour for all cases. Four material points per
 8 cell predicts a higher peak vertical kinetic energy in comparison with other cases, unlike the
 9 low energy case, no oscillations were observed for the high input energy.

10 The effect of mesh size on the flow kinematics is studied by comparing two mesh sizes:
 11 0.01 m and 0.0125 m (figure 3.9). It can be observed that the run-out distance converges with
 12 an increase in the number of material points per cell in both cases. Less than 1% difference
 13 in the run-out distance was observed between a mesh size of 0.0125 m and 0.01 m. The final
 14 run-out duration is almost unaffected by the increase in the number of material points per cell.

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

22 3.3.5 GIMPM

23 The shape functions used in the MPM are continuous, and hence penetration between bodies
 24 are handled automatically without the need for any supplemental contact algorithm ([Chen
25 and Brannon, 2002](#)). In the MPM, the continuum body deforms and moves in an arbitrary
 26 computation grid and all the boundary conditions are carried by the boundary particles. If a
 27 boundary particle is present in a cell, then the cell boundary becomes a part of the continuum
 28 body, and the cell size represents the thickness of the boundary. However, in certain cases both
 29 the boundary particle and an interior particle can be found in a cell, in which case the cell is
 30 still treated as a boundary cell, and the interior particle temporarily acts as a boundary particle.
 31 To avoid numerical errors, it is essential to consider smaller cell size along the boundary ([Chen
32 and Brannon, 2002](#)).

33 In the MPM simulations, numerical noises are observed when the material points crosses
 34 the cell boundaries during deformation of the material, this is termed as cell crossing noise. If a
 35 material point is located very close to the cell boundary, it results in discontinuous nature of

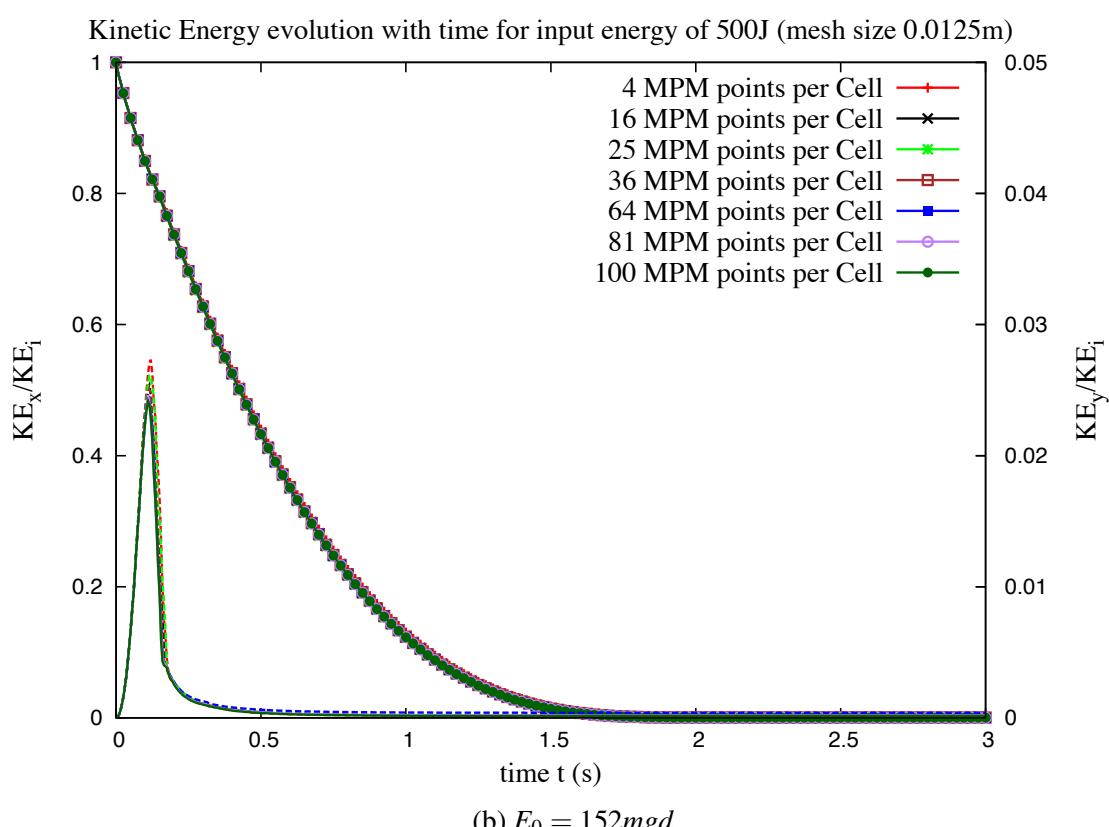
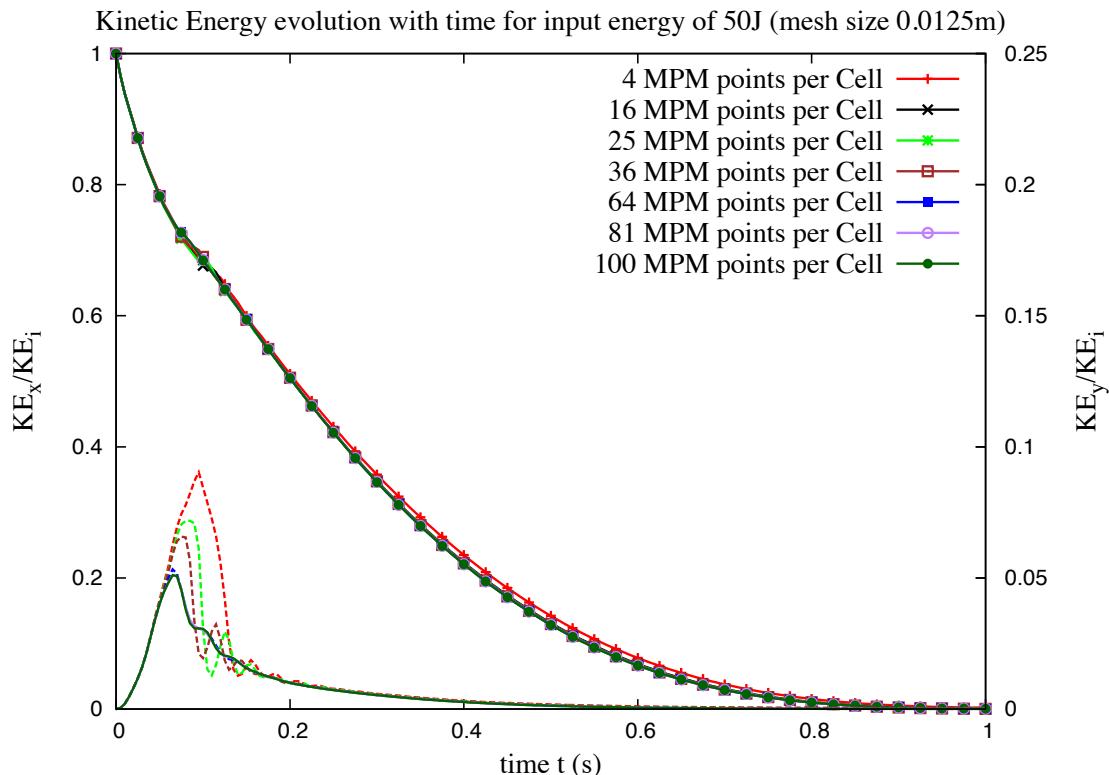


Figure 3.8 Evolution of kinetic with time for varying material points per cell

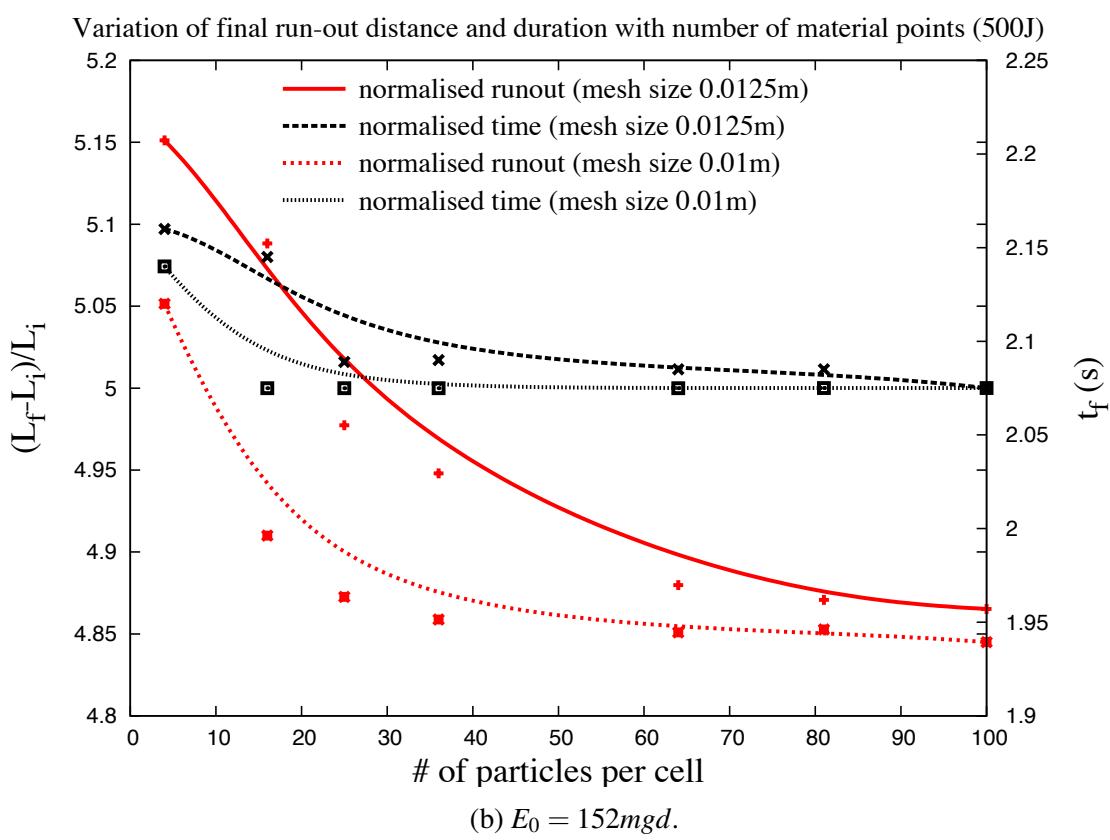
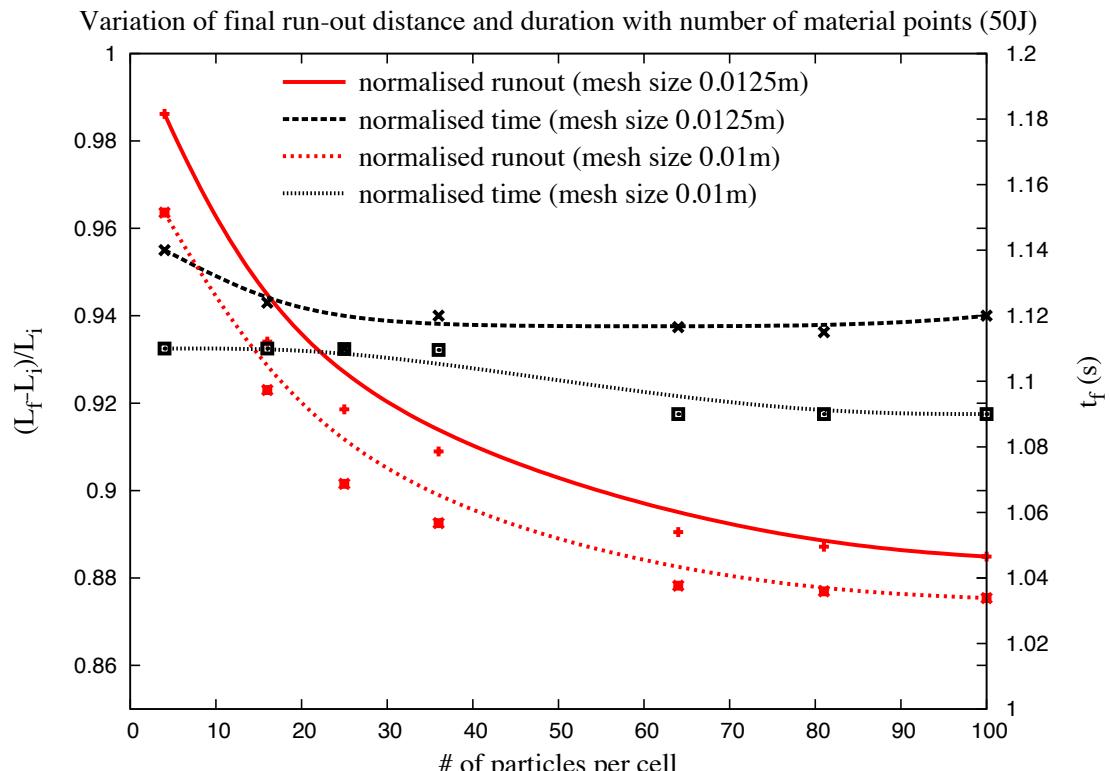


Figure 3.9 Evolution of run-out and duration of flow for varying material points per cell.

the gradient of the weighing function causing a force imbalance on the grid (Bardenhagen and Kober, 2004). This results in large non-physical acceleration values resulting in separation of material points from the continuum (Sulsky et al., 1995). figure 3.10 illustrates the problem of cell crossing noise. The main reason for the occurrence of cell crossing noise is the use of piecewise linear shape functions. However, this problem, which is predominant when using fine mesh size, can be overcome by changing the order of arithmetic operation as proposed by Sulsky et al. (1995). To overcome the problem of cell crossing noise, Bardenhagen and Kober (2004) proposed an alternate method called the Generalized Interpolation Material Point Method that uses smoother shape functions and a larger influence region for each grid node. This approach minimizes the cell crossing noise. However, special attention is required to simulate the boundaries in the Generalized Interpolation Material Point Method (GIMP).

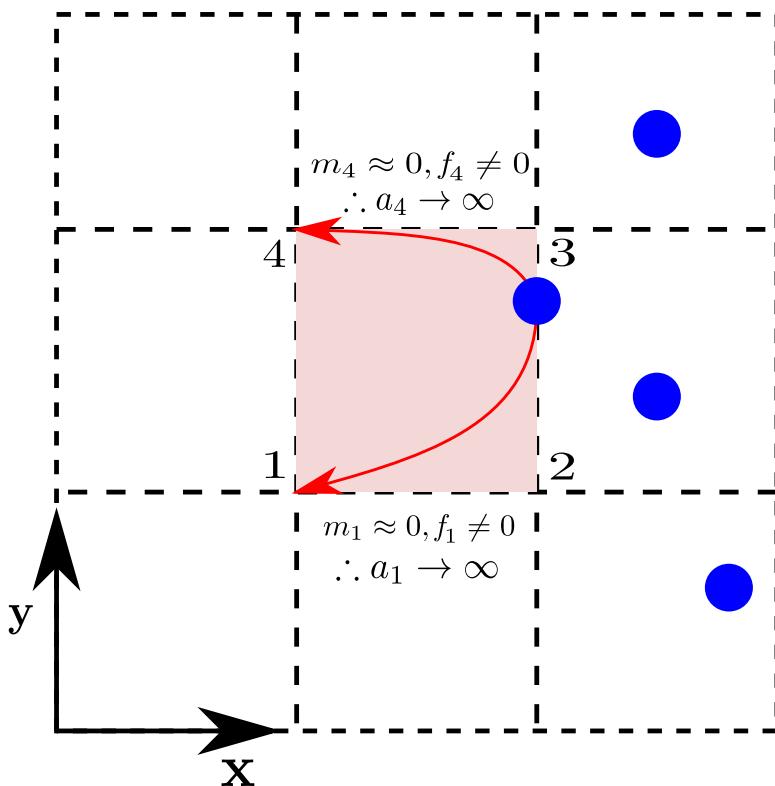


Figure 3.10 Cell crossing noise

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

1 wide variations in the physical and chemical properties of the grains, the discrete granular
2 structure has a rich generic phenomenology, which motivates us to understand the fundamental
3 behaviour of these materials. A granular material can be considered as a continuous material
4 if it is viewed at a macroscopic scale, ignoring the fact that it is composed of grains. On a
5 macroscopic scale, the behaviour of the granular material could be approximately defined
6 using continuum mechanics. However, On a grain level, the granular materials exhibit complex
7 solid-like and/or fluid like behaviour depending on the way the grains interact with each other.
8 The Analytical and Finite Element models, which consider granular materials as continuum
9 cannot take into account the local geometrical processes that govern the mechanical behaviour
10 of a non-homogeneous soil. The application of continuum models to describe granular flow
11 poses subtle problems for statistical analysis ([Mehta and Barker, 1994](#)). The grain level
12 description of the granular material enriches the macro-scale variables which poorly account
13 for the local rheology of the materials. Numerical models based on the Discrete Element
14 approach proposed by ? are capable of simulating the granular material as a discontinuous
15 system. Although, modern measurement techniques can probe into the local granular variables,
16 like grain position, velocities, contact forces, etc., they have inherent limitations in acquiring
17 those variables. The *discrete-element* approach is a powerful and reliable research tool to study
18 the behaviour granular materials at the grain-scale. This approach involves applying Newton's
19 equation of motion simultaneously to all grains described as rigid solid bodies by considering
20 the contact forces and the external forces acting on the grains. For a given boundary condition,
21 the collective mechanical response of grains to the external force leads to the relative motion
22 between grains constrained in a dense state and/or by in-elastic collisions in the loose state. ?
23 applied this method to granular geomaterials, and called it the *Distinct Element Method*, to
24 differentiate from the existing *Finite Element method* used in geomechanics. The attribute
25 "distinct" refers to the degrees of freedom of individual grains, but it was later replaced by
26 "discrete" to underline the discrete nature of the system. A similar method called *Discrete
27 Element Method* (DEM) was used at the same time for the simulation of molecular systems
28 with classical schemes that could be directly applied to granular materials. In-spite of the
29 formal analogy (grains and force laws) between granular and molecular systems, the physics is
30 fundamentally different. The interactions between individual grains are governed by unilateral
31 contact laws, and the mechanism of energy dissipation is through friction and inelastic collisions.
32 Moreover, granular materials have a wide variation in their grain shape and size distribution
33 that require appropriate numerical treatments. In the Molecular Dynamics approach, the normal
34 reaction force, which prevents the interpenetration of two grains is proportional to the depth
35 of penetration. Thus, frictional contact between grains can be expressed as a function of the
36 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.

$$m_i \frac{d^2x_i}{dt^2} = \mathbf{F}_i, (i = 1, \dots, N) \quad (3.20)$$

where N is the number of grains in the simulation, m_i is the mass of 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. 3.20. 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.

1 The computation of the forces and torques is the central part of the Discrete Element
 2 Method simulation. The dynamics of the granular material is governed by Newton's equation
 3 of motion which depends on the centre-of-mass coordinates and the Euler angles of the grains i
 4 ($i=1,\dots,N$):

$$5 \quad \frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{1}{m_i} \vec{\mathbf{F}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j)$$

$$6 \quad \frac{\partial^2 \vec{\varphi}_i}{\partial t^2} = \frac{1}{\hat{J}_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j), (j = 1, \dots, N)$$

8 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
 9 of inertia \hat{J}_i are (sometimes complicated) functions of the grain positions \vec{r}_j , their angular
 10 orientations $\vec{\varphi}_j$, and their corresponding velocities \vec{v}_j and $\vec{\omega}_j$. In a two-dimensional system,
 11 the angular orientation of a grain is described by a single (scalar) quantity φ_i and the moment
 12 of inertia reduces to a scalar value J_i . The Newton's equation of motion can be written as:

$$13 \quad \frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{1}{m_i} \vec{\mathbf{F}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j)$$

$$14 \quad \frac{\partial^2 \vec{\varphi}_i}{\partial t^2} = \frac{1}{J_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j), (j = 1, \dots, N)$$

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

$$19 \quad \vec{\mathbf{F}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{F}}_{ij}, \quad \vec{\mathbf{M}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{M}}_{ij} \quad (3.21)$$

21 The limitation to pairwise interaction is an abstraction, which is justified if the grains
 22 deformation at the contact is trivial. To describe the deformation of granular assemblies one
 23 has to take into account the effect of multi-grain interactions. This method is general and can
 24 be applied to a wide range of systems. The Discrete Element Method can be used to study the
 25 behaviour of grains in rapid flows to static assemblies. The method treats both the conditions
 26 in exactly the same way, it is not necessary to divide the system and then treat each condition
 27 differently. The simplest model for granular grain is a sphere. In a two-dimensions case, the
 28 sphere is reduced to a circular disk. Simulations using spherical grains are numerically very
 29 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. 3.20 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:

$$\mathbf{F}_i = \sum_{j \neq i} \mathbf{F}_{ij} + \mathbf{F}_{ext,i} \quad (3.22)$$

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 3.11). The contact force can be decomposed into two components, as the normal (F_n) and the tangential (F_t) components:

$$\mathbf{F}_{ij} = F_n \mathbf{n} + F_t \mathbf{t} \quad (3.23)$$

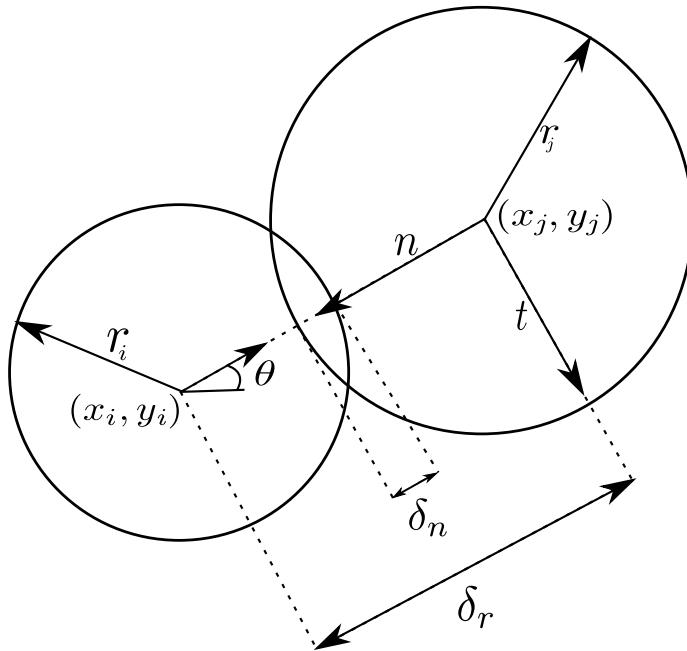


Figure 3.11 Grains i and j in contact, and the separation δ_n is used to calculate the normal force

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

1 Normal force

2 When granular grains collide, part of the kinetic energy is dissipated as heat and the other
 3 part causes deformation of the grain. These deformations generate interaction forces. The
 4 grains are considered to be rigid while their contact is assumed to be soft. Thus, the grains
 5 do not change their shape, instead they overlap. The shapes of the grains are conserved on an
 6 average, after many collisions. The overlap at the contact is limited to very small deformations,
 7 which are achieved by defining a repulsive normal force that opposes the overlap. The mutual
 8 compression (δ_n) of the grains i and j is defined as:

$$9 \quad \delta_n = |x_i - x_j| - r_i - r_j \quad (3.24)$$

11 where x_i and x_j are the centres of the grains and r_i and r_j are their radii (see Figure 3.11). When
 12 $\delta_n > 0$, the two grains are not in contact, and there is no interaction. When $\delta_n < 0$, the two
 13 grains overlap, and there is a repulsive normal force that pushes the two grains apart. The
 14 simplest model is to consider the contact as a linear spring with damping. The repulsive force
 15 depends linearly on δ_n , and is controlled by the stiffness of the grain. The energy dissipation
 16 due to the interaction between grains is an intrinsic characteristic of the granular material and
 17 is incorporated by adding a damping force that opposes the relative velocity for the duration of
 18 the contact. The interaction force at the contact is idealized as a simple spring-dashpot system,
 19 with elastic and dissipative constants (Luding et al., 1994).

$$20 \quad F_n = \begin{cases} 0, & \delta_n > 0 \\ -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, & \delta_n < 0 \end{cases} \quad (3.25)$$

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

$$28 \quad \hat{F}_n = -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, \quad F_n = \begin{cases} 0, & \hat{F} \leq 0 \\ \hat{F}_n, & \hat{F} > 0 \end{cases} \quad (3.26)$$

30 For pairwise collisions, the normal force (F_n) represented as $k_n \delta_n + \gamma_n$ causes a decrease in the
 31 relative normal velocity of the grains by a factor ε . This factor is the *coefficient of restitution*,

and is defined as $\varepsilon \approx g'/g$, 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. 3.24. Thus we obtain:

$$\frac{d\delta_n}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n} \quad (3.27)$$

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. 3.27 yields the separation δ_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, which is defined as:

$$\varepsilon = \exp\left(-\frac{\pi\gamma_n}{2m^{eff}} / \sqrt{\frac{Y}{m^{eff}} - \frac{\gamma^n}{2m^{eff}}}\right) \quad (3.28)$$

Tangential force

Granular grains are not perfect spheres, but have a complicated surface texture, therefore at oblique collisions, besides the normal force there is a tangential force too. Even perfectly smooth spheres exert a tangential force due to their bulk viscosity (?). To build a heap of spheres on a flat surface, the grains as well as the surface has to be sufficiently rough, indicating the dependence of the tangential force on the surface properties of the granular materials. For realistic simulation of granular materials, it is important to consider the tangential force in Discrete Element Method. The tangential force is considered in a similar fashion as the normal force, arising from a spring stretched by the relative motion of the grain. Tangential forces are modelled by considering the relevant relative tangential velocity of the grain surfaces at the point of contact. The point of contact is an approximation, as the description of the normal force assumes a compression δ_n , which implies a contact surface in 3-D or a contact line in 2-D. Assuming a tangential spring of length δ_t exerts an opposing force to the relative tangential displacements (ignoring the effect of relative rolling between the grains), the tangential force can be postulated similar to the normal force (eq. 3.27) as:

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{t} \quad (3.29)$$

This equation must also be numerically integrated, just like eq. 3.20. The grains are in contact when $\delta_t < 0$, and when $\delta_t = 0$, the grains no longer exert a force on each other. With these assumptions, δ_t can be calculated similar to the normal force. The tangential force is assumed

¹ to be governed by Coulomb's friction law.

² $|F_t| \leq \mu F_n$ (3.30)

³ where F_t is the tangential force and μ is the friction coefficient. It is therefore necessary to
⁴ constrain the tangential force to remain less than or equal to μF_N . To impose the condition
⁵ in eq. 3.30, two-stages similar to the normal force computation is adopted. The first step is to
⁶ evaluate the candidate force, and is then accepted if it obeys the condition in eq. 3.30.

⁷
$$\hat{F}_t = -k_t \delta_t - \gamma_t \frac{d\delta_t}{dt}, \quad F_t = \begin{cases} sgn(F_t), & |\hat{F}| \geq \mu F_n \\ \hat{F}_t, & |\hat{F}| < \mu F_n \end{cases}$$
 (3.31)
⁸

⁹ where k_t is the stiffness of the tangential spring and γ_t is the damping constant. If $|F_t| = \mu F_n$,
¹⁰ the contact is sliding, otherwise, it is non-sliding. It can be noted that the normal force (eq. 3.26)
¹¹ and the tangential force (eq. 3.31) are handled in the same way in Discrete Element Method.
¹² When the grains slide against each other, they do not retain any memory of their initial position,
¹³ and hence do not return to its original position. In order to model this behaviour, a limiting
¹⁴ value of δ_t is imposed. When the contact slides $\delta_t = \pm \mu F_n / k_t$ is imposed.

¹⁵ In addition to sliding, the grains can roll relative to one another about their centre of mass
¹⁶ due to the tangential force acting at their contact surface. In this case, $d\delta_t/dt = 0$. It is important
¹⁷ to assume that the grains touch at a single point instead of overlapping, i.e. $\delta_n = 0$. This point
¹⁸ 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
¹⁹ $v_i + r_i(\boldsymbol{\omega} \times \mathbf{n})$. If it belongs to grain j , its velocity is $v_j + r_j(\boldsymbol{\omega} \times \mathbf{n})$. The relative velocity is the
²⁰ difference between these two velocities.

²¹
$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{t} - (r_i \boldsymbol{\omega}_i + r_j \boldsymbol{\omega}_j) \times \mathbf{n}$$
 (3.32)

²² It should be noted that the eq. 3.32 is only an approximation, as the grains in Molecular
²³ Dynamics do not touch at points, but overlap. It is therefore an approximation that produces
²⁴ an error of order $O(\delta_n/r)$ (Radjai and Dubois, 2011). It is assumed that the contact forces are
²⁵ exerted at the point of contact. It implies that the tangential force is accompanied by torque
²⁶ acting on two grains. If the overlap is zero, these torques are:

²⁷
$$\tau_{ij} = -(a_i \mathbf{n}) \times (F_t \mathbf{t}), \tau_{ji} = -(a_j \mathbf{n}) \times (F_t \mathbf{t}),$$
 (3.33)
²⁸

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. 3.20:

$$I_j \frac{d\omega_i}{dt} = \sum_{j \neq i} \tau_{ij} \quad (3.34)$$

where I_j is the moment of inertia of grain j . The eq. 3.33 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 packing, a non-linear law can have significant influence on the acoustic properties (?), and on the global stiffness (?). However, in case of rapid granular flows, the interaction force between the grains has almost no effect on the phenomenon, and a linear law can be used to describe this kind of behaviour (Radjai and Dubois, 2011).

3.5.2 Numerical algorithm and integration scheme

The efficiency of a Discrete Element Method program is mainly determined by its efficiency to compute the interaction forces between grains. If we consider a model system with pairwise interactions, we have to consider the contribution of the force on grain i due to all its neighbours. If we consider only the interaction between a grain and the nearest image of another grain, then for a system of N grains, we must evaluate $N \times (N - 1)/2$ pair distances. Consider a system of 1000 grains, at every time step all possible pairs of grains have to be considered to compute the interaction forces, hence, $N(N - 1)/2 \approx 500,000$ force computations are required. For short-range grain interactions, the majority of these force evaluations is unnecessary as the corresponding grains are located far apart and do not necessarily touch each other. For a dense system of equally sized grains, the grains can have contacts with not more than 6 grains, this reduces the number of force computation required to $3N \approx 3000$. In the preliminary force computation scheme, at least 166 times more pair interactions are considered than necessary. Therefore, the numerical methods employed in the Discrete Element Method program should try to minimize the computation of interaction forces (?). There are three different methods for the efficient computation of the forces, the *Verlet* algorithm, the *link-cell* algorithm, and a

¹ lattice algorithm. The Verlet algorithm described in Grubmuller et al. (1991) is implemented in
² 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 3.12. 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*:

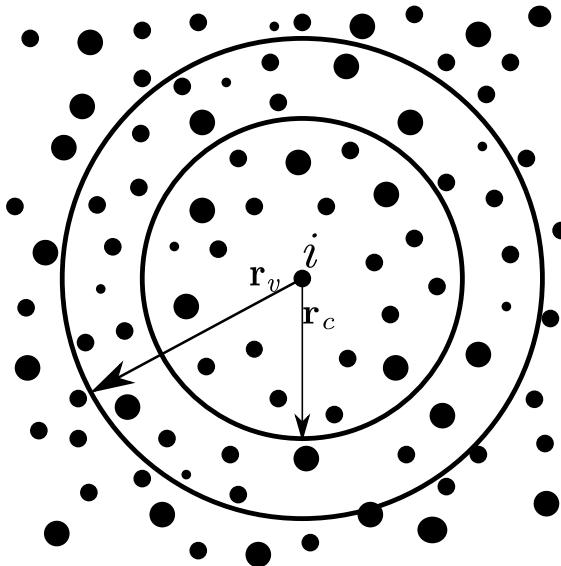


Figure 3.12 The Verlet list: a grain i interacts with those grains with the cut-off radius r_c , the Verlet list contains all the grains within a sphere with radius $r_v > r_c$

$$(|\vec{r}_i - \vec{r}_j| - R_i - R_j) < \text{Verlet distance} \quad (3.35)$$

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

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

$$\left| \vec{r}_i - \vec{r}_j \right| - R_i - R_j > 0 \quad (3.36)$$

¹⁴ The above condition is required for all pairs (i,j) of grains which are *not* known as neighbours.
¹⁵ This condition is a criterion to update the Verlet lists (?). Assume at the instant when the
¹⁶ Verlet lists are constructed, the surfaces of the grains have the distance $\left| \vec{r}_i - \vec{r}_j \right| - R_i - R_j >$
¹⁷ Verlet distance, i.e. they are not classified as neighbours. If the Verlet lists are updated
¹⁸ before one of these grains has travelled the distance *verletdistance*/2 since the lists were
¹⁹ constructed, they can never collide without being recognized as neighbours first. This is
²⁰ explained in figure 3.13. The impact of optimisation of the Verlet list algorithm has negligible
²¹ effect on the computation time, as the algorithm is quite efficient already and only consumes a
²² few percent of the total computation time in construction of the Verlet lists. The implementation
²³ of the Verlet list algorithm in force computation drastically reduces the computation time in
²⁴ comparison to the linear algorithm. The performance of the Verlet list algorithm is controlled
²⁵ by two crucial parameters: the number of cells N_c , for the construction of the Verlet lists and
²⁶ the Verlet distance r_v (?).

²⁷ Leap frog or Verlet integration algorithm

²⁸ Discrete Element Method involves numerically solving Newton's equation of motion eq. 3.20,
²⁹ which is an ordinary differential equation. Choosing an integration algorithm is important, as
³⁰ the forces are not always differentiable in time, and the temporary derivative of the force is
³¹ discontinuous when the contact splits. It is also very essential to numerically integrate eq. 3.32
³² with the same precision as eq. 3.20. At first, computational speed seems important. It is usually
³³ not very relevant because the fraction of time spent on integrating the equation of motions

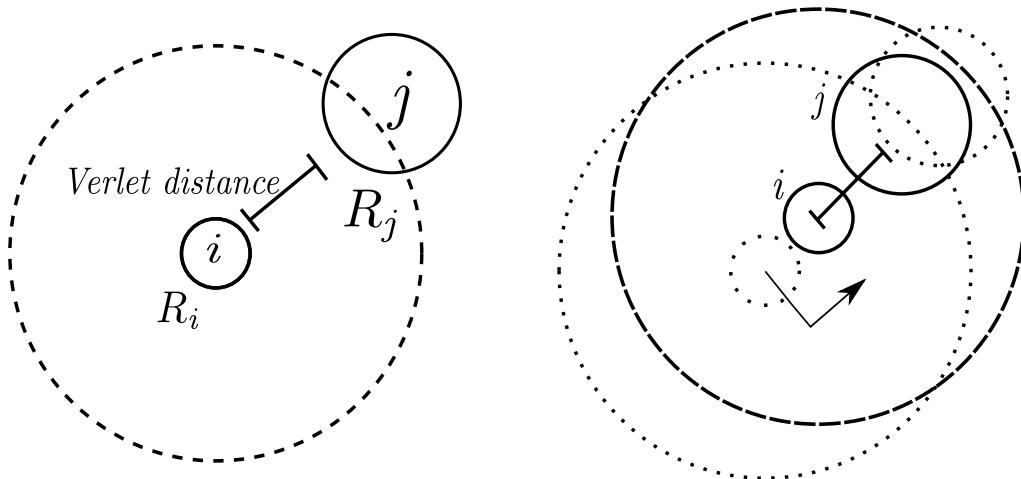


Figure 3.13 Checking the validity of Verlet lists. *Left:* the grains i and j are not recognized as neighbours since the distance of their surface is larger than the Verlet distance. The radius of the dashed circle is $R_i + R_{max} + \text{Verlet distance}$. *Right:* in the most unfortunate case the 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.

$$t(t + \Delta t) = rt + \mathbf{v}(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots \quad (3.37)$$

If we truncate this expansion beyond the term Δt^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 Δt is the time step. On the other hand, their velocities are calculated at intermediate times, that is, at $t = \Delta t/2, 3\Delta t/2, \dots$. Let the position of a grain at time $t = k\Delta t$ be written as x_k , and its velocity at time $t = \Delta t(k + 1/2)$ be written $\mathbf{v}_{k+1/2}$, and its acceleration at $t = k\Delta t$ be \mathbf{a}_k . Then the following equation is used to advance systematically:

$$\mathbf{v}_{k+1/2} = \mathbf{v}_{k-1/2} + \mathbf{a}_k \Delta t, \quad x_{k+1} = x_k + \mathbf{v}_{k+1/2} \Delta t \quad (3.38)$$

This algorithm determines the new grain position with an error of order $O(\Delta t^4)$. But [eq. 3.38](#) 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:

$$\mathbf{v}_k = \mathbf{v}_{k-1/2} + \mathbf{a}_{k-1}\Delta t/2 \quad (3.39)$$

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. 3.39 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. 3.38. 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. 3.40 and figure 3.14. To calculate the forces and acceleration, it requires the positions and velocities at time t:

$$\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t)\Delta t \quad (3.40)$$

$$x(t + \Delta t) = x(t) + \mathbf{v}(t + \Delta t/2)\Delta t \quad (3.40)$$

$$\mathbf{v}(t) = \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t - \Delta t)\Delta t/2 \quad (3.40)$$

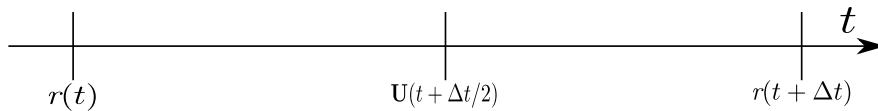


Figure 3.14 Verlet integration scheme

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 ω_N , the characteristic oscillation frequency of very short waves. This frequency is proportional to ω_0 , which is easier to estimate. Therefore, it is essential to choose a time step $\Delta t \approx \epsilon/\omega_0$, where ϵ 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.

3.5.3 Boundary conditions

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

21 Periodic boundary

The effect of a wall on the response of grains is very critical, especially in numerical simulation where the number of grains is relatively fewer in comparison to the experiments. The undesired effect of a wall can be eliminated using periodic boundary conditions, i.e. a periodic extension of the simulation area in one or more dimensions. Any grain leaving the system at one side is reintroduced at the opposite side, and correspondingly the interaction forces between grains at opposite sides of the simulation area are taken into account. In this framework, the simulation domain becomes a unit area containing grains with periodic copies paving the whole system. The periodic boundary conditions extend the system boundaries to infinity, so that the simulation cell simply plays the role of a coordinate system to locate grain positions [3.15](#).

The external stresses or displacements are applied on the simulation box by constraining the degrees of freedom of the wall, which are alternatively kept free or fixed depending on whether a stress or a displacement is monitored in a system. With periodic boundary conditions, this role is played by the collective degrees of freedom carried by the coordinate system, whose

basis vectors become dynamic variables, and their conjugate stresses are expressed as a state function of the granular configuration (?). 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.

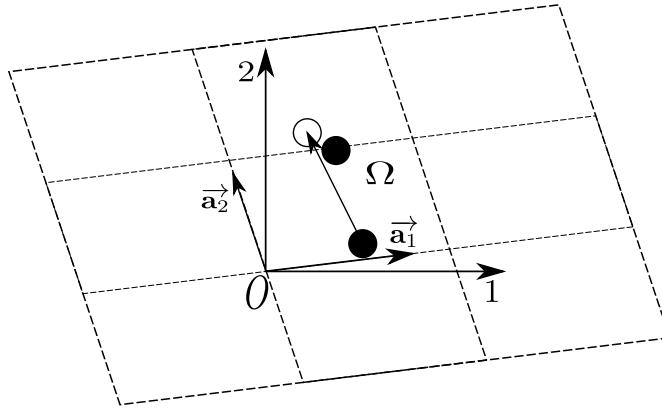


Figure 3.15 A 2D simulation cell Ω with its basis vectors in an absolute frame. A grain located at the right boundary interacts with the image of another grain located at the left boundary.

Let us consider a collection of N_p grains with their centres contained in a cell of volume V . The cell can have any shape allowing for a periodic tessellation of space. The simplest shape is a parallelepiped i.e. parallelogram in 2D. The cell and its replicas define a regular lattice characterized by its basis vectors (\vec{a}_1, \vec{a}_2). In the case of a parallelogram, the basis vectors may simply be the two sides of the parallelogram; figure 3.15. 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:

$$\vec{r}(i') = \vec{r}(i) + \sum_{k=1}^2 i_k \vec{a}_k \quad (3.41)$$

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

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

³ **3.5.4 Particle Assembling Methods**

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

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

$$p(R) = \frac{R_{min}R_{max}}{R_{max} - R_{min}} \frac{1}{R^2} \quad (3.42)$$

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

$$R = \frac{R_{min}R_{max}}{R_{max} - z(R_{max} - R_{min})} \quad (3.43)$$

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 3.16); 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 3.16 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 ???

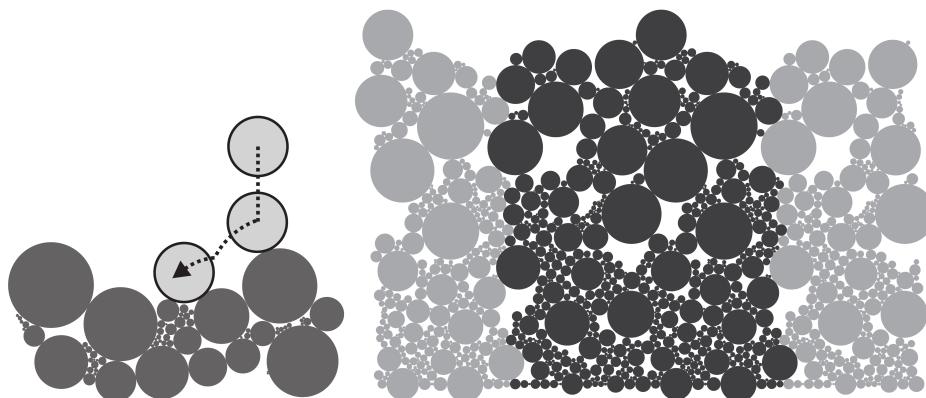


Figure 3.16 (a)Ballistic deposition: first contact followed by steepest descent; (b) small-scale periodic sample ([Radjai and Dubois, 2011](#))

13

14 Statistical analysis of prepared sample

15 In order to ensure the homogeneity of the sample prepared using the above technique, a statisti-
 16 cal analysis of the sample is performed. Various parameters such as grain size, coordina-
 17 tion number, contact normal direction, contact normal force, etc., can be used in the statistical
 18 analysis to verify the homogeneity of the prepared sample. Of the various parameters available,
 19 the coordination number, i.e. the average number of contacts per grain is chosen to study the
 20 homogeneity of the sample, because of its simplicity and its physical significance in represent-
 21 ing the density of the sample. With increase in the coordination number, the density of the
 22 granular assembly increases. The mechanically stable sample prepared by ballistic deposition
 23 technique presented in ?? is used for the statistical analysis. Representative elements of size

0.02 m × 0.01 m (highlighted segments in ??) were considered. Each representative element has approximately 40 grains. A histogram is plotted showing the number of grains having a particular coordination number for all the representative elements (??). A normal distribution of the number of grains and the coordination number can be observed from ?? . Most grains are found to have 2 or 3 contacts with its neighbours. The number of grains with higher coordination number, i.e. coordination number greater than 3, is found to increase as we go down the sample. This is attributed to the effect of gravity, which increases linearly as we go down. Representative elements at the top are found to have a shift in their peak towards lower coordination number, as they are less restrained in comparison to their counterparts at the bottom of the sample. Overall, the sample is found to be homogeneous, with representative elements having similar normal distribution of the coordination number. Thus, the prepared sample is a good representation of the actual granular assembly and the simulations based on these samples are found to be more realistic in comparison to crystalline/lattice packing of granular grains.

3.5.5 Voronoi Tessellation

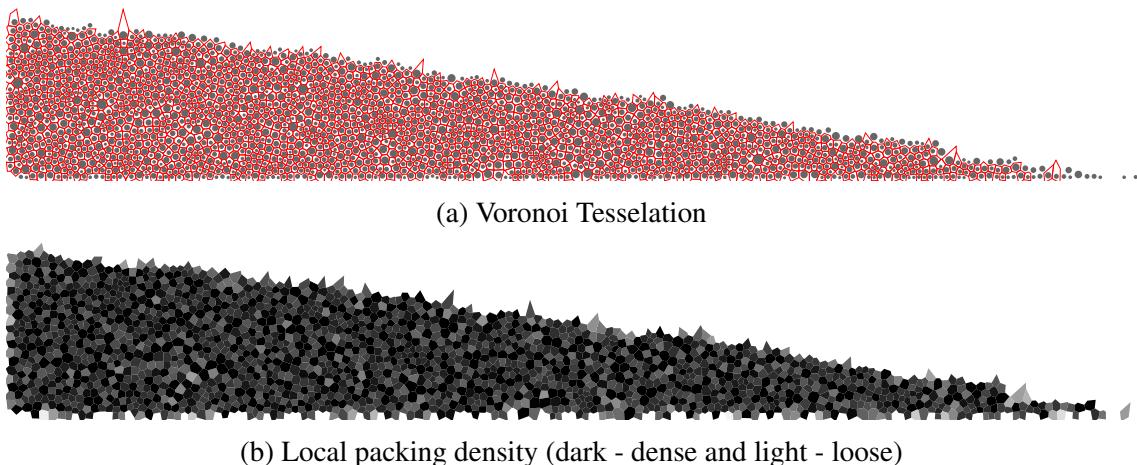


Figure 3.17 Voronoi tessellation to average bulk properties

Draft - v1.0

Thursday 25th September, 2014 – 21:16

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
- 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. 5
6
7
- Augarde, C. and Heaney, C. (2009). The use of meshless methods in geotechnics. In *Proceedings of the 1st International Symposium on Computational Geomechanics*, France. 8
9
- Bandara, S. (2013). *Material Point Method to simulate Large Deformation Problems in Fluid-saturated Granular Medium*. PhD thesis, University of Cambridge. 10
11
- Bardenhagen, S. and Kober, E. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496. 12
13
- 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. 14
15
16
- Belytschko, T., Lu, Y. Y., and Gu, L. (1994). Element-free Galerkin Method. *International Journal for Numerical Methods in Engineering*. 17
18
- 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. 19
20
- Bonet, J. and Kulasegaram, S. (2000). Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International journal for numerical ...*. 21
22
23
- Chen, Z. and Brannon, R. (2002). An evaluation of the material point method. *Sandia National Laboratories (SAND2002-0482)*. 24
25
- 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. 26
27
28
- 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 ...*. 29
30
31

- ¹ 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.
- ⁴ 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.
- ⁷ Gumhold, S. (2003). Splatting illuminated ellipsoids with depth correction. volume 2003, pages 245–252.
- ⁹ Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Computer Methods in Physics*, 3:319–343.
- ¹¹ Hertz, H. (1882). Über die Berührung fester elastischer Körper. *Journal für die Reinen und Angewandte Mathematik*, 92:156–171.
- ¹³ Jaeger, H., Nagel, S., and Behringer, R. (1996). Granular solids, liquids, and gases. *Reviews of Modern Physics*, 68(4):1259–1273.
- ¹⁵ Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):235–257.
- ¹⁷ 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.
- ¹⁹ Jullien, R., Meakin, P., and Pavlovitch, A. (1992). Random packings of spheres built with sequential models. *Journal of Applied Physics*, 25:4103.
- ²¹ Kafaji, I. (2013). *Formulation of a dynamic material point method (MPM) for geomechanical problems*. PhD thesis, University of Stuttgart.
- ²³ 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.
- ²⁶ Krogh, M., Painter, J., and Hansen, C. (1997). Parallel sphere rendering. *Parallel Computing*, 23(7):961–974.
- ²⁸ Kuester, F., Bruckschen, R., Hamann, B., and Joy, K. (2001). Visualization of particle traces in virtual environments. pages 151–157.
- ³⁰ Levoy, M. (1988). Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37.
- ³² Li, S. and Liu, W. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55:1.
- ³⁴ Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169.
- ³⁶ Luding, S., Clément, E., and Blumen, A. (1994). Anomalous energy dissipation in molecular-dynamics simulations of grains: The "detachment" effect. *Physical Review*, 50:4113.

- 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. 1
2
3
- Maeda, K. and Sakai, H. (2010). Seepage failure and erosion of ground with air bubble dynamics. *Geoenvironmental Engineering and ...*. 4
5
- Meakin, P. and Jullien, R. (1985). Structural readjustment effects in cluster-cluster aggregation. *Journal de Physique*, 46(9):1543–1552. 6
7
- Mehta, A. and Barker, G. (1994). The dynamics of sand. *Reports on Progress in Physics*, 57:383. 8
9
- Monaghan, J. (2005). Smoothed particle hydrodynamics. *Reports on progress in physics*. 10
- Mori, H. (2008). *The SPH method to simulate river levee failures*. Msc thesis, University of Cambridge. 11
12
- Qiu, G., Henke, S., and Grabe, J. (2011). Application of a Coupled Eulerian–Lagrangian approach on geomechanical problems involving large deformations. *Computers and Geotechnics*. 13
14
15
- Radjai, F. and Dubois, F. (2011). *Discrete-element modeling of granular materials*. ISTE Wiley, London; Hoboken, N.J. 16
17
- Radjai, F. and Richefeu, V. (2009). Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41(6):715–728. 18
19
- Radjai, F., Voivret, C., and McNamara, S. (2011). Discrete-element modelling of granular materials. chapter Periodic b, pages 181–198. ISTE Wiley. 20
21
- 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. 22
23
- 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. 24
25
- Shin, W. (2010). *Numerical simulation of landslides and debris flows using an enhanced material point method*. PhD thesis. 26
27
- 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. 28
29
30
- 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. 31
32
33
- 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. 34
35
- Sulsky, D., Zhou, S.-J., and Schreyer, H. L. (1995). *Application of a particle-in-cell method to solid mechanics*, volume 87. 36
37

- ¹ Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159(1):98.
- ²
- ³ Vold, M. J. (1959). A numerical approach to the problem of sediment volume. *Journal of Colloid Science*, 14(2):168–174.
- ⁴